Diss. ETH No. 19411

ANIMATION RECONSTRUCTION OF DEFORMABLE SURFACES

A dissertation submitted to
ETH Zurich

for the degree of
Doctor of Sciences

by

Hao Li
Dipl.-Inform., Universität Karlsruhe (TH)

born 17 January 1981
citizen of Saarbrücken, Germany

Committee in charge:

Prof. Dr. Mark Pauly, EPFL, Chair
Prof. Dr. Szymon Rusinkiewicz, Princeton University
Prof. Dr. Markus Gross, ETH Zurich / Disney Research
Dr. Kiran Bhat, Industrial Light & Magic, Lucasfilm Ltd.

2010

The dissertation of Hao Li is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____
Chair

ETH Zurich

2010

To my mother, my sister, and my father. Actually, Dad isn't in this world anymore, so he'll never know. But I'll still dedicate it to him...

You've got to be able to make
animation for much less...
—Don Bluth

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor and mentor Mark Pauly for instilling in me the joy of conducting outstanding research in computer graphics. These four years have been the most intense and successful times in my career. Your support and vigilance have allowed me to achieve results that I couldn't have thought of.

Thank you so much Committee for the direction, feedbacks, and all the enlightening advices. Thank you Szymon Rusinkiewicz, Kiran Bhat, and Markus Gross.

Furthermore I would like to acknowledge my close collaborators and friends: Thibaut Weise for the great partnership and your awesome 3D scanner. Thank you Bart Adams and Leonidas Guibas for the unforgettable times at Stanford, working with you has been a wonderful experience and a great source of inspiration. My thanks also go to Bob Sumner for the guidance and your pioneering work on mesh deformation. I really wonder how my thesis would be without these technological advances. I would like to extend my gratitude to Linjie Luo, Daniel Vlasic, Pieter Peers, and Jovan Popović for the excellent collaborative effort in my last research project.

I would like to acknowledge my lab mates for the exciting times at ETH Zurich: Bálint Miklós for the memorable (and destructive) parties, Michael Eigensatz, Camille Wormser, and of course the entire crew at CGL and Disney Research Zurich: Johannes Schmid, Nils Thürey, Cengiz Öztireli, Bernd Bickel, Manuel Lang, Marcel Germann, Simon Heinzle, and Alexander Hornung. I also had the pleasure to supervise some very talented Master students: Jens Puwein, Huw Bowles, Jeroen Dries, and Liana Manukyan.

A very special thank you to the Industrial Light & Magic folks: Kiran Bhat, Brett Allen, Kevin Wooley, Stephen Spencer, Chris Twigg, Cary Phillips, Steve Sullivan, Rob Levine, Yisheng Chen, Vivek Verma, Rony Goldenthal, Ronald Mallet, Fred Pighin, Jeff Smith, Oliver Franzke, David Lenihan, and Mike Jutan. My summer internship at ILM was one of the best times during my Ph.D., thank you for making that happen!

I am also much indebted to the insightful discussions and fun times with all the Siggraph friends: Carsten Stoll, Mario Botsch, Eitan Grinspun, Szymon Rusinkiewicz, Daniel Vlasic, Niloy Mitra, Peter Huang, Maks Ovsjanikov, Martin Wicke, Krystle de Mesa, Emily Whiting, Liliya Kharevych, Tilke Judd, Ilya Baran, Simon Pabst, Josiah Manson, Linjie Luo, Michael Wand, Martin Bokeloh, Will Chang, Jochen Süssmuth, Sylvain Paris, Olga Sorkine, Jessica Hodgins, Sang Il Park, Pieter Peers, Abhijeet Ghosh, Cyrus Wilson, and Alex Ma.

Thanks also go to two digital artists: Polina Tolkacheva and Tricia Barrett for sweating with me during the last Siggraph submission. I would like to thank Aguru Images, Inc. and Larry McCallister from Paramount Pictures for granting me the rights for using digital material from G.I. Joe: The Rise of Cobra in this dissertation. Thank you Tim Hawkinson for allowing me to use the "Bear" model in one of my figures and Jason Osipa for using your 3D face model in our project on facial rigging. Thanks also go to Volker Helzle for providing the full FACS expressions of the Nikita model.

I want to thank Yuanshan Lee, Krystle de Mesa, Duygu Ceylan, Etienne Vouga, Wolfgang Globke, and Oliver Franzke immensily for proofreading my publications and this dissertation as well as for all their suggestions for improvements.

Finally, I would like to take the opportunity to thank my mother, my sister, my father, and all my friends from high school and undergrad years.

My research is supported in part by the Swiss National Science Foundation (grants 20001-112122 and 200020-124738), NSF grants ITR 0205671 and ISS-1016703, FRG 0354543, FODAVA 808515, NIH grant GM-072970, the Fund for Scientific Research, Flanders (F.W.O.-Vlaanderen), the University of Southern California Office of the Provost, as well as the U.S. Army Research, Development, and Engineering Command (RDECOM). The content of the information does not necessary reflect the position or the policy of the U.S. Government, and no official endorsement should be inferred.

TABLE OF CONTENTS

ABSTRACT OF THE DISSERTATION

ANIMATION RECONSTRUCTION OF DEFORMABLE SURFACES

by

Hao Li

Doctor of Sciences

ETH Zurich, 2010

Prof. Dr. Mark Pauly, Chair

Accurate and reliable 3D digitization of dynamic shapes is a critical component in the creation of compelling CG animations. Digitizing deformable surfaces has applications ranging from robotics, biomedicine, education to interactive games and film production. Markerless 3D acquisition technologies, in the form of continuous high-resolution scan sequences, are becoming increasingly widespread and not only capture static shapes, but also entire performances. However, due to the lack of inter-frame correspondences, the potential gains offered by these systems (such as recovery of fine-scale dynamics) have yet to be tapped. The primary purpose of this dissertation is to investigate foundational algorithms and frameworks that reliably compute these correspondences in order to obtain a complete digital representation of deforming surfaces from acquired data. We further our explorations in an important subfield of computer graphics, the realistic animation of human faces, and develop a full system for real-time markerless facial tracking and expression transfer to arbitrary characters. To this end, we complement our framework with a new automatic rigging tool which offers an intuitive way for instrumenting captured facial animations.

We begin our investigation by addressing the fundamental problem of non-rigid registration which establishes correspondences between incomplete scans of deforming surfaces. A robust algorithm is presented that tightly couples *correspondence estimation* and *surface deformation* within a single global optimization. With this approach, we

break the dependency between both computations and achieve warps with considerably higher global spatial consistency than existing methods. We further corroborate the decisive aspects of using a non-linear space-time adaptive deformation model that maximizes local rigidity and an optimization procedure that systematically reduces stiffness.

While recent advances in acquisition technology have made high-quality real-time 3D capture possible, surface regions occluded by the sensors cannot be captured. In this respect, we propose two distinct avenues for dynamic shape reconstruction. Our first approach consists of a bi-resolution framework which employs a smooth template model as a geometric and topological prior. While large-scale motions are recovered using non-rigid registration, fine-scale details are synthesized using a linear mesh deformation algorithm. We show how a detail aggregation and filtering procedure allows the transfer of persistent geometric details to regions that are not visible by the scanner. The second framework considers temporally-coherent shape completion as the primary target and skips the requirement of establishing a consistent parameterization through time. The main benefit is that the method does not require a template model and is not susceptible to error accumulations. This is because the correspondence estimations are localized within a time window.

The second part of this dissertation focuses on the animation reconstruction of realistic human faces. We present a complete integrated system for live facial puppetry that enables compelling facial expression tracking with transfer to another person's face. Even with just a single rigid pose of the target face, convincing facial animations are achievable and easy to control by an actor. We accomplish real-time performance through dimensionality reduction and by carefully shifting the complexity of online computation toward offline pre-processing. To facilitate the manipulation of reconstructed facial animations, we introduce a method for generating facial blendshape rigs from a set of example poses of a CG character. The algorithm transfers controller semantics from a generic rig to the target blendshape model while solving for an optimal reproduction of the training poses. We show the advantages of phrasing the optimization in gradient space and demonstrate the performance of the system in the context of art-directable facial tracking.

The performance of our methods are evaluated using two state of the art real-time acquisition systems (based on structured light and multi-view photometric stereo).

KURZFASSUNG DER DISSERTATION

ANIMATIONSREKONSTRUKTION VON DEFORMIERBAREN FLÄCHEN

von

Hao Li

Doktor der Wissenschaften

ETH Zurich, 2010

Prof. Dr. Mark Pauly, Leiter

Die genaue und zuverlässige Digitalisierung von dynamischen Objektoberflächen ist ein wichtiger Bestandteil für die automatische Erstellung von realistischen Computeranimationen. Anwendungen befinden sich sowohl im Bereich der Robotik, Biomedizin und Bildung als auch bei der Produktion von interaktiven Computerspielen und Filmen. Markierungslose 3D-Scantechnologien die nicht nur eine statische Oberfläche erfassen sondern eine vollständige Sequenz von hochaufgelösten Scans aufnehmen finden immer häufiger Verwendung. Aufgrund der fehlenden Korrespondenzen zwischen der einzelnen Aufnahmen, ist eine Ausschöpfung deren Potenzials (z.b. die Gewinnung der Dynamik von feinen Details) bislang nicht möglich. Das primäre Ziel dieser Dissertation besteht darin fundamentale Algorithmen und Systeme zu untersuchen welche durch die Berechnung dieser Korrespondenzen eine vollständige digitale Rekonstruktion eines erfassten Objekts ermöglichen. Weiterhin untersuchen wir die realistische Animation von Gesichtern als wichtigen Aspekt der Computergrafik und entwickeln dabei ein in Echtzeit operierendes vollständiges System welches sowohl die markierungslose Verfolgung von Gesichtern als auch die Übertragung von Ausdrücken auf beliebige Gesichter ermöglicht. Anschliessend ergänzen wir das System mit einem neuartigen Rigging-Verfahren welches erfasste Gesichtsanimationen intuitiv Kontrollierbar macht.

Wir beginnen unsere Untersuchung mit der fundamentalen Problemstellung der nicht-starren Registrierung welches es möglich macht Korrespondenzen zwischen un-

vollständigen Oberflächenerfassungen von deformierbaren Objekten herzustellen. Wir stellen einen robusten Algorithmus vor welcher eine feste Kopplung zwischen geschätzte Korrespondenzen und Oberflächendeformation durch einen einzigen globalen Optimierung ermöglicht. Dieses Verfahren entfernt Abhängigkeiten zwischen den beiden Berechnungen und erlauben es Deformationen mit signifikant höherer räumlichen Kohärenz zu erzielen. Zu den weitere wichtigen Aspekten unseres Verfahrens gehören sowohl das nichtlineare Raum-Zeit-adaptives Deformationsmodell zur Maximierung der lokalen Starrheit als auch ein Optimierungsablauf welcher in der Lage ist systematisch Steifheitseigenschaften des Modells schrittweise zu reduzieren.

Währen neuartige Erfassungstechnologien die Aufnahme von hochwertigen 3D Daten ermöglicht, können verdeckte Oberflächen nicht gescannt werden. Wir stellen deswegen zwei unterschiedliche Rekonstruktionsverfahren für die Gewinnung von vollständigen Objektoberflächen vor. Der erste Ansatz besteht aus einem Zwei-Skalen-Systems das geometrisches und topologisches A-priori-Wissen durch ein geglättetes Template einsetzt. Während grobe Bewegungen durch eine nicht-starre Registrierungsverfahren berechnet werden, können feine Details durch einen linearen Deformationsmodell gewonnen werden. Wir zeigen wie diese Details durch eine Akkumulations-und Filterprozedur effektiv in verdeckten Regionen transferiert werden kann. Der zweite Ansatz besteht primär darin Löcher von verdeckten Oberflächen durch zeitlich-kohärente Geometrien zu vervollständigen. Der Vorteil dieses Verfahrens besteht vor allem darin das weder eine global konsistente Oberflächenparametrisierung noch ein Template-Modell benötigt wird. Da zusätzlich Korrespondenzen nur lokal innerhalb eines Zeitfensters berechnet werden können sich bei längeren Aufnahmen keine Fehler akkumulieren.

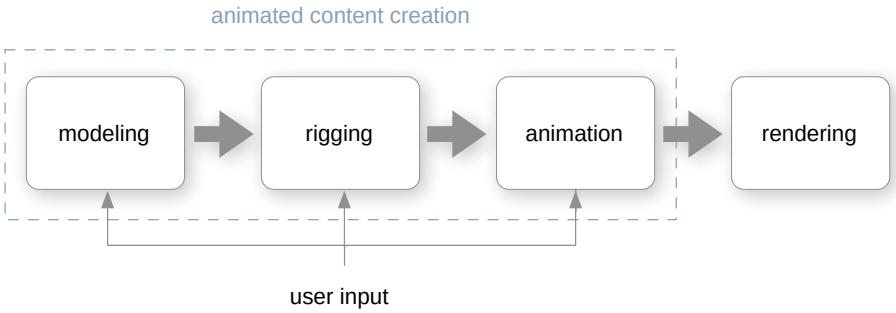
Der zweite Teil dieser Arbeit konzentriert sich auf die Animationsrekonstruktion von realistischen Gesichter. Wir stellen ein vollständig integriertes System für die Verfolgung von komplexen Gesichtsausdrücken und deren Übertragung auf anderen Gesichter vor. Glaubhafte Animationen von Gesichtern sind sogar dann möglich wenn nur eine einzige Zielpose zu Verfügung steht, wobei ein Schauspieler die erzeugte Animation sehr einfach steuern kann. Die Berechnungen erfolgen dabei in Echtzeit durch die Verwendung von Dimesionsreduktion und eines sorgfältigen Vorverarbeitungsschritts. Um die Manipulation der rekonstruierten Gesichtsanimationen zu ermöglichen, führen wir eine Methode zur Erzeugung von Gesichts-Blendshapes ein die lediglich einige wenige Beispieleposen eines computergenertiertedn Characters benötigt. Der beschriebene Algorithmus

überträgt semantische Bedeutung eines generischen Rigs auf einem Zielmodell, wobei die Beispielposen optimal nachgebildet werden. Ausserdem zeigen wir dass es vorteilhaft ist diese Optimierung im Gradientenraum durchzuführen und demonstrieren die Leistungsfähigkeit unseres Systems im Kontext einer, durch einen Künstler kontrolliertbaren, Gesichtsverfolgungsmethode.

Unsere Methoden werden mithilfe von zwei modernen Echtheit-Erfassungssysteme evaluiert (basierend auf strukturiertem Licht und Multi-View-Photometric-Stereo-Verfahren).

Preface

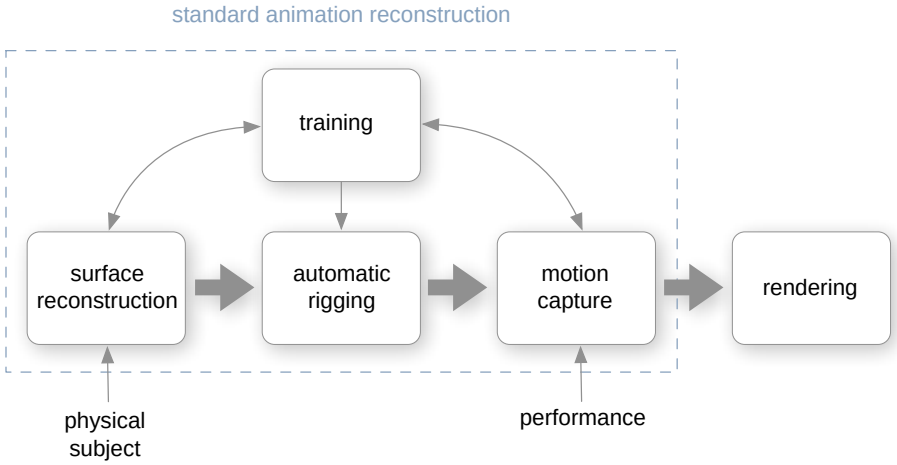
What is Animation Reconstruction? Animation reconstruction is the inverse process of generating computer animation. For example, instead of sculpting a person's face by hand, we may simply capture and reconstruct its shape using a 3D scanner (*surface reconstruction*). Rather than creating the movement of someone jumping and getting the timings right, we can directly record the animation using a motion capture system (*motion capture*). An alternative way to customizing animation controls by hand is to train the system with inputs of real artists so that it is capable of performing the same task automatically in the future (*training*).



Standard content creation pipeline for computer animation.

To illustrate some inefficiencies of the standard *animated content creation pipeline*, let us consider the simple example of animating a CG character. First, an artist is involved in the creative process of *modeling* a 3D object which consists of sculpting a surface. A careful placement and configuration of a skeleton (*rigging*) for the handcrafted model then allows the artist to intuitively create new poses. A frequently practiced way to *animate* a character consists of specifying key poses at specific time frames and

interpolate the motions (*keyframing*). Once the animation is ready, each frame of the animated 3D object may then be *rendered* as a two-dimensional image, given a virtual camera, light sources, textures, and more. Because each stage of the traditional pipeline involves a significant amount of manual work and artistic skills, recreating realistic animation is remarkably time-consuming and difficult.



Standard animation reconstruction pipeline. In the traditional setting, surface reconstruction and motion capture are separate stages.

Animation reconstruction considers any computational aspects that supports each stage of the standard pipeline through direct measurements from reality. Its pipeline is depicted in the above figure and proposes a shift from laborious *human interpretation* of real-world geometry and motion to an accurate and automatic *acquisition* process. At the very core of animation reconstruction are the design of computational models for effective processing of captured input data, the involvement of meaningful geometric and kinetic priors, and the investigation of algorithms that allow those models to evolve their behaviors based on sampled *training data*.

1 Introduction

Ever since the birth of computer animation, intuitive modeling and animation tools were developed to support *scientists*, *engineers*, *educators*, and *artists* in creating compelling *animated visual content*. Through *computer generated* (CG) animation, conveying the functionality of complex systems can be more accurate, learning experiences become more intuitive, and fascinating animated feature films are made. Because the traditional graphics pipeline relies on a considerable amount of human input, producing realistic animation remains a challenging and time-consuming process. As a result, the field of computer graphics has substantially expanded over the past ten years with techniques to automate this process. A predominant number of digital models and phenomena are *inspired* or *directly adopted* from the *real-world*. This observation has stimulated the development of sensing technologies that directly measure the shape and motion of actual dynamic objects—significantly reducing the effort required for a person to model and animate from scratch. However, obtaining a complete representation of the shape and motion of highly deformable objects (such as human bodies, faces, and cloths) remains a challenging problem because the subject may exhibit arbitrary *complex deformations* or have *large occlusions*. While resolution and accuracy are constantly

improving with each generation of new imaging sensors, capturing the entire shape at a single instance is generally impossible even when multiple viewpoints are used. We argue that increasing 3D scan coverage is therefore on a fundamentally different "technology curve," and is unlikely to be solved by improvements in scanning technology.

Hypothesis: *The premise of this work is that aggregating a continuously captured sequence of incomplete data through time can be appropriate to derive sufficient knowledge of an object's shape and deformation. This information can be further used to effectively support animators in creating and manipulating compelling CG animations for challenging dynamic subjects such as human faces.*

This dissertation investigates frameworks and geometric techniques that accurately reconstruct *dynamic three-dimensional models* of deforming surfaces captured with high-resolution real-time 3D scanners. While striving to develop robust and general purpose algorithms that can handle a wide range of deformations (such as human performances, skin deformation, garment wrinkling, etc. . .), we further emphasize on modeling highly complex *facial animation* and present tools for intuitive manipulation and transfer of facial expressions to other characters. The goal of this work is to establish a new foundation for *inverse engineering* computer animation and to push the boundaries of pure *geometric* and *data-driven* approaches developed over the past decade.

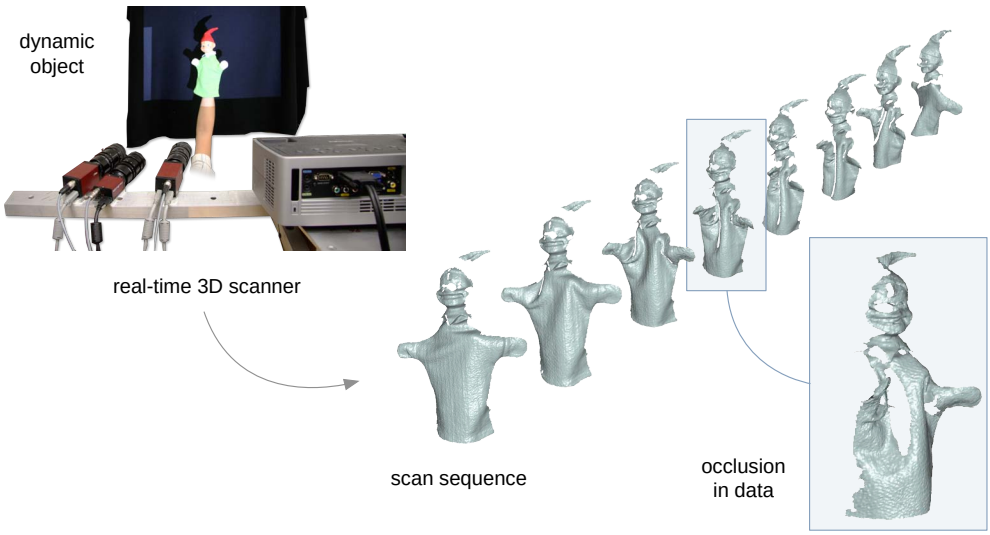


Figure 1.1: Our real-time structured light scanner based on active stereo delivers high resolution input scans from a single view.

Acquisition. The first step in animation reconstruction consists of capturing dynamic objects. Traditionally, shape and motion are both separately captured before being combined to create an animated model. In particular, shapes are obtained through *3D scanning* and motion is recovered typically by *tracking markers* that are placed on the subject. While optical shape acquisition has become widely accepted as a mature technology for digitizing static objects [FHM+96, MTSA97, NWN96, RTG97, Cur97, MBR+00, LPC+00, Li05, RGB, SCD+06a, BBB+10], only relatively recently can accurate and dense geometries be captured at sustained "video" rates [RHHL02, DRR03, ZSCS04, ZH04, MES, WLG07, HVB+07, BPS+08, VPB+09, BHPS10], enabling detailed acquisition of surfaces that undergo complex deformations (hundreds of thousands of surface samples per frame).

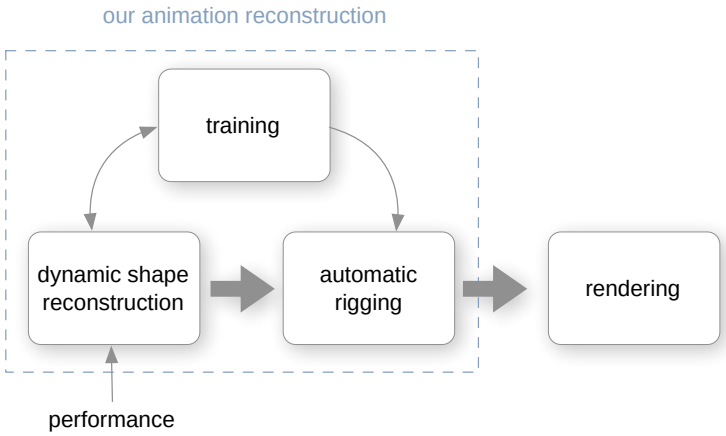


Figure 1.2: Our proposed animation reconstruction pipeline. Note how geometry and motion are captured within a single dynamic shape reconstruction stage.

The animation reconstruction algorithms in this dissertation are designed around two such real-time acquisition systems: one that is based on structured light [WLG07] and one on multi-view photometric stereo [VPB+09]. Both state of the art systems capture dense geometries at 30 frames per second (fps) and do not involve any markers (c.f., Figure 1.1). The main advantages over traditional marker-based motion capture systems [Vic, PH06] are as follows: the ability to recover fine-scale dynamics (since motion can be acquired at the same resolution as the geometry), no requirement to place and calibrate markers (which is impractical and time-consuming), and also, the ability to capture surface textures simultaneously. Consequently, these new acquisition technologies suggest a new animation reconstruction pipeline as illustrated in Figure 1.2.

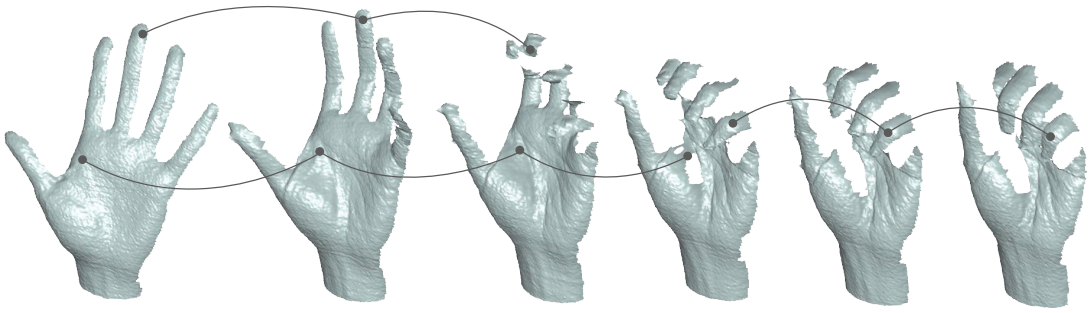


Figure 1.3: Incomplete correspondences. Because of occlusions and deformations in the subject, establishing surface correspondences across the entire recording is particularly challenging.

Correspondences. To recover the full motion without the use of markers, dense inter-frame *correspondences* need to be established across the captured data. Even though similar geometric features and reflection properties (such as color) are important indicators for matching surface regions, they can significantly differ when the subject is deforming. Furthermore, optical scanners can only acquire a portion of the full surface at each frame due to occlusions. For instance, when a hand is grasping, parts of the palm are visible in one frame but hidden at a later time as shown in Figure 1.3. Typically, real-time sensing devices also suffer from noise and outliers as a result of algorithmic and hardware limitations, and non-cooperative surface materials. A thorough discussion on issues with correspondence computation for continuous scans can be found in Li [LP07]

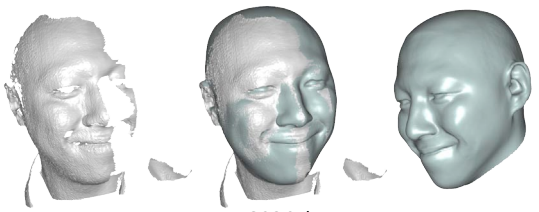
Dynamic Shape Reconstruction. This thesis investigates novel optimization techniques and the use of effective geometric and topological priors to establish dense *spatio-temporal* correspondences of deforming surfaces in the aforementioned ill-posed setting.

While no prior knowledge about correspondences or physical properties is needed, we do assume a moderate amount of temporal coherence in the input data. We develop a method that automatically computes correspondences and a warping field between pairs of scans by imposing a smooth and continuous detail preserving deformation model. Bringing two deformed and partially overlapping shapes into alignment is called *non-rigid registration* and it has long been believed that a fully automatic approach can only reliably handle small-scale warps such as those due to hardware calibration inaccuracies [IGL03, HTB03, BR04, BR07]. Larger deformations were typically recov-

ered using a complete model (*template*) of the scanned subject and often assisted with user-specified, sparse correspondences [ACP03, ASP+04, BBK06, ARV07]. Several researchers have identified the need for and importance of more robust and automatic techniques, which has led to a revival of research on pairwise non-rigid registration algorithms [HAWG08, LSP08, CZ08, CZ09, LAGP09, CLM+10] which can handle significantly larger deformations.

We further extend our non-rigid registration technique to robustly process longer sequences using only a coarse geometric template model as a prior, and scan sequences recorded from no more than a single view. In particular, our *geometry and motion reconstruction* framework [LAGP09] produces consistent dynamic meshes where geometric details hidden by occlusions are propagated from observations in other time instances. We also demonstrate that temporally coherent and hole-free mesh sequences can be computed [LLV+10] without involving any templates. These sequences enable valuable applications such as free viewpoint video.

Realistic Facial Animation. Having set the foundations for dynamic shape reconstruction, we can immediately apply our methods to realistic facial animation. Why faces? Humans are highly social animals—we interact with each other every day. As a result, we are particularly sensitive to the subtlest details that appear unnatural in CG faces. Creating compelling *facial expressions* is therefore a challenging and important aspect of computer graphics. Using motion capture data to produce realistic facial animation is generally more accurate and efficient than relying on traditional *keyframing* techniques, even though digital artists may be highly trained for this purpose. While the dense input data we use in this work captures the necessary fine-scale dynamics, as opposed to standard marker-based methods, it comes at the price of solving the significantly more challenging correspondence problem which is necessary for facial tracking.



On top of our exploration on geometry and motion reconstruction, this dissertation presents a complete and practical system [WLGP09] that covers two important aspects of facial animation, namely *markerless, real-time facial tracking* and *expression retargeting* to another person's face. We achieve convincing facial animations by careful

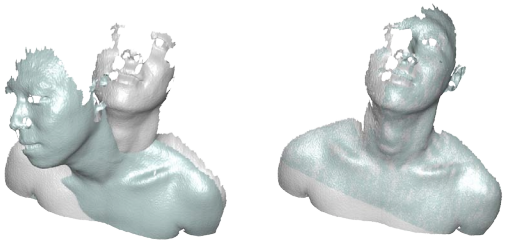
integration of state of the art registration and tracking techniques, efficient deformation models, and transfer algorithms. Furthermore, we investigate a novel approach for intuitive manipulation of reconstructed facial animations [LWP10]. Our approach consists of automatically generating a model (*facial rig*) for instrumenting *semantically meaningful* expression parameters such as "raise left eyebrow." Personalized rigs are obtained by providing example facial poses as training data. We show that our *rigging* technique may be easily integrated as a data-driven module for facial tracking and allows intuitive editing of facial animations via *blendshape controls*.

1.1 Objectives and Challenges

This dissertation investigates the fundamental question in animation reconstruction: *how can the full motion of deformable surfaces be accurately recovered from incomplete time-varying input data?* In particular, we are interested in knowing the positions of all surface points during the entire recording while the subject undergoes complex deformations. Because of occlusions, only a subset of the full geometry can be captured at a time, and as a result, surface regions disappear and (re-)appear. At the same time, surface reconstruction from scans captured at a particular frame can only deliver high resolution details in regions that are visible. The question arises as to whether geometric details that are hidden in one frame can be reconstructed once it is observed at another time as the subject exposes new surface. How can we distinguish between geometric details that are *persistent* or *transient* since the object deforms? When a full model (template) of the subject is unavailable, can we use recovered surface motion to better approximate missing geometry in hole regions?

Different geometric techniques will be presented in this thesis to address each of these questions. As we will further show, these foundational algorithms yield enabling technologies for realistic facial animation reconstruction and data-driven facial rigging. This thesis will find answers to the following problems:

Pairwise Non-Rigid Registration. To determine a dense motion field across a sequence of 3D input scans, we first need to develop an algorithm that automatically establishes full inter-frame correspondences. Given two consecutive scans,



initial alignment registration output

the problem consists of finding dense surface correspondences within overlapping regions and an optimal deformation that brings the source shape (frame t) into alignment with the target shape (frame $t+1$). As a result, we obtain a more complete surface at frame $t+1$ as additional geometry is propagated from frame t. However, the more the subject deforms the larger the difference becomes between the source and target shape. The problem becomes even more challenging as surface correspondences only exist within a common subregion which is not known *a priori*. Because pairwise registration will serve as a central building block for computing spatio-temporal correspondences of entire recordings, efficient computation will be a critical factor for practical considerations.

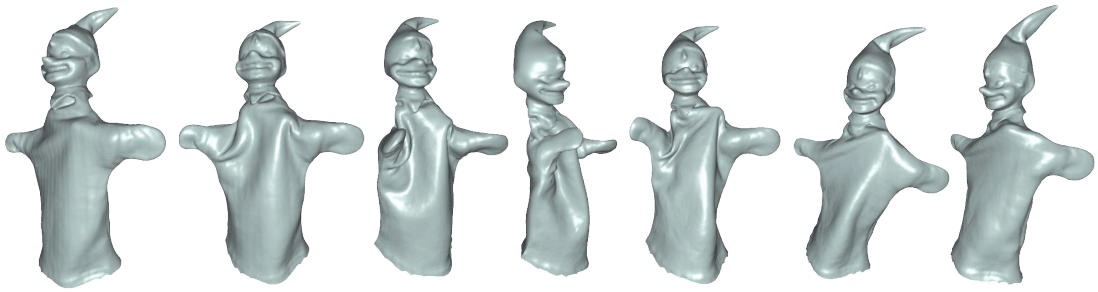


Figure 1.4: Full geometry and motion reconstruction computed from the single-view scans shown in Figure 1.1.

Geometry and Motion Reconstruction. From a sequence of partial scans acquired using a real-time 3D scanner, our goal is to reconstruct an animated sequence of a full digital model with consistent parameterization across the entire recording as illustrated in Figure 1.4. Since a full model can be easily obtained by a separate template building step using static surface reconstruction, pairwise correspondences may be directly used to track the template model. However, important geometric details that are hidden due to occlusions but exposed at a different frame should be reconstructed as well. Moreover, it is crucial to distinguish between *static* and *dynamic* details, since static ones will be persistent in the shape without being affected by the deformation of the object. We therefore consider both surface reconstruction and motion capture as a single reconstruction problem. The algorithm should be sufficiently resistant to error accumulations and robust enough to rely on observations from a single view where less than half of the object's surface is visible in each scan. In fact, most active optical acquisition systems (e.g., structured light scanner) are designed to capture from a single direction due to light interference issues.

Temporal-Coherent Shape Completion. Reconstructed meshes that are in full correspondence have the advantage for being ideal for editing operations throughout the motion such as texturing, shape editing, and deformation transfer. However, dynamic objects that involve topology changes cannot be represented by a single static template. Consider the example when a cloth is gliding on a human skin: two disconnected templates would be necessary to faithfully represent the process.

To deal with complex topology changes and still obtain a sequence of complete, watertight meshes, our goal consist of filling holes in occluded surface regions. Even when the subject is fully surrounded by 3D sensors, large holes cannot be avoided due to occlusions. Naively filling holes in each frame independently would however yield strong flickering in the output as no temporal information is taken into account. Our goal is therefore to develop a shape completion technique that is temporally coherent while accurate correspondences have to be reliably established across incomplete scans of topology changing subjects.

Figure 1.5: Accurate 3D facial expressions can be transferred in real-time from an actor (top) to a different face (bottom).

Real-time Facial Expression Tracking and Retargeting. Our findings on non-rigid registration and dynamic shape reconstruction will ensure direct impact in the field of facial animation. The ability to establish accurate correspondences between shapes helps to reliably track complex facial expressions and automate the process of building consistent parameterizations across faces of different people. To fully explore the potentials of our markerless, real-time acquisition system, we propose to develop a complete framework for real-time tracking of an actor specific facial model and expression retargeting to another person's face as illustrated in Figure 1.5. Facial tracking should be able to handle fast and instantaneous expression changes and be sufficiently accurate to capture any sub-

tle motion. We must also ensure that tracking remains robust for an indefinite length of input scans and does not suffer from error accumulations. Eventually, both, high-resolution facial tracking and expression transfer must be achieved in real-time to enable *live facial puppetry* as an integrated system for real, practical applications.

Automatic Facial Rigging Based on Examples. Let us consider the problem of how to manipulate reconstructed facial animations with intuitive expression controls. The process of manually *rigging* a custom character is time consuming, especially when we have realistic human expressions in mind. In film production, for instance, it is not atypical to build several hundreds of controls to animate or fine-adjust the expressions of a single digital face. Automatic facial rigging considers two objectives: equipping an input facial model with semantically meaningful expression controls and personalizing the model through training. The latter should be scalable in the sense that very few training samples (*examples*) would be sufficient to capture a person's facial characteristics. Figure 1.6 illustrates the influence of input examples on the generated facial blendshapes. Because personalized expressions may now be triggered by a set of controls (e.g., "move lower lip up"), we can simply transfer these semantic parameters to another rigged character instead of entire deformations as done traditionally. In particular, the generated rigs must be accurate enough to describe the "true" expressions of the target person so that more convincing retargeting can be achieved as when source expressions are being transferred. Furthermore, to allow intuitive editing of captured data, we also require seamless integration of personalized rigs into our facial tracking framework.

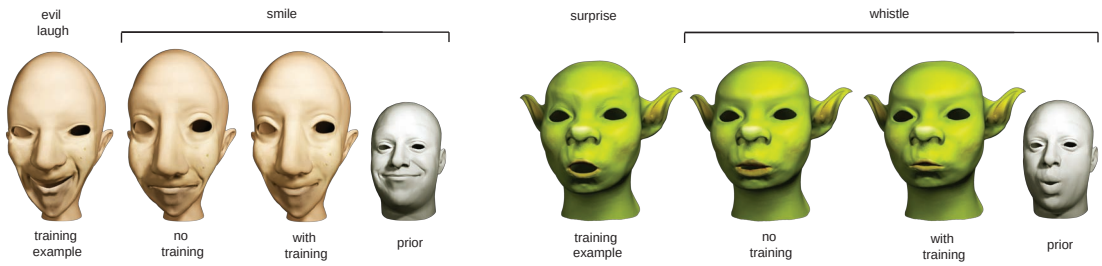


Figure 1.6: Example-based facial rigging allows transferring expressions from a generic prior to create a blendshape model of a virtual character. This blendshape model can be successively fine-tuned toward the specific geometry and motion characteristics of the character by providing more training data in the form of additional expression poses.

1.2 Motivating Applications and Impact

In addition to computer graphics, the methods unearthed through this dissertation have a wide range of applications and impact in other sciences and industries:

- **Robotics.** Computer vision systems of autonomous robots can benefit from faithful 3D reconstructions of dynamic shapes for a more complete understanding of scene events which may facilitate tasks such as interaction with humans. Additionally, our facial animation framework may support the development of lifelike humanoid robots where biomechatronic systems can be directly trained with accurately recorded human facial expressions.

- **Communication.** Compelling animated digital replicas of oneself provide new means for telepresence and virtual collaboration. For example, a full 3D footage of a virtual news correspondent who is stationed in a remote location can be directly broadcasted to the studio and interviews be conducted as if the person was actually there making the communication experience richer and more natural than conventional 2D videoconferences. A quasi-real-time holographic system for telepresence was recently introduced by Blanche and colleagues [BBV+10].

- **Medicine.** Physically accurate capture of human individuals in motion can aid physicians with surgery planning, improved medical diagnosis, and enable the design of advanced prosthetics. In oncology, when cancer patients undergo radiation therapies, the locations of pre-identified malignant tumors can be constantly updated using our reconstructions for accurate treatment.

- **Biology.** Biologists will have a powerful new tool for studying animals and complex ecosystems. For example, the shape and deformation of endangered animals can be digitized to provide compelling archives if they become extinct. Also, statistical analysis of humans can be used to explore shape changes as infants develop into adults.

- **Security.** Law enforcement agencies can benefit from digitized individuals for purposes such as criminal documentation: collecting motion biometrics for data-mining and surveillance services. In particular, geometric signatures (e.g., scars, tattoos, etc...) and motion patterns may help to identify suspicious persons.

- **Film Production.** Applications of our research carry over to feature film productions where real actors are being replaced by digital clones (c.f. Figure 1.7) and their performances captured at very high resolution without involving any markers. In addition, accurate pre-visualization of facial animations can be achieved with live feedbacks so that individual shots can be carefully planned before filming begins.

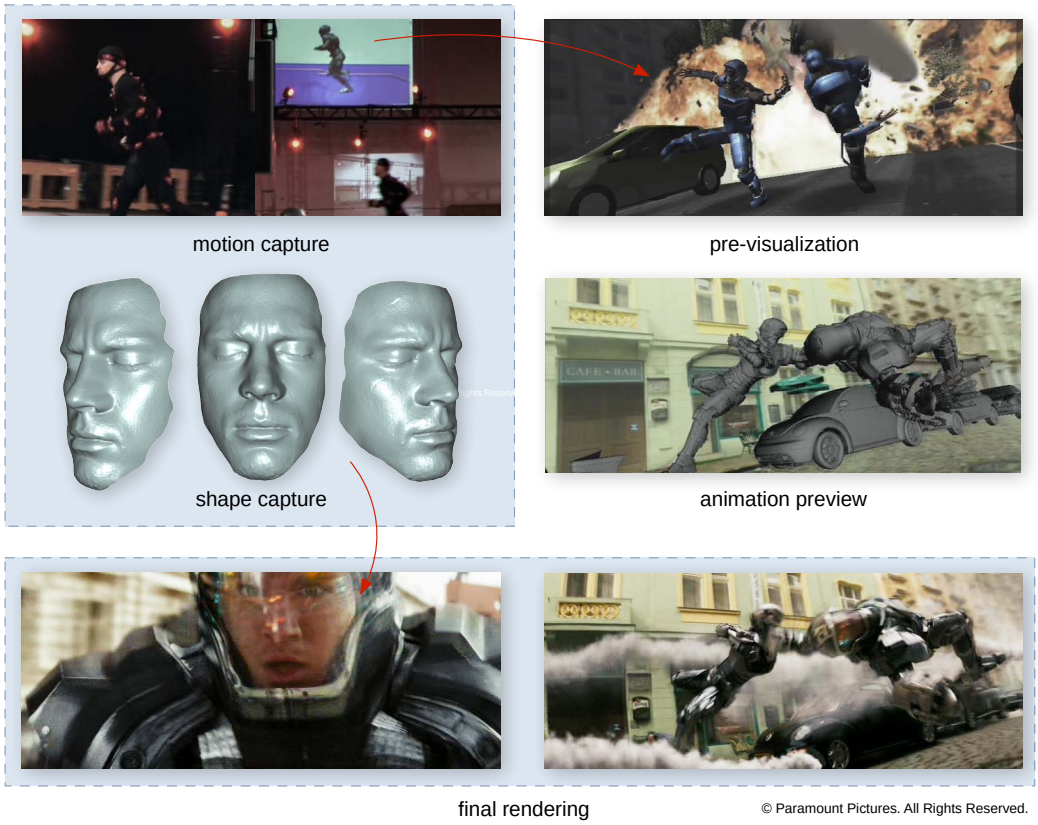


Figure 1.7: In feature films, actors are often replaced by digital doubles for shots that are impossible to realize. Capturing shape and motion from real actors is an important process to recreate compelling animated characters. Pre-visualization is becoming increasingly popular as an effective tool for planning and conceptualizing movie scenes.

1.3 Contributions

The principal contributions of this dissertation are:

- A non-rigid registration algorithm that automatically computes surface correspondences and a warping field between two partial scans of a deforming subject.

- A framework and algorithms for geometry and motion reconstruction of complex deforming shapes captured from only a single view of a real-time 3D scanner.
- A temporal-coherent shape completion technique for dynamic shapes captured using a multi-view acquisition system.
- A full integrated framework for markerless, real-time facial tracking and expression transfer to a different character's face using a structured light scanner.
- A method that automatically generates a facial blendshape rig for an input face model and personalizes them with user provided example expressions.

1.4 Organization

The remainder of this dissertation is organized as follows:

Chapter 2, Real-Time Data Capture Revisited. This chapter formalizes the notion of shape and motion in a discrete setting and provides an extensive overview of state of the art 3D acquisition techniques that are able to capture high-resolution scans of deforming subjects at "video" rates. We also describe several fundamental algorithms for post-processing of scanned data. Real-time 3D scanning is the first step for recovering high-quality shape and motion and provides the necessary input data for our *animation reconstruction* algorithms.

Chapter 3, Registration of Deformable Surfaces. In order to determine the motion of surface points, correspondences need to be established between partial data captured between two frames. This is equivalent to bringing a pair of 3D scans into alignment by warping one shape onto another. *Non-rigid registration* is a fundamental component for all reconstruction and tracking algorithms presented in the chapters ahead. Before introducing our novel registration algorithm, we begin this chapter with a comprehensive introduction to the subject of rigid registration, surface deformation, and non-rigid alignment.

Chapter 4, Dynamic Shape Reconstruction. This chapter covers a framework that simultaneously reconstructs detailed shape and motion of deforming objects captured from a single view. A robust *non-rigid registration* algorithm based on space-time adaptive deformation and techniques for effective detail propagation are presented here.

To deal with inevitable occlusions in multi-view acquisitions, we also introduce a *hole-filling* technique to obtain watertight temporally coherent meshes.

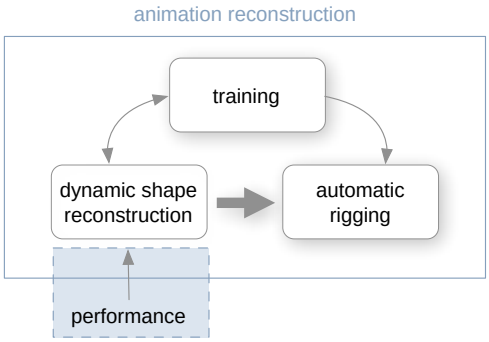
Chapter 5, Facial Animation Reconstruction. We introduce in this chapter a complete system for markerless, real-time facial expression tracking with transfer to the face of an arbitrary digital character. Efficiency is accomplished through a shift of costly computation toward offline pre-preprocessing. Furthermore, various specialized techniques for robust treatment of complex facial deformations are covered here.

Chapter 6, Directable Facial Animation. This chapter presents a scalable technique that automatically generates a personalized *facial rig* from a set of user provided example expressions. These examples may be both handcrafted or 3D scans of real actors. In particular, we demonstrate that the generated rig can be directly used for art-directable facial tracking.

Chapter 7, Conclusion and Future Directions. We summarize this dissertation with a few take-home messages and suggest ideas for future research.

2

Real-Time Data Capture Revisited



In animation reconstruction, captured dynamic data are the main driving force behind the creation of compelling animated digital models. This chapter formalizes the notion of shape and motion acquisition, and presents several state of the art techniques for real-time acquisition of deforming surfaces. Because of hardware and algorithmic restrictions, scans are typically affected by noise and outliers. To obtain clean input data, we will introduce several effective geometry post-processing tools. The goal is to produce high-resolution input data with negligible artifacts for our dynamic shape reconstruction stage (c.f., illustration on the right).

In general, we are interested in capturing the *shape* of a subject together with its *motion*. Traditionally, the relevance of the two aspects may vary depending on the nature of the problem. For example, when the purpose consists of cloning realistic digital human faces, both geometry and motion need to be captured at very high resolution. On the other hand, when retargeting the performance of an actor onto a different digital

character, the actor's geometry might not be required and sparsely captured motion data is often sufficient (e.g., when *skeleton rigs* are used). Hence, depending on the application, certain acquisition technologies may be more suitable than others.

Nevertheless, we argue that *data acquired at high spatial and temporal resolution can be pertinent for a wide range of purposes* other than recreating dynamic digital doubles. For instance, captured fine-scale deformations and second order dynamics (such as muscle jiggling) may be resynthesized onto other characters or used to produce large sets of dynamic shape priors for data-driven methods. The main advantage of using high resolution captured data over alternative animation techniques such as physical simulation or key-framing, is that realistic and complex surface dynamics come for free. While recent advances in 3D scanning facilitate the acquisition of detailed dynamic shapes, the motion is typically not given explicitly but can be robustly determined using *non-rigid registration* which we describe in more detail in Chapter 3.

As noted in Chapter 1, we focus on 3D range sensors that are able to continuously capture dense surface geometry at high frame rates. Although resolution and accuracy are constantly improving with each generation of new optical devices, image sensors, and scanning techniques, acquiring geometry remains an inverse problem and usually relies on a set of assumptions about the scanned subject and the scene. For example, stereo approaches generally require the shape of the subject to be locally continuous (for effective stereo matching). Methods with active illuminations often assume the surface reflectance to be close to *Lambertian* (i.e., free from specularities and non-linear color distortions). Hence, the scans are generally still affected by high-frequency noise and incomplete due to occlusions and non-cooperative surface materials. Even though multiple sensors can be placed around the subject to increase coverage, obtaining a hole-free mesh is generally not possible. Moreover, for interactive applications (e.g., live facial puppetry presented in Chapter 5), not only must the recording be in real-time, but a dense range map also has to be delivered instantly. In particular, passive stereo matching algorithms that involve costly off-line computations cannot be used in this scenario. Active illumination techniques such as structured light projection simplify the matching problem by changing the scene with a known signal. While these systems are able to generate a continuous stream of high-quality scans in real-time, they usually produce a strong distracting light and are often unsuitable for a multi-view setup due to light-interference.

Starting with Section 2.1, we formalize the concept of shape and motion, and describe acquisition as a mapping from a continuous to a discrete setting. Section 2.2 summarizes the most important acquisition techniques that are relevant in our animation reconstruction setting, namely 3D scanning with high spatial and temporal resolution. We compare these different approaches and discuss their advantages and disadvantages for different scenarios. Section 2.3 and 2.4 give a more detailed look into the two 3D scanners used in this work. After acquisition, we obtain a discretized 3D representation of dynamic shapes which are usually affected by noise and outliers. Section 2.5 presents basic tools for effective representation and post-processing of these data, such as outlier removal, Laplacian smoothing, and isotropic remeshing.

2.1 Formalizing Shape, Motion, and Acquisition

This section introduces a formal specification of our input data and their properties. We start by describing the notion of *shape, motion, and temporal correspondences* of the scanned subject using concepts from differential geometry. During acquisition, only *exterior surfaces* are observable. Hence, we dedicate a section discussing *topology changes* for these surfaces. While real-world performances take place in a continuous setting, our *captured depth maps* are discretized as well as incomplete and noisy. Here, we illustrate how real-time 3D sensors *sample* the dynamic surface and how temporal correspondences are lost during acquisition. In particular, we will define *overlapping regions* between scans of deforming objects which will play a central role for correspondence computation and non-rigid registration.

2.1.1 Scanned Subject

Shape and Motion. We describe the shape of a dynamic object as an orientable time-varying *two-manifold surface* $\mathcal{M}(t) \subseteq \mathbb{R}^3$ possibly with boundaries and t as the time axis (c.f., Figure 2.1). In particular, spatial local parameterization $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^2$ exists at any instance in time around each point $\mathbf{p}(\mathbf{u}, t)$ (c.f., DoCarmo [dC76] and Lee [Lee00]). Later on, we will discover that the notion of local parameterization will play a central role for surface processing algorithms which rely on the existence of tangent planes and (infinitesimal) *local geodesic neighborhoods*, and also for optimizations that are based on continuous surface representations. When the subject deforms, the position of a surface point $\mathbf{p}(t) \in \mathcal{M}(t)$ describes a continuous trajectory in the space-time domain. Each time curve $\mathbf{p}(t)$ characterizes a global continuous *motion* of a surface point. In particular, we

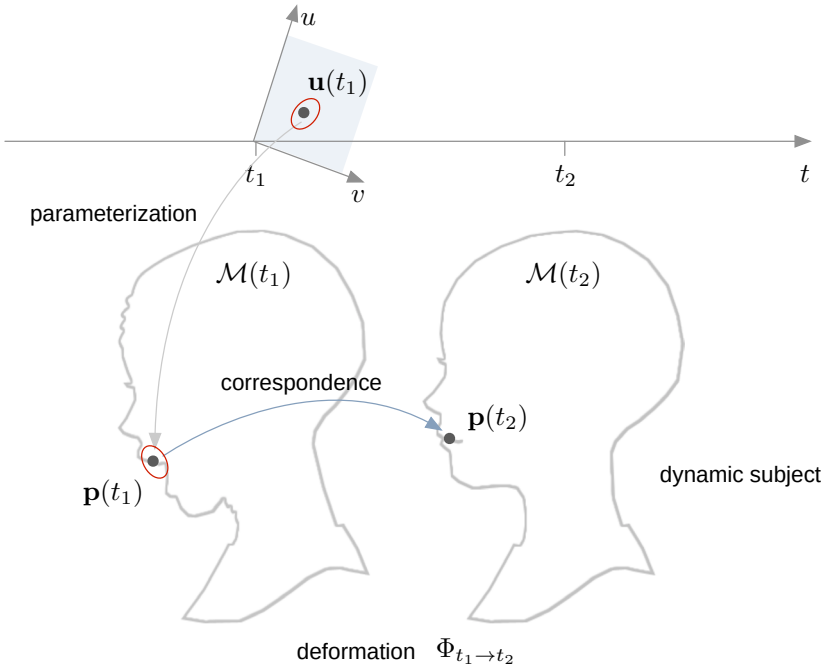
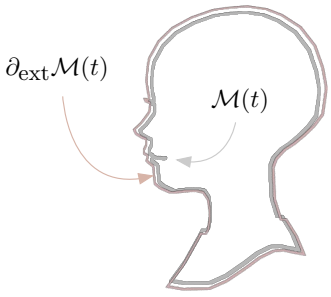


Figure 2.1: We represent the dynamic shape of a deforming object as a two-manifold embedded in a space-time continuum. Every surface point and its local neighborhood can be mapped from a parametric domain in $\mathbb{R}^2$ and has a corresponding point at any instance in time. The continuous mapping between both shapes is called deformation.

assume $\mathbf{p}(t)$ to be a $\mathcal{C}^\infty$ curve, but it is not *regular* as stationary surface points have vanishing derivatives. We call $\mathbf{p}(t_1)$ and $\mathbf{p}(t_2)$ *pairwise temporal correspondences* between two time instances t_1 and t_2. In the most general sense, we define the surjective and continuous mapping $\Phi_{t_1 \to t_2} : \mathcal{M}(t_1) \to \mathcal{M}(t_2)$ as the *deformation* (or *warping*) of $\mathcal{M}(t)$ from t_1 to t_2 where $\Phi_{t_1 \to t_2}(\mathbf{p}(t_1)) = \mathbf{p}(t_2)$. In animation reconstruction neither temporal correspondences nor the deformations are known in advance.

Topology. While $\mathcal{M}(t)$ can be of arbitrary genus $\mathcal{G}$, we only consider surfaces of solid matter where $\mathcal{G}$ remains constant through time (as opposed to liquid state objects for instance). Although most real objects may be represented by multiple two-manifolds that are *homeomorphic* to disjoint sets in parametric domains, recovering these separate manifolds is non-trivial. Con-



sider the example when two finger tips are touching: while in reality the hand $\mathcal{M}(t)$ would have a genus $\mathcal{G} = 0$, the only genus that can be deduced from an observable exterior surface $\partial_{\text{ext}}\mathcal{M}(t)$ is $\mathcal{G} = 1$. In addition, observations are typically incomplete due to occlusions which makes it even harder to extract the correct number of disconnected objects and their topologies. To simplify the problem, we typically consider any subject as a single connected manifold with predefined topologies (e.g., Section 4.2) which we will refer later as the *template* $\mathcal{T}(t)$. Many objects however may consist of multiple disconnected surfaces and cannot be represented by a single connected manifold (e.g., gliding surface sheets on human skin). To model these shapes, we will propose a technique in Section 4.3 that skips the requirement of using a prior template and facilitates modeling with a single connected manifold surface by allowing the shape to change its topology over time. In particular, we assume the subject to be simply represented by its exterior points where $\mathbf{p}(t) \in \partial_{\text{ext}}\mathcal{M}(t)$. In this case, surface points might not have temporal correspondences at certain time frames. In this case, the points that lie in the inside mouth region for example (c.f., Figure 2.1) would disappear when the mouth is closed.

2.1.2 Captured Data

Incomplete Scans. Ideally we would like to capture the entire manifold surface $\mathcal{M}(t)$ at any time instance t, i.e., recovering all temporal correspondences and deformations. Unfortunately, the continuous shape representation gets partly lost during the optical acquisition process. In general, only a non-occluded subset of the exterior surface $\partial_{\text{ext}}\mathcal{M}(t)$ can be acquired. We consider the subset $\mathcal{S}(t) \subseteq \partial_{\text{ext}}\mathcal{M}(t)$ as the *scanned* manifold surface visible to the sensors at time t. The amount of surface regions that can be captured also depends on the underlying scanning technology. For example, multi-view stereo approaches can only capture shapes that have sufficient surface albedo and are simultaneously visible in at least two pairs of sensors. When the scene is illuminated, shadows created by light sources also need to be taken into account. An extensive discussion of visibility issues and scan configurations for optical scanners can be found in Li [Li05] and Curless [Cur97]. We measure *surface area* of $\mathcal{S}(t)$ by integrating the length of the normal of each point $\mathbf{s}(\mathbf{u}, t) \in \mathcal{S}(t)$ over the *scan parameterization* region $\mathcal{U}_{\mathcal{S}}$:

$$A(\mathcal{S}(t)) = \int\int_{\mathcal{U}_{\mathcal{S}}} \|\mathbf{s}_u(\mathbf{u}, t) \times \mathbf{s}_v(\mathbf{u}, t)\|_2 \, \mathrm{d}u \, \mathrm{d}v \tag{2.1}$$

with $\mathbf{s}_u(\mathbf{u}, t)$ and $\mathbf{s}_u(\mathbf{u}, t)$ the partial derivatives in u and v directions respectively.

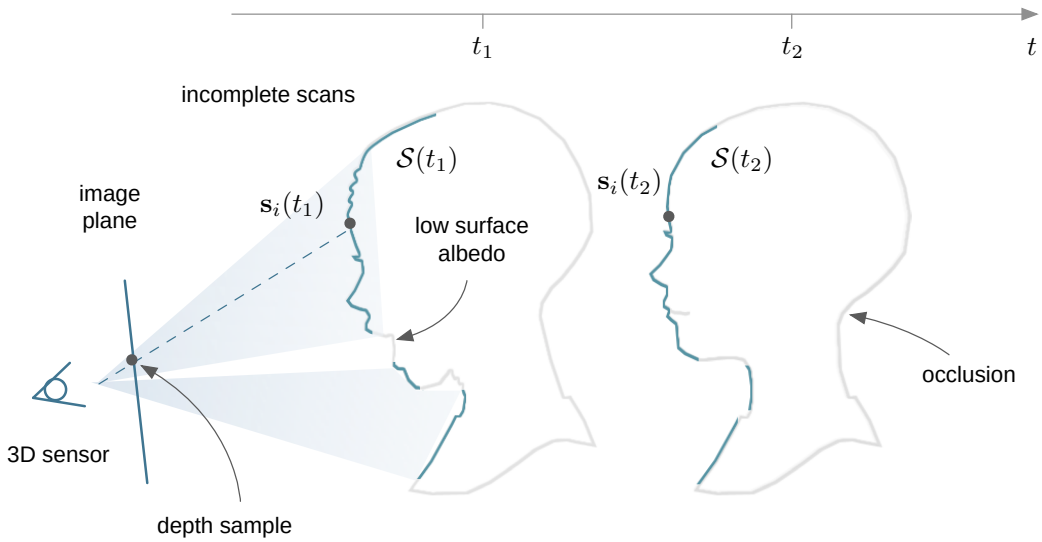


Figure 2.2: While the subject deforms, a 3D sensor captures the exterior surface $\partial_{\text{ext}}\mathcal{M}(t)$ and produces depth samples on the image plane. The resulting scans are typically surface samples $\mathbf{s}(t)$ that are incomplete due to occlusions and low surface albedo. Furthermore, the discretized depth samples are usually affected by quantization errors, noise, and outliers.

Spatial Discretization. We now describe how a continuous scan $\mathcal{S}(t)$ becomes discrete after acquisition. W.l.o.g., we consider a single-view acquisition setup which observes a sequence of continuous *depth maps* $f_s : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ of a deforming subject in real-time. In particular, $f_s(\mathbf{u}, t)$ is the depth measured at time t and position $\mathbf{u}$ on the image plane. For unobserved surface samples we set $f_s(\mathbf{u}, t) = \infty$. Note that $\mathcal{S}(t) = \{f_s(t) : \mathbb{R}^2 \to \mathbb{R} \setminus \{\infty\}\}$. Because digital sensors have finite resolutions, the actual captured depth map is a discretized two-dimensional sampling $f_d^t : \mathbb{R}^2 \to \mathbb{R}$ with step size l_u and l_v in u and v-direction respectively. We obtain:

$$f_d^t(\mathbf{u}) = f_s(\mathbf{u}, t) \sum_{m=0}^{N_u} \sum_{n=0}^{N_v} \delta(u - ml_u, v - nl_v) \tag{2.2}$$

with $\delta(\mathbf{u})$ a two-dimensional impulse function and $(N_u - 1) \times (N_v - 1)$ the image resolution. Hence, in order to capture the full geometric details of a continuous shape $\mathcal{S}(t)$ the sampling frequencies $\nu_u = \frac{1}{l_u}$ and $\nu_v = \frac{1}{l_v}$ must be at least twice as large as the horizontal and vertical frequencies of $f_s(t)$—satisfying the *Nyquist criterion*. When higher frequency details are present, the captured depth map $f_d^t(\mathbf{u})$ may exhibit artifacts be-

cause of *aliasing* and *quantization noise*. Aliasing can be suppressed by low-pass filtering the discretized shape (c.f., Section 2.5). Note that for most scanning systems based on image sensors (CCD or CMOS), each light detector captures more than the intensity of a single point $i(\mathbf{s}(t))$ because of diffraction, lens aberration, and inaccurate focussing. While this side-effect results in a slight deviation from an ideal-sampling, we obtain a natural low-pass filtering which is characterized by the so called *point spread function* of an optical system.

While the two-manifold surface of a depth map captured from a single view describes a discrete graph $\mathbf{s}(\mathbf{u}) = [\mathbf{u}, f_d^t(\mathbf{u})]^\top \in \mathbb{R}^3$, the integration of $2.5D$ shapes obtained from multiple views at time t becomes a dense *point cloud* of surface samples $\mathbf{s}_i(t) \in \mathcal{S}(t)$ as illustrated in Figure 2.2 with $i = (1, \ldots, N)$. When concatenating point samples obtained from multiple views, overlapping regions will have a denser sampling and thus capture more details. The sampling density in those regions is no longer measured by uv-step sizes but the average distance to point samples lying in the one-ring neighborhood (in case connectivity is given as for meshes) or k-nearest neighbors for point clouds (c.f. Section 2.5). Note that we can also compute a parameterization for $\{\mathbf{s}_i(t)\}_i$ as long as it remains a two-manifold surface [HLS07].

Temporal Discretization. Analogous to the spatial domain, we discretize the captured range map over time t as follows:

$$f_d(\mathbf{u}, t) = f_s(\mathbf{u}, t) \sum_{k=0}^{N_t} \sum_{m=0}^{N_u} \sum_{n=0}^{N_v} \delta(u - ml_u, v - ml_v, t - kl_t) \tag{2.3}$$

with N_t the length of the recording and $\nu_t = \frac{1}{l_t}$ the frame rate. In a real-time setting we typically assume $\nu_t > 25$ Hz. Note that for a sample point $\mathbf{s}_i(t_1) = \mathbf{p}(t_1)$ observed at t_1 it generally holds that $\mathbf{s}_i(t_2) \neq \mathbf{p}(t_2)$ as both points might not correspond. Hence, the motion of a surface sample $\mathbf{s}_i(t_1)$ can only be determined if a $\mathbf{s}_j(t_2)$ exists such that $\mathbf{s}_j(t_2) = \mathbf{p}(t_2)$. Because of possible topological changes in $\partial_{\text{ext}}\mathcal{M}(t)$, such a corresponding point $\mathbf{s}_j(t_2)$ might not even exist. Hence, the subset of $\mathcal{S}(t_1)$ that guarantees valid existing corresponding points $\mathbf{s}_j(t_2)$ is defined as:

$$\mathcal{S}_{\exists t_2}(t_1) = \{\, \mathbf{s}_i(t_1) \mid \mathbf{s}_i(t_1) \in \mathcal{S}(t_1) \wedge \Phi_{t_1 \to t_2}(\mathbf{s}_i(t_1)) \in \partial_{\text{ext}}\mathcal{M}(t_2) \,\} \quad . \tag{2.4}$$

Furthermore, we define the *overlapping* region $\mathcal{S}_{t_1 \cap t_2}$ at time t_1 between both scans $\mathcal{S}(t_1)$ and $\mathcal{S}(t_2)$ as follows:

$$\mathcal{S}_{t_1 \cap t_2}(t_1) = \{\mathbf{s}_i(t_1) \mid \mathbf{s}_i(t_1) \in \mathcal{S}(t_1) \wedge \Phi_{t_1 \to t_2}(\mathbf{s}_i(t_1)) \in \mathcal{S}(t_2) \,\} \subseteq \mathcal{S}(t_1) \quad . \tag{2.5}$$

Within this region, one-to-one surface correspondences exist between $\mathcal{S}(t_1)$ and $\mathcal{S}(t_2)$ and $\Phi_{t_1 \to t_2} : \mathcal{S}(t_1) \to \mathcal{S}(t_2)$ is *surjective* as multiple source points can be warped to the same position. Therefore, it follows that $\Phi_{t_1 \to t_2}(\mathcal{S}_{t_1 \cap t_2}(t_1)) = \mathcal{S}_{t_1 \cap t_2}(t_2)$ and $\Phi^{-1}_{t_1 \to t_2}(\mathbf{p}(t_2)) = \Phi_{t_2 \to t_1}(\mathbf{p}(t_2))$.

Correspondence Problem. We may now define a *pairwise correspondence problem* between $\mathcal{S}(t_1)$ and $\mathcal{S}(t_2)$ as the task of determining a one-to-one assignment for all samples $\mathbf{s}_i(t_1) = \mathbf{p}_i(t_1) \in \mathcal{S}_{t_1 \cap t_2}(t_1)$ where $i = 1, \ldots, N$ to the closest sample $\mathbf{s}_j(t_2)$. Note that $\mathcal{S}_{t_1 \cap t_2}(t_1)$ is generally not known in advance and needs to be determined as part of the pairwise correspondence computation. In Chapter 3, we will present *pairwise non-rigid registration* algorithms which, in addition to solving pairwise correspondences, compute all deformations $\Phi_{t_1 \to t_2}(\mathbf{s}_i(t_1))$ for $i =, \ldots, N$, provided $\mathbf{s}_i(t_1) \in \mathcal{S}_{\exists t_2}(t_1)$.

Captured Shape and Motion. Suppose we successfully compute non-rigid registration for time t_1. We obtain an *accumulated* shape represented by the samples $\{\mathbf{s}_i(t_1)\}_i \cup \{\Phi_{t_2 \to t_1}(\mathbf{s}_j(t_2))\}_j$. Motion can be represented by a dense motion displacement field $\{\mathrm{d}\mathbf{s}(t_1)\}$ with time step $\mathrm{d}t \approx t_2 - t_1$. For each original samples $\mathbf{s}_i(t_1)$ we obtain an instantaneous 3D velocity vector

$$\frac{\mathrm{d}\mathbf{s}_i(t_1)}{\mathrm{d}t} \approx \frac{\Phi_{t_1 \to t_2}(\mathbf{s}_i(t_1)) - \mathbf{s}_i(t_1)}{t_2 - t_1} \tag{2.6}$$

and for the accumulated ones $\Phi_{t_2 \to t_1}(\mathbf{s}_j(t_2))$, velocity is given by

$$\frac{\mathrm{d}\mathbf{s}_j(t_1)}{\mathrm{d}t} \approx \frac{\mathbf{s}_j(t_2) - \Phi_{t_2 \to t_1}(\mathbf{s}_j(t_2))}{t_2 - t_1} \quad . \tag{2.7}$$

When the subject undergoes a globally rigid motion we may express $\Phi_{t_2 \to t_1}$ as a simple Euclidean transformation Φ_{rigid} with rotation matrix $R \in \mathrm{SO}(3)$ and translation vector $\mathbf{t} \in \mathbb{R}^3$. Hence, for all $i = 1, \ldots, N$:

$$\Phi_{\text{rigid}}(\mathbf{s}_i(t_1)) = R\,\mathbf{s}_i(t_1) + \mathbf{t} \quad . \tag{2.8}$$

In particular, we may consider a global velocity field that is decomposed into a *rotational* and a *translational* component. Let us suppose that $\mathbf{s}_j(t_2) = \Phi_{\text{rigid}}(\mathbf{s}_i(t_1))$. The instantaneous velocity vector field of a rigid motion follows immediately from Equation 2.8 and is linear (c.f. [Bot79]):

$$\frac{\mathrm{d}\mathbf{s}_i(t_1)}{\mathrm{d}t} \approx \frac{\mathrm{d}\mathbf{s}_j(t_2)}{\mathrm{d}t} = \frac{\mathrm{d}R}{\mathrm{d}t}\,\mathbf{s}_i(t_1) + \frac{\mathrm{d}\mathbf{t}}{\mathrm{d}t} = \mathbf{w} \times \mathbf{s}_i(t_1) + \frac{\mathrm{d}\mathbf{t}}{\mathrm{d}t} \tag{2.9}$$

with $\mathbf{w}$ the *angular velocity tensor* and $\frac{\mathrm{d}\mathbf{t}}{\mathrm{d}t}$ the *translational velocity*.

Noise and Outliers. All stages in an optical 3D acquisition pipeline (from hardware calibration, scan configuration, scene geometry, surface properties, optical device, imaging sensor to scanning algorithm) can lead to measurement inaccuracies and produce noise and outliers in the scans $\mathcal{S}(t)$.

In a real-time setting, where the subject is moving, the problem of noise becomes even more prominent as a full scan has to be accomplished within milliseconds. In addition to imaging problems (e.g., short exposure, motion blur...), scanning methods that require multiple shots can only use limited frames and have to deal with deformations of the subject in the acquisitions.

Depending on the scanning technique the amount and distribution of noise can vary. As described in [HLP93], noise is often being modeled as an ellipsoidal distribution function with principal axis in the direction of the sensor's reference viewpoint. Outliers may also be modeled as samples which uncertainty ellipse does not intersect with the ray of sight. Consequently, we can incorporate measurement inaccuracies in the definition of captured depth map as follows:

$$\hat{f}_d(\mathbf{u}, t) = f_d(\mathbf{u}, t) + \epsilon_n(\mathbf{u}, t) + \epsilon_o(\mathbf{u}, t + \epsilon_s(\mathbf{u}, t) \tag{2.10}$$

where $\epsilon_n(\mathbf{u}, t)$ and $\epsilon_o(\mathbf{u}, t)$ are noise and outlier functions respectively. In many acquisition system we might observe an additional *structured* noise term $\epsilon_s(\mathbf{u}, t)$ that correlates over space and time. For example, scans produced by phase-shift methods [HZ06, ZH04, WLG07] typically exhibit unwanted vertical lines that remain over several frames for fast motions in z-direction (*temporal aliasing*). While it might be reasonable to assume $\epsilon_n(\mathbf{u}, t)$ to be *normal* distributed (*Gaussian noise*), modeling the statistical occurrence of $\epsilon_o(\mathbf{u}, t)$ is not straightforward and highly depends on the acquisition method. For the remaining of this dissertation, we assume that surface samples captured from a single-view are discretized as $\hat{f}_d(\mathbf{u}, t)$.

2.2 Dynamic Shape Acquisition Techniques

With our formal specification of shape, motion, and acquisition, we now explore different methodologies for real-time 3D capture. For an extensive survey on static scanning, we refer the reader to the following literature [Rus01, Cur97, SS01, Li05, SCD+06b]. The focus herein is on the real-time aspect and we propose a taxonomy that mainly distinguishes between marker-based and markerless methods as illustrated in Figure 2.3. For most techniques geometry is obtained through optical triangulation or

time-of-flight. In general, tracking is either performed in 3D after geometry acquisition or on each 2D sensor independently.

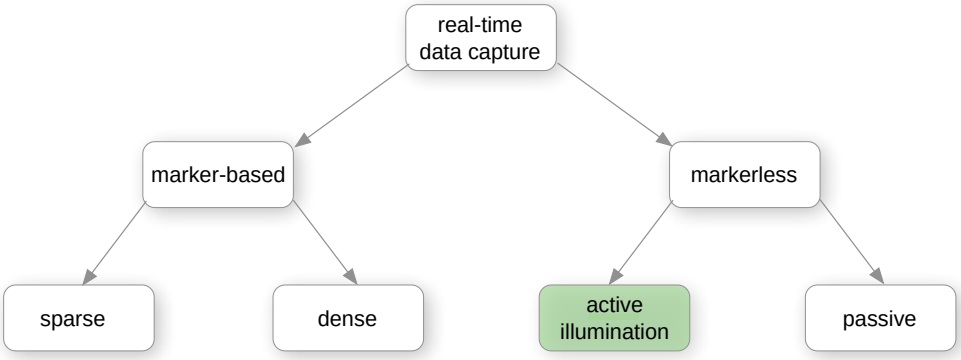


Figure 2.3: A taxonomy for real-time data capture. The aim is to facilitate shape and motion acquisition. Our work focuses on input scan sequences obtained from active illumination techniques.

For marker-based approaches, shape and motion are recovered by either placing landmarks on the subject or by manipulating surface properties other than geometry (typically texture). The advantage is that motion is immediately deduced by tracking the markers. Three-dimensional positions of the markers are typically obtained through optical triangulation but other techniques exist (e.g., inertial sensors). Because the subject is augmented with easily detectable *tracking features*, marker-based approaches are independent of surface textures and less susceptible too poor illumination conditions. Consequently, they are still widely accepted as the method of choice for pure motion capture purposes due to their robustness and accuracy. Nevertheless, using markers is unsuitable for simultaneous acquisition of dynamic shape and its original texture. Moreover, they typically involve expensive studio equipments and a time-consuming set up process (marker-calibration, applying make-up...). While technologies using sparse markers have limited geometry resolution, state-of-the-art dense marker based techniques can produce highly compelling results but are less reliable.

Markerless acquisition systems are divided into active and passive approaches. In both cases, real-time data capture can be described as in Section 2.1.2. Active techniques facilitate geometry acquisition by controlling illumination in a scene. Although shape is being continuously captured, no explicit correspondence information is available. Dense motion is typically recovered using non-rigid registration between consecutive 3D scans,

and/or by tracking features from the image recording. Passive acquisition methods are the least restrictive in the sense that no special light emission device is required and the subject does not need to wear markers. Another characteristic is that a single shot is usually sufficient for acquisition. Hence, passive methods are inherently suitable for multi-view and dynamic shape acquisition (provided exposure is sufficiently short). However, they are often less accurate and robust than active methods as they typically rely on the texture quality of the subject and lighting conditions. For example, the geometry of a diffuse lit surface that does not have any textures cannot be resolved. Most sophisticated passive methods (with global matching and occlusion handling) are also unable to produce high-resolution scene geometry in real-time as they generally involve prohibitively high computational costs [BBH03].

Regardless of versatility and cost, we are mainly interested in systems that are able to deliver high-resolution input scans (possibly textured) with minimal noise and outliers. In addition, we wish to process long 3D scan sequences that, in some cases, need to be produced in real-time. Less reliable technologies that require time-consuming manual data clean-up are therefore not suitable. Currently, the only techniques that satisfy all our requirements are markerless approaches based on active illumination. Nevertheless, recent progress in stereo algorithms [BBB+10, BHPS10] are likely to unleash the potential of passive systems as future game-changers.

Sparse Marker. Motion capture systems that track a *sparse* set of markers are characterized by their exceptional robustness and accuracy. The idea is to facilitate tracking by placing a set of markers $\mathbf{m}_i(t)$ with $i = 1, \ldots, N_m$ that maximizes invariance to the subject's texture and scene illumination. As opposed to markerless approaches, correspondences between $\mathbf{m}_i(t_1)$ and $\mathbf{m}_i(t_2)$ are easier to establish. Moreover, they usually employ sensors with very fast update rates resulting in highly accurate approximation of instantaneous velocity

$$\frac{\mathrm{d}\mathbf{m}_i(t_1)}{\mathrm{d}(t)} \approx \frac{\mathbf{m}_i(t_2) - \mathbf{m}_i(t_1)}{t_2 - t_1} \quad .$$

In the past two decades, several motion capture technologies have been proposed. An exhaustive survey of the most important techniques can be found in [MAB92, HB01, WF02].

Marker-based systems that use optical sensors [Wol74] achieve sub-millimeter accuracy but can only be detected if the markers are not (self-) occluded, i.e., if $\mathbf{m}_i(t) \in$

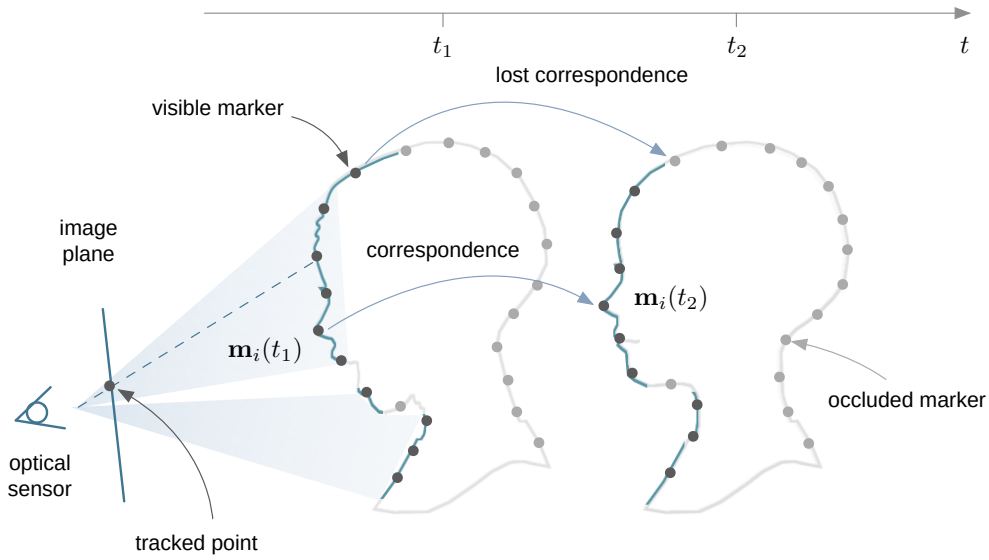


Figure 2.4: Dynamic data acquisition using sparse markers and optical sensors. Several non-optical motion capture systems do not lose track of correspondences but are less accurate than optical ones.

$\mathcal{S}(t)$. The precise positions of visible $\mathbf{m}_i(t)$ are typically computed through optical triangulation using images recorded from multiple views. Because of indistinguishable markers and occlusions, initial 3D positions reconstruction is often aided by a skeleton model and a rigid link assumption. To capture the motion, frame-to-frame marker correspondences can be either based on *tracking* or *identification*, depending on the technology. Markers ranging from painted dots [Wil90, cLO05] to *passive* retroreflective markers [Vic] and *active* light-emitting diodes (*LEDs*) may use high-speed acquisitions (several hundreds frames per second) and/or motion constraints (e.g., underlying skeleton) to disambiguate between identical looking markers. These methods are based on tracking and thus, "marker swapping" cannot be fully avoided. Markers are often used to extract skeletal motion. However, when several hundreds of markers are tracked, a rough geometry of the surface can be extracted together with highly accurate dynamics as demonstrated by Park and Hodgins [PH06]. In addition to being beneficial for long range acquisitions, active markers such as LEDs can be time modulated [Pha] in order to emit a unique signal to resolve the correspondence problem via identification. However, only limited number of LEDs can be uniquely identified and attaching them on certain surfaces can be difficult (e.g., faces, cloth...).

Non-optical marker-based systems such as inertial, magnetic, or acoustic systems are not restricted by marker visibility issues. In particular, the markers $\mathbf{m}_i(t)$s are fully visible anywhere on $\mathcal{M}_i(t)$ for any t. However, the main issue with these systems is the lack of precision. Inertial motion capture systems use gyroscopes or accelerometers to measure rotation on articulated joint angles from previous frames without direct 3D position computation [MJKM04]. As a result, measurements accumulate errors over extended time periods. Marker-based techniques based on magnetic systems [Asc] calculate position and orientation by relative magnetic flux on a transmitter and receiver. The accuracy of these systems is highly sensitive to the presence of metallic materials. Acoustic systems determine range using time-of-flight of an ultra-sonic signal [WJH97]. Nevertheless, pulses emitted by ultrasonic beacons are susceptible to deflections due to obstacles resulting again in lost of accuracy.

Dense Markers. We speak of dense markers, when carefully designed geometry independent properties such as surface textures are densely applied on the subject's surface to improve shape reconstruction and motion tracking. So far, only optical acquisition systems are used in connection with dense markers. Hence, they also suffer from occlusions but are generally less susceptible to low surface albedo than markerless methods. White and coworkers [WCF07] capture the shape of complex deforming cloth using custom color patterns directly printed on the cloth. The texture consists of densely tessellated triangles with random colors that maximize entropy per captured pixel. Using a custom texture with markers that are easy to distinguish facilitates optical triangulation as correspondences between multiple views can be determined with a known reference parametric domain.

In practice, these type of patterns are unsuitable for surfaces such as human skin. Several examples in Furukawa and Ponce [FP08, FP09a] adapted the idea of using "unstructured" stipple make-up, pioneered by Mova LLC [Mov], that are applied on faces. The method consist of exploiting high-frequency noise for more reliable multi-view stereo matching (dense shape reconstruction) and temporal correspondences (detailed motion recovery). Their approach captures shape and motion in three-steps. First, a multi-view stereo algorithm is used to reconstruct high-resolution input scans. The second stage uses a rigid motion model to locally align a large set of small surface patches with the scan of the next frame. Because erroneous motion estimates are likely to occur, the final step regularizes the overall deformation using a global non-rigid deformation

model. Due to the greedy nature of their inter-frame correspondences, the approach highly depends on the photo-consistency of local surface patches textures.

While these methods are highly reliable and accurate for capturing realistic deformations, simultaneous capture of dynamic shape and the subject's original texture is difficult if the pattern obstructs with the latter. To address this problem, Mova LLC [Mov] proposes a real-time acquisition system that applies fluorescent make-up on an actor's face (or cloth). The random patterns are only visible under fluorescent light (see illustration). The capture process consists of rapidly switching between pure UV-light and illumination in the visible spectrum. In this way, both problems, tracking and reconstruction, are facilitated while simultaneous capture of skin texture is possible.

The input data captured with dense marker-based systems are suitable for most of our animation reconstruction algorithms. Because shape acquisition is typically based on techniques used for passive markerless acquisition (enriched with evenly distributed dense texture features), they also involve a time-consuming off-line reconstruction process for high-resolution data and are not yet suited for interactive applications on commodity hardware. Nevertheless, due to the rapid growth of (consumer-level) graphics accelerators, it is likely that real-time performance is achievable in the near future.

Active Illumination. Active optical acquisitions are markerless and use controlled illuminations for geometry capture without (fully) relying on surface texture. Surface motion is typically recovered by first sampling the shape of each frame (i.e., determining $\{\mathbf{s}_i(t)\}_i$), followed by non-rigid registration between consecutive shapes yielding the velocity vectors $\mathrm{d}\mathbf{s}_i(t)/\mathrm{d}t$ as described in Equation 2.6 and 2.7. Different methodologies are used to determine the range. As opposed to static acquisition, sufficient information must be captured within a very short time (usually $t_2 - t_1 \leq 40$ ms) to reconstruct all samples covering $\mathcal{S}(t)$. Fast *multiple shots techniques* can also be used for real-time acquisition, but usually use very few frames for reconstruction and assume that motion within the frames are negligible. The most established avenues to compute continuous high-resolution depth maps are either based on optical triangulation, shape from shading, or time-of-flight, or combinations thereof.

Optical triangulation techniques sample the subject's geometry by intersecting rays of corresponding points between two cameras or between a camera and a light emitted as depicted in Figure 2.5. While the principles of camera-to-camera triangulation are the same as in passive acquisition, *active-stereo techniques* [SS01, DRR03,

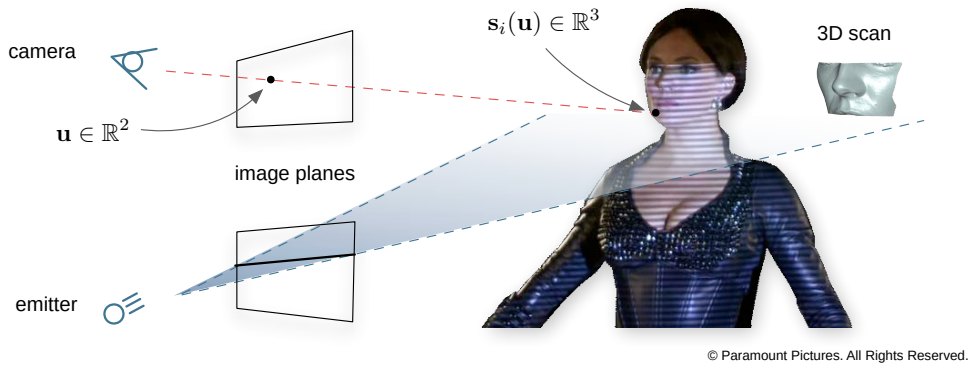


Figure 2.5: Camera-to-projector triangulation. When stripe patterns are projected as with phase-shift, a surface sample $\mathbf{s}_i$ is obtained by simple ray-plane intersection.

ZSCS04, Art, Kin] enrich the scene with the projection of detailed textures (often random noise) to facilitate stereo matching. The projected patterns are also called *unstructured light*. Real-time camera-to-projector triangulation techniques typically project a set of carefully designed stripes on the subject where 3D positions on captured stripes can be easily obtained by ray-plane intersection. As opposed to active-stereo methods, correspondences are determined between a known projected pattern and a camera (*structured light techniques*) which in many cases can be computationally more efficient [RHHL02, ZCS02, LSP06]. However, matching becomes ineffective for subjects with unknown and highly non-linear surface reflectance properties. Due to the limited resolution of projected patterns, a common technique (*three-phase shifting*) consists of projecting at least three sine wave modulated at different phases in order to resolve higher resolution subpixel accurate correspondences through phase unwrapping [HZ06]. The main disadvantage of solely relying on this method is that large depth discontinuities in the subject can yield outliers due to wrong correspondences. As shown in [ZH04, WLG07], phase unwrapping can be combined with active stereo, yielding accurate high-resolution shape reconstructions in real-time. Because of limited depth of field and diminishing light-levels with increasing distances, the working volumes of projector-camera systems are often limited to 1 m^3, hence not suitable for full-body performances. Moreover, because of possible light interferences, structured light scanners are not always suitable for acquisitions with many view-points. Nevertheless, we demonstrate in this work how single-view reconstruction of geometry and motion is possible using the low-cost structured light system of Weise and coworkers [WLG07]. Although surface textures can be

captured simultaneously, a strong distracting light needs to be projected in the scene, making those systems not always favorable for practical performance capture. Recent developments in high-speed IR emitters have enabled high-quality structured light acquisition with imperceptible light. Several systems such as the LogicDP projectors [Log] or Microsoft's *Kinect* 3D camera are readily available for the consumer market.

Another active range measuring technology that uses light invisible to the human visual spectrum are time-of-flight cameras [LSBS99, MES] . These cameras use LEDs to send a near IR light pulse to the subject and calculate the traveled distance by measuring the time difference between pulse emission and reception. Moreover, multiple time-of-flight sensors that are modulated at different frequencies can operate simultaneously. Although current systems lack in accuracy and resolution (320×240) due to their weak signal in the presence of background illumination, methods that use multiple slightly displaced shots were introduced to reduce noise and achieve super-resolution [STDT08]. The quality of scans produced by current time-of-flight scanners are still much inferior to those captured with active optical triangulation techniques, but we expect their resolution and accuracy to improve in the near future.

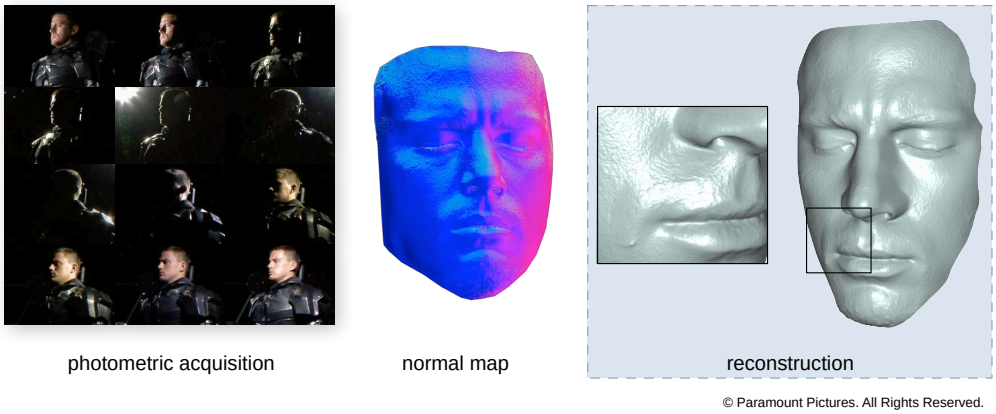


Figure 2.6: Photometric stereo estimates surface normals of an object observed under different lighting conditions. High-resolution reconstructions (down to the pore level) can be reliably obtained as shown on the right.

Photometric stereo techniques [Woo89] use light sources emitted from different directions to estimate surface normals. Assuming Lambertian surface, dense local surface orientations can be reliably determined by solving a simple linear system (c.f. Figure 2.6).

Surface shape can then be obtained using normal integration by adding depth constraints (e.g., on the occlusion boundaries [HVB+07]). Another approach consists of solving an linear optimization problem that uses soft constraints from an initial geometric prior such as the visual hull [VPB+09] or structure from motion [ZCHS03]. A survey on photometric stereo has been recently conducted by Barsky and coworkers [BP03]. While extremely high-frequency details such as pores can be reliably recovered, low frequency geometry is typically biased and inaccurate. For this reason, a more accurate but lower resolution geometry is often reconstructed separately [MHP+07, MJC+08, LLV+10] and then augmented with dense normal maps using the method proposed by Nehab and colleagues [NRDR05]. For fast acquisition, Hernandez and colleagues [HVB+07] propose a *one-shot technique* that simultaneously emits colored lights from different positions. However only surfaces with uniform albedo can be recovered. The *Light Stage* acquisition system described in [MJC+08, CEJ+06, VPB+09] uses synchronized and fast time-multiplexed lighting as well as high-speed cameras for recording. To our knowledge this is currently the only practical solution for real-time acquisition based on shape from shading. The advantages of such systems is the ability to capture high-resolution dynamic shapes from multiple views with a large working volume. As shown in [MHP+07], geometry acquired using normal maps can have superior surface details than laser scans. This dissertation will present a technique for multi-view shape completion of dynamic shapes from data captured using the technique presented in [VPB+09].

Passive Acquisition. Depth measurements with passive acquisition techniques are mostly based on (multi-view) stereo approaches. Similar to active stereo techniques, optical triangulation determines 3D positions of surface samples by finding corresponding points on two or more 2D images. The only difference is that no controlled illumination is used to promote correspondence finding. Correspondence computation for each pixel of a reference camera is typically reduced to a 1D search problem by exploiting epipolar geometry. Since a surface sample and its projection on two camera sensors are on a same plane (*epipolar plane*), the correspondence of each observed pixel lies on its *conjugate epipolar lines*. Epipolar lines can be easily determined once *extrinsic* and *intrinsic* camera parameters are computed (e.g., with an automatic calibration process [Tsa92, HS97, FP09b, Bou08]). Nonetheless, it generally holds that search becomes more difficult when a wide baseline between cameras is used. On the other hand, using a too small baseline decreases accuracy.

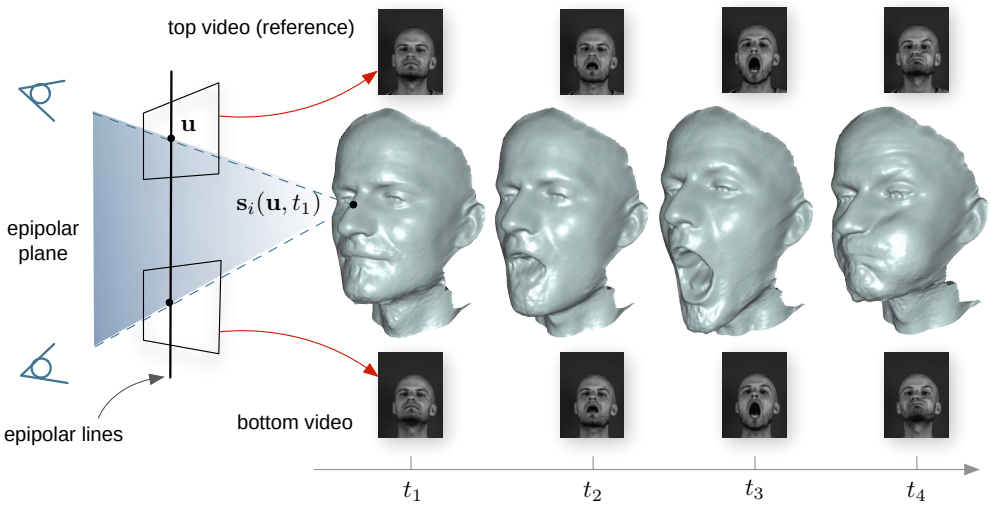


Figure 2.7: Passive stereo acquisition of dynamic shapes.

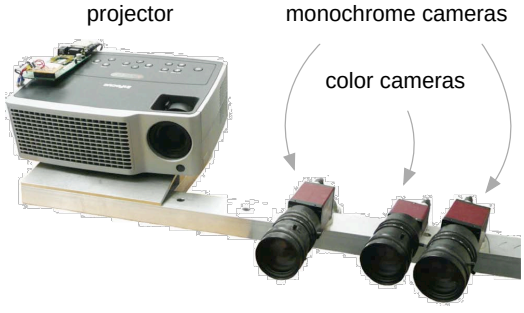
Because of possible suboptimal surface textures (e.g., homogeneous regions) and illumination conditions, determining correspondences is the most difficult element in passive stereo acquisition. A comprehensive overview of stereo matching algorithms can be found in [SS01, BBH03] (binocular stereo) and [SCD+06b] (multi-view stereo). After roughly three decades of research in stereo algorithms, we have witnessed a shift from pure local matching techniques to global optimization frameworks that are particularly effective in extracting continuous surfaces and handling occlusions. Some of the most popular approaches model the disparity map as a Markov Random Field (MRF) and use inference algorithms such as *Graph Cuts* [RC98, BVZ01, KZ02, FP10] or *Belief Propagation* [YFW03, SySnZ03, KSK06] to extract an optimal set of connected surfaces. A nice evaluation between both optimization techniques is given in [TF03]. Generally, passive acquisition methods are known to be less reliable and accurate than active techniques. Furthermore, many parameters need to be manually specified for optimal reconstruction which can result in long turn around times for parameter tuning. For certain scenarios however (such as facial reconstruction), some recent work have demonstrated that skin pores can be effectively used for stereo matching and generate results with accuracies comparable to those obtained from structured light scanners [BBB+10]. The requirements are very bright and diffuse lighting conditions as well as high-definition recordings [BBB+10, BPS+08]. In addition, band-pass filtered high-frequency details (*mesoscopic structures*) can be synthesized on top of the (multi-view) reconstruction for more compelling visual quality as demonstrated in Beeler and colleagues [BBB+10].

Because passive methods are one-shot techniques, they can be directly used for real-time dynamic capture [BPS+08, BPS+08, Ima] using at least two video cameras for recording. Some of the early work on 3D motion recovery from passive stereo systems involve rather naive local *scene flow* estimations [CK01, LS08, NA02, VBK05, KF06]. These methods independently track a densely sampled set of local surface patches across input frames using geometry and texture information. As opposed to registration algorithms for deformable surfaces, global constraints such as spatio-temporal consistency are not fully exploited. As a result, pure scene flow computation algorithms are limited to slowly deforming subjects with minimal occlusions, and are highly sensitive to error accumulations. Analogous to dense-marker based systems or active acquisition methods, surface motion can be recovered using non-rigid surface registration algorithms when high-quality 3D scan sequences can be reconstructed. For colored scans, texture-based tracking [BBPW04, BTVG08] that exploits local photo-consistency can be employed to promote tracking accuracy and robustness along surface tangents [FP08, FP09a, BPS+08, LLV+10]. The passive facial tracking system developed by Bradley and coworkers [BPS+08] for example exploits skin pores for optical-flow based tracking [HS81, BBPW04].

While most previous work on non-rigid registration relies on the quality of tracked features (geometry and/or texture based), we argue in Chapter 3 that a coupled optimization between surface deformation and 3D correspondences can effectively resolve for this dependency. While texture based constraints can be easily incorporated [LLV+10] (for input data with rich texture information), pure geometric correspondences can be robustly determined even for large deformations and occlusions. *Our key insight is that correspondence search that incorporates global deformation constraints achieves a very large funnel of attraction for the minimization problem.*

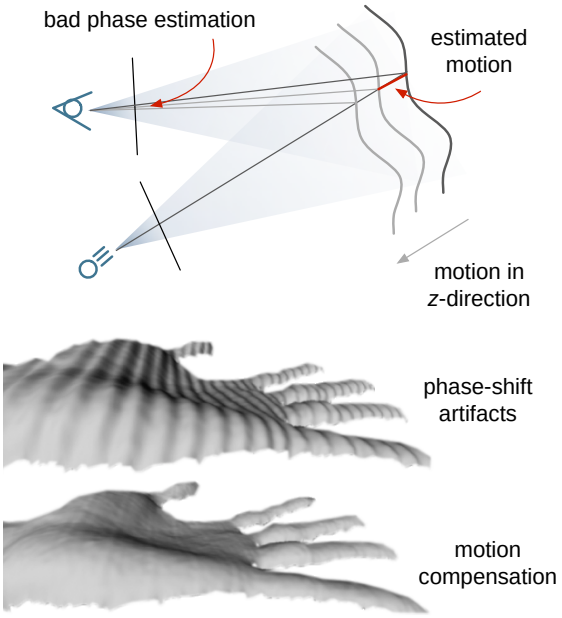
2.3 Single-View Structured Light Scanning

Our first source of input data is obtained from the *stereo phase-shift* structured light scanner developed by Weise an coworkers [WLG07] and is used throughout this work. The system combines the robustness of active stereo techniques and the accuracy of structured light techniques



with phase-shift patterns. Dense 3D scan sequences with textures (640 × 480 pixes) are produced at 30 fps from a single view-point. The capture volume is approximately $40 \times 30 \times 60\text{cm}^3$ which makes it suitable for facial performance capture and reconstruction of small dynamic objects. Depending on the subject, each scan has approximately 100 K vertices and achieves sub-millimeter accuracy when the subject is not moving. While acquisition works under normal lighting conditions, the brightness of the projected pattern needs to be sufficiently high.

The real-time 3D scanner uses off-the-shelf components and consists of a standard DLP projector (120 Hz), two high-speed monochrome cameras (200 fps) and color camera for texture recording. To fully exploit projection speed, the 4-segment RGBW color wheel of the projector has been removed—achieving effectively 360 Hz (W channel is only used to increase brightness). Consequently, each RGB channel can be used to project individual temporal patterns. The third color camera operates at longer exposure to capture textures free from sine patterns.

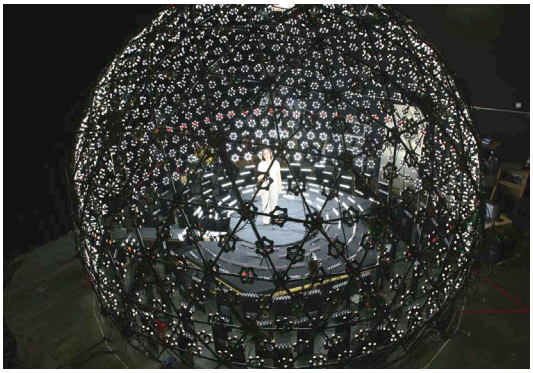


The structured light scanner sequentially projects three phase-shifted sinusoidal pattern which are used to uniquely determine the phase for each observed pixel. However, we need correspondences between projector and camera instead of phases for optical triangulation. When N_{phase} number of phases are being projected, an observed pixel may correspond to N_{phase} different possible positions (period) on the projector image plane. Hence the system uses a second monochrome camera to solve for the optimal period using stereo matching between two cameras (*phase unwrapping*). The proposed stereo correspondence technique performs a two step-algorithm. First, a greedy sum-of-squared-differences (SSD) matching is performed for each pixel (N_{phase} possibilities per correspondence). Due to mismatches, many discontinuities and holes appear in the temporary disparity map. The second step consist of maximizing local surface continuity which is equivalent to a labeling problem

(each surface segment being a label). *Loopy Belief Propagation* as described in [KSK06] is used for this optimization followed by a left-right consistency check of the disparity maps. An inherent problem of phase-shift techniques is that for fast motions in z-direction leads to aliasing artifacts (unwanted vertical line structures). A simultaneous estimation of surface motion compensates for these distortions as described in [WLG07].

The main advantage of single-view 3D scanners is the simplicity of the acquisition setup, requiring no calibration or synchronization of multiple sensing units. However, single-view reconstruction of dynamic shapes is particularly challenging, since every scan covers a small section of the object's surface. Our non-rigid registration algorithm in Chapter 3 and dynamic shape reconstruction framework in Chapter 4 are designed to deal with single-view data. To produce 3D point clouds in real-time, most of the computations run on graphics processing units (*GPU*) except for the Loopy Belief Propagation optimization.

2.4 Multi-View Photometric Stereo



In Section 4.3, we propose a shape completion algorithm that processes high-resolution scan sequences of full body performances captured from multiple views. We consider input data that emanate from the *Lightstage 6* acquisition system (originally proposed in [CEJ+06]) where high-resolution dynamic shapes are reconstructed using the photometric stereo method from Vlasic and colleagues [VPB+09]. In addition to detailed scans that are captured at 60 Hz from a 360° surrounding, the system delivers surface textures and orientations (normal maps) at 1024×1024 resolution captured from 8 high speed cameras. To enable photometric stereo, 901 uniformly-spaced light sources are placed on the top two-third of an 8 m tall geodesic sphere and 299 are placed on the floor. This spherical lighting configuration is able to produce sufficient lighting without blinding the actor and also facilitates the process of calibration and time-multiplexing. The Lightstage 6 has a large working volume for human-size acquisition and produces scans with millimeter accuracy.

Shape reconstruction for each frame is split into multiple stages. For each output frame, each synchronized camera captures 8 images at 240 fps with different lighting con-

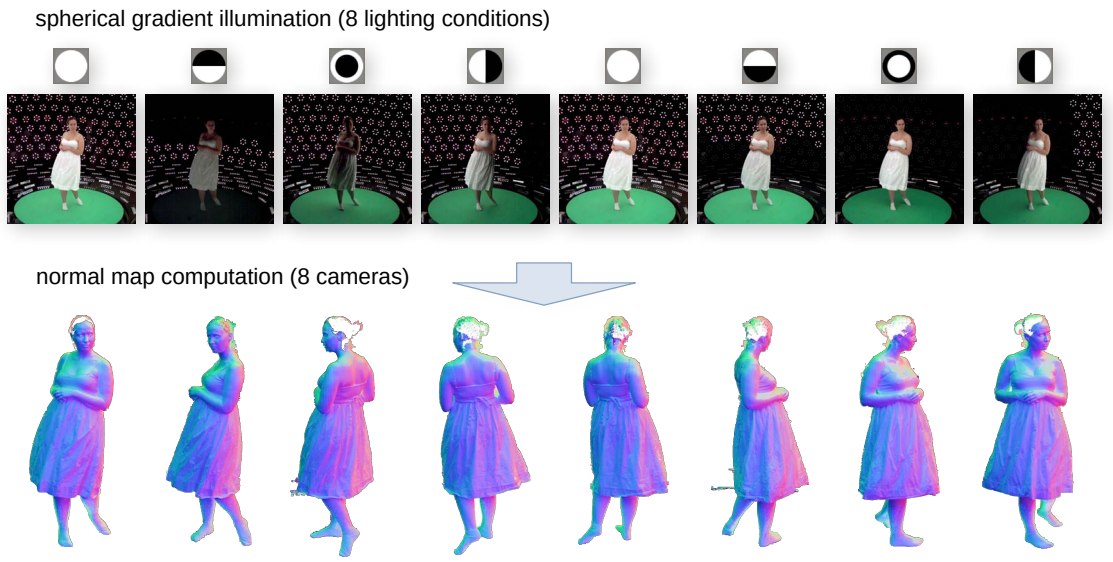


Figure 2.8: Under 8 spherical gradient illumination (top) we can extract dense normal maps from different views placed around the subject (bottom).

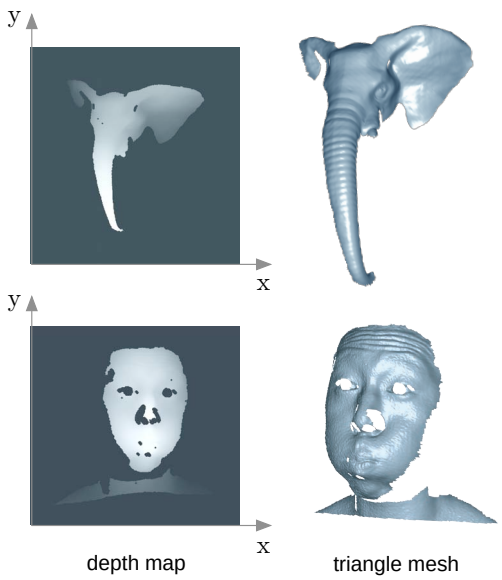
ditions (spherical gradient illumination as proposed in [MHP+07]). The time-multiplexed light patterns consist of 6 principal lighting directions and two interleaved full illuminations. A robust and efficient data-driven technique calculates a high-resolution normal map for each camera. Optical-flow [HS81, BBPW04] is used to compensate for subtle motions within the 8 frames. In parallel, the subject's silhouette is extracted from each view and combining them produces the shape's *visual hull* which forms a rough geometric prior. The next step consists of calculating a depth map from each normal map using normal integration. This under-constrained problem is solved by incorporating the visual hull as additional soft constraints which yields an over-determined linear system. Due to low frequency distortions in the scans (as the visual hull is only an approximation), the depth maps from each view are slightly misaligned. Hence, scans between adjacent views need to be non-rigidly aligned. In particular, the (one-step) non-rigid registration technique uses a thin-plate spline deformation model and correspondences that minimize the shape of local surface patches. Once the scans are aligned, a merging process produces a single consistent surface using the *VRIP* algorithm proposed by Curless and Levoy [CL96b]. Optionally, the remaining holes that are due to occlusions and low surface albedo can be filled using Poisson reconstruction [KBH06] and surface samples obtained from the visual hull. Because filling each frame independently causes strong flickering in those hole filled regions, we propose a temporally coherent shape completion

technique in Section 4.3 that considers scans from adjacent frames for spatio-temporal filtering.

The bulk of this engineering effort focusses on fast and detailed shape acquisition of human-size subjects. The advantage of using large spherical area light sources enables short exposure acquisitions (minimizing motion blur) while remaining sufficiently comfortable for the subject to perform (since irradiance is generally better distributed than for example point-light sources). Compared to pure passive techniques, shape reconstruction is more reliable especially for regions with fine geometric details and homogenous textures such as garment.

2.5 Data Representation and Processing

From Depth Maps to Triangle Meshes. Our captured dynamic surface data is discretized as either a sequence of depth maps or unorganized point clouds $\{\mathbf{s}_i(t)\}_i$ (c.f. Section 2.1). In both cases, polygonal mesh connectivity might or might not be provided by the scanning method. For range maps, adjacent vertices of a depth image are often connected to form a continuous piecewise linear surface (c.f. illustration). However, the shapes produced by many scanning technologies may be too strongly affected by noise and outliers to be reconstructed via simple data interpolation (e.g., time-of-flight scanners). In this case, it might be appropriate to strip away mesh connectivity and use a robust volumetric method [LC87, CL96b, KBH06, GG07] to reconstruct a smooth, two-manifold surface. Similarly, when merging range data from multiple-views (*surface integration*), mesh connectivity is often discarded and a consistent mesh extracted through surface reconstruction.



This thesis primarily considers input data in the form of dense, regularly sampled triangle mesh sequences. We directly obtain triangle meshes from the two scanning systems presented in Section 2.3 and 2.4. While a large variety of surface representation methods exist [Far02, PBP02] (higher order NURBS surfaces, subdivision surfaces,

implicit surfaces...), triangle mesh representations have the advantage of being (in our case) easy to obtain, flexible, and efficient to process. Triangle meshes have powerful shape approximation properties and are particularly effective in representing the exterior surface of complex geometries such as $\partial_{\text{ext}}\mathcal{M}(t)$. Compared to NURBS surfaces, triangles meshes can represent arbitrary topologies without being decomposed into multiple surface patches. Furthermore, the stability of many numerical optimization techniques for geometry processing relies on the fact that input shapes are densely and uniformly discretized (e.g., meshes with triangles that are close to equilateral).

A triangle mesh $\mathcal{S}_d \subseteq \mathbb{R}^3$ is an *explicitly* defined surface representation embedded in 3-space and, in contrast to spline surfaces (such as NURBS), not defined in terms of a surface parameterization. Nevertheless, triangle meshes that describe a two-manifold $\mathcal{S}(\mathbf{u})$ may be decorated with a surface parameterization $\mathbf{u} \in \mathcal{U}_{\mathcal{S}}$. Depth maps for example, inherently carry a surface parameterization and dedicated algorithms exist to generate parameterizations for general surfaces [HLS07]. In general, $\mathcal{S}_d$ discretizes a smooth manifold $\mathcal{S}$ into *geometric* and *topological* elements and describes a continuous piecewise linear surface. More specifically, $\mathcal{S}_d$ consists of a set of *vertices* (geometry):

$$\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_{N_\mathcal{V}}\} \quad \text{with} \quad \mathbf{v}_i \in \mathbb{R}^3 \tag{2.11}$$

and a set triangular faces (topology or connectivity):

$$\mathcal{F} = \{f_1, \ldots, f_{N_\mathcal{F}}\} \quad \text{with} \quad f_i \in \mathcal{V} \times \mathcal{V} \times \mathcal{V} \quad . \tag{2.12}$$

Alternatively, we may describe mesh connectivity using edges which, in some cases, can be a more efficient:

$$\mathcal{E} = \{e_1, \ldots, e_{N_\mathcal{E}}\} \quad \text{with} \quad e_i \in \mathcal{V} \times \mathcal{V} \quad . \tag{2.13}$$

The beauty of using triangle meshes for approximating smooth geometries lies in its quadratic approximation power. In particular, halving the edge lengths would reduce the error by a factor of $\frac{1}{4}$ which can be shown using Taylor expansion. At the same time, the number of faces $N_\mathcal{F}$ is inversely proportional to the discretization error of $\mathcal{S}_d$.

As pointed out in Section 2.1, we consider surfaces that are 2-manifolds with possible boundaries since our algorithms rely on the existence of local geodesic neighborhoods and tangent planes. To test whether a triangle mesh is locally homeomorphic to a disc (or half-disc at boundaries) in a parametric domain $\mathcal{U}_d$, it is sufficient to verify if $\mathcal{S}_d$ is free from non-manifold edges (more than 2 incident triangles), non-manifold vertices (multiple incident triangle fans), and self-intersections.

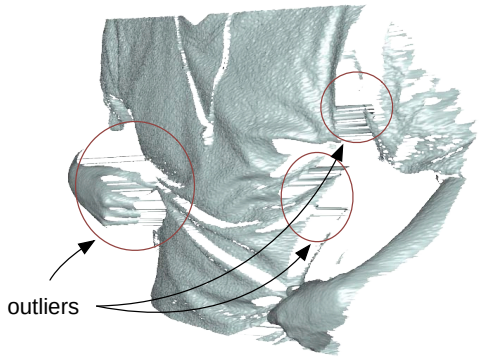
In addition to efficient random element access and fast mesh traversal, many fundamental geometry processing algorithms require a local neighborhood query of an arbitrary surface point. For triangle meshes, a frequently performed access is the one-ring neighborhood $\mathcal{N}(i)$ of a vertex $\mathbf{v}_i \in \mathcal{S}_d$. The neighborhood $\mathcal{N}(i)$ consists of all vertices, edges, and faces incident to $\mathbf{v}_i$. Depth maps for example have neighborhood information implicitly encoded in their parametric domain where the neighborhood of a surface sample is determined by adjacent pixels. However, we need to consider general surfaces and fast one-ring neighborhood access can be achieved in $\mathcal{O}(1)$ using a suitable polygonal mesh data structure (e.g., face-based representation, winged-edge data structure, half-edge data structure...). All implementations in this work use the efficient (computation and memory-wise) directed half-edge data structure [CKpS] as underlying representation for triangle meshes. For sufficiently dense meshes, we may accurately approximate the *normal* $\mathbf{n}_i$ of a vertex $\mathbf{v}_i$ by averaging the unit normals of all triangles $f_j \in \mathcal{N}(i)$.

Another important geometric operation is the distance query of an arbitrary point in space $\mathbf{p} \in \mathbb{R}^3$ to a triangle mesh $\mathcal{S}_d$ which is analogous to determining the closest point $\mathbf{c} \in \mathcal{S}_d$. The point $\mathbf{c}$ may lie exactly on a vertex, inside a triangle, or on an edge. Hence, naively querying each closest point would result in a linear search in the elements of $\mathcal{S}_d$. Typically, acceleration data structures for spatial query (e.g., uniform grids, kd-tree, hash data structures, BSP trees, octrees, or bounding volume hierarchies,...) are used to significantly speed up the closest point computation. Throughout this dissertation, we employ a kd-tree data structure [Ben75] for triangle primitives which achieves a search performance of $\mathcal{O}(\log(N_\mathcal{F}))$ per query. Note that building the data structure involves a computational cost of $\mathcal{O}(N_\mathcal{F} \log(N_\mathcal{F}))$. For many applications (outside this work) that involve regular updates of dynamic geometry, acceleration methods with faster construction may be more suitable (bottom-up techniques such as bounding volume hierarchies, GPU parallelized kd-trees [ZHWG08]...). In the case of dense meshes, a simpler kd-tree that only determines the closest vertex $\mathbf{v}_j$ (as opposed to the closest point on $\mathbf{c} \in \mathcal{S}_d$) may also be considered. Because the set of vertices $\mathcal{V}$ is finite, the query can be further generalized to determine the k closest points with the same run-time complexity.

Outlier Removal. We pointed out in Section 2.1 that raw scan data are often affected by a certain amount of outliers (especially in early generations of scanning technology).

In general, it is rather difficult to specify a criterion for detecting outliers as they depend on shape and material properties of the subject and the scanning technology. An effective, semi-automatic tool for treating outliers based on classification heuristics can be found in [WPH+04]. Fortunately, due to the robustness of the acquisition techniques [WLG07, VPB+09], outliers can be easily detected and removed in our input scans.

We mostly observe outliers in the form of false mesh triangulations at occlusion boundaries where there is a large depth disparity (see figure on the right). Several strong artifacts such as jumping peaks can also be identified in regions which surface normals that are mostly perpendicular to the sensor's viewing direction (unconfident surface regions). Because our input scans are densely and uniformly sampled, triangles with an extremely large edge can be regarded as those outliers. Consequently, we simply discard triangles with edge length greater than a threshold of 0.5 cm. Additionally, we aggressively delete all fragmented components in $\mathcal{S}_d$ with fewer than 200 triangles within a single connected surface patch. The fragments are mainly caused by noise in the volumetric reconstructions [KBH06] when combining multiple range maps [VPB+09].



Mesh Smoothing. The use of mesh smoothing (or *fairing*) in a surface reconstruction pipeline is typically associated with noise removal of captured input data (c.f. Figure 2.10), but also finds its place within the context of geometric modeling and multiresolution techniques. Similar to low-pass filtering in signal processing, the purpose of mesh smoothing is to reduce unwanted high-frequency details while preserving low-frequency components of a surface (its global shape). Unlike for example image signals, a triangle mesh may not have a parameterization (except for depth maps). We therefore require an efficient low-pass filtering technique that reduces high curvature variations by simply moving the vertices without changing mesh connectivity. Mesh smoothing based on signal processing analysis is a well-understood topic [Tau95, FDCO03] and it has been shown that it is directly related to discrete Laplacian diffusions defined on meshes [DMSB99]. We now summarize the basic concepts of *Laplacian smoothing*

which we describe as stationary surfaces of Laplacian flows. We will later extend these ideas to hole-filling and surface deformation techniques based on linearized membrane or thin-plate energy minimization. From a signal processing standpoint, the first step

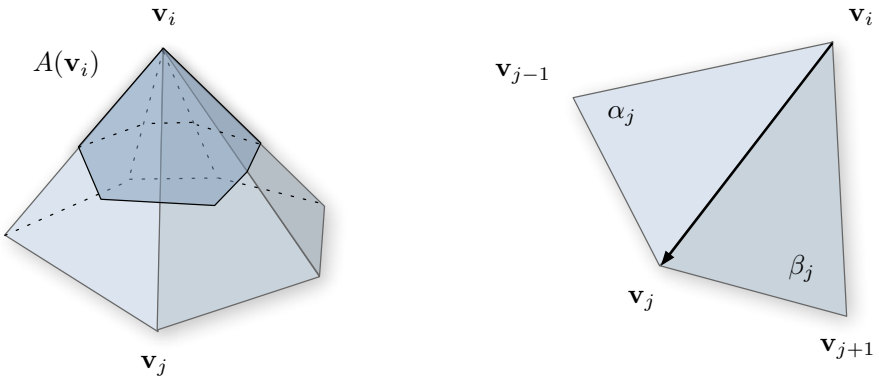


Figure 2.9: A vertex $\mathbf{v}_i$ and its incident faces (left) and the terms of the cotangent weights (right).

of an ideal low-pass filter consists of converting a discrete signal from spatial domain to frequency domain where high-frequency components are discarded. This is typically achieved using a discrete (fast) *Fourier* transform (*DFT*) in which an orthonormal basis of shifted and scaled sine waves is constructed. Once the signal is truncated in Fourier domain, the inverse DFT is performed to obtain the low-pass filtered result. In the case of triangle meshes, the formulation of discrete Fourier transform is different from parametric functions. Note that the second derivatives of these basis functions in Fourier domains are multiples of themselves. As demonstrated in [Tau95], we may construct an orthonormal basis function that shares the exact same properties by taking the eigenvectors of a discretized *Laplace* operator $\Delta_{\mathcal{S}}$. The discrete surface signal of a triangle mesh $\mathcal{S}_d$ is described by concatenating the vertices into the matrix $V = [\mathbf{v}_i, \ldots, \mathbf{v}_{N_\mathcal{V}}]^\top$ where a symmetric neighborhood structure is defined by the one-ring neighborhood $\mathcal{N}(i)$. The discrete Laplacian (*Laplace-Beltrami* operator [dC76]) defined on the signal V can then be formulated as a weighted average over the neighborhood:

$$\Delta_{\mathcal{S}}\mathbf{v}_i = \sum_{\mathbf{v}_j \in \mathcal{N}(i)} w_{ij}(\mathbf{v}_j - \mathbf{v}_i) \tag{2.14}$$

where $\sum_{\mathbf{v}_j \in \mathcal{N}(i)} w_{ij} = 1$ and $w_{ij} \geq 0$. A good choice for these weights w_{ij} are the cotangent-weights described in [PJP93, MDSB02, DMSB99] as they preserve local ge-

ometry properties such as edge lengths and angles. The discretized Laplace-Beltrami operator becomes:

$$\Delta_{\mathcal{S}}\mathbf{v}_i = \frac{2}{A(\mathbf{v}_i)} \sum_{\mathbf{v}_j \in \mathcal{N}(i)} (\cot\alpha_j + \cot\beta_j)(\mathbf{v}_j - \mathbf{v}_i) \tag{2.15}$$

where $\alpha_j = \angle(\mathbf{v}_i, \mathbf{v}_{j-1}, \mathbf{v}_j)$, $\beta_j = \angle(\mathbf{v}_i, \mathbf{v}_{j+1}, \mathbf{v}_j)$, and $A(\mathbf{v}_i)$ the Voronoi area around $\mathbf{v}_i$ as depicted in Figure 2.9. In theory, we may consider the matrix form of the discrete Laplacian $\Delta_{\mathcal{V}} = -K\,V$ and compute its eigenvectors E, i.e., $-K\,E = D\,E$ with D the diagonal matrix of eigenvalues. This step can be followed by discarding the eigenvalues in D corresponding to the high frequencies and transforming back in spatial domain. While this approach is computationally equivalent to performing a DFT on V, there is no known extension of fast Fourier transform algorithm in this setting. A practical solution consists of taking a convolution approach with a smoothing kernel which is linear in the number of vertices $N_{\mathcal{V}}$. We observe that the following update rule:

$$\mathbf{v}_i \leftarrow \mathbf{v}_i + \lambda\,\Delta_{\mathcal{S}}\mathbf{v}_i \tag{2.16}$$

with time-step $0 < \lambda < 1$ is equivalent to a projection onto the low frequencies. In fact, Equation 2.16 can be written in matrix form yielding $V \leftarrow (I - \lambda K)V = E(I - \lambda D)E^{-1}V$. In particular, the *damping factor* λ attenuates the high-frequency components of V. Note that repeating this Laplacian update is equivalent to performing an explicit forward Euler integration solving the following (heat) diffusion equation:

$$\frac{\partial \mathbf{v}_i}{\partial t} = \lambda \Delta_{\mathcal{S}}\mathbf{v}_i \quad . \tag{2.17}$$

Using a sufficiently small time-step λ ensures convergence to the steady state of a diffusion flow $\Delta_{\mathcal{S}}\mathbf{v}_i = \mathbf{0}$ when (numerically!) integrating this 2nd order linear PDE over time. For arbitrarily large time-steps, Desbrun and coworkers [DMSB99] use an implicit fairing approach and successively solve the following sparse linear system:

$$(I - \lambda K)V^{n+1} = V^n \tag{2.18}$$

where V^n is the nth iteration of the diffusion process. This discrete diffusion flow can also be regarded as the *mean curvature flow* [dC76] since the following relation holds:

$$\Delta_{\mathcal{S}}\mathbf{v}_i = -2H(\mathbf{v}_i)\mathbf{n}_i \tag{2.19}$$

with mean curvature $H(\mathbf{v}_i) = \frac{\kappa_1 + \kappa_2}{2}$, maximum curvature κ_1, and minimum curvature κ_2. In particular, mesh smoothing through mean curvature flow is the same as moving

each vertex $\mathbf{v}_i$ along the surface normals $\mathbf{n}_i$ with a speed equal to $H(\mathbf{v}_i)$. Notice that each integration step solving the unbounded diffusion equation in Equation 2.17 causes shrinkage of the two manifold. Anti-shrinking is typically accomplished through volume normalization with the initial shape [DMSB99] or by carefully amplifying low frequencies with the $\lambda|\nu$ method [Tau95].

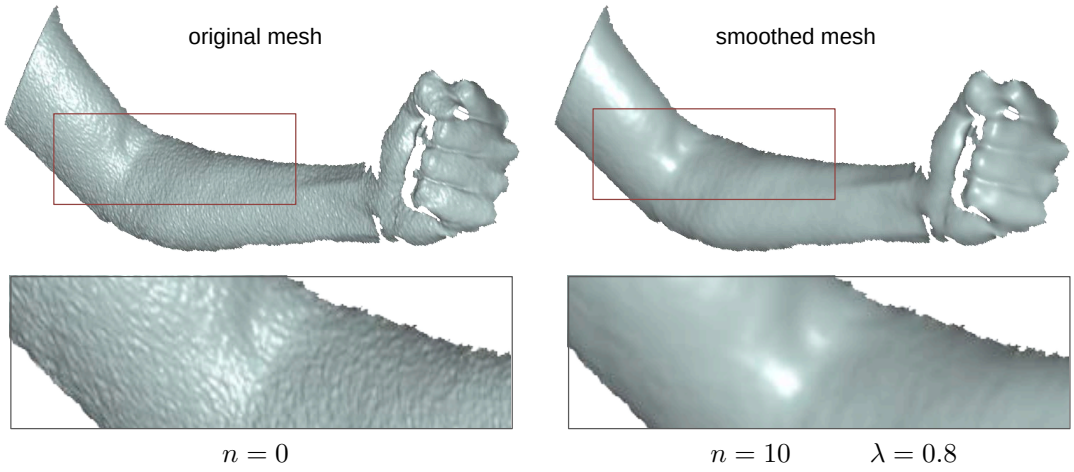


Figure 2.10: Laplacian smoothing used to low-pass filter a 3D scan affected by high-frequency noise. This example shows $n = 10$ iterations of explicit Laplacian updates.

Isotropic Remeshing. While depth-scans are dense and regular surface samplings, robustly merging them into a consistent surface generally involves a volumetric mesh extraction step based on the *marching cubes* algorithm [LC87, CL96b, KBH06, GG07]. For sufficiently high resolution of the grid discretization of marching cubes, the correct topology of the subject can be extracted (except when the subject exhibits (self-) contacts). However, because mesh vertices are connected at intersections between an (implicit) surface and a regular grid, the extracted mesh has highly varying edge lengths which are often close to zero. Recall that non-uniform and singular edge lengths are likely to cause numerical instabilities for many geometric optimizations on discrete surfaces (whenever $\mathcal{N}(i)$ is involved). Another important application of isotropic remeshing is to convert resolutions between uniformly sampled meshes. When fitting a template mesh $\mathcal{T}_d$ to a target scan $\mathcal{S}_d$ (e.g., for tracking or shrink-wrapping purposes), the sampling density of both meshes must be compatible. Although the Nyquist criterion suggests a higher

resolution sampling of $\mathcal{T}_d$, we typically choose the same resolution for efficiency reasons.

The aim of isotropic remeshing is to resample a given polygonal mesh and reconnect the vertices in a topologically consistent way. For triangle meshes, we eventually obtain triangle faces that are close to equilateral. While methods exist that locally adapt the edge length according to the scale of geometric details [SAG03], we focus on remeshing algorithms that produce uniform samplings (i.e., homogenous edge lengths) according to a user-specified target edge length l. Additionally, we require vertex positions to closely stay on the original mesh surface so that shape is being preserved after remeshing.

A number of methods have been proposed that exploit surface parameterization [AMD02, AVDI03, ACSD$^+$03] to produce high-quality remeshing. In particular, resampling and tesselation is efficiently computed in the two-dimensional parameter domain. Since obtaining a global parameterization for $\mathcal{S}_d$ is known to be an expensive step, several techniques were introduced that only use local mesh operations and/or local patch parameterizations [SG03, SAG03, VRpS03]. These algorithms either impose hard error bounds for highly uniform triangulation or locally adapt the resolution according to the mesh curvature. Since we are mainly interested in obtaining isotropic meshes with uniform edge lengths, we resort to a fast and easy-to-implement remeshing algorithm proposed by Botsch and Kobbelt [BK04]. The isotropic remeshing algorithm unifies

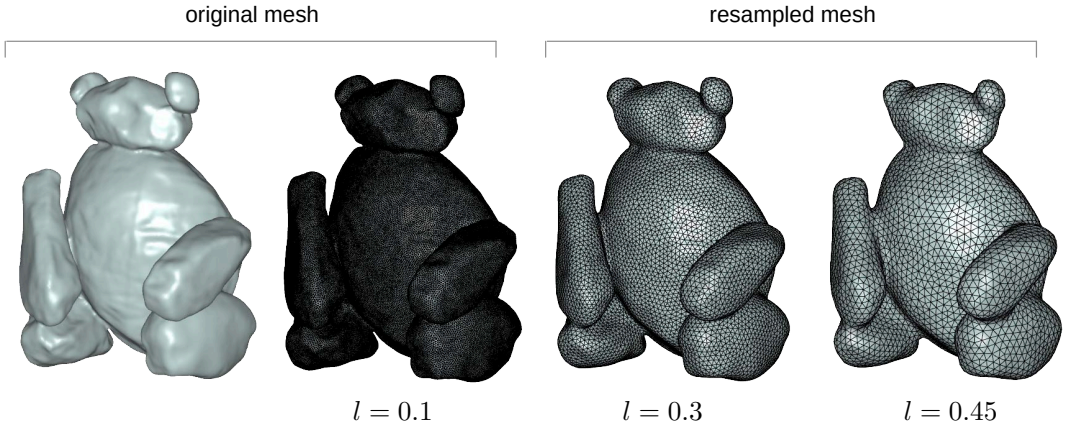


Figure 2.11: Isotropic remeshing of the UCSD Bear.

edge length equalization, vertex valence optimization, and vertex distribution given an input target edge length l (c.f. Figure 2.11). Each criterion is optimized separately, but

a tight interleaved process approximates a more difficult to solve coupled optimization. In particular the following steps are iteratively performed:

1. All edges are split at their midpoint if their length exceeds $l_{\max} = \frac{4}{3}l$.
2. All edges are collapsed if their edges are shorter than $l_{\min} = \frac{4}{5}l$.
3. All edges are flipped if they improve the valence of the influenced vertices. Vertex valence should be 6 (or 4 at mesh boundaries).
4. To improve uniformity of the vertex distribution, area-weighted tangential smoothing is performed for each vertex.

Very few iterations (generally only 5) are sufficient to produce triangle meshes with edge lengths close to l and triangles with inner angles close to $60\,^\circ$. Note that the local mesh operations (edge split, edge collapse, and edge flipping) only change the local mesh connectivity but preserve the global mesh topology. The maximum edge lengths $l_{\max} = \frac{4}{3}l$ can be derived from the fact that an edge split operation only improves the tesselation if $|l_{\max} - l| > |\frac{1}{2}l_{\max} - l|$. Simply consider the case when $\frac{1}{2}l_{\max} < l$. The lower threshold $l_{\min} = \frac{4}{5}l$ can be derived analogously.

We perform valence equalization in a greedy fashion. We iterate over each edge $e_i \in \mathcal{E}$ and evaluate the local energy E_{val} that measures deviation from the optimal valence $\text{valence}_{\text{opt}}$ before and after the flip. If E_{val} decreases after the flip, we keep the new connectivity, otherwise we revert to the original local triangulation. We may derive $\text{valence}_{\text{opt}} = 6$ (and $\text{valence}_{\text{opt}} = 4$ on mesh boundaries) from the Euler characteristic for triangle meshes $\chi(S_d)$ [Cox89]. The valence deviation energy is defined as follows:

$$E_{\text{val}}(e_i) = \sum_{j=1}^{4} (\text{valence}(\mathbf{v}_{i_j}) - \text{valence}_{\text{opt}})^2 \tag{2.20}$$

where $\mathbf{v}_{i_j}$ are the 4 vertices of the two incident triangles of e_i.

The last step of the iterative remeshing procedure consists of a continuous edge length equalization (mid-point splits and collapses are discrete operations). In this refinement step, each vertex is locally relocated using an area-weighted tangential smoothing process. We may describe an Voronoi area-weighted smoothing with the following update rules:

$$\mathbf{v}_i \leftarrow \mathbf{v}_i + \lambda \, \Delta_{\text{grav}} \mathbf{v}_i \tag{2.21}$$

with

$$\Delta_{\text{grav}}\mathbf{v}_i = \frac{1}{\sum_{\mathbf{v}_i \in \mathcal{N}(i)} A(\mathbf{v}_i)} \sum_{\mathbf{v}_i \in \mathcal{N}(i)} A(\mathbf{v}_i) \quad . \tag{2.22}$$

As opposed to the cotangent-based Laplace-Beltrami $\Delta_{mathcalS}$ operator, this smoothing operation causes the vertices to move toward a *gravity*-weighted centroid and involves a normal and tangential surface motion. To enforce a purely tangential motion (since the aim is to equalize edge lengths), the update rule can be extended with a projection on the tangent plane of $\mathbf{v}_i$:

$$\mathbf{v}_i \leftarrow \mathbf{v}_i + \lambda \, (I - \mathbf{n}_i \, \mathbf{n}_i^\top)(\Delta_{\text{grav}}\mathbf{v}_i - \mathbf{v}_i) \quad . \tag{2.23}$$

In particular, vertices with larger Voronoi area $A(\mathbf{v}_i)$ have a higher gravitational force, attracting those with smaller Voronoi areas. Restricting the motion on tangent planes (defined by $\mathbf{v}_i$ and $\mathbf{n}_i$) is a linear discretization of tangent motion on a parameterized smooth surface. Due to blurring caused by this linear approximation, the relocated vertices are reprojected onto the original surface. Reprojection only needs to be performed once after several cycles of iterations and can be efficiently computed using a kd-tree data-structure as discussed previously.

3

Registration of Deformable Surfaces

Given partial acquisitions of deforming objects, we consider the fundamental problem of recovering full *3D models in motion*. Even when multiple range sensors are simultaneously used in order to maximize coverage, the captured shape is generally still incomplete because of (self-) occlusions or low surface albedo. To obtain a complete digital representation, the subject has to move around in space and expose new surface geometry to the sensors. While the subject is changing its position and undergoing deformations, the surface portions that are previously captured must be repositioned and non-rigidly aligned to the current scan. As mentioned in Chapter 2, the purpose of *non-rigid registration* algorithms is exactly to compute these alignments and to establish surface correspondences between partial scans.

In addition to being a key element for dynamic shape reconstruction, non-rigid registration is an essential tool for dense markerless motion capture, template tracking, shrink-wrapping (deforming a generic 3D model to fit scan data), and establishing correspondences between different objects. A large variety of non-rigid registration techniques exist, each of them typically being attuned to a specific application, scenario, or acquisition technology. This chapter reviews the basic concepts of surface registration

for *static* (Section 3.1) and *deformable objects* (Section 3.3). To better illustrated the latter, we provide an extensive overview of the most established surface deformation techniques developed over the past few years (Section 3.2). Since the right choice of the deformation model is an essential step for a successful non-rigid registration, we will focus on explaining their strengths, weaknesses, and how they relate to each other from a geometrical and physical perspective. Additionally, a thorough investigation on different non-rigid registration techniques is covered in Section 3.3.2.

Our goal is to develop a fully unsupervised, pairwise non-rigid registration algorithm that is robust and accurate enough to process long scan sequences without severe accumulation of errors. In particular, it is highly desirable that the proposed method is able to handle significantly larger deformations than existing techniques. We propose in Section 3.4 a unique approach that unifies deformation and correspondence computation within a single non-linear optimization framework. Our initial method proposes a continuous optimization of correspondence positions which requires the target scan to have a *surface parameterization* which is inherently given for single-view depth map acquisitions.

To remove some of the implementation headaches when dealing with general shapes that are not directly equipped with surface parameterization (e.g., multi-view acquisitions), we derive in Section 3.5 an easier to implement non-rigid iterative closest point (ICP) variant that can be equally effective and accurate (in practice!). We establish a link between the two algorithms and explain why they both solve correspondences and deformations simultaneously.

The registration of deformable surfaces is one of the most difficult and crucial steps in animation reconstruction. To give a better intuition and to highlight some of the challenges, let us discuss everything again, but in more detail and with the notations introduced in Section 2.1:

What is Surface Registration? Consider the example where two scans, $\mathcal{S}(t_1)$ and $\mathcal{S}(t_2)$, are captured at two different time instances, t_1 and t_2, while the object $\mathcal{M}(t)$ is deforming (c.f. Figure 3.1). The surface region $\mathcal{S}(t_1) \setminus \mathcal{S}_{t_1 \cap t_2}(t_1)$ that is exposed in t_1 but not in t_2 should supplement $\mathcal{S}(t_2)$. On the other hand, common subregions, $\mathcal{S}_{t_1 \cap t_2}(t_1)$ and $\mathcal{S}_{t_1 \cap t_2}(t_2)$, between both scans must perfectly overlap. The goal is therefore to determine the warping $\Phi_{t_1 \to t_2}$ of $\mathcal{S}(t_1)$ toward $\mathcal{S}(t_2)$ as if both scans were captured at the same time t_2. Computational methods to this problem are called *registration*

(or *alignment*) algorithms and we mainly distinguish between *rigid* and *non-rigid* ones. We call them *pairwise* when dealing with two input shapes and *multi-frame* otherwise. Rigid registration problems are easier to solve because only 6 parameters of a Euclidean transformation Φ_{rigid} need to be computed (c.f. Equation 2.8).

Depending on the purpose, non-rigid registration algorithms may involve complex deformation models Φ_{deform} with several orders of magnitude the number of unknowns as for the rigid case. Hence, we generally consider $\Phi_{t_1 \to t_2} = \Phi_{\text{deform}}$ unless explicitly specified to be rigid.

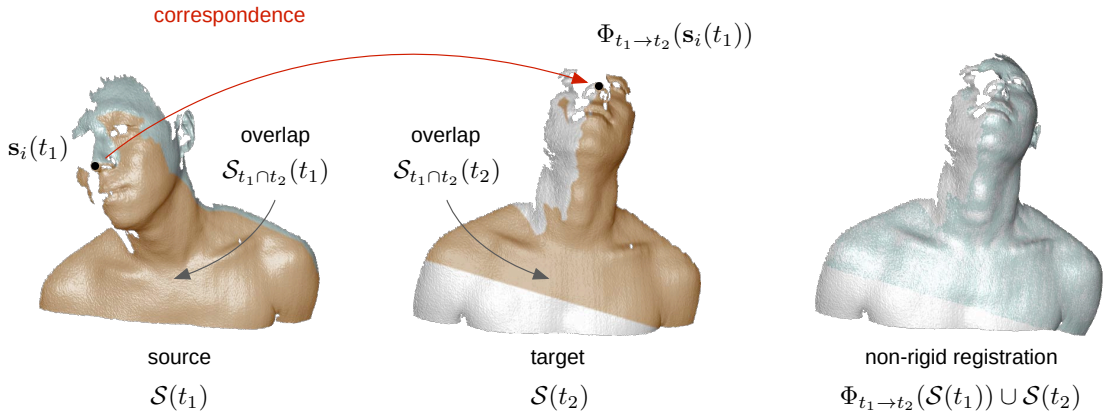


Figure 3.1: Correspondences reside within the overlapping regions (depicted in orange). After pairwise non-rigid registration, the shape becomes more complete (right).

Link to Surface Correspondences. As seen in Section 3.5, surface registration relies on establishing *correspondences* between the common subregions $\mathcal{S}_{t_1 \cap t_2}(t_1)$ shared by two independently captured shapes. To find the deformation Φ_{deform} between two shapes, we must ensure that corresponding surface points, $\mathbf{s}_i(t_1)$ and $\Phi_{t_1 \to t_2}(\mathbf{s}_i(t_1))$, coincide after the warp. Unless provided through human intervention or explicitly tracked using marker based systems, correspondences are generally not known *a priori*. From a purely geometric standpoint, the problem of automatically finding correspondences between $\mathcal{S}(t_1)$ and $\mathcal{S}(t_2)$ is rather non-trivial. In fact, only a limited number of surface points may have local geometries that are sufficiently unique and similar for being matched without ambiguity. Moreover, we have no prior knowledge about where the common subregion $\mathcal{S}_{t_1 \cap t_2}(t_1)$ is and how the subject would deform (i.e., the mapping $\Phi_{t_1 \to t_2}$). Consequently, the common approach to this ill-posed problem is to incorporate effective (but limiting) prior assumptions about the shape of the subject and how it may deform.

Remark: *Note that knowing the surface deformation $\Phi_{t_1 \to t_2}$ of a source scan $\mathcal{S}(t_1)$ toward its target $\mathcal{S}(t_2)$ trivially yields the full correspondence between both shapes and vice versa. Likewise, the more correspondences we are able to establish, the easier the problem of determining the deformation of the subject.*

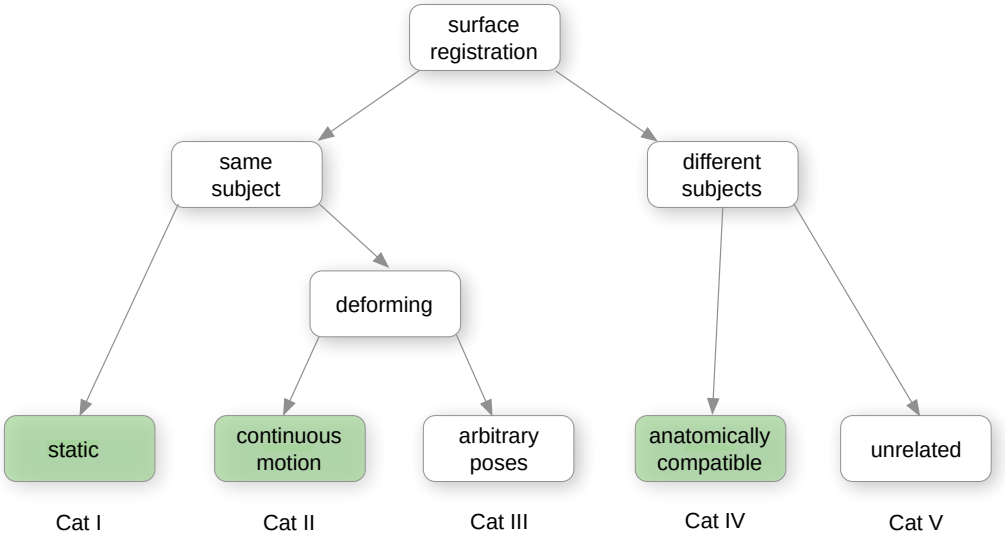
Link to Surface Motion. In animation reconstruction, we are interested in capturing the motion of deforming surfaces in addition to reconstructing the full geometry. As we continuously capture scans at short and regular time intervals $\mathrm{d}t$, a dense motion field $\mathrm{d}\mathbf{s}_i(t)$ can be immediately inferred from correspondences (source and target positions) as described in Equation 2.6 and 2.7. As a result, non-rigid registration is often regarded as the key element for markerless motion capture of deforming geometries.

How to Classify Registration Problems? Surface registration appears frequently in geometric problems where shape matching is involved. Due to the fast growing development in novel non-rigid registration techniques, many methods are difficult to categorize since they are often combinations of others. We therefore propose a *taxonomy* that is motivated by the *nature of the input data* instead of their computational methodologies. As illustrated in Figure 3.2, we can classify registration problems into the following categories:

- **Cat I (Static):** Registration is performed between scans of a same static subject (such as a building, a statue, or a person who attempts to hold still). Only small scale warps are allowed in this scenario. More specifically, we may assume $\max_{i,t}\{\|\mathrm{d}\mathbf{s}_i(t)\|_2^2\} \leq \epsilon$ for outlier-free $\mathcal{S}(t)$.

- **Cat II (Continuous Motion):** Registration is consecutively performed between pairs of scans (of the same subject) that are continuously captured using a real-time acquisition system. Unfortunately, it is non-trivial to quantify the amount of allowed deformation Φ_{deform}. Consequently, we assume a sufficiently small temporal sampling $t_{j+1} - t_j \leq \sigma_t$, a reasonable amount of temporal coherence in the motion $\max_{i,t}\{\|\mathrm{d}\mathbf{s}_i(t)\|_2^2\} \leq \sigma_\Phi$, and sufficiently large common subregion $A(\mathcal{S}_{t_j \cap t_k}(t)) \geq \sigma_\mathcal{A}$.

- **Cat III (Arbitrary Poses):** Registration is performed between shapes of different poses but originating from the same subject (e.g., registration between an angry and smiling face or sitting and standing person). In this scenario the input



	Cat I	Cat II	Cat III	Cat IV	Cat V
general applications	rigid surface reconstruction	dynamic shape reconstruction	reconstruction from arbitrary poses of same object	correspondence computation across different objects	object recognition
		template tracking		shrink-wrapping	
		motion capture			
applications in facial animation	building a static face model	reconstruction of full, animated head models	corresponding different facial expressions of the same person	corresponding faces of different people	retrieving a face from a database that contains other objects
		facial expression tracking		warping a generic model to an arbitrary face scan	
state of research and open problems	mature technology	robust technique presented in this dissertation	involves human input and/or template model		coarse global shape matching possible
	challenges with very incomplete data with large global motions	challenges in handling very large deformations, long sequences, and topology changes	challenges in minimizing computational costs and manual parameter adjustment open problem with incompatible topologies		challenges in non-rigid shape descriptors

Figure 3.2: A taxonomy on registration problems based on the nature of input data (top) and important applications for each class of problem (bottom). Our proposed non-rigid registration algorithm is general and effective for the scenarios shown in green.

data are independently captured, where t_j and t_{j+1} can be arbitrary time instances, and may exhibit large pose variations $\|\mathrm{d}\mathbf{s}_i(t)\|_2^2$.

- **Cat IV (Anatomically Compatible):** Registration is performed between shapes of different subjects that have the same anatomy (e.g., a generic face model and the face scan of a specific person). For simplicity, the same pose (e.g., neutral facial expression) is usually used for both the source and the target. However, semantically meaningful correspondences are sometimes hard to define. For example, how can we correspond a human head to that of a chameleon (the ears are missing)? Similar to Cat III, there is no temporal element in this type of registration. However, when source and target shapes are bounded by thresholds σ_Φ and $\sigma_\mathcal{A}$, the algorithms from Cat II may be applied here as well.

- **Cat V (Unrelated):** Registration (or rather global shape matching) is performed between completely unrelated entities (e.g. faces, cars, buildings, trees, etc...). This class of registration problem does not make any assumption about the shape or pose of the objects. While having potential impacts in animation reconstruction, the applications here are primarily focussed on shape retrieval.

Remark: *The categories of registration problems listed above are special cases of one another and gradually less restrictive w.r.t. the amount and variability of deformation. In practice, additional assumptions need to be made before these techniques to become truly robust and effective for real applications. For example, articulated motion may be imposed to facilitate matching between shapes in arbitrary poses.*

3.1 Rigid Registration

A substantial amount of research has been devoted to the registration of rigid objects. In analogy to the more general non-rigid setting, rigid registration is primarily used for shape completion and (rigid) motion tracking from incomplete 3D input scans. There are three possible scenarios: either the object is moving, the 3D sensor, or both simultaneously. In all cases, the relative motion between two captured scans, $\mathcal{S}(t_1)$ and $\mathcal{S}(t_2)$ can be described by a single Euclidean transformation ϕ_{rigid} which aligns their overlapping regions $\mathcal{S}_{t_1 \cap t_2}$. As mentioned above, it only requires us to solve for 6 parameters and, w.l.o.g., we assume only the subject undergoes a rigid transformation. Because many important ideas can be extended to the non-rigid case, we now summarize

the basic concepts of rigid shape alignment and use dense uniform triangle meshes to describe our surfaces. For an extensive overview we refer the reader to the excellent course presented by Rusinkiewicz and coworkers [RBK05].

Let us consider the general case when two scans are captured at arbitrary time instance, t_1 and t_2, the objective consists of determining ϕ_{rigid} such that the distance between $\mathcal{S}_{t_1 \cap t_2}(t_1)$ and $\mathcal{S}_{t_1 \cap t_2}(t_2)$ are minimized (since both regions might be affected by noise and outliers). More specifically, we are dealing with a minimization problem with energy functional:

$$E_{\text{fit}} = \sum_{\mathbf{v}_i \in \mathcal{V}(t_1) \cap \mathcal{S}_{t_1 \cap t_2}(t_1)} \|(R\,\mathbf{v}_i + \mathbf{t}) - \mathbf{c}_i\|_2^2 \quad \text{with} \quad \mathbf{c}_i \in \mathcal{V}(t_2) \cap \mathcal{S}_{t_1 \cap t_2}(t_2) \quad . \tag{3.1}$$

While the correct solution minimizes this equation, minimizing E_{fit} does not in general yield the correct transformation ϕ_{rigid}. Consider the simple example when $\mathcal{S}_d(t_1)$ and $\mathcal{S}_d(t_2)$ are two planes where no unique solution can be found. In practice, we assume the existence of a certain amount of discriminating geometric features to facilitate the computation of $\underset{R,\mathbf{t}}{\operatorname{argmin}} E_{\text{fit}}$. Furthermore, recall that neither corresponding points $\mathbf{c}_i$ nor overlapping regions $\mathcal{S}_{t_1 \cap t_2}$ are known. It becomes obvious that optimizing for all possible unknowns is intractable over all possible correspondence combinations and the group of rigid body transformations.

The general approach to this alignment problem is to reduce the search space by decoupling the optimization into three steps. First, (*salient*) features are independently identified in $\mathcal{S}_d(t_1)$ and $\mathcal{S}_d(t_2)$. These features between both scans are then matched to produce a set $\mathcal{C} = \mathbb{R}^3 \times \mathbb{R}^3$ of point-to-point correspondence, $(\mathbf{v}_i, \mathbf{c}_i) \in \mathcal{C}$ where $i = 1, \ldots, N_{\mathcal{C}}$. When more than three pairs of correspondences are found, the optimal transformation ϕ_{rigid} can be uniquely determined by minimizing the sum of squared distances between corresponding points.

3.1.1 Closed Form Solution

A closed form solution to this last step is described in Horn [Hor87] and uses quaternions [Sal79] obtained via spectral decomposition of cross-correlated data (in a 4D quaternion space) to describe the rotation. Alternatively, one may consider a simpler approach based on SVD proposed by Arun and colleagues [AHB87] which solves the exact same problem. This algorithm is a special case of the well-known Procrustes analysis method [Ber98] which also handles isomorphic scaling. Let us summarize the

SVD approach using the set of pairwise correspondences $\mathcal{C}$. The cross-covariance matrix between the points $\mathbf{v}_i$ and $\mathbf{c}_i$ is defined as:

$$\Sigma = \frac{1}{N_\mathcal{C}} \sum_{\mathbf{v}_i, \mathbf{c}_i} (\mathbf{v}_i - \mathbf{g}_v)(\mathbf{c}_i - \mathbf{g}_c)^\top \tag{3.2}$$

with source centroid $\mathbf{g}_v = \frac{1}{N_\mathcal{C}} \sum_{i=1}^{N_\mathcal{C}} \mathbf{v}_i$ and target centroid $\mathbf{g}_c = \frac{1}{N_\mathcal{C}} \sum_{i=1}^{N_\mathcal{C}} \mathbf{c}_i$. From the SVD $\Sigma = U\,\Lambda\,V^\top$, we may extract the optimal rotation $R = U\,V^\top$. To ensure that $R \in \mathrm{SO}(3)$, we detect unwanted reflections whenever $\det(R) = -1$. In case reflection occurs, we invert the sign of the jth column vector in V if the diagonal component $\lambda_j \in \Lambda$ is zero. Eventually, we obtain the optimal translation $\mathbf{t} = \mathbf{g}_c - R\,\mathbf{g}_v$.

Because of the discrete nature of identifying features and computing correspondences, it is, in most cases, practically impossible to determine the correct solution in a single step. This forcefully means that even when the local features are perfectly discriminating, each vertex $\mathbf{v}_i$ might have multiple ambiguous correspondences due to incompleteness or local symmetries in the shape. Consequently, the registration problem is further divided into a *coarse alignment step* and a *refinement step*. As opposed to coarse registration, the refinement process assumes ϕ_{rigid} to be rather small and tightly couples correspondence estimation and transformation computation (typically an iterative process).

3.1.2 Coarse Alignment

Bringing two scans (that are separated by a large rigid motion) into rough alignment requires the identification of sufficient corresponding feature points between both surfaces. Since only a subset $\mathcal{S}_{t_1 \cap t_2}$ is shared by the two scans, globally aligning both data, $\mathcal{S}_d(t_1)$ and $\mathcal{S}_d(t_2)$, is generally not possible by simply looking at the distribution of the entire surface geometry (e.g., via PCA normalization). However, when looking at a small geodesic patch $\mathcal{P}_r(\mathbf{v}_i) \subseteq \mathcal{S}_d(t_1)$ of geodesic radius r and center $\mathbf{v}_i$, a matching patch $\mathcal{P}_r(\mathbf{c}_i) \subseteq \mathcal{S}_d(t_2)$ could be determined if $\mathcal{P}_r(\mathbf{v}_i) \subseteq \mathcal{S}_{t_1 \cap t_2}(t_1)$. From this observation, many coarse alignment techniques sparsely sample both triangle meshes and use shape descriptors to characterize local surface geometries about each of these samples. Similar local shape descriptors between two scans form a set of candidate correspondences. Due to noise, incomplete data, incoherent sampling, and possible false correspondences, the estimated motion Φ_{rigid} usually remains suboptimal. Note that more discriminative descriptors produce less ambiguous matches, but are also more sensitive to noise and partial data. On the other hand, less discriminative descriptors (e.g., by simply choosing

a small radius r) are more robust but also introduce additional ambiguity, increasing the difficulty of correspondence computation.

Pose-Invariant Shape Descriptors. Many types of shape descriptors have been investigated, some being directly adapted from global shape matching algorithms. The goal is to define a local signature for a patch $\mathcal{P}_r$ in the form of a compact feature vector with fixed dimensions. Similarity between two feature vectors is typically measured by their $\mathcal{L}^2$ distance. Global shape matching techniques are generally designed to be invariant w.r.t. Euclidean transformations. For instance, the shape histograms from Ankerst and colleagues [AKKS99] characterize 3D geometry by decomposing the enclosing space into a set of concentric shells. Since a local patch is described by a statistical distribution, their shape descriptor is invariant of the underlying coordinate frame. Shape descriptors based on spherical harmonics [KFR03] store a set of rotation invariant frequency components and represent another pose-invariant approach. Translation invariance can be obtained by translating the centroids to the origin. Note that feature vectors of large dimensions can be assisted with dimension reduction techniques (such as PCA) or indexing techniques for better efficiency.

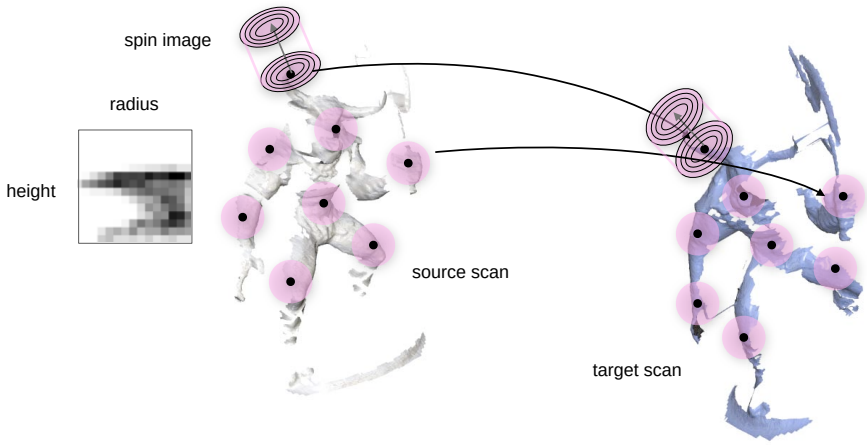


Figure 3.3: Example of global rigid motion invariant shape descriptor based on spin images.

Surface Oriented Shape Descriptors. A different type of descriptor considers surface orientation and measures local geometry distribution about a surface sample and uses its normal to normalize for two of the three degrees of rotational freedom. In particular, transformation invariance is reduced to a single angular dimension (around

the normal) leading to potentially higher discriminative power than pose-invariant descriptors. Johnson and Herbert's spin images [JH97], for example, store the average of the surface area of a normal ring with fixed radius and height (c.f. Figure 3.3). The method of Frome and coworkers [FHK$^+$04] suggests storing amplitudes of frequency components of each normal ring. Finally, a more descriptive method, known as 3D Shape context [BMM00], performs an exhaustive 1D search over all normal angles of rotation to determine the alignment with the maximum response. Note that the latter approach should be used carefully as it is highly sensitive to noise and local symmetries. While originally designed for approximate and partial symmetry detection, the curvature-based descriptors used in [MGP06] can also be used for rigid motion estimation. The method densely samples the surface and defines a two-dimensional local signature simply as a pair of principal curvatures $(k1, k2)$. Although each signature is highly ambiguous, a clustering of a large set of possible correspondences in a Euclidean transformation space may reveal the most likely rigid motion. This approach provides a more refined and complete description of the underlying surface but also incurs a higher computational cost (due to the clustering step).

Pairwise Correspondence Assignment. Shape descriptors characterize local geometric features and establish potential candidates for correspondences between two different scans. Potential candidates are samples on the target mesh with feature vector distance below a certain threshold. We now describe several fundamental techniques for establishing one-to-one correspondences between two sets of surface samples. As mentioned above, searching over all possible candidate correspondences and determining the transformation that minimizes E_{fit} leads to an exponential explosion in complexity. While a greedy approach of iteratively picking the best possible assignment among the candidates (and discarding this assignment from subsequent matches) would be most efficient, it is in practice not robust enough.

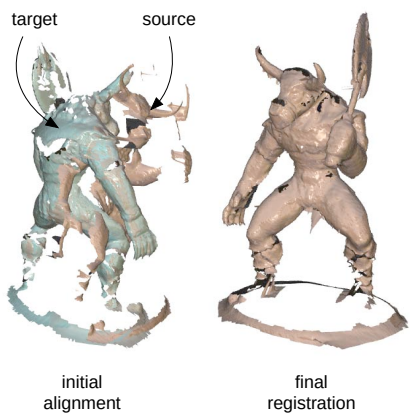
Branch and Bound Approach. One way to leverage the combinatorial intractability when matching independent features is to exploit spatial consistency of feature locations. Gelfand and colleagues [GMGP05] propose a *branch and bound* approach that incrementally adds two pairs of correspondences, $(\mathbf{v}_i, \mathbf{c}_i)$ and $(\mathbf{v}_j, \mathbf{c}_j)$ into a decision tree. Early termination can be predicted in the hierarchy when inter-feature distances are not preserved, i.e., $\|\mathbf{v}_i - \mathbf{v}_j\|_2^2 - \|\mathbf{c}_i - \mathbf{c}_j\|_2^2 > 2r$.

RANdom SAmple Consensus. In the presence of multitudinous outliers, robust estimators such as random sample consensus (*RANSAC*) approach [FB87] could be more effective. The idea consists of randomly picking three feature points on the source scan and candidates on the target. These three pairs of correspondences form a unique *candidate* rigid body motion $(\tilde{R}, \tilde{\mathbf{t}})$. For every other source samples $\mathbf{v}_i$, we determine the candidate correspondences $\mathbf{c}_i$ that are closest to $\tilde{R}\,\mathbf{v}_i + \tilde{\mathbf{t}}$ and measure E_{fit}. The entire process is repeated m times and the resulting rigid motion is the one with minimal E_{fit}. Lately, Aiger and colleagues [AMCO08] extended the idea of RANSAC with the extraction of coplanar 4-points sets that are approximatively congruent under rigid body motion. This method enables robust alignment with an order of magnitude faster than previous randomized algorithms.

Spectral Approach for Spatially Consistent Correspondences. An alternative to RANSAC based matching that also considers spatial consistency is to use the spectral correspondence approach developed by Leordeanu and Hebert [LH05]. In this method, a graph adjacency matrix (*affinity*) matrix is constructed that takes into account matching scores between candidate features as well as how compatible pairs of these correspondences are. Using Raleighs ratio theorem, it can be shown that the principal eigenvector of the affinity matrix maximizes the score of the matrix in a continuous setting. To extract a discrete assignment, they employ a greedy algorithm that iteratively picks the assignment of the maximum eigenvector and discards all potential candidates in conflict with this assignment. The process is repeated until no correspondences are possible. The accuracy and robustness of this method has been successfully demonstrated on a variety of 3D registration problems [LH05, HAWG08, dAST+08].

3.1.3 Registration Refinement

Having described the fundamental algorithms for coarse shape registration, we may ask ourselves why the refinement computation would take a conceptually different approach. How should refinement algorithms be designed to maximize effectiveness? Firstly, coarse alignment methods discretize a pair of scans by sparsely sampling surfaces and quantifying the local neighborhoods with shape descriptors. While a dense sampling would



better approximate the surface, it is computationally intractable. Since both shapes are being sampled independently, both surfaces have a different discretization. Even when the solution of the combinatorial correspondence problem is globally optimal, it is, in practice, unlikely that their (higher resolution) representative surfaces are optimally aligned (*discretization error*). Furthermore, since individual features are described by pose-independent shape descriptors, the optimal transformation is the same regardless of the initial orientation of the source scan. On one hand, a coarse alignment algorithm has the ability to resolve for arbitrarily large rigid transformation, on the other hand, the correspondence search of each feature is completely independent of the global transformation, resulting in possible outlier correspondences (*decreasing reliability*). Note that the latter observation is partly leveraged (in a combinatorial sense) by RANSAC and spectral methods but not fully exploited as in the continuous setting of refinement techniques. In particular, none of the coarse alignment methods determine correspondences based on the quality of the initial pose, while registration refinement methods do.

The structure of refinement techniques differs from their sister algorithms for coarse alignment in two ways: they involve dense surface sampling and unify correspondence and transformation optimization during alignment. To enable the latter, they approximate a continuous optimization problem with a tight coupling between correspondence estimation in small steps (similar to gradient descent methods) and optimal rigid motion computation. This coupling is typically achieved using an iterative approach interleaving the two steps. Ideally, both subproblems should improve each other in each iteration. Note that methods exist that solve both steps within a single optimization [Fit01]. In this way, outlier correspondences can be better prevented as each optimal transformation step introduces a certain regularization into the optimization (in terms of global spatial consistency). To summarize, we need a correspondence estimation method that is efficient enough to enable dense surface sampling and can be improved with each step of transformation optimization. The general approach herein is to use correspondences based on proximity heuristics (e.g., closest points) and to assume that the transformation Φ_{rigid} that separates source and target scans are within a certain threshold.

Iterative Closest Point Algorithm. The most widely adapted pairwise refinement technique is the iterative closest point (*ICP*) algorithm originally developed by Besl and McKay [BM92]. The algorithm is conceptually trivial and is guaranteed to converge to a

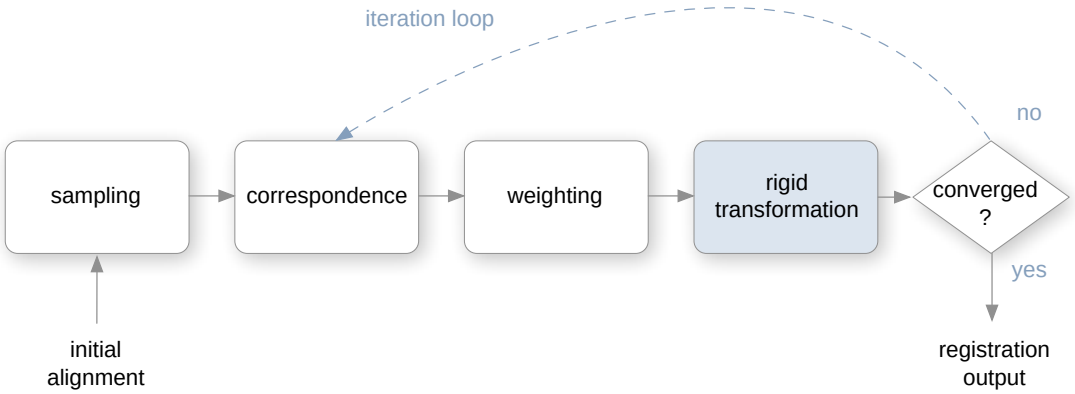


Figure 3.4: Rigid ICP pipeline for registration refinement.

local optimum. Assuming source and target are sufficiently close, ICP first computes for each vertex $\mathbf{v}_i \in \mathcal{S}_d(t_1)$ the closest point $\mathbf{c}_i$ on the target surface $\mathcal{S}_d(t_2)$. For very large meshes, a random resampling of surface points might suffice. The second step consists of solving for the optimal rigid motion $\Phi_{\text{rigid}} = (R, \mathbf{t})$ by minimizing E_{fit} from Equation 3.2. Note that correspondence pairs that are very far apart can be regarded as outliers and rejected. One option is to discard pairs $(\mathbf{v}_i), \mathbf{c}_i)$ with distances larger than a scalar multiple of their median, i.e., $\|\mathbf{v}_i - \mathbf{c}_i\| > k\, l_{\text{median}}$. The entire process is repeated until E_{fit} converges. A simple convergence criteria can be $|E^k_{\text{fit}} - E^{k-1}_{\text{fit}}| < \epsilon\,(1 + E^k_{\text{fit}})$ where E^k_{fit} is the energy in the kth iteration. The success of ICP mainly relies on how well the geometric features between both scans can be used for matching (c.f. Figure 3.5). For featureless regions and ambiguous features, the algorithm simply converges to the closest matching one.

Since its introduction, a large number of ICP variants have been proposed and their convergence behavior well-studied [PHYH06]. An extensive survey on different ICP adaptations can be found in Rusinkiewcz and Levoy [RL01]. Following their classification methodology, most rigid registration refinement algorithms can be described with the pipeline shown in Figure 3.4. We will now recapitulate the most important techniques. Most ICP methods are characterized by the following stages:

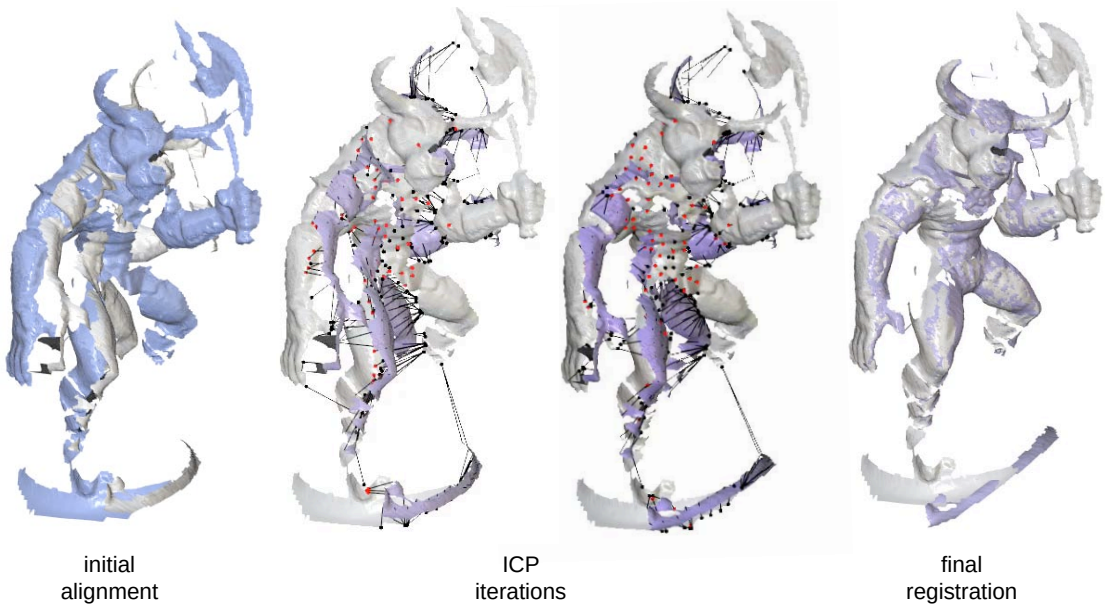


Figure 3.5: The iterative closest point algorithms alternates between closest point computation and optimal transformations.

Sampling. Before computing correspondences, the source scan may be resampled with a lower vertex density $\hat{\mathcal{V}} \subseteq \mathcal{V}(t_1) \subset \mathcal{S}_d(t_1))$ in order to avoid dealing with overly high resolution meshes. Note that the target scan is not resampled as opposed to coarse alignment techniques. While random resampling is often used due to its efficiency, there might still be chances that $\hat{\mathcal{V}}$ does not fully capture all the features that are crucial for accurate matching with the target scan. Since it is unknown in advance which geometric features should be used for matching (i.e., those residing in $\mathcal{S}_{t_1 \cap t_2}$), a good solution consists of simply uniformly resampling the original mesh. While the isotropic remeshing algorithm in Section 2.5 may be used to obtain a uniformly sampled mesh with user specified density, the algorithm does not perform well for disconnected and cluttered surfaces which is common for scanned data. Additionally, since all vertices are relocated, an extra closest point search to the original mesh needs to be performed in order to keep the original vertex positions. A straightforward resampling method designed for dense regular meshes (with possible clutters and holes) consists of iteratively discarding all vertices within a ball of radius r and keeping the center position of the ball. This algorithm can be summarized as follows:

1. Initialize resampling list $\hat{\mathcal{V}} = \{\mathbf{v}_1\}$.

2. For all vertices $\mathbf{v}_i \in \mathcal{V}(t_1)$ with $i = 2, \ldots, N_{\mathcal{V}}$:

   - If $\|\mathbf{v}_i - \hat{\mathbf{v}}_j\|_2^2 > r$ for all vertices $\hat{\mathbf{v}}_j \in \hat{\mathcal{V}}$, add $\mathbf{v}_i$ into $\hat{\mathcal{V}}$.

It becomes obvious that for a reasonably small size of r, the algorithm performs linearly on average (worst case scenario is $\mathcal{O}(N_{\mathcal{V}}\, N_{\hat{\mathcal{V}}})$). In some cases the input mesh discretization might be highly irregular possibly due to mesh simplification algorithms or because it emanates from a handcrafted 3D model. For these polygonal surfaces, uniform resampling can be achieved using Turk's particle repulsion approach [Tur92]. Note that in some applications with highly imbalanced ratio of (very little) geometric details and (large) featureless regions, the sampling strategy can have a significant impact in the alignment accuracy and convergence speed. In the context of cultural heritage, Gelfand and coworkers [GRIL03] propose a stability analysis which maximizes the relevance of surface samplings to constraint all different degrees of freedom of the transformation. A simpler and more efficient technique based on normal space sampling is presented in [RL01] and selects surface samples in such way that the resulting points have a uniform distribution of their normals.

Correspondence. Since we are attempting to minimize the distance of overlapping surface regions, a natural choice for proximity heuristics for a vertex $\mathbf{v}_i \in \mathcal{V}(t_1)$ are the closest vertices on the target mesh $\mathbf{c}_i \in \mathcal{V}(t_2)$. They can be efficiently determined in $\mathcal{O}(\log N_{\mathcal{V}})$ using a spatial data-structure such as a kd-tree [Ben75]. Note that for target meshes with irregular samplings or low polygonal count, it is often more convenient to compute the closest point on the target triangle mesh $\mathbf{c}_i \in \mathcal{S}_d(t_2)$. For simplicity, this thesis mainly considers high-resolution uniform meshes as input.

Certain applications, such as real-time shape completion [RHHL02] or online facial tracking [WLGP09], require closest point queries at interactive rates. Being the slowest component of the ICP pipeline, using a kd-tree would be still too slow. Alternatively, we may assume the scans, $\mathcal{S}_d(t_1)$ and $\mathcal{S}_d(t_2)$, to be sufficiently close such that the closest point can be approximated with a projection in z direction. For depth maps, we simply assume $\mathbf{v}_i = \mathbf{s}_i(\mathbf{u})$ to have the closest point $\mathbf{c}_i = \mathbf{s}_j(\mathbf{u})$ with same uv coordinates. This projection is constant in time as it only consists of evaluating the depth value on $\mathcal{S}_d(t_2)$.

Weighting. Especially for uniformly resampled source scans $\mathcal{S}_d(t_1)$, using all correspondences for the minimization $\underset{R,\mathbf{t}}{\operatorname{argmin}}\, E_{\text{fit}}$ would generally yield a suboptimal rigid

motion. In fact, only correspondences within $\mathcal{S}_{t_1 \cap t_2}$ should be used for the matching. Since this region of overlap is unknown, we may assign discrete weights $w_i \in \{0, 1\}$ for each correspondence pair $(\mathbf{v}_i, \mathbf{c_i})$ and optimize for those as well. The energy functional becomes:

$$E_{\text{fit}} = \sum_{\mathbf{v}_i \in \mathcal{V}(t_1)} w_i \|(R\,\mathbf{v}_i + \mathbf{t}) - \mathbf{c}_i\|_2^2 \quad . \tag{3.3}$$

The discrete optimization of w_i is typically performed after correspondence estimation and before rigid transformation computation. In this way, we tightly couple correspondence estimation, overlapping regions computation, and optimal rigid transformation. Note that setting $w_i = 0$ is equivalent to pruning the correspondences (i.e., removing them from the equation of E_{fit}). The only difference lies in the implementation trading-off between pre-processing and run-time efficiency [RBK05]. While a combinatorial approach such as for coarse rigid alignment can be employed, simple heuristics are usually sufficiently effective. The most utilized pruning strategies include setting $w_i = 0$ when:

- The corresponding points are further away than a prescribed threshold as in the original ICP formulation using medians or simply by discarding correspondences with $\|\mathbf{v}_i - \mathbf{c}_i\|_2^2 > \sigma_{\text{distance}}$.

- The corresponding point $\mathbf{c}_i$ lies on the boundary of the target mesh $\mathcal{S}_d(t_2)$. This heuristic is based on the observation that, for scanned data, source vertices outside the overlap regions $\mathbf{v}_i \in \mathcal{S}_d(t_1) \backslash \mathcal{S}_{t_1 \cap t_2}(t_1)$ often have their closest points mapped on the target mesh boundary. In practice, this pruning criterion should be combined with distance thresholding since the closest point outside the overlap region might also lie inside $\mathcal{S}_d(t_2)$ (especially when the source shape is a closed manifold).

- The normals of source and target vertex, $\mathbf{n}(\mathbf{v}_i)$ and $\mathbf{n}(\mathbf{c}_i)$, are incompatible, i.e., $\mathbf{n}(\mathbf{v}_i)^\top\, \mathbf{n}(\mathbf{c}_i) > \sigma_{\text{angle}}$. For noisy data, robustly estimated surface normals should be used such as those described in Mitra and Nguyen [MN03]. When mesh connectivity is available, we may simply use normals obtained through mesh smoothing.

Note that in each ICP iteration these weights are re-evaluated. In addition to identifying the region of overlap, these heuristics also improve the robustness of the closest point heuristics. Additionally, an interesting pruning technique based on bi-directional reprojection of closest points has been introduced in [PMG+05]. The method reprojects the closest point $\mathbf{c}_i \in \mathcal{S}_d(t_2)$ of a source vertex $\mathbf{v}_i \in \mathcal{S}_d(t_1)$ onto the closest point

of the original surface $\mathcal{S}_d(t_1)$ and evaluates the new distance. If this distance is larger than a threshold, the correspondence is considered incompatible and discarded. While faster convergence can be achieved in some cases, this approach comes at the price of constructing a new kd-tree construction on the source scan in each iteration.

Rigid Transformation. Once the correspondences are specified and weighted (or pruned), we may update the pose of the source scan $\mathcal{S}_d(t_1)$ by minimizing the fitting energy of Equation 3.3. As described in the original version of ICP [BM92], the closed form solution from Section 3.1.1 minimizes this integration of squared *point-to-point* distances between the correspondences. Although correspondence and transformation optimizations are tightly coupled in this iterative framework, it is well-known that, for point-to-point error metric, large featureless regions may penalize tangential surface motion and, hence, cause low convergence rate. Consider the example where a large number of correspondences are being found on a relatively large region that does not exhibit significant details. Even if some correspondences suggest the source scan to glide in this region to match a certain feature, E_{fit} would be dominated by positional constraints in those featureless regions. This observation suggests the use of an energy term that does not penalize gliding of correspondence points on the target surface.

Point-to-Plane Distance. A common approach is to use the *point-to-plane* error metric for optimal rigid body alignment introduced by Chen and Medioni [CM92]. Instead of minimizing the distance between each source vertex $\mathbf{v}_i$ and its corresponding point $\mathbf{c}_i$, the idea is to locally approximate the shape of the target surface at each point $\mathbf{c}_i$ by the tangent plane $\mathcal{T}(\mathbf{c}_i) = \{\mathbf{x} \in \mathbb{R}^3 | \mathbf{n}_i^\top (\mathbf{x} - \mathbf{c}_i) = 0\}$ and minimize the distance of $\mathbf{v}_i$ to this plane. Here, $\mathbf{n}_i$ is the normal of $\mathbf{c}_i$. The point-to-plane metric yields the following energy functional term:

$$E_{\text{fit}} = \sum_{\mathbf{v}_i \in \mathcal{V}(t_1)} w_i \left(\mathbf{n}_i^\top \left(R\,\mathbf{v}_i + \mathbf{t} - \mathbf{c}_i \right) \right)^2 \quad . \tag{3.4}$$

Notice how transformation computation is now coupled with (approximate) correspondence optimization within a single step of E_{fit} minimization. This coupling and first order approximation of the target surface drastically improves convergence rate as well as robustness as demonstrated in the thorough analysis by Pottmann and colleagues [PHYH06]. We now look at how to solve for the optimal rigid motion $(R, \mathbf{t})$ subject to the minimization of the energy term in Equation 3.4.

Since no closed form solution exist for this non-linear optimization problem, we may directly linearize E_{fit} and solve for an over-constrained linear system. Linearization is obtained by assuming rotations to be sufficiently small and taking the first-order approximations of the sine and cosine functions in the rotation matrix R. In particular, we make the assumption that $\sin\theta \approx \theta$ and $\cos\theta \approx 1$ for small angles $\theta = \epsilon$. Substituting the matrix representation of Euler angles, i.e., $R = R_y(\theta_y)\, R_x(\theta_x)\, R_z(\theta_z)$, we obtain the following approximation:

$$R \approx \hat{R} = \begin{bmatrix} 1 & -\theta_z & \theta_y \\ \theta_z & 1 & -\theta_x \\ -\theta_y & \theta_x & 1 \end{bmatrix} = I + \begin{bmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{bmatrix} = I + S \tag{3.5}$$

with S the corresponding skew-symmetric matrix of R. The energy term with linearized rotation matrix and Euler angles $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)^\top$ becomes:

$$\begin{aligned}
\hat{E}_{\text{fit}} &= \sum_{\mathbf{v}_i \in \mathcal{V}(t_1)} w_i \left(\mathbf{n}_i^\top \left(\hat{R}\, \mathbf{v}_i + \mathbf{t} - \mathbf{c}_i \right) \right)^2 && (3.6) \\
&= \sum_{\mathbf{v}_i \in \mathcal{V}(t_1)} w_i \left(\mathbf{n}_i^\top \left(I + S \right) \mathbf{v}_i + \mathbf{n}_i^\top\, \mathbf{t} - \mathbf{n}_i^\top\, \mathbf{c}_i \right)^2 && (3.7) \\
&= \sum_{\mathbf{v}_i \in \mathcal{V}(t_1)} w_i \left(\mathbf{n}_i^\top\, \mathbf{v}_i + \mathbf{n}_i^\top \left(-\mathbf{v}_i \times \boldsymbol{\theta} \right) + \mathbf{n}_i^\top\, \mathbf{t} - \mathbf{n}_i^\top\, \mathbf{c}_i \right)^2 && (3.8) \\
&= \sum_{\mathbf{v}_i \in \mathcal{V}(t_1)} w_i \left(\left(\mathbf{v}_i \times \mathbf{n}_i \right)^\top \boldsymbol{\theta} + \mathbf{n}_i^\top\, \mathbf{t} - \left(\mathbf{n}_i^\top \left(\mathbf{c}_i - \mathbf{v}_i \right) \right) \right)^2 && (3.9)
\end{aligned}$$

Setting $\mathbf{x} = (\boldsymbol{\theta}^\top, \mathbf{t}^\top)^\top$, the matrix notation of the linearized energy term becomes $\hat{E}_{\text{fit}} = \|A\,\mathbf{x} - \mathbf{b}\|_2^2$ which is simply an *ordinary least squares* problem. Hence, the minimizer of $\hat{E}_{\text{fit}}$ is the solution of the *overdetermined linear system* $A\mathbf{x} = \mathbf{b}$ which can be solved using *normal equation* (provided $A^\top A \in \mathbb{R}^{6\times 6}$ is invertible), i.e., $\mathbf{x} = (A^\top A)^{-1} A^\top \mathbf{b}$. However, it is well-known that the normal equation may be ill-conditioned in some cases (squaring very small numbers may cause numerical instabilities). Alternatively a QR-factorization approach can be numerically more stable and equally efficient using implicit solvers based on Householder transformations. To obtain $R \in \mathrm{SO}(3)$ we simply orthonormalize it using polar decomposition [FAT07].

Putting It All Together. When only mesh vertices on the target surface are considered, computing the closest point becomes a discrete process. Consequently, undesirable oscillations between the rigid motion estimation may appear toward the end of the ICP

refinement. To improve stability, a typical approach consists of combining the point-to-plane and point-to-point metric, E_{plane} (c.f. Equation 3.3) and E_{point} (c.f. Equation 3.4), and solve for the motion $\mathbf{x}$ that minimizes $E_{\text{plane}} + \alpha\, E_{\text{point}}$. In particular, incorporating the point-to-point constraint helps to promote convergence of the overall energy. We typically choose a small weight $\alpha = 0.1$. To summarize, when carefully combining all the fundamental techniques discussed in the ICP stages, extremely large rigid motions can be accurately recovered—even without intervention of coarse rigid alignment algorithms. In particular, our robust non-rigid registration framework in Section 3.4 will set its foundations based on these key insights.

3.2 Surface Deformation

Before we proceed to non-rigid registration algorithms, we first summarize the most important surface deformation techniques that are relevant for non-rigid alignment and other problems in animation reconstruction. In general, we may describe surface deformation of a triangle mesh $\mathcal{S}_d$ as a mapping $\Phi : \mathcal{S}_d \to \mathbb{R}^3$ where $\mathbf{v}_i \mapsto \Phi(\mathbf{v_i}) = \tilde{\mathbf{v}}_i$. If we knew the mapping of all vertices, the deformation for $\mathcal{S}_d$ would be fully defined. However, this is not the case for many applications such as mesh editing where users provide only a few geometric constraints to manipulate the shape (e.g., vertex displacements, orientation of local frames...). For non-rigid registration problems, only a subset of correspondences can be reliably determined within the overlap $\mathcal{S}_{t_1 \cap t_2}$—again, how should the remaining surface $S(t_1) \setminus \mathcal{S}_{t_1 \cap t_2}$ deform? When some of the correspondences are wrong, can we compensate those with a plausible deformation model? It becomes apparent that finding the right surface deformation is closely related to an interpolation or data-fitting problem. Since it is hard to characterize the deformation of an arbitrary subject, our deformation model must be flexible enough to capture a maximum range of shapes but also has to be resistant to unnatural distortions.

Regularization. Deformation models are typically associated with a *regularization* or *smoothness* component that specifies the overall change of shape subject to certain geometric and/or physical surface properties (e.g., as smoothness, lengths, curvature...). The characterization of regularization varies largely depending on the underlying model. In the context of geometric design for example, spline and subdivision surfaces can be directly deformed through control vertices [PBP02, Far02] to produce smooth surfaces. However, it is difficult to automatically generate such surface representation from scanned data as a large number of carefully laid out patches would be necessary to

model complex shapes. We therefore consider deformation models that are decoupled from the surface representation. Consequently, prescribed geometric constraints (either from captured data or user guided) should prevent neighboring parts of vertices to get mapped to disparate positions. As we will see later, certain deformation models allow us to specify the *amount of regularization* during deformation which often helps to model the "stiffness" of a certain surface.

Hard vs. Soft Constraints. While for mesh editing purposes, it is generally desirable to exactly interpolate user-prescribed constraints, fitting scanned data might not be the case. Hard constraints are appropriate to use when the data is highly reliable such as with sparse markers obtained from motion capture [BLB+08]. However, in many scenarios, imposing soft constraints can be crucial for several reasons. For example, since a reliable and accurate correspondence estimation on markerless data is hard, we may not want to fully rely on these matches, but rather trust the deformation prior. Furthermore, we also know that the scanned subject may change its shape in unknown ways and its scans are generally affected by noise and outliers. Using soft constraints, deformations usually do not exactly interpolate the input data, but rather attempt to satisfy the regularization imposed by the deformation model. Depending on the approach and the surface representation, soft constraints may also be considered to avoid over-fitting issues.

Local vs. Global Deformation. We may further distinguish between deformation models that operate only within a region of interest (local support) and some that change the entire shape (global support). When aligning scans, it is important to not only warp the common subregion between the two surfaces, but also define the deformation for $S(t_1) \setminus \mathcal{S}_{t_1 \cap t_2}$. In these cases, we often use the regularization parameter to describe how spatially consistent the overall deformation should be. Notice that a global support may be defined in space or on the surface. Deformation models with global support are crucial elements in a registration framework since they promote a tighter coupling between global transformation and correspondence computation. Intuitively, when a large number of good correspondences are found, the motion of non-overlapping regions will follow these matchings. On the other hand, errors due to false correspondences would be evenly distributed across the entire shape instead of being localized.

Space Deformation. Some of the first deformation algorithms that are designed to handle arbitrary input surface representations are based on spatial deformations. A survey is presented in Bechmann [Bec94]. The idea is to first define a warping field on

the ambient space $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ which, in a second step, infers the deformation on the points of the embedded surface $\mathcal{S} \subset \mathbb{R}^3$. Early work on spatial deformation techniques include the intuitive local and global geometric operations used for designing new solid shapes by Barr [Bar84]. For general deformations, these operations are too restrictive and unsuitable for data-fitting or interpolation problems.

One way to improve flexibility consists of using a control object (3D lattice) as a proxy to parameterize spatial deformations. While early *free form deformation* techniques [SP86] use lattices that simply consist of a discretized bounding box, most modern cage-based techniques provide a better approximation of the underlying surface [JSW05, LKCOL07, JMD+07]. These methods differ mainly in the way they interpolate the interior of the cage and how well they preserve local features. However, most cage-based deformation algorithms require carefully hand-crafted lattices around the underlying subject.

Approaches based on radial basis function (*RBF*) [CFB97, KSSH02, BK05] are often used to warp 3D models to captured data [CLK01, CK05] due to their efficiency and stability. The drawback however is that their influence regions do not capture surface geodesics and local features may not be well preserved. Hence, they are suitable for handling extremely large data but are restricted to small deformations. For the alignment of very large scan data, the use of more sophisticated spatial technique based on thin-plate splines [SS91] was first proposed in Brown and Rusinkiewicz [BR04] where a global deformation is computed that minimizes spatial curvature of the warping field. A regularization term allows the user to specify the optimal balance between interpolation and smoothness depending on the amount of noise and distortion in the input data. Because the method uses soft-constraints, it permits inaccurate correspondence estimates as opposed to RBF methods. However, regularization is expressed by a global support in space which makes it less suitable for large deformations. Since spatial deformation techniques are particularly effective and robust for partial and cluttered data (such as scans) we will present an efficient graph-based method in Section 3.2.4 which defines global regularization over the surface and can handle large deformations.

Surface-Based Deformation. Surface-based deformation techniques directly operate and define regularization on the surface. While being inappropriate for warping sets of disconnected surface portions (such as scans), they optimally exploit surface topology and are remarkably effective in preserving intrinsic surface properties. In data fitting

and tracking problems, surface deformation models are typically employed whenever templates are involved as they generally represent a complete surface. The most prominent techniques include physically-inspired models and several approaches based on differential coordinates (Laplacian and gradient-based representations). A recent survey on linear surface-based methods is described in Botsch and Sorkine [BS08]. We will review these methods in more detail in Section 3.2.1, 3.2.2, and 3.2.3. Let us first analyze two simple deformation models that are often used in shape matching problems with dense triangles meshes.

Suppose the deformation of a triangle mesh with vertices $\mathbf{v}_i \in \mathcal{S}_d$ is described by an additive displacement field $\tilde{\mathbf{v}}_i = \Phi(\mathbf{v}_i) = \mathbf{v}_i + \mathbf{d}_i \in \tilde{\mathcal{S}}_d$ with $i = 1, \ldots, N_{\mathcal{V}}$. An uncomplicated way to deform the mesh would be to specify a few positional constraints $\mathbf{c}_j \in \mathbb{R}^3$ such that the distance between $\mathbf{v}_j$ and $\mathbf{c}_j$ is minimized. W.l.o.g., we assume the first $N_{\mathcal{C}} \leq N_{\mathcal{V}}$ vertices to have correspondences, i.e., $j = 1, \ldots, N_{\mathcal{C}}$. To this end, we formulate the data fitting as an energy minimization of point-to-point squared distances:

$$E_{\text{fit}} = \sum_{i=1}^{N_{\mathcal{C}}} \|\mathbf{v}_i + \mathbf{d}_i - \mathbf{c}_i\|_2^2 \tag{3.10}$$

For a fitting problem with soft constraints, we would like to determine the displacement $\mathbf{d}_i$ of all vertices $\mathbf{v}_i$ subject to a global regularization term. Obviously, fitting the data by ignoring regularization and keeping the unconstrained vertices in their original positions would not result in an attractive mesh. One straightforward approach consists of encouraging neighboring vertices to displace similarly, as described in the facial tracking framework [ZSCS04]. This regularization is described with the following energy term:

$$E_{\text{reg}} = \sum_{i=1}^{N_{\mathcal{V}}} \sum_{\mathbf{v}_j \in \mathcal{N}(i)} \frac{1}{\|\mathbf{v}_i - \mathbf{v}_j\|_2^2} \|\mathbf{d}_i - \mathbf{d}_j\|_2^2 \quad . \tag{3.11}$$

Notice how neighboring displacements are normalized with the distance of the edges that connect their source vertices. In this way, the displacements of vertices that are closer have more influence on each other. The final step consists of solving for the $\mathbf{d}_i$ by minimizing the combined energy:

$$E_{\text{tot}} = E_{\text{fit}} + \alpha_{\text{reg}}\, E_{\text{reg}} \tag{3.12}$$

with regularization parameter α_{reg}. Choosing a large value for α_{reg} would increase stiffness of the deformation and a small value would better interpolate the constraints. From

a differential geometry stand point, we may note that this simple formulation is equivalent to an approximation of a (linearized!) bending energy and the same as performing a deformation using Laplacian coordinates (c.f. Section 3.2.2). In fact, the minimizer of Equation 3.11 alone are the roots of a sum of discrete Laplacians weighted with inverse edge lengths $\Delta_{\mathcal{S}}\mathbf{d}_i = \sum_{\mathbf{v}_j \in \mathcal{N}(i)} \frac{1}{\|\mathbf{v}_i - \mathbf{v}_j\|^2}(\mathbf{d}_i - \mathbf{d}_j)$, as originally proposed by Fujiwara [Fuj95]. However, determining the least squares minimizer of the overdetermined system in Equation 3.12 requires solving a normal equation. Consequently, the solution closely approximates a bi-Laplacian equation which minimizes surface bending. In Section 3.2.1, we will present a more rigorous derivation for bending-minimizing surfaces which clarifies how this solution relates to the minimization of change in curvature in unconstrained regions. While this particular choice of discrete Laplacian attempts at preserving the edge length ratios before and after deformation, it does not consider unevenly distributed angles of the triangles as with cotangent weights (c.f. Equation 2.15). This simple deformation model is particularly effective when dense and reliable correspondences are available such as for template tracking with high-resolution input scans. When correspondences are sparse and not evenly distributed, larger regions without vertex constraints may in certain scenarios deform unnaturally. In particular, when the deformed subject is supposed to change its volume, this deformation model would loose its accuracy. Nevertheless, we call this type of deformation *physically-inspired* since it mimics real-world elastic behaviors.

Alternatively, several researchers [ACP03, PMG+05, SP04] suggest a purely geometric approach where deformation regularization is accomplished by enforcing the affine transformations of neighboring vertices to be alike. The difference with the previous approach lies in the deformation representation which is now generalized to $\Phi(\mathbf{v_i}) = A_i\,\mathbf{v}_i + \mathbf{a}_i$ where $A_i \in \mathbb{R}^{3\times3}$ and $\mathbf{a} \in \mathbb{R}^3$. Describing a deformation with local affine transformations is more flexible than with pure displacements, since, in addition to a shift component, rotation, shear, and scale are implicitly expressed in the model. Let us be more concrete. The new fitting term is now formulated as:

$$E_{\text{fit}} = \sum_{i=1}^{N_{\mathcal{C}}} \|A_i\,\mathbf{v}_i + \mathbf{a}_i - \mathbf{c}_i\|_2^2 \tag{3.13}$$

and the regularization term:

$$E_{\text{reg}} = \sum_{i=1}^{N_{\mathcal{V}}} \sum_{\mathbf{v}_j \in \mathcal{N}(i)} \|\,[A_j|\mathbf{a}_j] - [A_i|\mathbf{a}_i]\,\|_F^2 \tag{3.14}$$

where $\|\cdot\|_F^2$ denotes the *Frobenius* norm. When minimizing the new combined term $E_{\text{tot}} = E_{\text{fit}} + \alpha_{\text{reg}} E_{\text{reg}}$, the deformation energy required to minimize E_{fit} is no longer concentrated in the translational component $\mathbf{a}$ but evenly diffused over the linear component A. Simultaneously, E_{reg} promotes unconstrained surface regions to scale, rotate, and shear according constrained areas. As a result, deformation models that are based on affine transformations are able to capture more general deformations than the previous one. For example, even when the regularization parameter α_{reg} is set to be a large value, it would not penalize a global linear transformation (such as scaling). One drawback however is that over-fitting is likely to occur for large unconstrained regions. This problem is often exhibited by drastic shape distortions as observed in the original work of Sumner and Popović [SP04]. The authors tackle this issue by introducing a stabilization term which encourages the non-translational component to be the identity:

$$E_{\text{stab}} = \sum_{i=1}^{N_{\mathcal{V}}} \|A_j - I\|_F^2 \quad . \tag{3.15}$$

We immediately see that this stabilization causes the overall deformation to minimize bending, similarly to the first technique (since it forces E_{fit} and E_{ref} to be similar to the earlier formulation). To summarize, when E_{stab} is added to the total energy $E_{\text{tot}} = E_{\text{fit}} + \alpha_{\text{reg}} E_{\text{reg}} + \alpha_{\text{stab}} E_{\text{stab}}$ with a control parameter α_{stab}, we may interpolate between the behavior of an affine transformation-based deformation and a bending minimization model. In practice, a more general deformation model can capture a wider range of shapes without being penalized, but is also more susceptible to the problem of over-fitting. At this point, it becomes clearer why, for purely bending-minimizing models, transformations such rotations, scaling, and shearing are not explicitly captured by a linear matrix (such as A_i) but implicitly encoded in each displacement $\mathbf{d}_i$. As we will clarify in Section 3.2.1, the regularization term is only a linear approximation of a true (non-linear) bending-minimization constraint which may cause local geometric details to be distorted for deformations with large rotations.

Linear vs. Non-Linear. The two deformation models presented above are linear, since they can be solved by setting up a large (but sparse) linear system. Notice that even for a linear deformation model, the minimization of E_{tot} may become non-linear as it is the case when using a point-to-plane distance metric for E_{fit}, similarly to the case of ICP algorithms. Consequently, we call a deformation model linear if its characterizing regularization energy E_{reg} is linear. As mentioned above, linear deformation techniques

which attempt to approximate non-linear deformations may fail at capturing important local transformations (such as rotations in the case of bending minimizing energies).

While certainly less efficient, non-linear deformations can be solved to some extent depending on the optimization formulation and the underlying non-linear solver. Most of the practical and reliable solutions perform a linearization at a more refined stage (e.g., Gauss-Newton method). Instead of linearizing a continuous shell energy using discrete Laplacians, Grinspun and coworkers [GHDS03] propose to directly define a discrete energy with similar non-linear properties such as stretching and bending. Their approach is especially well suited for simulating dynamic behaviors because of the physical accuracy. However, for large positional constraints, numerical instabilities may occur. A numerically stable non-linear deformation technique, named *PriMo*, was introduced by Botsch and colleagues [BPGK06] and trades off physical accuracy with physical plausibility. The idea is to simulate thin-shell behaviors by discretizing the surface into fine volumetric rigid cells that are coupled through non-linear, elastic forces. While having a complex structure and being computationally more expensive than many other techniques, this method can achieve highly appealing results. Similar results can also be achieved using a purely geometric approach. In fact, we may consider the previously presented model based on affine transformations and locally maximize its rigidity (instead of only prescribing local smoothness). As-rigid-as possible deformations were introduced around the same time by Sumner and coworkers [SSP07] and Sorkine and Alexa [SA07]. In particular, local rigidity can be accomplished by enforcing the non-translational components A_i to maximize their rotations (which introduces non-linear terms). Consequently, local features can be better preserved and over-fitting problems avoided without involving physically-inspired regularization terms. Section 3.2.4 will be present one efficient variant, called *embedded deformation* [SSP07], which yields comparable results to PriMo. Since this method is based on space deformations, it is immediately applicable to fragmented surfaces such as scanned data. This thesis will show that locally as-rigid-as-possible deformations are particularly important for achieving highly accurate and robust non-rigid registrations since they encourage detail preservation.

3.2.1 Physically-Based Linear Deformation

This section condenses the linear surface-based deformation model, beautifully assessed in Botsch and Sorkine [BS08], which is characterized by the minimization of elastic energies known from physics-based simulations [TPBF87]. The idea is to regular-

ize the deformation by penalizing stretching and bending energies given an initial (rest shape) configuration. By doing so, we simulate the deformation behavior of a continuous *thin shell* surface.

Again, we consider a smooth two-manifold surface $\mathcal{S}$ with surface parameterization $\mathbf{p} : \mathcal{U} \subset \mathbb{R}^2 \rightarrow \mathcal{S}$ and its deformed state $\tilde{\mathcal{S}}$. A smooth deformation $\Phi(\mathbf{p})$ transforms a point $\mathbf{p} \in \mathcal{S}$ such that $\tilde{\mathbf{p}} = \mathbf{p} + \mathbf{d} \in \tilde{\mathcal{S}}$. In particular, the displacement $\mathbf{d}(\mathbf{u})$ also has a parameterization and $\tilde{\mathbf{p}}(\mathcal{U}) = \tilde{\mathcal{S}}$.

The thin shell energy that is used to regularize $\Phi(\mathbf{p})$ can be phrased as a measure based on parameterization independent (i.e., intrinsic) surface properties derived from the first and second fundamental forms, $\mathbf{I}(\mathbf{u})$ and $\mathbf{II}(\mathbf{u}) \in \mathbb{R}^{2\times 2}$. More concretely, stretching is described by the change of surface area after deformation and bending by the change of curvature. Following [TPBF87], we may formulate the (rigid motion invariant) thin shell energy as follows:

$$E_{\text{shell}}(\mathbf{d}) = \int_{\mathcal{U}} \alpha_s \|\tilde{\mathbf{I}} - \mathbf{I}\|_F^2 + \alpha_b \|\widetilde{\mathbf{II}} - \mathbf{II}\|_F^2 \tag{3.16}$$

with $\tilde{\mathbf{I}}$ and $\widetilde{\mathbf{II}}$, the fundamental forms of $\tilde{\mathcal{S}}$. The stiffness parameters for stretching and bending are α_s and α_b. Linearization of this regularization term can be accomplished by substituting the change of the first and second fundamental forms with the first and second order partial derivatives. This approximation leads to the following quadratic energy:

$$\hat{E}_{\text{shell}} = \int_{\mathcal{U}} \alpha_s \left(\|\frac{\partial}{\partial u}\mathbf{d}\|_2^2 + \|\frac{\partial}{\partial v}\mathbf{d}\|_2^2 \right) + \alpha_b \left(\|\frac{\partial}{\partial u\partial u}\mathbf{d}\|_2^2 + 2\,\|\frac{\partial}{\partial u\partial v}\mathbf{d}\|_2^2 + \|\frac{\partial}{\partial v\partial v}\mathbf{d}\|_2^2 \right) \mathrm{d}u\mathrm{d}v \tag{3.17}$$

This energy term is minimized when its derivative becomes zero. More specifically, it can be shown through variational calculus that the minimizer is exactly the solution of the following (fourth order) Euler Lagrange PDE:

$$-\alpha_s \Delta\mathbf{d} + \alpha_b \Delta^2\mathbf{d} = \mathbf{0} \tag{3.18}$$

with Laplacian and bi-Laplacian operator:

$$\Delta\mathbf{d} = \text{div}\nabla\mathbf{d} = \frac{\partial}{\partial u\partial u}\mathbf{d} + \frac{\partial}{\partial v\partial v}\mathbf{d} \quad , \tag{3.19}$$

$$\Delta^2\mathbf{d} = \Delta(\Delta\mathbf{d}) = \frac{\partial}{\partial u\partial u\partial u\partial u}\mathbf{d} + 2\,\frac{\partial}{\partial u\partial u\partial v\partial v}\mathbf{d} + \frac{\partial}{\partial v\partial v\partial v\partial v}\mathbf{d} \quad . \tag{3.20}$$

Solving Equation 3.18 subject to some boundary constraints (e.g., hard constraints) yields the deformed surface $\tilde{\mathcal{S}}$ with stretching and bending regularization. Notice that

the solutions of $\Delta\mathbf{d} = \mathbf{0}$ and $\Delta^2\mathbf{d} = \mathbf{0}$ are each the minimizers of the pure stretching and bending energies. In general, the solution of the Euler-Lagrange PDE $(-1)^k\Delta\mathbf{d}^k = \mathbf{0}$ provides a C^{k-1} continuous surface deformation.

Since our triangle meshes $\mathcal{S}$ may not have a surface parameterization, we perform the same Laplace discretization as for Laplacian mesh smoothing (c.f. Section 2.5). We use the same discrete Laplace-Beltrami operator from Equation 2.15 which is now defined on displacement vectors instead of surface points:

$$-\alpha_s\Delta_{\mathcal{S}}\mathbf{d} + \alpha_b\Delta_{\mathcal{S}}^2\mathbf{d} = \mathbf{0} \quad . \tag{3.21}$$

Link to diffusion on meshes. Observe that this variational minimization is closely linked to the steady-state of the diffusion equation defined on two-manifolds (c.f. Equation 2.17). In fact, we may translate the linearized stretching and bending energies to surface points and obtain the linearized *membrane* and *thin-plane* energies:

$$\hat{E}_{\text{memb}} = \int_{\mathcal{U}} \|\frac{\partial}{\partial u}\mathbf{p}\|_2^2 + \|\frac{\partial}{\partial v}\mathbf{p}\|_2^2 \tag{3.22}$$

$$\hat{E}_{\text{plate}} = \int_{\mathcal{U}} \|\frac{\partial}{\partial u\partial u}\mathbf{p}\|_2^2 + 2\,\|\frac{\partial}{\partial u\partial v}\mathbf{p}\|_2^2 + \|\frac{\partial}{\partial v\partial v}\mathbf{p}\|_2^2 \quad . \tag{3.23}$$

The minimum of the linearized membrane energy $\hat{E}_{\text{memb}}$ (which measures surface area) is obtained when the steady state of the diffusion equation is reached, i.e. $\Delta\mathbf{p} = \mathbf{0}$ (for unbounded systems!). In particular, this observation explains the shrinking effect of Laplacian mesh smoothing.

Solving the PDE with hard constraints. After discretization with the Laplace-Beltrami operator, we may directly solve Equation 3.21 subject to some (hard) boundary constraints. Notice the difference between the problem of mesh deformation and Laplacian smoothing. The latter typically considers an unbounded system and user-specified time step λ as the steady state might not be the desired solution. In mesh deformation, we wish to directly minimize the linearized thin shell energy $\hat{E}_{\text{shell}}$ given certain constraints. When no constraints are provided, the mesh remains in its rest pose.

Let us rephrase the discretized Laplace-Beltrami operator $\Delta_{\mathcal{S}}$ (from Equation 2.15) applied to the deformation of the entire mesh $\Phi(\mathcal{S}_d)$ in matrix notation:

$$\begin{bmatrix} \Delta_{\mathcal{S}}\mathbf{d}_1 \\ \vdots \\ \Delta_{\mathcal{S}}\mathbf{d}_{N_{\mathcal{V}}} \end{bmatrix} = M^{-1}L_s \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{N_{\mathcal{V}}} \end{bmatrix} \tag{3.24}$$

where M is a diagonal "mass" matrix containing per vertex weights and L_s a symmetric matrix containing per edge weights. The Euler-Lagrange PDE from Equation 3.21 then becomes a sparse $N_{\mathcal{V}} \times N_{\mathcal{V}}$ linear system:

$$(-\alpha_s L + \alpha_b L^2)\mathbf{d} = \mathbf{0} \tag{3.25}$$

where $L = M^{-1}L_s$. When incorporating *hard constraints*, i.e., $\mathbf{d_j} = \tilde{\mathbf{v}}_j - \mathbf{v}_j$ for $j = 1, \ldots, N_{\mathcal{S}}$, we may move each column of L with a constrained vertex to the right-hand side and remove the respective rows of the system. The resulting system becomes:

$$(-\alpha_s \bar{L} + \alpha_b \bar{L}^2)\mathbf{d} = \mathbf{b} \tag{3.26}$$

with non-zero right-hand side $\mathbf{b} \in \mathbb{R}^{N_{\bar{\mathcal{V}}} \times 3}$ and submatrix $\bar{L} \in \mathbb{R}^{N_{\bar{\mathcal{V}}} \times N_{\bar{\mathcal{V}}}}$. While the linear system is still sparse, it is not symmetric anymore which is a problem for fast linear systems solvers. This can be easily fixed by pre-multiplying the above system by M which leads to the following symmetric and positive definite system:

$$(-\alpha_s \bar{L}_s + \alpha_b \bar{L}_s \bar{M}^{-1} \bar{L}_s)\mathbf{d} = M\ \mathbf{b} \tag{3.27}$$

which can be efficiently solved using a sparse direct Cholesky solver [SG04]. The advantage of using a Cholesky solver (as opposed to for example multi-grid methods) is that by pre-factorizing the matrix, only a back-substitution need to be performed whenever the right-hand side changes. In many real-time applications (such as mesh editing or template-based facial tracking), only the right-side is updated since the vertices that have constraints remain the same (only their positions change). Nevertheless, even when the matrix needs to be updated, a sparse symmetric system can still be solved efficiently since both, factorization and back-substitution, can be computed in linear time.

Solving the PDE with soft constraints. We illustrated an efficient solution for linearized shell energy-minimizing deformation bounded by hard constraints. We now look at how to incorporate soft constraints. Soft constraints are relatively easy to integrate into a deformation framework that solves the bi-Laplacian equation (bending) but surprisingly harder for the case of a Laplacian equation (stretching). As we will see next, this observation is due to the least-squares nature of the minimization. A bending energy-based deformation with soft constraints can be obtained by minimizing the following energy term:

$$E_{\text{tot}} = \alpha_{\text{fit}} E_{\text{fit}} + \ E_b = \alpha_{\text{fit}} \sum_{i=1}^{N_{\mathcal{C}}} \|\mathbf{d}_i - (\mathbf{c}_i - \mathbf{v}_i)\|_2^2 + \sum_{i=1}^{N_{\mathcal{V}}} \|\Delta_{\mathcal{S}} \mathbf{d}_i\|_2^2 \tag{3.28}$$

where the weight α_{fit} is used to control how close the deformation should interpolate the constrained points. Notice that since the Laplacian $\Delta_{\mathcal{S}}\mathbf{d}_i$ is linear, the minimizer of its squared norm is the solution the bi-Laplacian equation $\Delta^2_{\mathcal{S}}\mathbf{d}_i = 0$. The minimizer of the above energy term is then given by the following over-determined system:

$$\begin{bmatrix} \alpha_{\text{fit}}\, I^{N_\mathcal{C}\times N_\mathcal{C}} \mid \mathbf{0} \\ L \end{bmatrix} \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{N_\mathcal{V}} \end{bmatrix} = \begin{bmatrix} \alpha_{\text{fit}}\,(\mathbf{c}_1 - \mathbf{v}_1) \\ \vdots \\ \alpha_{\text{fit}}\,(\mathbf{c}_{N_\mathcal{C}} - \mathbf{v}_{N_\mathcal{C}}) \\ \mathbf{0} \end{bmatrix} \tag{3.29}$$

The corresponding normal equation is represented by the following $N_\mathcal{V} \times N_\mathcal{V}$ system:

$$\left[L^\top L + \begin{bmatrix} \alpha^2_{\text{fit}} I^{N_\mathcal{C}\times N_\mathcal{C}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right] \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{N_\mathcal{V}} \end{bmatrix} = \begin{bmatrix} \alpha^2_{\text{fit}}(\mathbf{c}_1 - \mathbf{v}_1) \\ \vdots \\ \alpha^2_{\text{fit}}(\mathbf{c}_{N_\mathcal{C}} - \mathbf{v}_{N_\mathcal{C}}) \\ \mathbf{0} \end{bmatrix} . \tag{3.30}$$

Notice that for a large fitting weight α_{fit}, the solution of this equation converges to the (bending energy minimizing) solution of $L^\top L\,\mathbf{d} = \mathbf{0}$ with hard constraints. However, when α_{fit} is too large, numerical problems may arise as observed by Botsch and Sorkine [BS08]. In those cases, using hard constraints with exact interpolations might be a better choice.

By looking at Equation 3.30, it becomes clear that in order to minimize stretching, we must compute the square-root of the Laplacian matrix L. However, we know that L is symmetric and positive definite and there exists $2^{N_\mathcal{V}}$ distinct square root matrices. We may simply ignore this ambiguity and consider the unique positive semi-definite matrix $L^{\frac{1}{2}}$. After diagonalizing $L = E\,D\,E^\top = E\,D^{\frac{1}{2}}\,E^\top\,E\,D^{\frac{1}{2}}\,E^\top$, we immediately obtain the square root $L^{\frac{1}{2}} = E\,D^{\frac{1}{2}}\,E^\top$ where $D^{\frac{1}{2}}$ is simply the square root of each diagonal components of the diagonal matrix D. For large and sparse matrices, special diagonalization solvers are required. A widely adapted approach is the iterative Davidson method [Saa92, PTVF97]. However, as documented in Taubin [Tau95], the spectral decomposition of large Laplacian matrices is generally impractical.

3.2.2 Laplacian Deformation

Surface deformation based on Laplacian differential coordinates was first sketched in Alexa [Ale01] and further pursued in [SCOL+04, LSCO+04]. These deformation models were later shown in Botsch and Sorkine [BS08] to be conceptually equivalent to a

linearized bending-energy minimizing deformation. As opposed to thin shell techniques, Laplacian deformations can be controlled by manipulating *differential coordinates* instead of only spatial positions. The idea here is to operate and define regularization directly over an intrinsic surface representation which can help to preserve geometric details as much as possible. Since Laplacians $\Delta_{\mathcal{S}}\mathbf{v}_i$ encode vertices relative to the centroids of their one-ring neighborhoods, they can be seen as a form of differential coordinate—we call them Laplacian coordinates (or differentials). Notice that these coordinates are invariant to translations but sensitive to linear transformations (rotations, scaling, shearing).

Again, given an initial triangle mesh $\mathcal{S}_d$ and positional constraints $\mathbf{c}_i$ for some vertices $\mathbf{v}_i$, we wish to find the deformed mesh $\tilde{\mathcal{S}}_d$ by minimizing an energy functional $E_{\text{tot}} = E_{\text{fit}} + \alpha_{\text{reg}}E_{\text{reg}}$. As before, we consider the point-to-point fitting term $E_{\text{fit}} = \sum_{i=1}^{N_C} \|\mathbf{c}_i - \tilde{\mathbf{v}}_i\|_2^2$ where $\tilde{\mathbf{v}}_i = \mathbf{v}_i + \mathbf{d}_i \in \tilde{\mathcal{S}}_d$. In its most basic formulation, the regularization is defined to minimize the *change of Laplacian* (hence, preserving surface laplacian):

$$\begin{aligned} E_{\text{reg}} &= \sum_{i=1}^{N_{\mathcal{V}}} \|\Delta_{\mathcal{S}}\tilde{\mathbf{v}}_i - \Delta_{\mathcal{S}}\mathbf{v}_i\|_2^2 && (3.31)\\ &= \sum_{i=1}^{N_{\mathcal{V}}} \|\Delta_{\mathcal{S}}(\mathbf{v}_i + \mathbf{d}_i) - \Delta_{\mathcal{S}}\mathbf{v}_i\|_2^2 && (3.32)\\ &= \sum_{i=1}^{N_{\mathcal{V}}} \|\Delta_{\mathcal{S}}\mathbf{d}_i\|_2^2 \quad . && (3.33) \end{aligned}$$

We immediately see that the minimizer of E_{reg} is the bi-Laplacian equation $\Delta_{\mathcal{S}}^2\mathbf{d}_i = \mathbf{0}$. Consequently, the regularization in Laplacian coordinates minimizes the linearized bending energy. However, in order to promote rotation and isotropic scale invariance, Sorkine and coworkers [SCOL+04] suggests to couple the basic Laplacian representation with an implicit transformation derived from the one-ring neighborhood with the following regularization functional:

$$E_{\text{reg}} = \sum_{i=1}^{N_{\mathcal{V}}} \|\Delta_{\mathcal{S}}\tilde{\mathbf{v}}_i - T_i(\tilde{\mathcal{V}}, \Delta_{\mathcal{S}}\mathbf{v}_i)\|_2^2 \tag{3.34}$$

where $\tilde{\mathcal{V}}$ are the vertices of $\tilde{\mathcal{S}}_d$ and T_i a *similarity* transformation which linear components consist of a linearized rotation (with small angle assumption) and isotropic scaling. It can be shown that T_i linearly depends on $\tilde{\mathcal{V}}$ and, hence, E_{tot} can be minimized by solving for $\tilde{\mathbf{v}}_i$ using simply normal equations. In short, the difference between deformation in

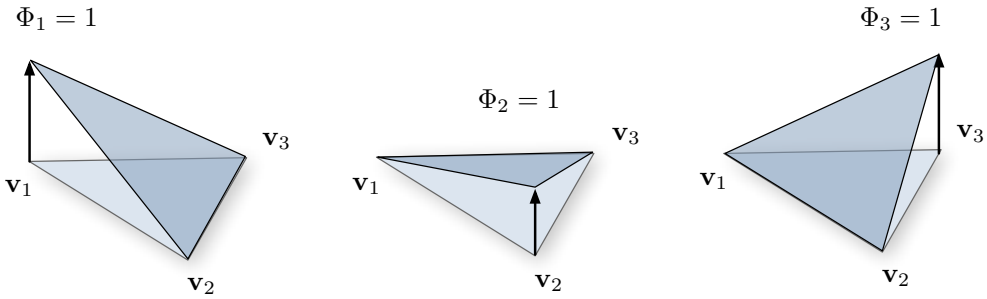


Figure 3.6: The "hat" basis functions are used to derive the gradients of the mesh coordinate functions.

Laplacian representation and linearized bending minimization lies in the way details are being preserved. In the former case, implicitly incorporating T_i in the regularization term helps to rotate and scale the detail structure of the shape according to the prescribed constraints. Instead of implicitly solving for T_i (by linearly linking it with $\tilde{\mathcal{V}}$), it is often explicitly specified through user-specified affine transformations in the context of mesh editing. For further details on how to manipulate T_i, we refer to the work on Laplacian surface editing [SCOL$^+$04].

3.2.3 Gradient-Based Deformation

Like Laplacian-based deformations, gradient-based techniques also set their foundations on differential coordinates [YZX$^+$04, BSPG06]. Instead of manipulating Laplacians coordinates, they operate on the mesh gradient field and fit a surface to mach a transformed gradient field. We first introduce some basic differential calculus for triangle meshes before we explore deformations. Let $\mathbf{p}(\mathbf{u})$ be the parameterization of a surface $\mathbf{S} \subset \mathbb{R}^3$, the gradient of the surface's coordinate function is given by $\nabla\mathbf{p}(\mathbf{u}) = \left[\frac{\partial}{\partial x}\mathbf{p}, \frac{\partial}{\partial y}\mathbf{p}, \frac{\partial}{\partial z}\mathbf{p}\right] \in \mathbb{R}^{3\times 3}$. Consider the continuous representation of its corresponding triangle mesh $\mathcal{S}_d$ where its piecewise linear coordinate function $\mathbf{v}(\mathbf{u})$ is simply expressed by barycentric interpolation of vertex coordinates $\mathbf{v}_i = \mathbf{v}(\mathbf{u}_i) \in \mathcal{V}$:

$$\mathbf{v}(\mathbf{u}) = \sum_{i=1}^{N_\mathcal{V}} \phi_i(\mathbf{u})\mathbf{v}_i \tag{3.35}$$

with $\phi_i(\mathbf{u})$ the per vertex, piecewise linear "hat" basis functions where $\phi_i(\mathbf{u}_k) = \delta_{ik}$ (c.f. Figure 3.6). To this end, we may associate a gradient $\nabla\mathbf{v}(\mathbf{u})$ per triangle face f_j where $j = 1, \ldots, N_\mathcal{F}$. The gradient $\nabla\mathbf{v}(\mathbf{u})$ is constant within f_j and may be defined w.r.t. the

vertices $\mathbf{v}_1 = \mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ of f_j:

$$\nabla \mathbf{v}(\mathbf{u}) = \nabla\phi_1(\mathbf{u})\mathbf{v}_1^\top + \nabla\phi_2(\mathbf{u})\mathbf{v}_2^\top + \nabla\phi_3(\mathbf{u})\mathbf{v}_3^\top \tag{3.36}$$

$$= [\nabla\phi_1, \nabla\phi_2, \nabla\phi_3] \begin{bmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \mathbf{v}_3^\top \end{bmatrix} = G_j \tag{3.37}$$

with $G_j \in \mathbb{R}^{3\times 3}$ the (now discrete) *surface gradient* of the triangle f_j. As described in Botsch and coworkers [BSPG06], the basis functions $\phi_i(\mathbf{u})$ within a triangle f_j can be deduced using its local frame:

$$[\nabla\phi_1, \nabla\phi_2, \nabla\phi_3] = \begin{bmatrix} (\mathbf{v}_1 - \mathbf{v}_3)^\top \\ (\mathbf{v}_2 - \mathbf{v}_3)^\top \\ \mathbf{n}^\top \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.38}$$

where $\mathbf{n}$ is the normal of f_j. Using all triangles $f_j \in \mathcal{N}(i)$ incident to the vertex $\mathbf{v}_i$, we may phrase the discrete Laplacian as the divergence of piecewise constant gradient fields:

$$\Delta_{\mathcal{S}}\mathbf{v}_i = \mathrm{div}\nabla\mathbf{v}_i = \sum_{f_j \in \mathcal{N}(i)} A(f_j)\,(\nabla\phi_{i,j})^\top G_j \tag{3.39}$$

where $A(f_j)$ is the area of f_j and $\phi_{i,j}$ the gradient of the basis function corresponding to the vertex $\mathbf{v}_i$ and face f_j.

By concatenating all per-face gradients G_j, we may establish a linear relationship with the vertices $\mathbf{v}_i$ using a $3\,N_{\mathcal{F}} \times N_{\mathcal{V}}$ matrix G which we call the gradient operator on the triangle mesh $\mathcal{S}_d$:

$$\begin{bmatrix} G_1 \\ \vdots \\ G_{N_{\mathcal{F}}} \end{bmatrix} = G \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{N_{\mathcal{V}}}^\top \end{bmatrix} \tag{3.40}$$

Returning to our original task of surface deformation, we first examine an energy functional that minimizes the change in surface gradients. Let $\tilde{\mathbf{v}}(\mathbf{u}) = \mathbf{v}(\mathbf{u}) + \mathbf{d}(\mathbf{u})$ be the surface coordinate function of the deformed surface $\tilde{\mathcal{S}}_d$, the gradient-based regularization term is then given by:

$$E_{\mathrm{reg}} = \int_{\mathcal{U}} \|\nabla\tilde{\mathbf{v}}(\mathbf{u}) - \nabla\mathbf{v}(\mathbf{u})\|_2^2 \mathrm{d}u\mathrm{d}v \tag{3.41}$$

Again, with added positional constraints $\mathbf{c}_i$ we may solve for $\mathbf{d}_i$ by minimizing $E_{\mathrm{fit}} = \sum_{i=1}^{N_C} \|\mathbf{c}_i - \tilde{\mathbf{v}}_i\|_2^2$ regularized with E_{reg} as before. In this setting, the mesh deformation with point constraints has a serious handicap as it preserves the original mesh gradients. Since

the surface gradients are defined w.r.t. the global coordinate frame, the deformation will, for example, tend to keep the original orientation, hence its rotation. Especially for constraints where local surface rotation is expected, the deformation will ignore this fact and result in unintuitive shapes.

However, we may consider surface deformations where local differentials (i.e., mesh gradients) are directly manipulated. Because local coordinate frames are directly encoded in the gradients G_j, any (non-translational) linear transformation $A_j \in \mathbb{R}^3$ that is applied to the *local frames* can be captured while keeping the resulting mesh connected. To be more concrete, we simply minimize the following energy term:

$$E_{\text{reg}} = \int_{\mathcal{U}} \|\nabla\tilde{\mathbf{v}}(\mathbf{u}) - T(\nabla\mathbf{v}(\mathbf{u}))\|_2^2 \mathrm{d}u\mathrm{d}v \tag{3.42}$$

where

$$T(\nabla\mathbf{v}(\mathbf{u_j})) = [\nabla\phi_1, \nabla\phi_2, \nabla\phi_3]\ (A_j\ [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3])^\top = \bar{G}_j \tag{3.43}$$

for $\mathbf{v}(\mathbf{u_j}) \in f_j$. Observe that the basis function matrix $[\nabla\phi_1, \nabla\phi_2, \nabla\phi_3]$ remains unchanged as we wish to express the transformation w.r.t. the original local frame. In particular, we obtain a right multiplication with the transposed linear transformation $\bar{G}_j = G_j A_j^\top$. To compute the deformed vertex positions $\tilde{\mathbf{v}}_i$, we simply solve the following linear least squares system:

$$(G^\top\ D)\ G \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{N_\mathcal{V}}^\top \end{bmatrix} = (G^\top\ D) \begin{bmatrix} \bar{G}_1 \\ \vdots \\ \bar{G}_{N_\mathcal{F}} \end{bmatrix} \tag{3.44}$$

where D is a diagonal weighting matrix containing the triangle areas. Notice that $G^\top\ D\ G\ = L$ is exactly the Laplacian and $G^\top\ D$ the divergence operator. In particular Equation 3.44 is a standard *Poisson equation* and we may substitute $L = M^{-1}L_s$ with the matrix form of our previous cotangent-weighted discrete Laplace-Beltrami operator. To summarize, gradient-based deformations are exceptionally effective when local transformations $T(\cdot)$ (with user-prescribed orientations) are provided. Furthermore, the reconstructed meshes are mostly free of local self-intersections. In fact, solving the (global) Poisson equation encourages local errors in the resulting gradient field to evenly spread over the entire surface.

Deformation Transfer. One important application of a gradient-based approach is *deformation transfer* which has been introduced by Sumner and Popović [SP04]. A

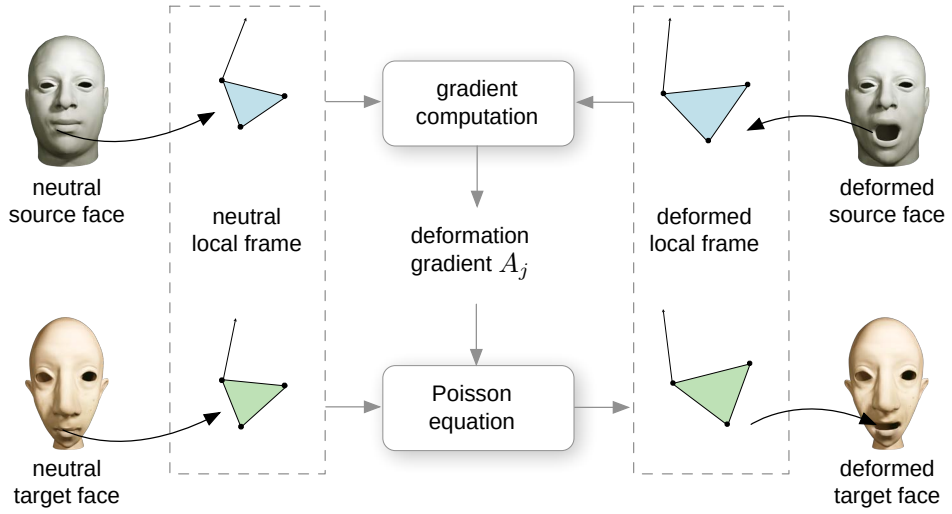


Figure 3.7: Deformation transfer can be divided into a deformation gradient computation followed by solving a Poisson equation.

direct connection with gradient-based representations was later established by Botsch and colleagues [BSPG06]. The aim of deformation transfer is to map the deformation of a source mesh $\mathcal{S}_d$ onto an arbitrary target mesh $\mathcal{S}'_d$. The local deformation is expressed by a linear transformation, i.e., *deformation gradients*, between the source mesh in its rest pose $\mathcal{S}_d$ and its deformed state $\tilde{\mathcal{S}}_d$. The deformation gradients are then applied to the local frames of $\mathcal{S}'_d$. Thus we perform a right-multiplication with the transposed of the deformation gradients as described above. To obtain the resulting mesh $\tilde{\mathcal{S}}'_d$, we perform a linear least-squares optimization that solves the Poisson equation in gradient space (c.f. 3.44) which enforces mesh connectivity in $\tilde{\mathcal{S}}'_d$.

Let us be more concrete. The deformation gradients between a source mesh $\mathcal{S}_d$ and its deformed pose $\tilde{\mathcal{S}}_d$ are defined as a linear mapping between the local frames of their corresponding triangles:

$$A_j = [(\tilde{\mathbf{v}}_1 - \tilde{\mathbf{v}}_3), (\tilde{\mathbf{v}}_2 - \tilde{\mathbf{v}}_3), \tilde{\mathbf{n}}]\ [(\mathbf{v}_1 - \mathbf{v}_3), (\mathbf{v}_2 - \mathbf{v}_3), \mathbf{n}]^{-1} \tag{3.45}$$

Note that $A_j = (G_j^{-1}\,\bar{G}_j)^\top \neq \tilde{G}_j\,G_j^{-1}$ as the name deformation gradient might misleadingly suggest. To transfer A_j to the target mesh in rest pose $\mathcal{S}'_d$ we simply compute the surface gradients of the deformed target mesh $\bar{G}'_j = G'_j A_j^\top$. Finally, we may efficiently

solve for $\tilde{\mathcal{S}}'_d$ using the following sparse linear system:

$$(G'^{\top}\, D')\, G' \begin{bmatrix} \mathbf{v}'^{\top}_1 \\ \vdots \\ \mathbf{v}'^{\top}_{N_\mathcal{V}} \end{bmatrix} = (G'^{\top}\, D') \begin{bmatrix} \bar{G}'_1 \\ \vdots \\ \bar{G}'_{N_\mathcal{F}} \end{bmatrix} \tag{3.46}$$

where G' is the mesh gradient operators and D' the diagonal matrix with area weights of the undeformed target mesh. Since the deformation gradients do not encode translational components, the resulting target mesh might be translated in space from the position of the source mesh. Hence, deformation transfer is translation invariant. Moreover, deformation transfer is general not invariant to rigid transformations w.r.t. the global coordinate system since the linear transformation A_j depends on the coordinate system, i.e. $A_j \neq R^{-1}\, A_j\, R$ where $R \in \mathrm{SO}(3)$ is a rotation matrix. However, the closer the deformation gradient gets to a true rotation, the less distortions we obtain between results of different global coordinate frames. One way to avoid this dependency is to perform a polar decomposition on the deformation gradients as described in [FAT07] and treat the non-rotation components separately.

3.2.4 Embedded Deformation

In non-rigid surface registration, we need to deform a scan in order to align with another scan. However, the scanned data may exhibit holes and consist of multiple disconnected surface fragments. Additionally, in the presence of largely incomplete and possibly inaccurate correspondences, we require a deformation model that preserves local shape details as much as possible, even under drastic deformations. The above presented deformation models are therefore only suitable when the undeformed mesh is a single connected surface (e.g., a template) and both scans are sufficiently close for reliable estimation of dense correspondences.

This section presents *embedded deformation* which is an efficient non-linear space-deformation technique introduced in the context of direct shape manipulation by Sumner and coworkers [SSP07]. The deformation model favors high-quality, natural shape deformations by locally maximizing rigidity in the transformation. The algorithm was shown to produce results that are comparable to another non-linear technique, PriMo [BPGK06]. Compared to the latter, embedded deformation can effectively handle multiple disconnected surfaces as it is designed to accommodate any type of geometric primitive (polygon soups, mesh animation, particle systems...).

In contrast to other space-deformation methods, this technique avoids a domain-specific solution, such as a kinematic skeleton that would have to be customized a priori for each source model. Instead, a *deformation graph* is (automatically) constructed from the embedded surface where each node defines an affine transformation that induces a warping on the nearby space. The nodes are sparsely distributed over the surface to decouple surface deformation from mesh complexity while retaining the global shape. In particular, surface deformation is obtained by solving for the graph nodes instead of the usually much denser mesh vertices, largely reducing the computational complexity typically associated with non-linear approaches.

Using this reduced model, the embedded mesh is then deformed by blending the transformations with overlapping influence. Further, the (undirected) edges of this graph are used to form a neighborhood structure to enable regularization on the surface deformation. In particular, globally consistent deformation can be achieved by connecting nodes of overlapping influence regions. An important feature of embedded deformation is that the optimization procedure maximizes rigid motion in the affine transformations of the nodes which naturally preserves details in largely unconstrained regions. Let us now describe the framework in more detail.

Deformation Graph. The source scan $\mathcal{S}_d$ is first augmented with a reduced deformable model in the form of a deformation graph. Graph nodes, defined by their positions $\mathbf{x}_1, \ldots, \mathbf{x}_{N_\mathcal{G}}$, may be chosen by uniformly sampling the mesh as discussed in Section 3.1.3. One affine transformation is associated with each node and induces a deformation on the nearby space. The influence of nearby nodes is blended by the embedded deformation algorithm in order to deform the scan vertices $\mathbf{v}_1, \ldots, \mathbf{v}_{N_\mathcal{V}}$ and the graph nodes themselves. Undirected edges connect nodes of overlapping influence to indicate local dependencies. The affine transformation for node $\mathbf{x}_i$ is specified as before by a matrix $A_i \in \mathbb{R}^{3\times3}$ and a translation vector $\mathbf{b}_i \in \mathbb{R}^3$. In this way, the collection of all per-node affine transformations expresses a non-rigid deformation of the graph and the scan. The number of nodes characterizes the degrees of freedom of this specific deformation model.

Along the lines of linear blend skinning [MTLT88], a vertex $\mathbf{v}_j$ is transformed to $\tilde{\mathbf{v}}_j$ by the nodes $\mathbf{x}_i$ according to a weighted linear combination of affine transformations:

$$\tilde{\mathbf{v}}_j = \Phi(\mathbf{v}_j) = \sum_{i=1}^{N_\mathcal{V}} w_i(\mathbf{v}_j)\ [A_i(\mathbf{v}_j - \mathbf{x}_i) + \mathbf{x}_i + \mathbf{b}_i] \quad . \tag{3.47}$$

The weights $w_i(\mathbf{v}_j)$ are nonzero for the k-nearest nodes (typically $k \geq 4$) and defined by

$$w_i(\mathbf{v}_j) = \frac{1 - \|\mathbf{v}_j - \mathbf{x}_i\|/d_{\max}}{\sum_{p=1}^{k} 1 - \|\mathbf{v}_j - \mathbf{x}_p\|/d_{\max}}, \tag{3.48}$$

where $d_{\max}$ is the distance to the $k+1$-nearest node which can be efficiently determined using a kd-tree or any spatial data structure for fast query (c.f. Section 2.5). Once the weights are computed, we may connect the nodes with edges. More specifically, two nodes share an edge if there exists a vertex which has nonzero weights to the nodes. In practice, we reduce the edge complexity by setting a small threshold σ_{edge} and only connect an edge if both weights are above this threshold (we typically choose $\sigma_{\text{edge}} = 0.15$). To summarize, the initial process of deformation graph construction can be divided into the following stages: uniform node sampling, assigning the $k+1$ closest nodes to each vertex, computing the $k+1$ influence weights for each vertex, and, finally, determining the graph edges by evaluating these weights.

Local Rigidity Maximization. Once the deformation graph is initialized, we may specify a regularization that prescribes a globally consistent deformation. In embedded deformation, two energy functionals control the deformation. The E_{rigid} term penalizes the deviation of each transformation from a pure rigid motion. Consequently, local features deform as rigidly as possible avoiding shearing or stretching artifacts. This is accomplished by minimizing the deviation of A_i from orthogonality and unit length:

$$E_{\text{rigid}} = \sum_{i=1}^{N_{\mathcal{V}}} \text{Rot}(A_i) \tag{3.49}$$

where

$$\begin{aligned} \text{Rot}(A) \quad = \quad & (\mathbf{a}_1^\top \mathbf{a}_2)^2 + (\mathbf{a}_1^\top \mathbf{a}_3)^2 + (\mathbf{a}_2^\top \mathbf{a}_3)^2 + \\ & (1 - \mathbf{a}_1^\top \mathbf{a}_1)^2 + (1 - \mathbf{a}_2^\top \mathbf{a}_2)^2 + (1 - \mathbf{a}_3^\top \mathbf{a}_3)^2 \end{aligned}$$

and $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$ are the column vectors of A_i.

A second energy term, E_{smooth}, serves as a regularizer for the deformation by indicating that the affine transformations of adjacent graph nodes should agree with one another:

$$E_{\text{smooth}} = \sum_{i=1}^{N_{\mathcal{V}}} \sum_{\mathbf{v}_j \in \mathcal{N}(i)} \gamma_{ij} \|A_i(\mathbf{x}_j - \mathbf{x}_i) + \mathbf{x}_i + \mathbf{b}_i - (\mathbf{x}_j + \mathbf{b}_j)\|^2 \tag{3.50}$$

where $\mathcal{N}(i)$ consists of all nodes that share an edge with node i. The weight γ_{ij} should be proportional to the degree to which the influence of nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ overlap. For

uniformly sampled nodes, we simply use $\gamma_{ij} = 1.0$. While other weighting schemes were assessed in [SSP07], no significant differences were observed in their experiments.

The smoothing term E_{smooth} is similar to the regularization term used for deformation based on affine-transforms (c.f. Section 3.2). The main difference here is that we compare transformed node positions rather than the transformations themselves which leads to fewer equations when solving for the minimizer of E_{smooth}. Another difference is that the per-node transformations are centered around the node positions instead of the global coordinate system which simplifies E_{smooth} and also improves numerical accuracy.

Non-Linear Optimization. Finally, a fitting term $E_{\text{fit}} = \sum_{i=1}^{N_C} \|\mathbf{c}_i - \tilde{\mathbf{v}}_i\|_2^2$, where $\tilde{\mathbf{v}}_i$ is defined as in Equation 3.47, provides the desired positional constraints and is the impetus that induces deformation. Once again, we combine the different energy terms to form a global objective function:

$$E_{\text{tot}} = E_{\text{fit}} + \alpha_{\text{rigid}} E_{\text{rigid}} + \alpha_{\text{smooth}} E_{\text{smooth}} \quad . \tag{3.51}$$

Both terms, E_{rigid} and E_{smooth}, serve as deformation regularization. Ultimately the shape deformation framework minimizes E_{tot} in order to solve for the 12 $N_{\mathcal{G}}$ unknowns in the per-node affine transformations $\Phi_i = (A_i, \mathbf{b}_i)$ where $i = 1, \ldots, N_{\mathcal{G}}$. We represent the optimization variables of Φ_i by a vector with concatenated components $\boldsymbol{\gamma}_i = [\mathbf{a}_{1,i}^\top, \mathbf{a}_{2,i}^\top, \mathbf{a}_{3,i}^\top, \mathbf{b}_i^\top]^\top$. Since the partial derivatives of E_{rigid} are quadratic in the optimization variables, the overall minimization of E_{fit} is non-linear.

A popular approach to this unconstrained non-linear least squares problem is to use the Gauss-Newton method [MNT04] which takes multiple linear steps to approximate the solution. The Gauss-Newton algorithm iteratively solves for $\boldsymbol{\gamma}$ by considering the first-order Taylor approximation of $\mathbf{f}(\boldsymbol{\gamma})$:

$$E_{\text{tot}}(\boldsymbol{\gamma}^{k+1}) = \|\mathbf{f}(\boldsymbol{\gamma}^{k+1})\|_2^2 \approx \|\mathbf{f}(\boldsymbol{\gamma}^k) + J_{\mathbf{f}}(\boldsymbol{\gamma}^{k+1} - \boldsymbol{\gamma}^k)\|_2^2 = \|\mathbf{f}(\boldsymbol{\gamma}^k) + J_{\mathbf{f}}(\boldsymbol{\gamma}^k)\Delta(\boldsymbol{\gamma}^k)\|_2^2 \tag{3.52}$$

where $\boldsymbol{\gamma}^k = [\boldsymbol{\gamma}_0^k \ldots \boldsymbol{\gamma}_{N_{\mathcal{G}}}^k]$ is the minimizer in the kth iteration, $J_{\mathbf{f}}(\boldsymbol{\gamma}^k)$ the Jacobian matrix of $\mathbf{f}(\boldsymbol{\gamma}^k)$, and $\Delta(\boldsymbol{\gamma}^k)$ the forward difference vector of $\boldsymbol{\gamma}^k$. The optimization is initialized with $\boldsymbol{\gamma}_i^0 = [[1,0,0],[0,1,0],[0,0,1],[0,0,0]]$. Since $J_{\mathbf{f}}(\boldsymbol{\gamma}^k)$ is linear in its unknowns $\boldsymbol{\gamma}$, the linear approximation of the Taylor expansion becomes particularly accurate. Each Gauss-Newton step results in a linear least squares problem:

$$J_{\mathbf{f}}^\top(\boldsymbol{\gamma}^k) J_{\mathbf{f}}(\boldsymbol{\gamma}^k) \Delta(\boldsymbol{\gamma}^k) = -J_{\mathbf{f}}^\top(\boldsymbol{\gamma}^k) \mathbf{f}(\boldsymbol{\gamma}^k) \tag{3.53}$$

which solves for $\Delta(\boldsymbol{\gamma}^k)$. Because $J_{\mathbf{f}}^{\top}(\boldsymbol{\gamma}^k)J_{\mathbf{f}}(\boldsymbol{\gamma}^k)$ is sparse, we may efficiently solve the normal equation in each iteration using a direct solver that employs Cholesky factorization [SG04]. We update $\boldsymbol{\gamma}^{k+1} = \boldsymbol{\gamma}^k + \Delta(\boldsymbol{\gamma}^k)$ until convergence, i.e., $(E_{\text{tot}}(\boldsymbol{\gamma}^{k+1}) - E_{\text{tot}}(\boldsymbol{\gamma}^k)) < \epsilon(1 + E_{\text{tot}}(\boldsymbol{\gamma}^k))$.

Taken as a whole, embedded deformation offers a number of advantages. It naturally generates plausible deformations that maximize rigidity and avoid unrealistic scaling or shearing. The deformation graph is a reduced deformable model that dissociates the complexity of the source scan from the complexity of the deformation system, enhancing performance greatly. The flexible nature of the deformation graph allows to deform scans that contain many different disconnected components in a coordinated fashion.

3.3 Non-Rigid Registration

We develop the concept of *non-rigid registration* based on the foundations on correspondence computation, rigid registration, and surface deformation, discussed in Section 3.1 and 3.2. Naively, the notion of rigid registration can be directly translated to the non-rigid case by replacing a rigid transformation Φ_{rigid} with a surface deformation Φ_{deform} and using correspondences as position constraints. However, it is rarely the case that correct correspondences can be found by simply comparing the geometry of both shapes. Even when sophisticated shape descriptors are used, false correspondences are likely to occur. However, unlike the case of rigid transformations, a single wrong correspondence can create large distortions in the deformation if its regularization is too weak. On the other hand, with a regularization being too strong (high stiffness), the source shape would end up behaving like a rigid transformation and not be able to accurately match the target surface. We observe that deformation computation is tightly coupled with correspondence estimation as both measurements contribute to the ill-posed question of what is a correct match. For this reason, non-rigid alignment algorithms often employ multiple passes of interleaved correspondence and deformation computations similar to registration refinement. Let us discuss some of the important design decisions when addressing a non-rigid registration problem:

3.3.1 Design Decisions

Correspondences. While correspondences computation methods that are invariant of the initial source pose (e.g., shape descriptors) can capture extreme pose variations

in the scans, they should be used with extra care. In fact, most shape descriptors illustrated in Section 3.1 are designed to capture rigid shapes, i.e., local patches that are matching should be almost identical. In a non-rigid setting, the captured data is subject to unknown deformations. Consequently, a lower threshold needs to be set for potential correspondence candidates which leads to higher ambiguity and more false positives. Additionally, smaller (unfortunately, also less discriminative) patches should be considered as opposed to the rigid case, since details are generally less distorted in the high-frequency components. Although in many cases, they might define a good initial set of correspondences, further refinement based on the measurement of deformation smoothness is usually necessary to improve accuracy and prune wrong estimates.

Within the context of iterative (or global correspondence optimization) methods, correspondence computations that take into account the previous pose during optimization (such as the closest point estimation) are particularly interesting for non-rigid registration. Firstly, dense correspondences can be efficiently computed as opposed to pose-invariant methods which increases the overall alignment accuracy. Secondly, the correspondences are no longer searched independently of each other but a globally consistent regularization can be introduced through the deformation and promote correspondence computations in later iterations. Note that an elastic deformation model should be considered (rather than a plastic one) where the initial shape would represent the rest state of the deformation.

Deformation Model. As described in 3.2, a large variety of deformation models exist. The right choice depends on the underlying surface representation, the availability of prior knowledge about the scanned subject, as well as the accuracy and density of the correspondence computation. If the deformed surface is a template model, surface-based deformation techniques are suitable since the surface topology is implicitly defined. On the other hand, space deformation techniques are necessary when dealing with fragmented surfaces such as scanned data.

Most deformation models allow the specification of a regularization (stiffness) parameter. For shapes, wherein small warps are expected, the use of a strong regularization is encouraged as it prevents arbitrary distortions due to possible false correspondences. In general, we also wish to capture large deformations which require more flexibility in the deformation. In this case, even with a reduced regularization, largely unconstrained regions should deform in a natural way, i.e., preserve details. This becomes an impera-

tive requirement whenever correspondences are sparse or the regions of overlap between both scans are small. Nevertheless, even when dense correspondences are available, detail preserving deformations help to compensate for inaccurate matches in the case of iterative non-rigid registration methods.

We have seen in Section 3.2 that space deformation techniques, such as embedded deformation, decouple the degrees of freedom of the deformation from the surface complexity by sparsely resampling the surface. Another example are kinematic skeletons which are often used in performance capture to restrict the tracking to plausible poses. The observation here is that natural deformations can be assumed to be smooth, i.e., large deformations are more likely to occur in low-frequencies of the surface, rather than high-frequency ones. A deformation model that has a large number of degrees of freedom increases the number of optimization variables and, thus, introduces more local minima in the optimization landscape. Consequently, using the right amount of degrees of freedom not only improves efficiency, but also increases robustness during optimization. The idea of using a coarser level representation can also be translated to surface-based deformation using mesh simplification procedures as described in [KCVS98, BK03, BS08].

Optimization. Non-rigid registration is an inherently ill-posed problem and in contrast to the rigid case, it is hard to define whether the determined correspondences are correct or not. More specifically, when aligning rigid shapes, the problem can be simply phrased as minimizing the distance between their overlapping regions. For deformable shapes, using the same criteria would lead to the trivial solution of minimizing the correspondence distances and using a very low regularization. Obviously, this would not yield very meaningful alignments. Regularization energies of (elastic) deformation models characterize the plausibility of a specific shape deformation by measuring its deviation from its rest state (which represents a known prior shape). Therefore, a non-rigid registration problem should be formulated as one that simultaneously minimizes a deformation energy and a fitting metric (which is specified by correspondences) .

Regardless of the employed deformation model and technique for correspondence computation, a common strategy for improving robustness in an iterative optimization framework is to manipulate the energy landscape. The idea consists of gradually changing certain global parameters in the problem formulation to effectively avoid local minima in a coarse-to-fine fashion. While in general a global optimum cannot be guaranteed, the impact of such scheduling procedure can be dramatic in finding the correct solution. One

important approach consists of progressively reducing the regularization of the deformation model. Here, the global shape is being aligned first, followed by more and more local registrations. Another effective coarse-to-fine strategy considers the amount of degrees of freedom. Starting with a low resolution deformation model, a scheduling procedure gradually adds new degrees of freedom whenever convergence is detected. Instead of homogeneously introducing resolution or diminishing regularization, a more optimal solution consists of adaptively modifying these global parameters according to how much they contribute to the optimization. For example, we may wish to introduce additional degrees of freedom only in specific regions where strong deformation has been detected. An important observation here is that a coarse-to-fine approach also improves the coupling between correspondences and deformation as a coarse energy landscape promotes global consistency during optimization. Besides improving robustness, a multi-resolution approach can be important for improving computational efficiency. For instance, a non-linear registration technique could be performed at a coarse resolution level and further refined with a more efficient linear method at higher resolution.

3.3.2 Related Work

A substantial amount of research has been devoted to non-rigid registration impacting dynamic shape reconstruction, motion capture, and shape analysis. Most approaches are specifically designed for particular applications and make special prior assumptions about the scanned subjects.

Following the taxonomy presented in Figure 3.2, methods for *non-rigid alignment of rigid objects* fall into problems of Cat I. *Template-based registration methods* appear in a variety of registration problems ranging from Cat II to Cat IV. All registration techniques designed for input scans that are captured from a *real-time 3D scanner* address problems of Cat II. Finally, several *advanced techniques* which make very specific prior assumptions about the deformation belong to problems of Cat III.

Non-Rigid Alignment of Rigid Objects. Non-rigid scan registration was first introduced to align rigid objects which are affected by low frequency warps, such as those caused by device non-linearities and calibration errors. To correct such distortions, Ikemoto and coworkers [IGL03] introduce a non-rigid registration technique that decomposes the input scans using a coarse-to-fine hierarchy of locally rigid pieces that are allowed to translate and rotate with respect to one another. The advantage of this

method is that no specific characterization of the warp is required since a continuous deformation is approximated from the convergence of the piecewise rigid model. However, the running time of this technique is quadratic in the number of patches.

Brown and Rusinkiewicz [BR04] address this scalability issue using thin-plate splines to represent smooth warps and a hierarchical ICP method to find good feature correspondences between subdivided patches. An extension of this method for the simultaneous alignment of a large number of scans with locally weighted ICP matching has been recently presented by the same authors [BR07].

Remark: *This class of registration assumes little deformation in the scanned subject. For improved robustness and efficiency, very few degrees of freedom (less unknowns) are typically used in the deformation model. A strong regularization in the deformation also helps to avoid outlier correspondences to tear the shape apart.*

Template-Based Registration for Large Deformations. The registration of scans with large-scale deformations, such as those of an articulated body, requires a more general and flexible deformation model. Moreover, local shape matching techniques might fail if the shape is distorted beyond a certain limit. Earlier solutions to this problem commonly involve the use of template models that are warped toward the input scans [BV99, ACP03, PMG+05, ASK+05, ARV07]. The template model provides a strong geometric prior and thus leads to high-quality reconstructions with automated hole-filling and noise removal.

Correspondence estimation is often facilitated by the use of tracked marker points or hand-selected feature correspondences. Park and Hodgins [PH06] propose using a large set of markers to accurately capture the dynamic motion of human bodies.

An algorithm that does not require hand selected markers has been developed by Anguelov and colleagues [ASP+04] where a joint probabilistic model over all point-to-point correspondences is optimized between two shapes. While the method is fully automatic and is able to recover significant movements of articulated parts and non-rigid deformations, it requires that one of the input shapes is a subset of the other.

A different approach that also requires a template model has been proposed by Bronstein and colleagues [BBK06]. They address the partial matching problem with a multi-dimensional scaling algorithm that aims at minimizing the distortion of the mapping between two surfaces.

Remark: *Because the scanned subject may deform arbitrarily, deformation models used for this type of registration use a large number of degrees of freedom. While a weak regularization in the deformation allows the registration to capture large motions, the correspondence problem becomes more challenging. As a result, these registration methods either rely on user assistance or automatic but computationally expensive correspondence algorithms which typically also require a template model.*

Registration of Real-time Range Scans. Real-time 3D scanners as used in this dissertation enable the continuous capture of deforming objects and produce dense sequences of range scans. In the most general setting, no prior template shape is given and no markers or explicit feature point correspondences are available. In this case, a true partial matching problem must be solved, as opposed to a part-in-whole matching as for template-based techniques. Existing work on the pairwise alignment of dense scans typically assumes that pairwise scans undergo small deformations while having a significant amount of overlap. Mitra and colleagues [MFO+07] present a registration method for dense time-series of point clouds that does not explicitly compute correspondence. Instead, they aggregate all scans into a 4-D space-time surface and estimate inter-frame motion from kinematic properties of this surface. This technique requires the deformation of adjacent frames to be sufficiently small as it is designed primarily for articulated motions.

Wand and coworkers [WJH+07] introduce a statistical framework that computes a globally optimal shape and deformation of the complete model over every frame. The method relies on an initial pairwise registration of all adjacent scans using a non-rigid ICP variant based on a deformation model proposed in [HTB03]. Pairwise correspondences are then iteratively improved during the optimization assuming the input scans deform smoothly over time. Because the surfaces in [HTB03] are represented by point clouds, deformation is achieved using a skeletal link structure which connects neighboring points.

For manifold surfaces, non-rigid ICP algorithms often use a deformation model based on smooth local affine transforms. As many degrees of freedom are introduced in the deformation model, a procedure that iteratively reduces the stiffness improves robustness to local minima [ACP03, SP04]. While a template model is still required, the optimal step non-rigid ICP proposed by Amberg and colleagues [ARV07] demonstrates several successfully aligned examples without the use of hand selected correspondences.

Remark: *Non-rigid registration techniques that are designed to find correspondences in scan sequences also use a large degree of variability in the deformation model. Because the recording is achieved in real-time, most parts of the surface can be assumed temporally coherent. While spatial proximity heuristics can be used for correspondence search between pairs of scans, accuracy is crucial to avoid accumulation of errors when an entire recording is processed.*

Advanced Methods for Non-Rigid Registration. Several researchers have proposed automatic non-rigid registration algorithms that are specifically designed to handle large deformations. Based on recent work on symmetry detection [MGP06] and extending the caveats of the correlated correspondence algorithm [ASP+04] Chang and Zwicker [CZ08] solve a discrete labeling problem to detect the set of optimal correspondences and apply graph cuts to optimize for a consistent deformation from source to target. This global optimization entails a high computational cost (more than an hour for pairwise registration of meshes with less than 10k vertices) which renders the method impractical for multi-frame alignment of continuous input scans.

They extend their scheme in [CZ09] using a reduced space deformation model represented by a volumetric grid that encloses the underlying scan. Linear blend skinning is used to embedded the underlying surface deformation. A decoupled optimization approach solves for deformation and skinning weights in an interleaved way which makes the approach particularly well suited for handling articulated subjects. Although significant motion and occlusions can be handled, their deformation field representation breaks down for topologically difficult scenarios such as shapes with nearby or touching surfaces.

Huang and colleagues [HAWG08] suggested a registration technique that finds an alignment by diffusing consistent closest point correspondences over the target shape while preserving isometries as much as possible. Their implementation has proved to be efficient for large isometric deformations, yet the correspondence search is sensitive to topological changes and holes that commonly occur in partial acquisition systems.

Salzmann and coworkers [SPIF07] propose a model for isometric deformation based on dihedral angles of a triangle mesh. Using dimensionality reduction techniques, they obtain a reduced deformable model that yields excellent results for shape recovery of inextensible surfaces such as cloth or paper from video sequences.

Remark: *Advanced techniques for non-rigid registration are designed to deal with significantly larger deformations while no templates are involved. These techniques often make higher level geometric assumptions about the deformation such as isometry preservation or quasi-articulated motion. To capture more general deformations, one common strategy consists of using a tighter coupling between shape matching and warping in the optimization process.*

3.4 Global Correspondence Optimization

Building on the foundational algorithms, introduced in earlier chapters, we now present a general registration algorithm for partial scans of deforming shapes. We address the challenges of non-rigid registration within a single non-linear optimization. Our algorithm simultaneously solves for:

- Correspondences between points on source and target scans.
- Confidence weights that measure the reliability of each correspondence and identifies non-overlapping areas.
- A warping field that brings the source scan into alignment with the target geometry.

The optimization maximizes the region of overlap and the spatial coherence of the deformation while minimizing registration error. Poor local minima are avoided with an iterative execution schedule that detects sub-optimal convergence and repeats the optimization with improved initial conditions so that a better result is obtained. This method employs the embedded deformation model, introduced in Section 3.2.4, which separates the geometric complexity of the scans from the complexity of the optimization, thereby enhancing performance and robustness. The non-linear deformation energy avoids unnatural shearing artifacts by maximizing local rigidity in the deformation.

The proposed approach is robust to considerable deformations and does not require high-speed acquisition. The method is not restricted to part-in-whole matching, but addresses the general problem of partial matching where the overlap region is a subset of both shapes.

Finally, this algorithm requires no explicit prior correspondences or feature points, which makes it more robust to settings where markers are hard to place and track on the scanned subject or when feature extraction methods yield unreliable key points.

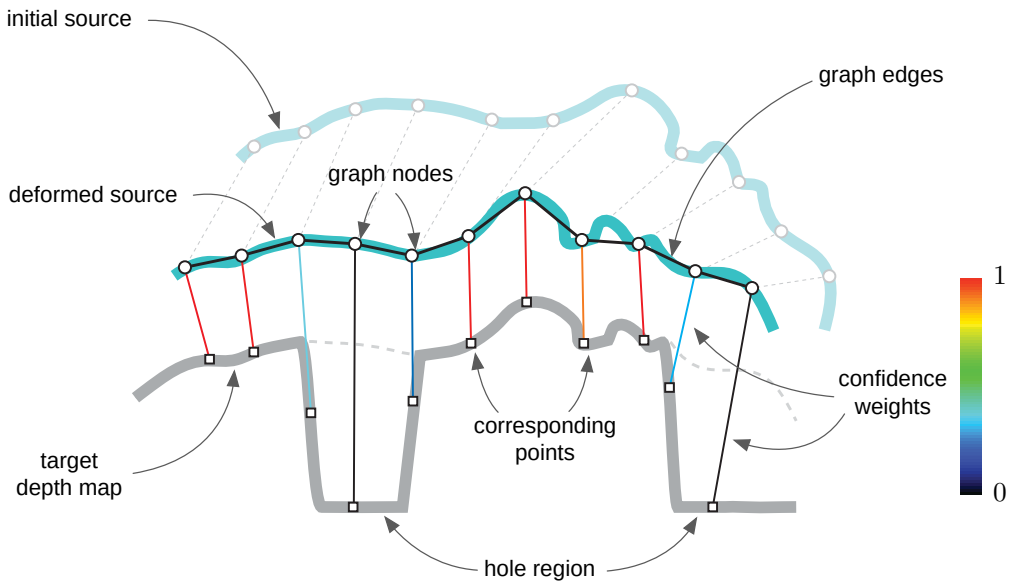


Figure 3.8: Our non-rigid registration algorithm simultaneously solves for the affine transformations of each node of the deformation graph, the corresponding points on the target shape, and the confidence weights. The latter evolve during the optimization to identify the region of overlap.

Overview. The core task of our system is the registration of two partial scans of a deforming object: a source scan is registered to a target scan captured at a different point in time (c.f., Figure 3.8). Each scan represents only a portion of the entire object. Since new parts of the object may have come into view and other parts may have become occluded in between the two captures, the region of overlap is a subset of both scans. Furthermore, the object may have undergone both rigid and non-rigid deformation, such as global Euclidean motion, pose changes, or changes in facial expression.

We employ a non-linear deformation system that favors natural deformations by maximizing both rigidity and consistency. The deformation is controlled by correspondences expressed at points distributed evenly over the source so that each point has a corresponding position on the target shape. The optimization solves simultaneously for both the deformation parameters as well as the correspondence positions. Since the deformation algorithm is designed to favor the most natural deformations, the optimizer will update the target correspondence positions so as to achieve a natural deformation. Inconsistent correspondences are penalized and the features of the source and target are

naturally aligned with one another, since such an alignment leads to a lower deformation energy state. Since some source points have no corresponding position on the target due to partial overlap, we augment each correspondence with a weight that is also solved for by the optimizer. We design an energy functional so that this weight is naturally brought to zero when an appropriate correspondence cannot be found. In doing so, the zero-weighted correspondences indicate non-overlapping regions and do not influence the deformation.

Depth Map Representation. We first develop our work in the context of depth maps, in which a 2D image in the xy-plane stores a depth value along the z-direction. A more general technique will be presented in Section 3.5. Our data is acquired using the range scanner based on structured light [WLG07] described in Section 2.3. Please note that, although the scanner has a high frame rate, our registration algorithm is robust under much lower frame rates. We demonstrate examples where the registered scans are spaced as many as 20 frames apart. A different 3D triangle mesh is extracted for both the source and target scans from their respective depth maps by triangulating the pixel grid and assigning the z-direction of each vertex to be the corresponding depth value. More details on how we process the data can be found in Section 2.5.

3.4.1 Coupled Global and Local Deformation

Although the embedded deformation method (c.f. Section 3.2.4) can, in principle, represent both rigid and non-rigid deformations, the performance of our registration system is enhanced by modeling these two quantities separately. Thus, we augment the embedded deformation framework with a global rigid transformation defined by a rotation matrix $\mathbf{R}$ (parameterized in axis-angle form) and a translation vector $\mathbf{t}$. The rotation is relative to the center-of-mass $\mathbf{g}$ of the scan. The source graph nodes and mesh vertices are deformed by first applying the local non-rigid embedded deformation routine and then the rigid transformation so that a vertex $\mathbf{v}_j$ is transformed to $\tilde{\mathbf{v}}_j$ according to:

$$\tilde{\mathbf{v}}_j = \Phi_{\text{global}} \circ \Phi_{\text{local}} \left(\mathbf{v}_j \right), \tag{3.54}$$

where

$$\Phi_{\text{global}} \left(\mathbf{v}_j \right) = \mathbf{R}(\mathbf{v}_j - \mathbf{g}) + \mathbf{g} + \mathbf{t} \tag{3.55}$$

and

$$\Phi_{\text{local}} \left(\mathbf{v}_j \right) = \sum_{i=1}^{n} w_i(\mathbf{v}_j) \left[\mathbf{A}_i(\mathbf{v}_j - \mathbf{x}_i) + \mathbf{x}_i + \mathbf{b}_i \right]. \tag{3.56}$$

This deformation model forms the first building block in our overall optimization strategy. Both the global rigid transformation and the per-node affine transformations are treated as unknowns in the optimization and only affect the fitting energy as shown in Section 3.4.2. The two regularization functionals E_{rigid} and E_{fit} remain unchanged from this original formulations in Section 3.2.4.

3.4.2 Correspondences

For each graph node, we associate one correspondence value that indicates the corresponding position on the target shape. This position is initialized via a closest point computation and subsequently updated by the optimizer. Since our range scans are created from captured depth maps, the depth map itself provides a natural parameterization for each scan. Thus, we represent the correspondence position for node i by its (u_i, v_i) values in the parameter domain of the target depth map. The function $\mathbf{c}(\mathbf{u}_i)$ maps from the parameter domain back to the 3-D position:

$$\mathbf{c}(\mathbf{u}_i) = \begin{bmatrix} \mathbf{u}_i \\ c(\mathbf{u}_i) \end{bmatrix}, \tag{3.57}$$

where $\mathbf{u}_i = [u_i, v_i]^{\mathrm{t}}$ and $c(\mathbf{u}_i)$ is a scalar function that gives the z-value of the mapped point.

In practice, we obtain better performance by allowing (u_i, v_i) to be transformed by the deformation model's rigid transformation and define the transformed coordinates $\tilde{\mathbf{u}}_i$ according to:

$$\tilde{\mathbf{u}}_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} (\mathbf{R}(\mathbf{c}(\mathbf{u}_i) - \mathbf{g}) + \mathbf{g} + \mathbf{t}) + \mathbf{d}_i. \tag{3.58}$$

We introduce the energy term E_{fit}, which strives to move each source graph node to its corresponding position on the target shape:

$$E_{\text{fit}} = \sum_{i=1}^{n} \|\tilde{\mathbf{x}}_i - \mathbf{c}(\tilde{\mathbf{u}}_i)\|_2^2, \tag{3.59}$$

where $\tilde{\mathbf{x}}_i$ is the deformed position of node i. The (u_i, v_i) parameters for each graph node become unknowns of the optimization, which allows the corresponding points to move along the surface of the target scan. This parameterization of the target scan is a key ingredient of our method, since it automatically constrains the corresponding points to lie on the target scan and avoids the need for re-projection during the optimization as in the case of non-rigid ICP approaches.

Subsequent numeric computations (Section 3.4.4) require computing partial derivatives with respect to the target scan's parameter domain. For efficiency and numeric robustness, we precompute the required derivatives by first building a continuous approximation of the target shape. Since our shape is defined on a function graph rather than a manifold, we favor a weighted least squares (WLS) approximation using a 2-D quadratic polynomial basis $[1, u, v, uv, u^2, v^2]$ and a Wendland function of degree 5 as a weighting function (see [Wen05] for details). Partial derivatives are precomputed for each pixel in the depth map and bilinearly interpolated at runtime.

3.4.3 Partial Overlap

One principle challenge of our registration framework is the ambiguity introduced by scans that only partially overlap one another. For some portions of the source mesh, no corresponding point exists on the target and this region of overlap is not known a priori. Instead, our system computes it automatically. We accomplish this task by making a modification both to the data representation and the correspondence energy functional.

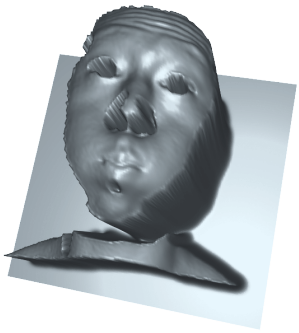


Figure 3.9: Weighted least squares (WLS) approximation.

Each range image contains portions where object measurements were obtained and "empty" regions where no object was detected. We preprocess the target range image by filling each empty pixel with a large value l so that the empty regions are replaced by deep holes after the mesh is reconstructed (see Figure 3.9). We set l to be twice the maximum depth value measured by the scanner so that hole regions lead to a large penalty in the fitting energy E_{fit}. One consequence of this change is that the hole regions will be treated as outliers in the WLS reconstruction due to the disparity between the object depths and the hole depth, leading to artifacts in the reconstruction. Thus, we perform a feathering operation in which a morphological erosion detects the border of the object region, and a smoothing filter is applied to the hole and border regions to ensure a smooth blend between the two [HSZ87]. The closest point variables (Section 3.4.2) do not distinguish between object geometry and hole geometry and are free to move between the two via the (u, v) mapping. Second, we associate a weight parameter ω_i

with each correspondence and modify E_{fit} as follows

$$E^*_{\text{fit}} = \sum_{i=1}^{n} \omega_i^2 \left\| \tilde{\mathbf{x}}_i - \mathbf{c}(\tilde{\mathbf{u}}_i) \right\|_2^2 . \tag{3.60}$$

Finally, we introduce the confidence energy term

$$E_{\text{conf}} = \sum_{i=1}^{n} (1 - \omega_i^2)^2. \tag{3.61}$$

Values of ω_i close to one indicate a reliable correspondence, while values close to zero indicate that no appropriate correspondence is found. The E_{conf} energy aims at maximizing the number of reliable correspondences by pushing the weights towards one, and thus maximizes the region of overlap. Now we consider what the optimization may do when a portion of the source mesh has no match on the target. First of all, we force each graph node to have some correspondence on the target regardless of whether there actually is an overlap between that portion of the source and the target. Without the modifications presented in this section, partial overlap would result in many bad correspondences and cause significant artifacts in the computed deformation, since the bad correspondences would pull the target shape in incompatible directions. Such unnatural deformations are high energy states, since the deformation model favors smooth deformations that maximize rigidity.

However, using our ω formulation, the source regions that are not present in the target can freely match to the hole regions. There is a high cost in terms of E^*_{fit} to such matches as the hole region is far away. Again, deforming the source to the position of correspondence in the hole also yields a high energy, since such large deformations are penalized. Thus, the minimum energy configuration naturally occurs when the ω_i parameter is reduced to zero by the optimizer. While this incurs some cost from E_{conf}, the cost is less than the alternatives. As a consequence, the optimizer naturally detects non-overlapping regions via the ω_i parameters (Figure 3.8).

3.4.4 Optimization

We sum the individual energy terms from the previous sections to form the full objective function of our optimization:

$$E = \alpha_{\text{rigid}} E_{\text{rigid}} + \alpha_{\text{smooth}} E_{\text{smooth}} + \alpha_{\text{fit}} E^*_{\text{fit}} + \alpha_{\text{conf}} E_{\text{conf}}. \tag{3.62}$$

The unknowns comprise the global rigid transformation, the affine transformations of the deformation graph, the (u, v) parameter domain coordinates for each graph node

correspondence, and the confidence weights ω_i for each node. The number of optimization variables is thus $15n + 6$ with n the number of deformation graph nodes.

We solve this nonlinear least-squares problem using the Levenberg-Marquardt algorithm [MNT04]. Since the system matrix is sparse, we solve the normal equations in each iteration using a direct solver that employs sparse Cholesky factorization [SG04]. A simple heuristic is employed to automatically adapt the optimization weights. Initially, $\alpha_{\text{rigid}} = 1000$, $\alpha_{\text{smooth}} = 100$, and $\alpha_{\text{conf}} = 100$. Each value is halved whenever $|F_k - F_{k-1}| < 10^{-5}(1 + F_k)$, with $F_k = E(\theta_k)$, until $\alpha_{\text{rigid}} < 1$, $\alpha_{\text{smooth}} < 0.1$, and $\alpha_{\text{conf}} < 1$. The weight α_{fit} is held constant at 0.1 during the optimization. The adaptation of weights initially favors global rigid alignment and subsequently lowers the stiffness of the object to allow increasing deformation as the optimization progresses. This automatic procedure is used for all shown examples.

Iterative Improvement. We detect convergence when $|F_k - F_{k-1}| < 10^{-6}(1 + F_k)$. As with any non-linear optimization, our system converges to a local optimum that may not represent the best possible global solution. We employ an iterative improvement algorithm to find a better local minimum by teleporting the solution to a different position in the energy landscape and restarting the optimizer from this new position. When the system converges, all correspondences are recalculated via a closest-point computation. Next, poor correspondences are detected using three criteria. A correspondence is poor if it is in a hole region, if the distance from the source graph node to the corresponding point on the target is greater than 2 cm, or if normals are inconsistent. A surface normal is maintained for each graph node and transformed along with the graph deformation. If the dot product of this transformed normal with the surface normal at the corresponding position on the target shape is less than 0.6, then the normals are considered inconsistent. The ω value for each poor correspondence is set to zero, and the ω value for all others is set to one. These heuristics are executed every time the optimization converges, and the optimizer is restarted with the new correspondence and ω values. The entire process (including the iterative improvement) converges when $|F_k - F_{k-1}| < 10^{-8}(1 + F_k)$.

3.4.5 Results

To illustrate the performance of the coupled correspondence optimization method, we conduct a series of experiments on synthetic data and real scans. The results are then compared with those of two recent non-rigid ICP variants. Our proposed global corre-

spondence optimization algorithm and N-ICP 1 implementation were both performed on a 3.0 GHz Quad-Core Intel Xeon machine with 8 GB RAM. The longest computation was for the torso example from Figure 3.14 and required 219 iterations until convergence which took 2 min 19 s in total.

Convergence and Robustness. Our first test case (Figure 3.10) evaluates the performance of our method on synthetic data with given ground truth correspondence. The source depth map is created by sampling a digital model of an elephant. We simulate a pose change by applying a warp to this model in order to obtain the deformed target shape. In addition, parts of both surfaces have been removed so that only subsets of both models are in correspondence.

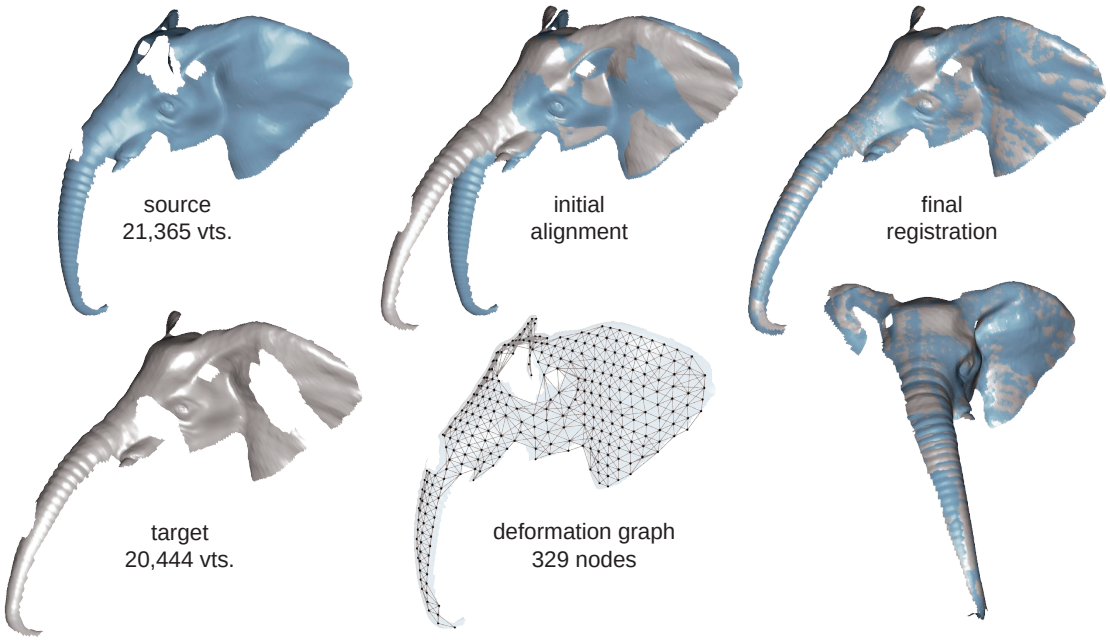


Figure 3.10: Evaluation with synthetic data.

The optimization correctly classifies this region of overlap by solving for appropriate correspondences and confidence weights as indicated by the color-coded energy terms in Figure 3.11. Large fitting errors E_{fit} are balanced by low confidence weights and hence a high value in E_{conf}. Since the objective function includes the augmented fitting term E^*_{fit}, where the squared distances to the corresponding points are scaled by the confidence weights, an optimal trade-off between alignment and deformation is found.

The graph in Figure 3.12 illustrates how the distance of the corresponding points with respect to the ground truth data evolves as the optimization progresses.

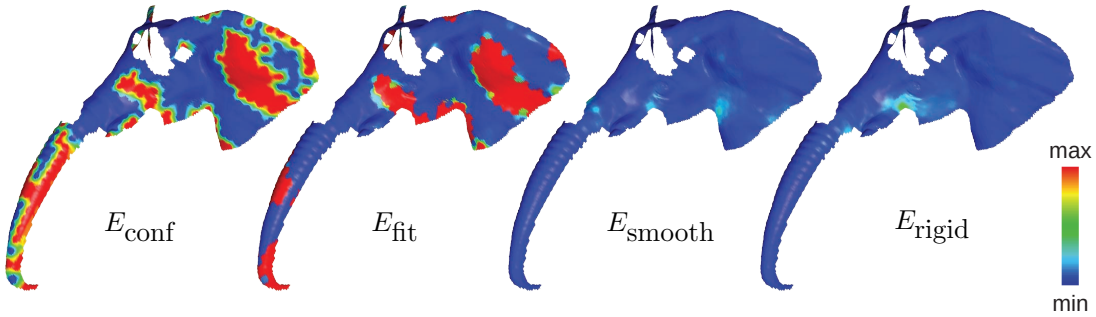


Figure 3.11: Different energy terms of the objective function.

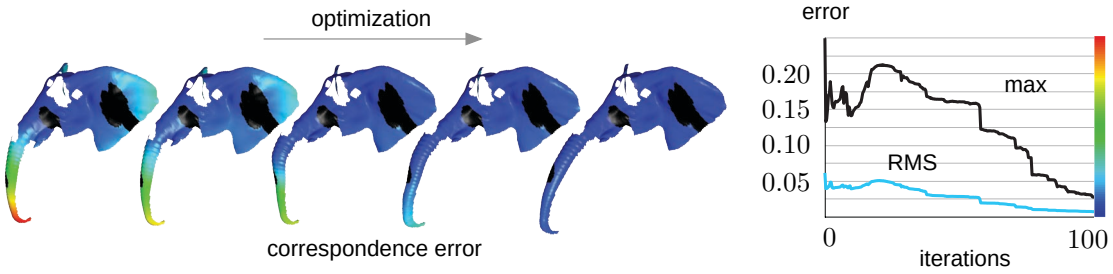


Figure 3.12: Left: Evolution of distance to ground truth correspondences during optimization. Right: Maximum and RMS error relative to the bounding box diagonal.

In Figures 3.13 to 3.15 we apply our method to real range scans. To simulate the effects of fast motion, we skipped several frames for the target shape. Figure 3.13 shows non-rigid registration of scans of a human face.

Our algorithm accurately captures the deformation on the cheeks, a mostly feature-less region. At the same time the relevant features of the face, such as nose, mouth, eyes, and ears are correctly aligned. Figures 3.14 and 3.15 show registration of scans of articulated objects, where most of the deformation is concentrated on a small region of the model. These examples are challenging for a marker-less algorithm, since the surface parts that contain most of the important features, the face or the fist, are substantially different in both scans and overlap only partially. Most of the correspondences are located on the torso and the arm, so that these regions dominate the energy terms of the objective function. Still, our algorithm is capable of aligning the facial

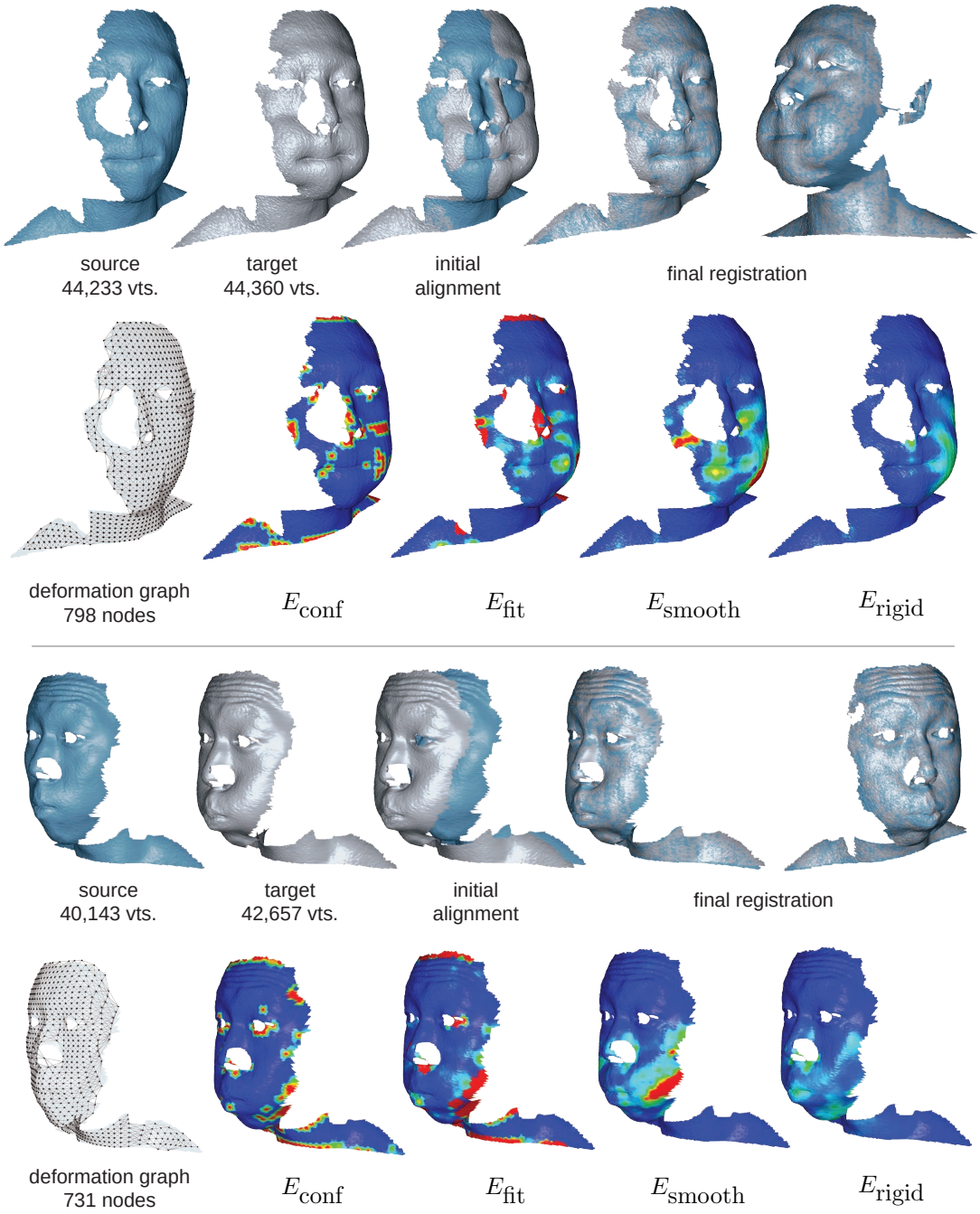


Figure 3.13: Registration of face scans. The deformation energies E_{smooth} and E_{rigid} illustrate that most of the deformation is concentrated on the cheeks. Both examples also contain a substantial rigid motion that is accurately solved for by the optimization.

features and accurately determines the regions of overlap.

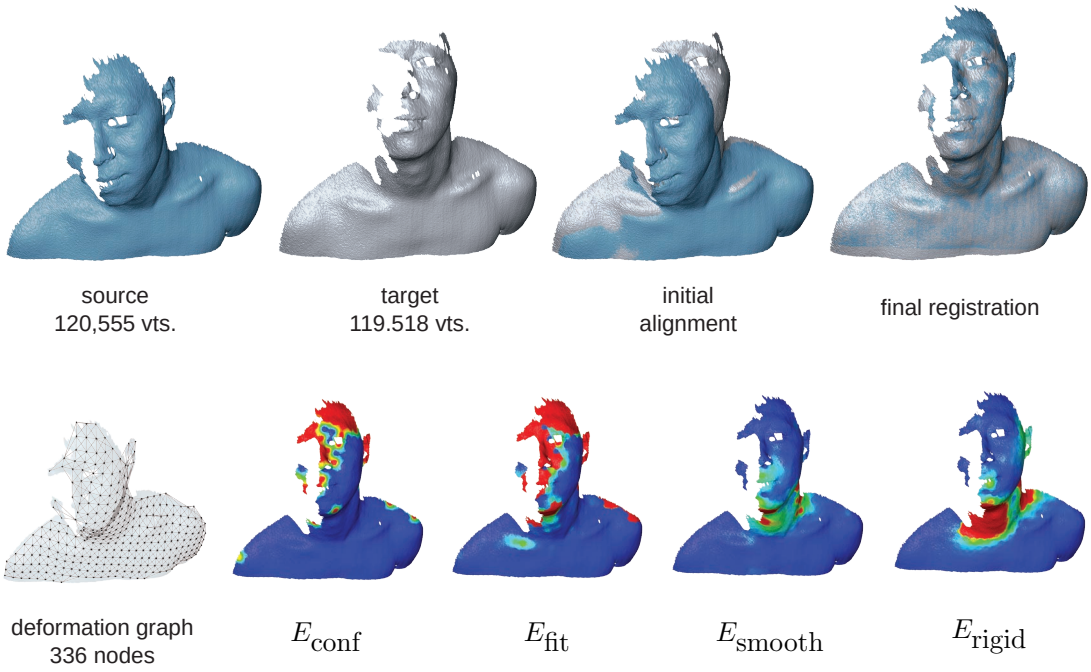


Figure 3.14: Registration of a torso. As indicated by E_{conf}, large regions in the face are only present in one of both models.

In this respect, the coupled optimization of correspondence points and deformation can be seen as a form of point-to-plane metric used for rigid ICP (c.f., [CM92]) where sliding along the surface is allowed and featureless regions would not penalize the optimization. The main difference is that in this optimization, the correspondence points with high confidence weights remain on the target surface, reducing the effect of approximation errors using a point-to-plane metric. Section 3.5 will show that for sufficiently high resolution of the target scan mesh, the point-to-plane strategy can achieve similarly accurate results.

In Figure 3.15, the algorithm correctly captures the bending of the arm but produces a slight misalignment in the fist. As the visualizations of the E_{conf} energy show, few reliable correspondences have been found in this region. This is mostly due to the inferior quality of the input data that leads to a poor WLS approximation of the depth map.

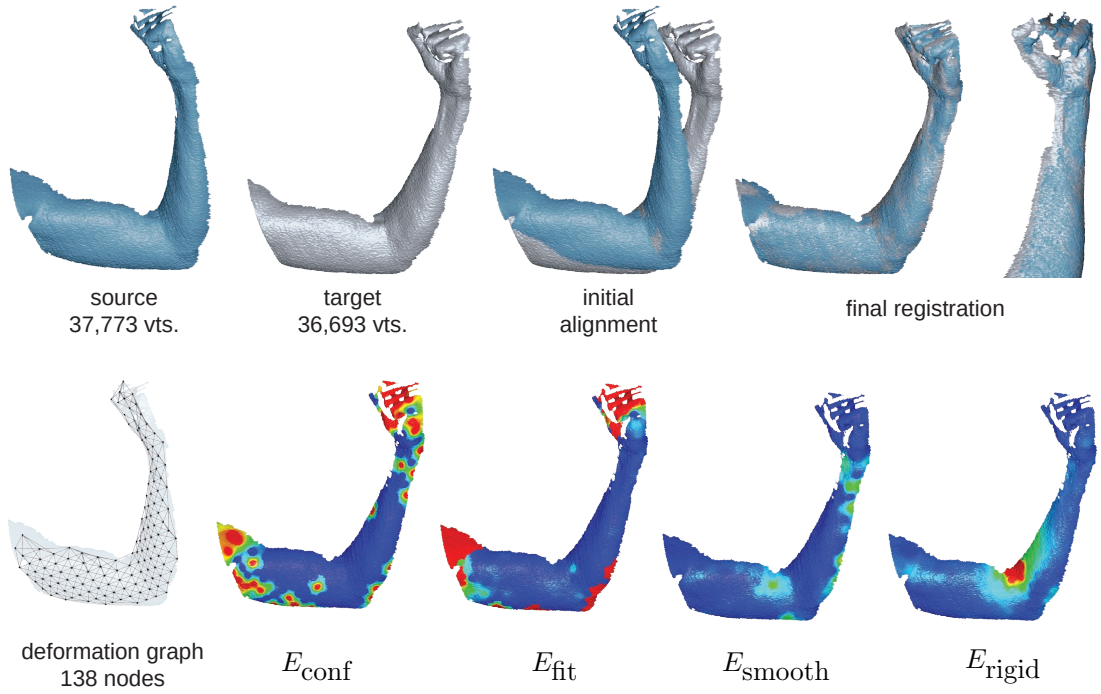


Figure 3.15: Registration of a bending arm. From this acquisition direction, the motion of the subject is considerably tangential w.r.t. to the surface.

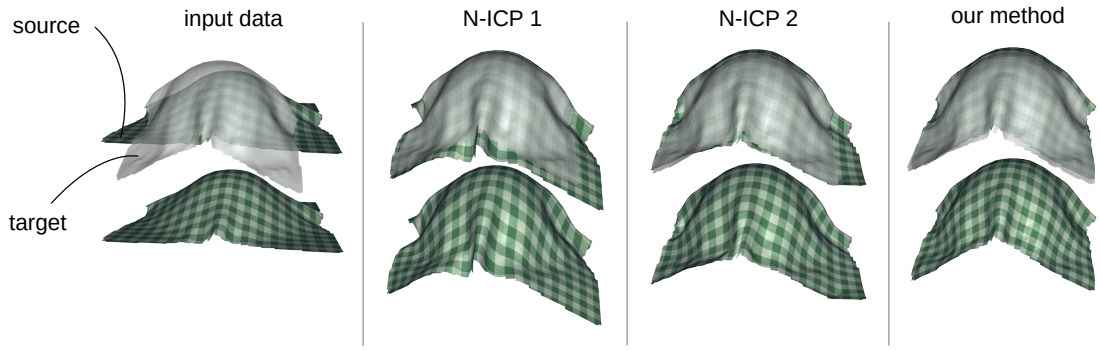


Figure 3.16: Performance comparison of the registration between two depth scans of a draping table cloth. While non-rigid ICP methods fail to preserve isometry, our method converges to the correct result.

Comparison to Related Work. The performance of different registration methods is shown in Figure 3.17. For each algorithm, we examine four test cases on the torso in order to visualize how different the poses of the input data can be, so that the registration is still able to converge successfully. We increase the number of frames between the

source and target input scans for each test as shown in the first four columns. The last column shows the first occurrences of notable misalignments for each approach. The initial alignment of our input data as shown in the first row is followed by a sequence of rigid alignments performed using rigid ICP with geometric stable sampling [GRIL03], normal compatibility pruning, and the point-to-plane metric. Subtle misalignments are already visible between frame 0 and 2.

N-ICP 1: The third row of Figure 3.17 shows the best results we could achieve using our implementation of the optimal step non-rigid ICP method (N-ICP 1) from [ARV07]. Instead of using the deformation model from [ACP03] as in the original work, we employed the the model from [SP04] which is known to produce comparable results The input meshes were decimated to 5% of the original size using the algorithm from [GH97] as the registration algorithm relies on dense correspondences which cannot be handled efficiently by the deformation model for our high resolution input scans.

Although a correct alignment could be found for the easiest test case, the registration fails for a registration between frame 0 and 6. The primary issue here is that the employed point-to-point metric in N-ICP 1 penalizes sliding of correspondences during the deformation process. A noticeable shrinking can also be observed which is due to the employed deformation model which only enforces smoothness over the local affine transforms.

N-ICP 2: The non-rigid ICP method described in [PHYH06, PMG+05] uses a combination of point-to-plane and point-to-point metric is used. The deformation model is also based on smooth affine transformations [ACP03]. The artifacts on the boundaries are due to the fact that correspondences to boundary edges were not pruned.

Besides the slight misalignment of the arm region for the pair of frames 0 and 6, the method breaks down more severely on the face region for the frames 0 and 12. The warped source shape is geometrically closer to the target scan than our approach. However the correspondences are semantically wrong as illustrated on the texture visualization in last row of Figure 3.17. We can depict semantic regions such as the mouth, the nose, the eyes, and the ears on the deformed source mesh. Ideally, the texture of each semantic region should correspond to its geometry. In addition to the slight geometric distortions on the nose, we observe that N-ICP 2 fails in matching the mouth region. The texture of the upper lip is matched to the geometry of the lower lip. Our method does not show any visible misalignments.

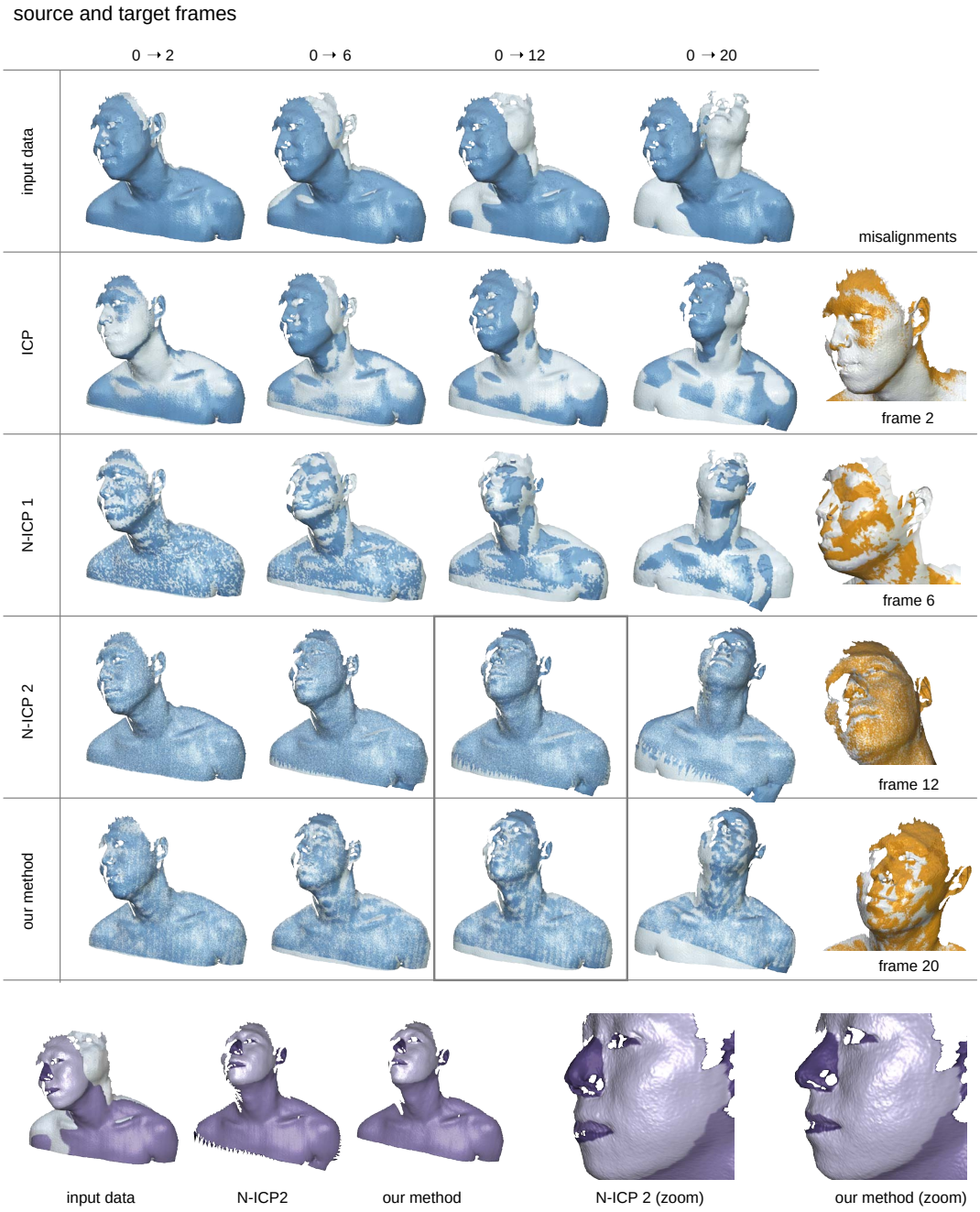


Figure 3.17: We compare the performance of our method with two recent non-rigid ICP methods (blue) and highlight misalignments (yellow). While for the frames 0 and 12 the results of N-ICP 2 are geometrically closer to the target, the texture visualization below (purple) shows that our method yields a semantically more correct alignment.

All methods break down for the registration between frame 0 and 20. As the amount of overlaps is too small for this pair of input data, our method cannot find reliable correspondences and a faithful alignment is not returned.

Table Cloth: We perform a comparison for the registration between two frames of a draping cloth falling onto an invisible sphere as shown in Figure 3.16. This example was created using a cloth simulation [GHF+07] and then depth sampled. The cloth input is exceptionally challenging as the deformation is more complex than for articulated objects and no stationary rigidness is present. Correspondence search is therefore more difficult and a correct registration would basically rely on an appropriate regularization during the deformation process. Unlike earlier non-rigid ICP methods, the alignment of our registration algorithm is able to recover the amount of isometry between the source and target shapes. The superior quality of our method is amplified when comparing the distortion of the checkers around the folded regions.

Final Warp. The continuous WLS approximation of the target scan is essential for the optimization, since it allows a unified treatment of valid depth samples and holes, and gives access to derivative information of the corresponding points $\mathbf{c}(u_i, v_i)$ with respect to the unknowns (u_i, v_i). As a result, the above registration procedure computes a warp between the source scan and the WLS approximation of the target. Since the WLS approximation smooths some features, our algorithm performs one final step to find a deformed source scan that matches the target scan more accurately. The correspondence positions are projected from the WLS approximation onto the target scan. Any projected correspondence that is farther than 2 cm from its source graph node is discarded. The remaining correspondences are used to solve one final deformation problem in which they are fixed as constants and are not controlled by the optimization. The per-node ω values are also removed from the optimization, so that only the deformation model parameters are solved for. This essentially warps the source scan directly to the target scan using the valid correspondences found during the optimization.

Limitations This framework relies on the fact that the target surface is parameterized. Although most acquisition systems provide a parameterization implicitly (e.g. depth maps or 2D sweep patterns), we will generalize this coupled optimization in Section 3.5 to handle manifold surfaces without the requirement of computing a parameterization on the target surface. In particular, we will replace the explicit correspondence optimization by an implicit point-to-plane error metric in the deformation computation.

Combining the correspondence and deformation estimation into a single, non-linear optimization is essential for the effectiveness of this method. However, this global scheme leads to a comparatively high computational cost for real-time acquisitions.

One important means to improve performance is the reduced deformable model that decouples the computational complexity from the size of the input scans. We observe that a uniformly sampled graph does not adapt to the geometry or the deformation of the processed data. Consequently, the smallest feature that we want to capture determines the resolution of the graph and thus leads to highly over-sampled graphs in mostly rigid regions. On the other hand, if the graph is too coarse, small-scale deformations cannot be captured accurately. This effect is noticeable for the bending arm example (see Figure 3.15), where the fingers are not appropriately matched due to inadequate resolution of the deformation graph. Section 4.2.3 will introduce an adaptive graph refinement method to overcome this limitation.

Another limitation is that our deformation complexity is decoupled from the mesh resolution. This fact prevents us from capturing very fine detail changes such as wrinkles in facial expressions. To overcome this problem, we consider an additional detail synthesis pass that uses dense correspondences and a linear deformation model for better efficiency. One such method is described in Section 4.2.5.

3.5 A Robust Non-Rigid ICP Algorithm

To ground our preliminary findings, we now derive a simpler algorithm that consolidates all the important ideas presented in earlier sections. In short, we wish to extend our global correspondence optimization [LSP08] to handle general surface (i.e., not a depth map) but, at the same time, avoid the headaches associated with the construction of a parameterized proxy. To this end, we develop a special variant of non-rigid ICP that achieves comparable accuracy and robustness for (smooth!) polygonal meshes. Additionally, the new algorithm is considerably easier to implement and can handle incomplete general surfaces. Finally, we successfully demonstrate the effectiveness of this method for solving non-rigid registration problems in several applications: geometry and motion reconstruction [LAGP09], dynamic shape completion [LLV+10], and even facial rigging [LWP10].

3.5.1 Requirements

Global Consistency. The key insight given by our previous global correspondence optimization framework is the importance of coupling correspondence and deformation optimization. When both elements are coupled, correspondences can be determined in a globally spatial consistent fashion, i.e., matching a point would affect all other points. While earlier non-rigid ICP algorithms interleave the two procedures in an attempt to achieve the same effect, each deformation step is bounded by the closest point estimate (or some other proximity heuristic). Our correspondence optimization framework, however, overcomes this issue by allowing the correspondences to move along the target surface within the continuous optimization.

Local Rigidity Maximization. As-rigid-as possible deformations are essential for non-rigid matching. Since we consider an iterative approach, any surface distortion during the optimization procedure, would prevent accurate matches in further iterations. We discovered that non-linear deformation models that locally maximize rigidity (such as embedded deformation) are remarkably effective in preserving details for accurate surface matching.

Stiffness Reduction. Our experiments show that non-rigid registration should generally begin with a strong stiffness which is progressively reduced whenever convergence is detected. Conceptually, this strategy follows a *coarse-to-fine pattern* (similar to a branch and bound approach) in the energy landscape where additional degrees of freedoms are introduced step-by-step. Even when the stiffness is known a-priori, this heuristic is able to greatly avoid local minima.

3.5.2 Implementation

The non-rigid ICP pipeline is directly derived from the rigid case as detailed in Section 3.1.3. Instead of a rigid transformation, we now have a deformation optimization stage (c.f. Figure 3.18) in which we can can control its regularization (i.e., global consistency). The non-rigid framework is divided into an inner an outer loop. The rationale behind this design principle is associated with the stiffness reduction strategy. The rule consists of keeping the same deformation regularization (i.e. same energy landscape) and solve the best we can using multiple iterations of closes point estimation until we reach convergence (inner loop). Once convergence is detected we may reduce the stiffness and repeat the entire procedure until the regularization parameter has reached a user prescribed threshold.

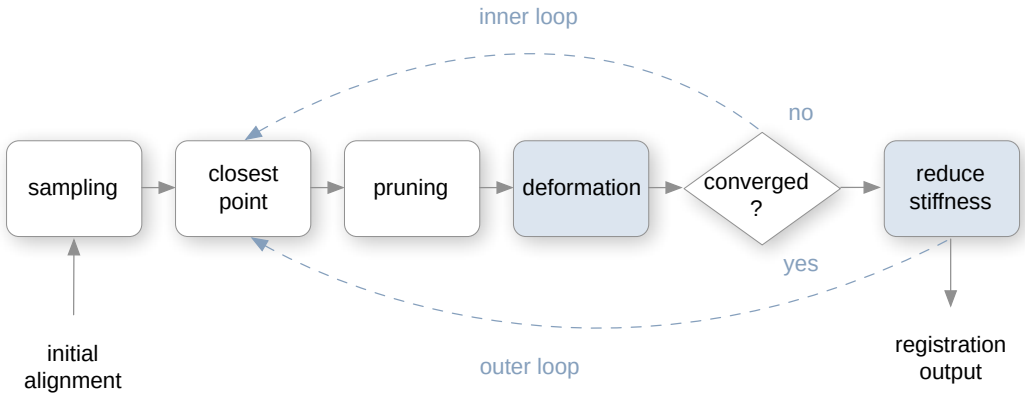


Figure 3.18: Our robust non-rigid ICP algorithm follows the design principles of a standard rigid ICP pipeline. As opposed to the rigid case, the transformation stage is replaced with a deformation optimization and regulates the stiffness of the deformation in the outer loop.

Starting with an initial alignment (can be quite far apart in practice) between a source and target mesh, $\mathcal{S}_d(t_1)$ and $\mathcal{S}_d(t_2)$, the objective is again to compute the deformed mesh $\tilde{\mathcal{S}}_d(t_2) = \Phi(\mathcal{S}_d(t_1)$ such that $\mathcal{S}_{t_1 \cap t_2}(t_1)$ overlaps and $\mathcal{S}(t_1) \setminus \mathcal{S}_{t_1 \cap t_2}(t_1)$ deforms in an "natural" way. Depending on the density of $\mathcal{S}_d(t_1)$, we may wish to downsize the resolution in order to reduce redundant computation for more efficiency. The uniform resampling algorithm presented in Section 3.1.3 is generally a good choice and produces a new set $\hat{\mathcal{V}}$ containing $N_{\mathcal{C}}$ vertices. We usually choose a target sampling distance $l_{\mathcal{C}}$ equal to half of the average edge length distance for a uniformly sampled target mesh $\mathcal{S}_d(t_2)$. Once the closest points $\mathbf{c}_i \in \mathcal{S}_d(t_2)$ are determined for each source vertex $\mathbf{v}_i \in \mathcal{S}_d(t_1)$, we obtain a set of correspondence pairs expressed by the tuple $(\mathbf{v}_i, \mathbf{c_i})$ where $i = 1, \ldots, N_{\mathcal{C}}$. We decorate each correspondence $\mathbf{c}_i$ with a discrete confidence weight $w_i \in \{0, 1\}$ and initialize ot with $w_i = 1$. Next, a pruning procedure sets $w_i = 0$ for all correspondences $\mathbf{c}_i$ that lie on the mesh boundaries of $\mathcal{S}_d(t_2)$, are too far from their source vertices $\mathbf{v}_i$, or disagree with their normals (c.f. Section 3.1.3). The pruning step is an essential for handling the general part-in-part problem and can be viewed as a discrete form of overlap region $\mathcal{S}_{t_1 \cap t_2}(t_1)$ optimization. So far, we *reused* the same components of a standard rigid ICP implementation.

Deformation Optimization. We consider embedded deformation as the underlying deformation model (c.f. Section 3.2.4) and use the closest points $\mathbf{c}_i \in \mathcal{S}_d(t_2)$ as initial constraint estimates. Graph nodes should sample the target surface with a slightly lower density than $\hat{\mathcal{V}}$. We generally use the same uniform sampling algorithm as for $\hat{\mathcal{V}}$ and use a target sampling distance $l_{\mathcal{G}} = 2\, l_{\mathcal{C}}$. The regularization energies E_{rigid} and E_{smooth} remain unchanged. We simply replace the fitting term which originally prescribes positional constraints with a combined point-to-plane and point to point energy:

$$\begin{aligned} E_{\text{fit}} &= E_{\text{plane}} + \alpha_{\text{point}}\, E_{\text{point}} \\ &= \left(\mathbf{n}^\top (\mathbf{c}_i - \mathbf{v}_i)\right)^2 + \alpha_{\text{point}} \left\|\mathbf{c}_i - \mathbf{v}_i\right\|_2^2 \quad , \end{aligned} \tag{3.63}$$

where $\mathbf{n} \in \mathbb{R}^3$ is the unit normal vector of $\mathbf{c}_i$. The point constraint E_{point} is only used to stabilize convergence since the correspondence point estimations are discrete. We typically choose a small weight $\alpha_{\text{point}} = 0.1$. The point-to-plane term E_{plane} allows the correspondences $\mathbf{c}_i$ to glide along the surface tangents of $\mathcal{S}_d(t_2)$ during the optimization. In this way we approximate the coupling of correspondence and deformation optimization within a single continuous optimization. We accomplish global consistency by incorporating the regularization terms into a global objective function:

$$E_{\text{tot}} = E_{\text{fit}} + \alpha_{\text{rigid}}\, E_{\text{rigid}} + \alpha_{\text{smooth}}\, E_{\text{smooth}} \quad , \tag{3.64}$$

where the choice of α_{rigid} and α_{smooth} is subject to a stiffness relaxation procedure explained in the next paragraph. Like E_{rigid}, the partial derivatives of the point-to-plane energy E_{plane} are also quadratic in their optimization variables. Similarly, the Gauss-Newton method is ideal for this non-linear optimization. Notice that the non-linear optimization of E_{tot} might itself take multiple cycles depending on the scale of α_{rigid} and α_{smooth}. Therefore, it is essential to let E_{tot} converge before recomputing the closest points (omitting this step is a common mistake!). We refer to Section 3.2.4 for more details on how to efficiently solve for the deformation $\Phi(\mathcal{S}_d(t_1)$.

Stiffness Reduction. For each pairwise alignment, we initialize the registration with high stiffness weights $\alpha_{\text{smooth}} = 100$ and $\alpha_{\text{rigid}} = 1000$. We then alternate in each iteration between correspondence computation and deformation by minimizing E_{tot}. If the relative total energy did not change considerably between iterations j and $j+1$ (i.e., $|E_{\text{tot}}^{j+1} - E_{\text{tot}}^{j}| / E_{\text{tot}}^{j} < \sigma$), we relax the regularization weights to $\alpha_{\text{smooth}} \leftarrow \frac{1}{2}\, \alpha_{\text{smooth}}$ and $\alpha_{\text{rigid}} \leftarrow \frac{1}{2}\, \alpha_{\text{rigid}}$. As mentioned previously, this relaxation strategy effectively improves

the robustness by avoiding suboptimal local minima and allows handling pairs of scans that undergo significant deformations. In all our experiments we use $\sigma = 0.005$. The iterative optimization is repeated until $\alpha_{\text{rigid}} < 0.1$ or until a maximum number of iterations $N_{\max} = 100$ is reached.

3.5.3 Results and Discussion

This simple non-rigid formulation of our global correspondence framework faithfully translates the concept of correspondence optimization to an ICP pipeline. As before, this method is designed for acquisition settings where the region of overlap is not known a priori, no explicit correspondences are provided, and minimal assumptions on the type of deformation are made. Although this approach involves additional closest point computations, it is in practice more efficient than performing a global correspondence optimization since surface parameters $\mathbf{u}$ need not to be solved. Additionally, the construction of a continuous proxy representation (WLS approximation) can also be disregarded. Large deformations can be recovered using this algorithm without the help of any (usually less reliable) high level and pose invariant shape descriptors. The performance of our non-rigid ICP algorithm is depicted in Figure 3.19, 3.20, and 3.21.

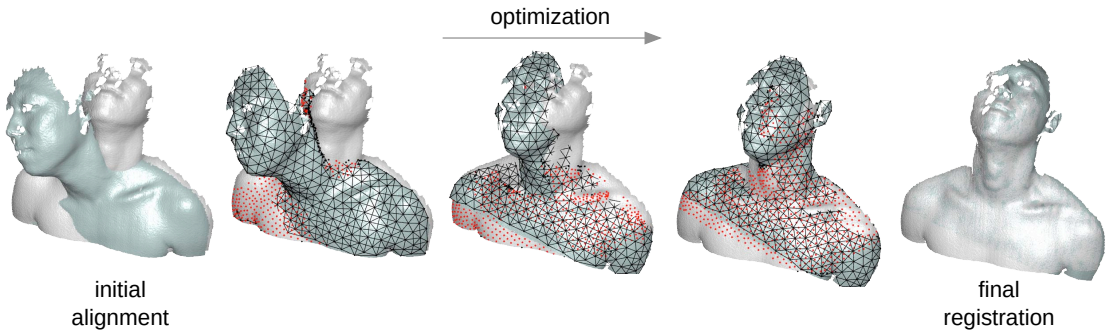


Figure 3.19: Pairwise non-rigid registration of torso. The red dots describe the corresponding points $\mathbf{c}_i$ on the target surface where red means $w_i = 1$ and black $w_i = 0$

The only requirement here is that the target mesh $\mathcal{S}_d(t_2)$ should be smooth and dense enough such that its tangent planes $\mathbf{n}^\top \mathbf{x} = 0$ provide a good local approximation of the surface, i.e., $\mathcal{S}(t_2)$ should be differentiable. Note that this local linearization is just the first order bi-variate Taylor approximation of $\mathcal{S}(t_2)(\mathbf{u})$ about $\mathbf{c}_i$. Although higher-order Taylor expansions may be considered for better approximation, we have not yet fully examined this option.

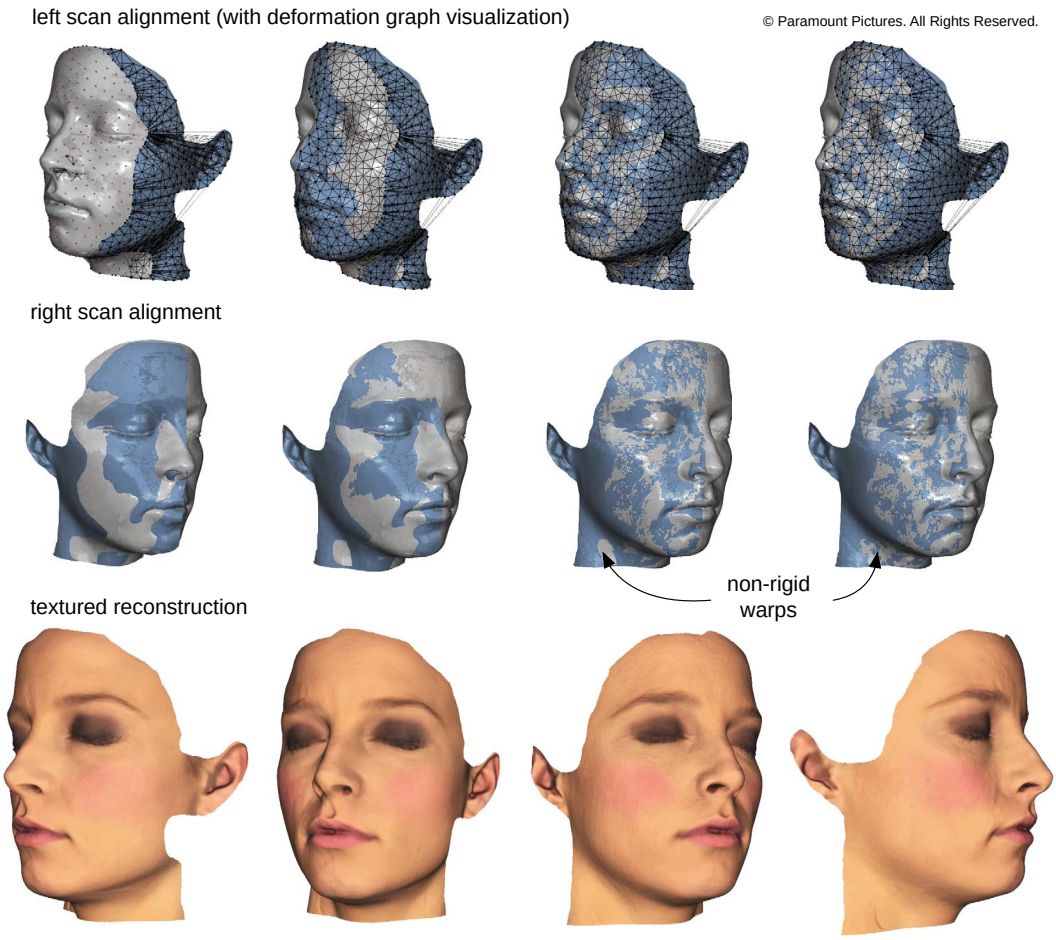


Figure 3.20: High accuracy non-rigid registration for ear-to-ear reconstructions. Three Light Stage scans (left, center, right) are aligned with each other in the first two rows. The third row shows a result after mesh integration using Poisson reconstruction where the textures are simply blended by linear combination.

While E_{rigid} itself enforces detail preservation through local rigidity maximization, combining it with E_{smooth} can be interpreted as a form of elasticity constraint w.r.t. a rigid motion invariant rest-state pose $R\,\mathcal{S}_d(t_1) + \mathbf{t}$. This is a reasonable assumption for performing registration between deformed shapes of the same subject (problems of type Cat II). For shrink-wrapping purposes (Cat IV problems), such as warping a generic template model onto a custom scan, one might consider reinitializing the initial pose in an interleaved fashion in order to achieve a plastic deformation behavior. This can be easily achieved by simply restarting the entire process and updating the new initial pose $\mathcal{S}_d(t_1)^{m+1} \rightarrow \tilde{\mathcal{S}}_d(t_2)^m$.

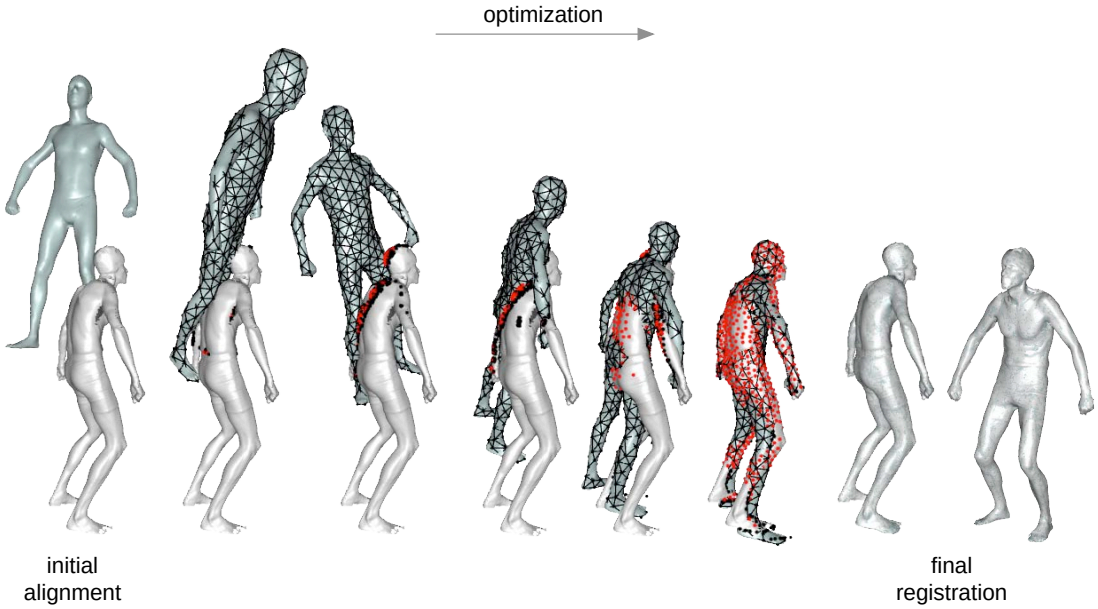


Figure 3.21: Fully unsupervised shrink-wrapping of a generic human body template to a target scan. Even from an extremely different initial alignment (shapes are looking in opposing directions), the registration can be successful. Notice that no sparse feature were involved.

This non-rigid registration framework is flexible in the sense that arbitrary sparse constraints (user-guided, marker-based, or automatically computed) can be trivially incorporate using an additional point-to-point constraint. However, for many applications such as multi-frame tracking, adding positional constraints based on shape descriptors can be highly unreliable as single wrong correspondence may hinder a successful convergence. We therefore argue that reliability can be effectively satisfied only based on a combination of proximity heuristics and global consistency.

Limitations. As highlighted above, this non-rigid ICP requires the target surface to be smooth and densely sampled. If this is not the case, a surface parameterization is required and we may resort to our more complex global correspondence optimization framework. Furthermore, depending on the initial pose, the algorithm does not guarantee convergence as we use a standard Gauss-Newton solver. It also remains unclear, how far apart source and target mesh can be, such that the optimization still leads to the correct answer. Unfortunately, for very large deformations, convergence to a suboptimal local minimum cannot be fully avoided as illustrated in Figure 3.22. Ultimately, the

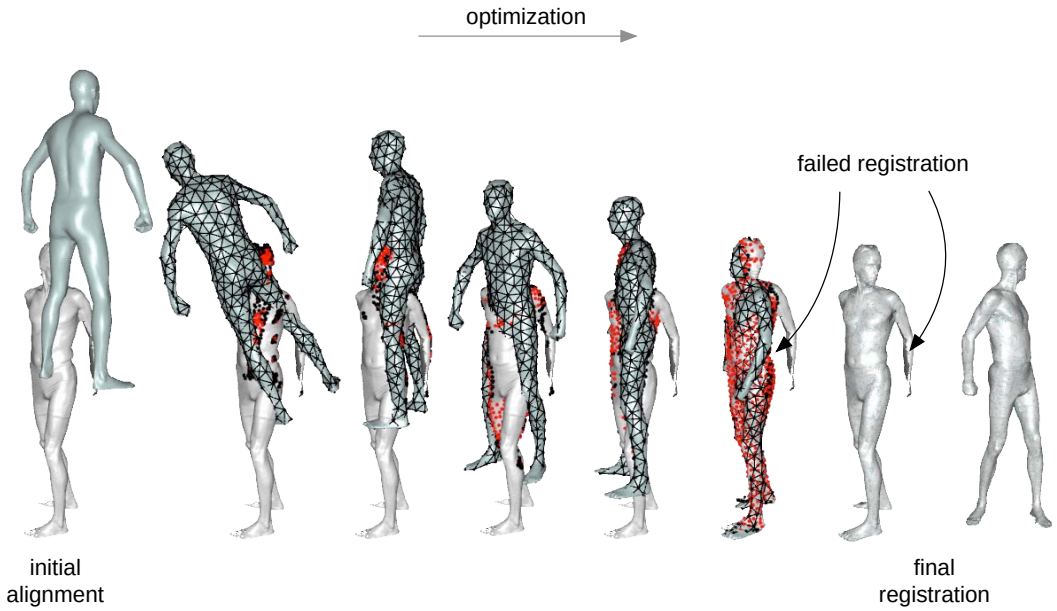
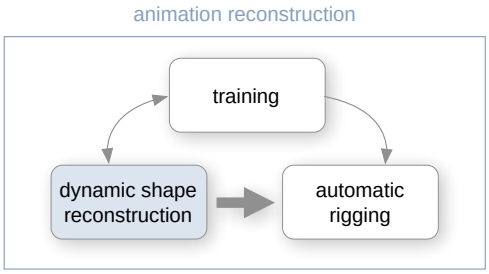


Figure 3.22: Failed shrink-wrapping example. While the right arm is well-aligned, the left one converged to the body instead of the arm.

problem of partial registration of deforming scans is inherently ill-posed and no algorithm will be applicable for all acquisitions scenarios. Hence, the performance of different algorithms is generally assessed through empirical experiments using appropriate ground truth information (such as texture markers).

4

Dynamic Shape Reconstruction



Real-time acquisition techniques produce continuous sequences of incomplete scans; non-rigid registration methods establish correspondences between these consecutively captured scans by warping one shape onto another. All in all, these algorithms are foundational building blocks for the central goal of this thesis, namely the reconstruction of complete digital models in motion. Until now, we have been only looking at pairs of shapes. Now we will extend our ideas to process longer recordings with numbers of frames up to two orders of magnitude larger. Consequently, this step will allow us to capture dense surface motion in addition to high-resolution shapes. Having the goal in mind to recreate an animated, complete shape and to fulfill the first stage of our animation reconstruction pipeline (figure on the right) we propose the following:

Hypothesis: *Through transitivity, high-resolution geometric details can travel across multiple frames to fill in all the occluded regions (where detail is missing) and better estimate the surface dynamics in those regions.*

This chapter investigates two important problems in dynamic shape reconstruction:

Problem 1: The first objective consists of recreating a full digital representation of a deforming subject that is consistent across a sequence of input scans. In particular, *high frequency* dynamics have to be accurately captured and separated from acquisition noise, and fine-scale geometric details must be reproduced in occluded areas. For regions with largely missing data due to insufficient coverage or (self-) occlusions, we need to resort to additional geometric priors. We show that tracking a coarse template model (using our correspondences) in an initial pass considerably simplifies the problem of hole-filling and dealing with complex deformations.

Problem 2: While being often difficult to build, template models also have the disadvantage of having a *fixed topology*. In particular, they do not explicitly model multiple surface layers (such as cloth gliding on a human body) and topology variations in the subject. Therefore, our second goal consists of simply filling holes in incomplete acquisitions with challenging topology changes by skipping the requirement of tracking surface points through the entire recording. While shape completion is a well-studied problem for static surfaces, we require in a dynamic setting to fill holes with patches that naturally deform with the rest of its surroundings. Hence, temporally coherent shape completion requires accurate correspondences to be established (at least) within a short temporal window and should be sufficiently robust to handle severe topology variations.

Both problems highly depend on the quality of inter-frame correspondences which we obtain through non-rigid registration. Thus, we are faced with the same challenges as those presented in the previous chapter, namely deformations that are too large and overlapping regions that are too small. Unfortunately, processing *multiple frames* consecutively brings additional headaches. While we assume that our input scan sequences are reasonably coherent over time (moderate deformations), we cannot fully avoid misalignments in each non-rigid registration step. By successively computing correspondences through the entire recording, tracking accuracy may deteriorate and result in *accumulation of errors* and *drifts*.

We begin this chapter with an overview of recent techniques (c.f., Section 4.1) developed for reconstructing dynamic shapes from sequences of real-time input scans. For completeness, we also discuss several hole-filling techniques designed for subjects

that are statically captured (i.e., no temporal coherence).

The framework in Section 4.2 addresses the first problem and uses input data captured from the single-view real-time structured light scanner presented in Weise and coworkers [WLG07]. The approach solely relies on a crude approximation of a template model and uses it to reconstruct large scale deformations. Aggregated fine-scale dynamics are being reintroduced in a second pass. While primarily designed for a single-view setup, this algorithm can be easily extended to multi-view reconstruction.

To tackle the second problem, we develop a system in Section 4.3 that generates a sequence of watertight meshes by filling large holes in scanned data with temporally coherent patches. To eliminate the requirement of using a template, our framework processes input data obtained from the multi-view photometric stereo system introduced in Vlasic and colleagues [VPB+09].

4.1 Related Work

There are generally three ways to obtain a complete, animated digital model from a sequence of incomplete scans captured in real-time:

- **Template-based methods** inherently produce a complete representation of the scanned subject. The dynamics of the template are typically inferred using tracking or non-rigid registration techniques. In particular, a full motion field can be directly deduced from the surface points on the template.

- **Methods that do no involve a template** address a significantly more challenging problem of 4D space-time surface reconstruction with time being treated as an additional dimension to a given 3D point cloud. This approach is typically characterized by requiring the subject to deform smoothly over time and not permitting fast complex motions.

- **Filling holes** in each frame independently would also produce to a sequence of watertight meshes. However, because of large (self-) occlusions (even in a multi-view acquisition setup), hole-filling patches will have inconsistent dynamics. While most shape completion techniques are limited to static objects, some recent methods were introduced to lift these restrictions. These algorithms can handle very large pose variations but are limited to quasi-articulated motions. Approaches in this category do not deliver any motion vectors.

Let us review some of the major research dedicated to these three different avenues in more detail:

Template-Based Methods. Template models are particularly useful in closing large holes and handling complex deformations during registration as highlighted in Section 3.3.2 for the pairwise case [BV99, ACP03, PMG+05, ARV07]. The same holds in a multi-frame setting where surface data is tracked or correspondences established in order to animate the template. As opposed to a pairwise setting, manual intervention between each frame is impractical when processing entire recordings. Also, because some of these automatic methods are prohibitively expensive in terms of computation [ASP+04, BBK06], they are usually not suitable for long scan sequences.

While motion capture systems [Vic] are still widely spread in the industry, Park and Hodgins [PH06, PH08] developed a system that uses a very dense and large set of markers to capture and synthesize dynamic motions such as muscle bulging and flesh jiggling. While high resolution motions can be captured accurately, marker-based motion capture systems typically have a time-consuming calibration process and high hardware cost, and require actors to wear unnatural skin-tight clothing with optical beacons.

Marker-less methods are widely used in the acquisition and modeling of facial animations. In [ZSCS04], the deformation of an accurate face template is driven by time-coherent optical flow features and geometric closest point constraints. Since many features in a human face are persistent, their system can robustly handle long sequences of facial animations.

More recently, several papers avoid the use of markers to reproduce complex animations of human performances and cloth deformations from multi-view video [BPS+08, dAST+08, VBMP08]. The latter two methods initialize the recording process with a high resolution full-body laser scan of the subject in a static pose. A low-resolution template model is created to robustly recover complex motions by combining various tracking and silhouette fitting techniques. Details of the high resolution models are then transferred back to the animated template. While large-scale deformations such as flowing garments are nicely captured, fine-scale geometric details such as folds that are not persistent in the surface are captured in the high-resolution model, remaining permanently throughout the reconstructed animation and possibly yielding unnatural deformations.

An extension of this approach has been presented in [ATD+08] that follows a similar rationale to our method presented in Section 4.2. A low-resolution template is

tracked and subsequently enriched with local detail extracted from the acquired data. However, the specifics of this system differ substantially from our solution. The input stems from a multi-view acquisition system using eight video cameras, the template tracking is based on a shape-skeleton and silhouette matching, and the detail synthesis is performed based on surface normals reconstructed using shape from shading.

Registration Without A Template. Since creating an accurate and sufficiently detailed template of a deforming object can be difficult, various methods have been proposed that do not rely on a complete model.

Explicitly computing correspondences over long sequences is an error-prone process. To avoid these issues Mitra and colleagues [MFO+07] cast the problem of computing hole-free surfaces from unregistered dynamic performance geometry as a spatio-temporal 4D interpolation problem.

Süssmuth and coworkers [SWG08] introduced a space-time approach that first computes an implicit 4D surface representation. A template is extracted from the initial frame and warped to the subsequent frames by maximizing local rigidity. These methods require adjacent frames to be sufficiently dense in space and time and are mainly designed for articulated motions.

Similarly, the method described in [WJH+07] uses a statistical framework to solve for the dynamic shape under an as-rigid-as-possible motion and impose temporal smoothness.

Significant performance improvements were achieved in a follow-up work using a volumetric meshless deformation model [WAO+09]. Here, they globally solve for an optimal deforming representative shape and minimize the effects of drift by employing a hierarchical scheme to register pairs of surfaces.

Sharf and colleagues [SAL+08] introduced a volumetric space-time reconstruction technique that represents shape motion as an incompressible flow of material through time. This strong regularization makes the method particularly suitable for very noisy input data. However, this introduces noticeable flickering in the reconstructions. Moreover, the deformation of most real-world objects do not exactly preserve volume (e.g., loose clothing). As opposed to the methods heretofore presented, this technique does no provide correspondences between frames. Hence, temporal smoothing is non-trivial.

Hole-Filling Techniques. Hole filling is commonly used in surface reconstruction for static objects. Various strategies exist that either explicitly operate on polygonal meshes [Hel98, Lie03] or implicitly via a volumetric representation [CL96c, CFB97, DMGL02, KBH06]. Regardless of the heuristic used to fill in the missing geometry, the end result is a hole-free surface. We refer to [Ju09] for an in-depth discussion of various hole filling strategies and heuristics. While these methods generate compelling hole-free static surfaces, naively applying them to every surface in a dynamic performance separately will result in topological incorrectness and temporally incoherent sequence of surfaces.

For dynamic objects, shape completion techniques were introduced that aggregate incomplete static scans that are arbitrarily captured. These methods are characterized by strong assumptions imposed on the deformation model (often quasi-articulation) and by the limited number of scans being used.

Pekelny and Gotsman [PG08] assume that the dynamic performance consists of articulations of rigid parts. Starting from a manual segmentation, an optimal rigid motion is computed for each part. Finally, information is accumulated (forward in time) for each rigid part to fill holes and improve the quality of the reconstructed surface.

Chang and Zwicker [CZ09] propose a method that does not require any manual segmentation or template. However, their method is limited to quasi-articulated motion of the subject.

Zheng and colleagues [ZST+10] automatically extract a consensus skeleton to derive a consistent temporal topology. However it also assumes that the underlying shape is clearly articulated, which is not always the case for subjects wearing loose clothing.

4.2 Geometry and Motion Reconstruction

We extend our findings on robust pairwise non-rigid registration developed in Chapter 3 and introduce a novel template-based dynamic registration algorithm that offers significant improvements in terms of accuracy and robustness over previous methods. A key feature of our approach is the separation of large-scale motion from small-scale shape dynamics. We introduce a time- and space-adaptive deformation model that robustly captures the large-scale deformation of the object with minimal assumptions about the dynamics of the motion and without requiring an underlying physical model or

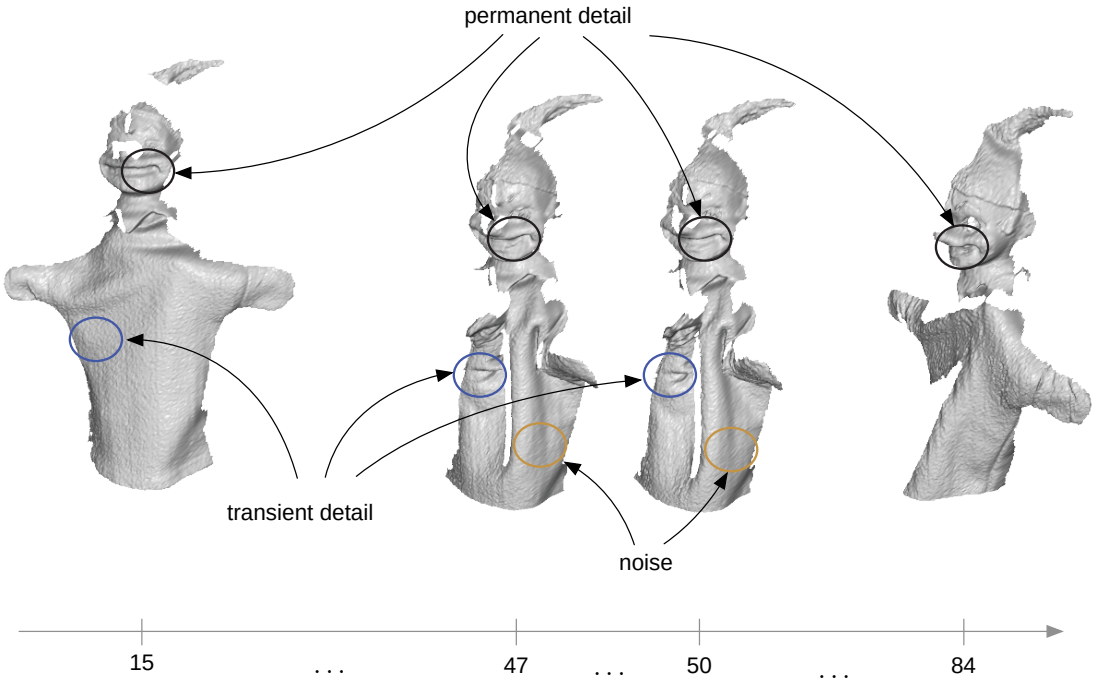


Figure 4.1: Deforming shapes typically contain both permanent detail, such as the face region of the puppet, and transient detail, such as the dynamic folds in the cloth. Transient detail still persists over a number of adjacent frames and can thus be distinguished from temporally incoherent noise.

kinematic skeleton. Our method dynamically adds degrees of freedom to the deformation model where needed, effectively extracting a generalized skeleton for the acquired shape. Small-scale dynamics are handled by a novel detail-synthesis method that computes a displacement field to adjust the deformed template to match the high-resolution input scans. The combination of these tools allows the efficient processing of extended scan sequences and yields a complete high-resolution geometry representation of the scanned object with full correspondences over all time instances.

We make a clear distinction between *static* and *dynamic* detail. Static detail includes all small-scale geometric features that are *persistent* in the shape and are not affected by the motion of the object. In the example shown in Figure 4.1, the mouth, eyes, and nose of the hand-puppet are static detail, since the entire face region is rigid. Dynamic detail consists of features that are *transient*. Deformation of the object can cause dynamic detail to appear and disappear, such as the folds in the body of the puppet. Our non-rigid registration method makes use of a template model to reconstruct the overall

motion of the shape and provide a geometric prior for shape completion and topology control. In contrast to recent methods in performance capture [dAST+08, VBMP08], we deliberately remove fine-scale detail from the template to avoid confusing static detail with dynamic detail. High-resolution templates from rigid scans typically have all detail "baked in", even transient features that are then erroneously transferred to all reconstructed surfaces (see also Figure 4.12). Our detail synthesis method automatically extracts detail from the high-resolution 3D input scans, propagates detail into occluded regions, and separates salient features from high-frequency noise.

The methods introduced in this framework are general in that they are not specifically designed for a certain acquisition setup or particular motion models. Our tool requires no user interaction beyond aligning the template with the first scan and specifying a few global parameters.

The reconstructed surface meshes come with temporally consistent correspondences, which enables further applications such as mesh editing, texturing, or signal processing to be applied to the animation sequence. We demonstrate the versatility of our approach by showing high-resolution reconstructions of highly deformable shapes such as cloth, as well as the more coherent motion of articulated shapes. In addition, our purely data-driven algorithm is able to accurately reproduce subtle secondary motions such as hand tremor, or the behavior of complex materials such as the crumpling of a paper bag.

4.2.1 Overview

We perform our reconstructions on the data obtained from the real-time acquisition system of [WLG07]. The scanner provides dense depth maps with a high spatial resolution of 0.5 mm. This allows us to capture fine-scale geometric detail of deforming objects at high temporal resolution. As highlighted in Section 2.1.2, input scans are typically highly incomplete and contain considerable amounts of measurement noise. We found that a template model is essential as a geometric and topological prior for the robust reconstruction of shapes that undergo complex deformations, in particular for single-view acquisition, where large parts of the object are occluded.

Figure 4.2 gives an overview of our processing pipeline. Static acquisition is used to reconstruct the initial template. We remove all high-frequency detail from the template using low-pass filtering, as described in Section 2.2, to avoid transferring poten-

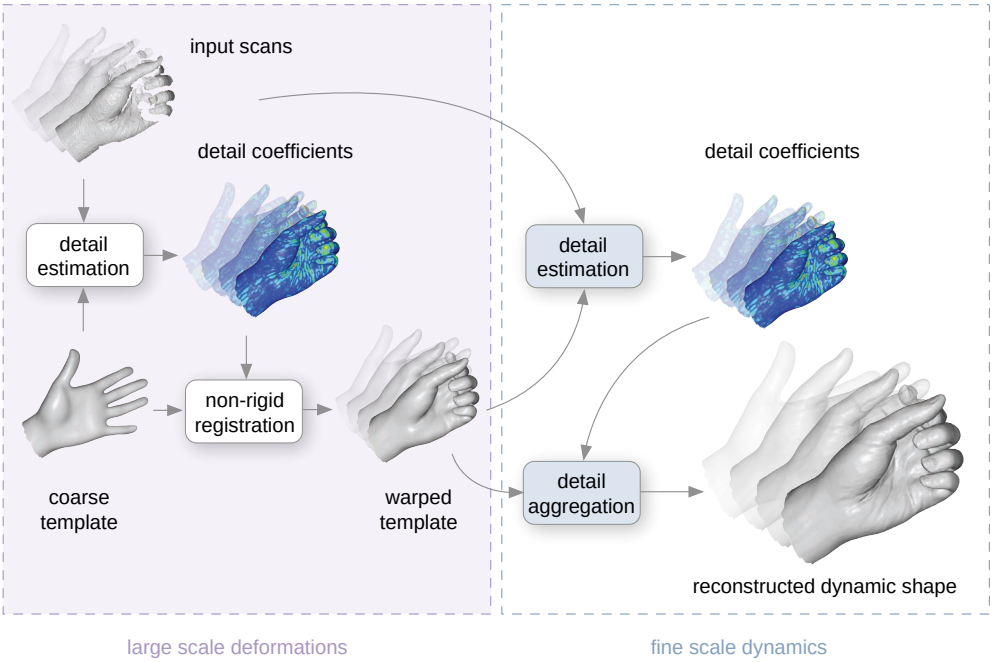


Figure 4.2: Bi-resolution geometry and motion reconstruction framework overview.

tially transient features to future scans. This significantly simplifies template construction since we do not require high geometric precision. To initialize the computations, we manually specify a rigid alignment of the template to the first frame of the scan sequence and apply one step of the pairwise non-rigid registration method described in Section 4.2.2.

Our *bi-resolution approach* (see Figure 4.2) reconstructs a complete and consistent surface for each frame. Template registration uses a non-linear reduced deformable model to recover the large-scale motion and align the template to each of the input scans (Section 4.2.2). The template-to-scan registration makes use of detail coefficients estimated in the previous frame to enable feature locking and improve the alignment accuracy. This requirement is crucial since establishing correspondence at the resolution of fine scale details is highly susceptible to ambiguous matches (see illustration). The final reconstruction is then obtained using a separate detail synthesis pass that runs once forward and once backward in time to aggregate and propagate detail into occluded regions (Section 4.2.5).

4.2.2 Template Registration

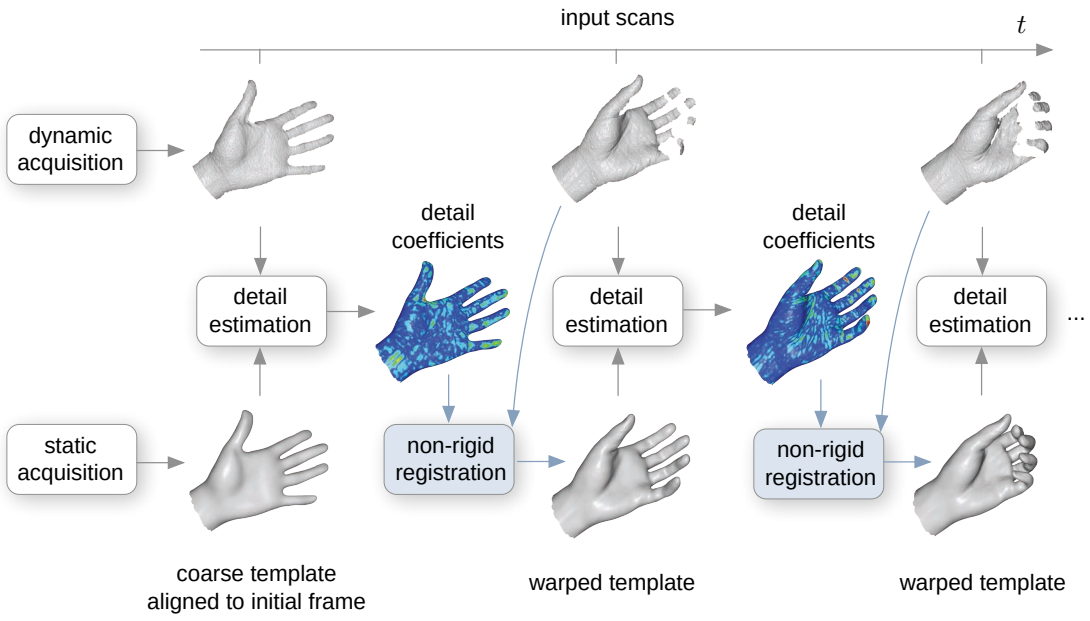


Figure 4.3: Template registration is used to reconstruct large scale deformations.

The registration stage captures the large-scale motion of the subject by fitting a coarse template shape to every frame of the scan sequence. Scans do not have to be a subset of the geometry described by the template, as in most previous methods (e.g. [ACP03]). Our method robustly handles part-in-part registration, as opposed to the simpler part-in-whole matching (see e.g. Figure 4.10). We assume minimal prior knowledge about the acquired motion and thus employ a general deformation model to capture a sufficiently large range of shape deformations. We extend the non-rigid registration framework presented in Section 3.5 to automatically adapt to the motion of the captured data. This allows recovering unknown complex material behavior and improves the robustness and efficiency of the registration.

Surface-Based Embedded Deformation. The embedded deformation algorithm presented in Section 3.2.4 computes a warping field using a deformation graph to discretize the underlying space. We now exploit the topological prior of the template and replace Euclidean distances in the original formulation by geodesic distances measured on the template mesh. This improvement avoids distortion artifacts that often occur

when geodesically distant parts of the object come into close contact (Figure 4.8).

Recall that each node $\mathbf{x}_i$ of the graph induces a deformation within a local influence region of radius r_i. Again, local deformation are represented as an affine transformation specified by a $\mathbf{A}_i \in \mathbb{R}^{3\times3}$ and $\mathbf{b}_i \in \mathbb{R}^3$ and graph nodes are connected by an edge whenever two nodes influence the same vertex of the mesh. Our surface-based formulation suggests that the vertex $\mathbf{v}_j$ should now be mapped to the following position:

$$\tilde{\mathbf{v}}_j = \sum_{\mathbf{x}_i} \overline{w}(\mathbf{v}_j, \mathbf{x}_i, r_i) \left[\mathbf{A}_i(\mathbf{v}_j - \mathbf{x}_i) + \mathbf{x}_i + \mathbf{b}_i\right], \tag{4.1}$$

where $\overline{w}(\mathbf{v}_j, \mathbf{x}_i, r_i)$ are the normalized weights $w(\mathbf{v}_j, \mathbf{x}_i, r_i) = \max(0, (1 - d^2(\mathbf{v}_j, \mathbf{x}_i)/r_i^2)^3)$ with $d(\mathbf{v}_j, \mathbf{x}_i)$ the distance between $\mathbf{v}_j$ and $\mathbf{x}_i$. We use a variant of the fast marching method to efficiently compute approximate geodesic distances [KS98].

During non-rigid registration we solve for the unknown transformations $(\mathbf{A}_i, \mathbf{b}_i)$. While local rigidity is maximized using the same energy E_{rigid}, we extend the smoothness term E_{smooth} using the geodesic distance weights to handle non-uniformly sampled graph nodes:

$$E_{\text{smooth}} = \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j} \overline{w}(\mathbf{x}_i, \mathbf{x}_j, r_i + r_j) \left\|\mathbf{A}_i(\mathbf{x}_j - \mathbf{x}_i) + \mathbf{x}_i + \mathbf{b}_i - (\mathbf{x}_j + \mathbf{b}_j)\right\|_2^2. \tag{4.2}$$

Minimizing these combined energies with the fitting term defined below yields affine transformations for each node, which in turn define a smooth deformation field on the template mesh. We solve this non-linear problem using the standard Gauss-Newton algorithm described in Section 3.2.4.

Robust Pairwise Registration. Since our input data is sufficiently coherent in time, we repeatedly perform pairwise non-rigid registration between the template and each input scan to determine the optimal deformation. Except for the reworked energies E_{smooth} and E_{fit}, the registration procedure is identical to our robust non-rigid ICP algorithm from Section 3.5.

In order to obtain an accurate fit, we augment the smooth template with detail information extracted from the previous frame. Template vertices $\mathbf{v}_i^j$ of frame j are displaced in the direction of the corresponding surface normal $\mathbf{n}_i^j$ yielding $\tilde{\mathbf{v}}_i^j = \mathbf{v}_i^j + d_i^{j-1}\mathbf{n}_i^j$, where d_i^{j-1} is the detail coefficient of frame $j-1$ (see Section 4.2.5). The correspondence energy combines the point-to-point and the point-to-plane metric to

avoid incorrect correspondences in large featureless regions:

$$E_{\text{fit}} = \sum_{(\mathbf{v}_i^j, \mathbf{c}_i^j) \in \mathcal{C}} \alpha_{\text{point}} \left\| \tilde{\mathbf{v}}_i^j - \mathbf{c}_i^j \right\|_2^2 + \alpha_{\text{plane}} \left(\mathbf{n}_{\mathbf{c}_i^j}^{\mathrm{T}} (\tilde{\mathbf{v}}_i^j - \mathbf{c}_i^j) \right)^2, \tag{4.3}$$

where $\mathbf{c}_i^j$ denotes the closest point on the input scan from $\tilde{\mathbf{v}}_i^j$ with corresponding surface normal $\mathbf{n}_{\mathbf{c}_i^j}$. We use $\alpha_{\text{point}} = 0.1$ and $\alpha_{\text{plane}} = 1$ in all our experiments. Again, correspondences are discarded if they are too far apart, have incompatible normal orientations, lie on the boundary of the partial input scans, or stem from back-facing or self-occluded vertices of the template.

Notice that detail information of the previous frame is only used to improve the accuracy of the registration by enabling geometric feature locking. The resulting continuous space deformation is applied to the template vertices without added detail. As discussed in Section 4.2.5 the final detail coefficients are obtained through a separate detail synthesis pass.

4.2.3 Dynamic Graph Refinement.

We replace the static, uniform sampling of the deformation graph with a spatially and temporally adaptive node distribution. While the idea of adaptive mesh deformation has been explored in previous work, for instance in the context of multi-resolution shape modeling from images [ZS00], we propose to adapt the degrees of freedom of the deformation model instead of the geometry itself in order to improve registration robustness and efficiency.

A hierarchical graph representation is pre-computed from a dense uniform sampling of graph nodes by successively merging nodes in a bottom-up fashion. The initial uniform node sampling corresponds to the highest resolution level $l = L_{\max}$ of the deformation graph that we restrict to roughly one tenth of the number of mesh vertices. We thus avoid over-fitting in regions of small-scale deformations, which are instead captured by our detail synthesis method (Section 4.2.5). We uniformly sub-sample the nodes of each level by repeatedly increasing their average sampling distance $r_{l-1} = 4\, r_l$ until l reaches $L_{\min}$. Each of the remaining nodes $\mathbf{x}_i^l$ from level $l \in L_{\min} \ldots L_{\max}$ form a cluster $\mathcal{C}_i^l$ which contains every node from the level below $\mathbf{x}_i^{l+1}$ that is not closer to any other cluster from l. The resulting cluster hierarchy is then used for adaptive refinement. We choose $L_{\min} = L_{\max}/2$ for all our experiments.

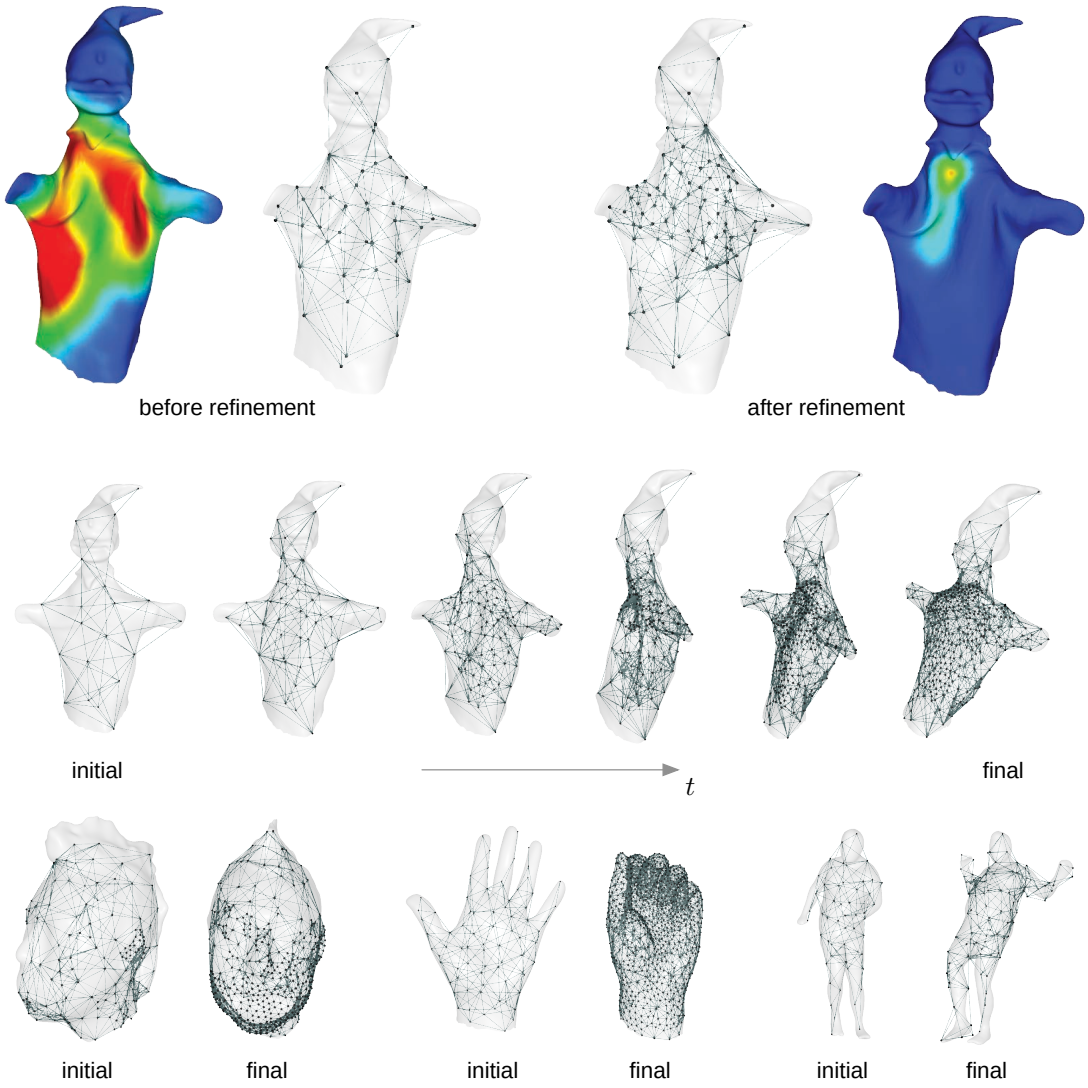


Figure 4.4: The deformation graph is dynamically refined during non-rigid registration to adapt to the deformation of the scanned object. Color-coded images indicate the regularization energy that determines where new nodes are added to the graph. The bottom row shows the initial and final deformation graphs for the hand and the sumo reconstruction.

Refinement Criterion. Registration starts with a coarse uniform graph at level L_{min} and dynamically adapts the graph resolution by inserting nodes in regions with high regularization residual (E_{smooth}), which indicates a strong discrepancy of neighboring node transformations (see Figure 4.4). In all our examples we set the threshold for refinement

to 10% of the highest regularization value. One step of refinement substitutes every node $\mathbf{x}_i^l$ that exhibits high regularization with all nodes contained in $\mathcal{C}_i^l$. To avoid unnecessary refinements for every new upcoming target frame, adaptive refinement is only performed if the global regularization term is still above a certain threshold, i.e. $E_{\text{smooth}} > 0.01$, for the maximum number of iteration $N_{\max} = 100$ of pairwise registration.

The dynamic refinement effectively learns an adaptive deformation model that is consistent with the motion of the scanned object. Additional nodes will be inserted automatically in regions of high deformation, while large rigid parts can be accurately deformed by a single graph node. In addition to being less susceptible to local minima, this leads to significant performance improvements (up to a factor of four in our examples) as compared to a uniform sampling with a high level of node redundancy. As illustrated in Figure 4.4, our adaptive model is suitable for a wide variety of dynamic objects, from articulated shapes to complex cloth folding.

4.2.4 Multi-Frame Stabilization.

The warped template $\mathcal{T}^{j-1}$ obtained after alignment to scan $j-1$ is the zero-energy state when aligning to scan j for each frame of the entire template warping process. For surface regions that are visible in the scan, dynamic details, such as cracks and fissures in paper-like materials can be accurately captured, since the method prevents the template from deforming back to its initial undeformed state. However, unobserved template parts are inherently prone to accumulation of misalignments, especially for lengthier scan sequences as illustrated in Figure 4.5. In contrast to our formulation, classical template fitting methods [ZSCS04, dAST$^+$08, VBMP08] warp the same initial template to each recorded frame and thus, use a deformation model that behaves globally elastic in time. For complex articulated subjects, such as human bodies, missing data in occluded regions would pull the template back to its original shape, which can be very different to the one of the current frame. Therefore, multi-view acquisition systems are usually used in combination with sparse and robust feature tracking [dAST$^+$08] and sometimes enhanced with manual intervention [VBMP08] to ensure reliable tracking.

In our dense acquisition setting, the surface coverage of the template by the input scans is spatially and temporally coherent over time. Thus, for non-occluded regions, the template shape from a closer time instance represents in general a more likely shape prior than the initial template $\mathcal{T}_{\text{init}}$. On the other hand, we make the assumption that

no better knowledge exists than $\mathcal{T}_{\text{init}}$ for template regions that are never observed or not seen for an extended period.

To address this issue we introduce a time-dependent combination of plastic and elastic deformation to accurately track exposed surface regions and reduce the accumulation of errors in less recently observed parts of the scanned object. After the pairwise registration of $\mathcal{T}^{j-1}$ to scan j as presented in Section 4.2.2, we obtain the plastically deformed template $\mathcal{T}^j$. A weight c_i^j for visibility confidence can then be defined for each vertex $\mathbf{v}_i^j \in \mathcal{T}^j$ as $c_i^j = \max\{0, (P + j_i^{\text{last}} - j)/P\}$ with j_i^{last} the last frame where $\mathbf{v}_i$ has been observed, and P a constant (we chose $P = 30$ in all our examples) that defines a temporal confidence range of visibility. All template vertices with $c_i^j = 1$ are visible in the current frame, while $c_i^j = 0$ represent those that are no longer considered confident. For the same frame, an elastically deformed template $\tilde{\mathcal{T}}^j$ with vertices $\tilde{\mathbf{v}}_i^j$ is created by warping $\mathcal{T}_{\text{init}}$ to the current frame j using the linearized thin-plate energy as described in [BS08]. Hard positional constraints are defined for all vertices with confidence $c_i^j = 1$. The resulting template $\bar{\mathcal{T}}^j$ with vertices $\bar{\mathbf{v}}_i^j$ is obtained by linearly blending $\mathcal{T}^j$ and $\tilde{\mathcal{T}}^j$ with the confidence weights for visibility yielding the vertices $\bar{\mathbf{v}}_i^j = c_i^j \mathbf{v}_i^j + (1 - c_i^j)\tilde{\mathbf{v}}_i^j$.

4.2.5 Detail Synthesis

Non-rigid registration aligns the template sequentially with all input scans. The resulting deformation fields induced by the graph capture the large-scale deformation but might miss small deformations that give rise to dynamic detail such as wrinkles and folds. To recover fine-scale detail at the spatial resolution of the scanner, we perform a separate detail synthesis stage that is composed of two steps: First, a per-vertex optimization from local correspondences is applied to estimate detail coefficients for each vertex of the template. These preliminary detail coefficients are the ones used for template alignment as detailed in Section 4.2.2. After the template has been registered to the entire scan sequence, we perform an additional pass that exploits the temporal coherence of the scan sequence to improve the reconstruction quality by propagating detail into occluded regions.

Linear Mesh Deformation. Since the deformed template is already well-aligned with the input scan, we employ an efficient linear mesh deformation algorithm similar to [ZSCS04] to estimate detail coefficients. For each vertex $\mathbf{v}_i$ in the template mesh, we trace an undirected ray in normal direction $\mathbf{n}_i$ and find the closest intersection point on the input scan. In case an intersection point $\mathbf{c}_i$ is found, a point-to-point correspondence

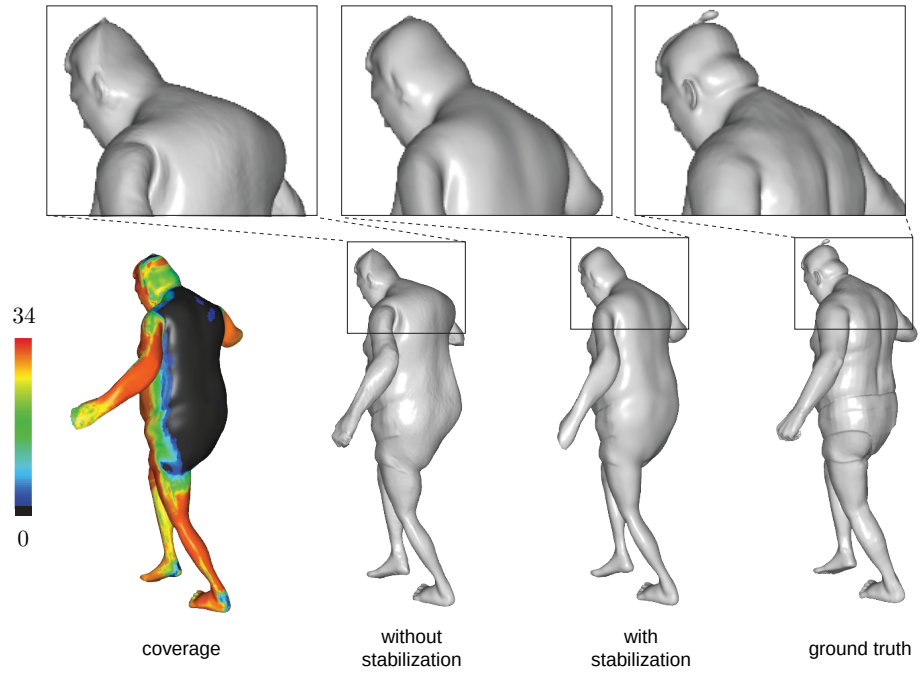


Figure 4.5: A hybrid plastic and elastic deformation model is used to stabilize the registration for multiple input frames as repeated pairwise alignment is susceptible to error accumulation. The accumulation of misalignments is shown on frame 30 of the sumo sequence.

constraint is created, if both points have the same normal orientation and are sufficiently close. Since the template has no high-frequency detail, its normal vector field is smooth, leading to spatially coherent correspondences. We compute the detail coefficients d_i by minimizing the energy resulting from the extracted correspondences subject to a regularization constraint

$$E_{\text{detail}} = \sum_{i \in \mathcal{V}} \|\mathbf{v}_i + d_i \mathbf{n}_i - \mathbf{c}_i\|_2^2 + \beta \sum_{(i,j) \in \mathcal{E}} |d_i - d_j|^2, \tag{4.4}$$

where $\mathcal{V}$ and $\mathcal{E}$ are index sets of mesh vertices and edges, respectively. The parameter β balances detail synthesis with smoothness and is set to $\beta = 0.5$ in all our experiments. The resulting system of equations is linear and sparse and can thus be solved efficiently. Notice that the rationale behind the regularization term of Equation 4.4 is essentially the same as the smooth displacement approach from Section 3.2. The only difference here is that the displacement is constrained to the surface normal.

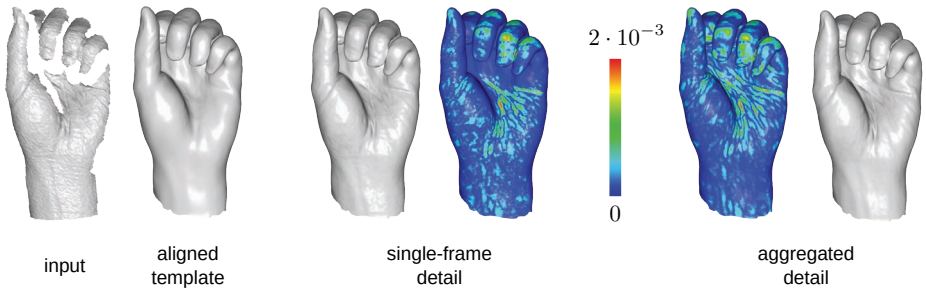


Figure 4.6: Detail synthesis. Reconstructing detail from the current frame leads to lack of detail in occluded regions. Aggregating detail over temporally adjacent frames propagates detail into hole regions and reduces noise. The color-coded images show the magnitude of the detail coefficients relative to the bounding box diagonal.

Aggregation. The linear mesh deformation method described above estimates detail coefficients independently for each frame in those regions of the object that are observed by a particular scan. To transfer detail to occluded regions we perform a separate processing pass that aggregates detail coefficients using a so-called *exponentially weighted moving average*. We use the formulation of Roberts [Rob59] and define this moving average as

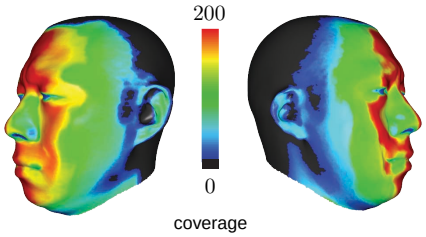


Figure 4.7: The color-coded images show the number of frames a certain region has been observed.

$$\overline{d}_i^j = (1-\gamma)\overline{d}_i^{j-1} + \gamma d_i^j \tag{4.5}$$

with γ set to 0.5 in all our examples. The influence of past detail coefficients decays quickly in this formulation, which is important, since transient or dynamic detail such as wrinkles and folds might not persist during deformation. Note that details in the template only disappear when they vanish in the input scans of succeeding frames. For instance, the details of a rigid object will persist and not fade toward zero coefficients since only observed coefficients are combined during detail synthesis. When processing scan j, we first update the vertices $\mathbf{v}_i^j \leftarrow \mathbf{v}_i^j + \overline{d}_i^{j-1}\mathbf{n}_i^j$ and perform the linear mesh deformation described in the previous section. This yields the new detail coefficients d_i^j that are then used to update the moving average $\overline{d}_i^j$, which will in turn be employed to process the subsequent scans.

The entire detail aggregation process is performed by running sequentially once forward and once backward through the scans while performing the linear mesh deformation and updating the moving averages. Going back and forth allows us to back-propagate persistent details seen at future instances to earlier scans (see Figure 4.6). As a final step, we apply a band-limiting bilateral filter [AW95] that operates in the time domain and detail range to further reduce temporal noise.

4.2.6 Results

We show a variety of acquired geometry and motion sequences processed with our system that exhibit substantially different dynamic behavior. Accurate reconstruction of these objects is challenging due to the high noise level in the scans, missing data caused by occlusions or specularity, unknown correspondences, and the large and complex motion and deformations of the acquired objects. The statistics for the results are shown in Table 4.1.

All templates were constructed by performing an online rigid registration technique similar to [RHHL02] on our acquired data, followed by a surface reconstruction technique based on algebraic point set surfaces described in [GG07]. Given the roughly aligned template mesh, our system runs completely automatically without any user intervention. Only few parameters (such as the weighting coefficients of the different energy terms) have to be chosen manually. For all examples, we use the same initial parameter settings. During optimization we automatically adapt the parameters using the approach detailed in Section 4.2.2.

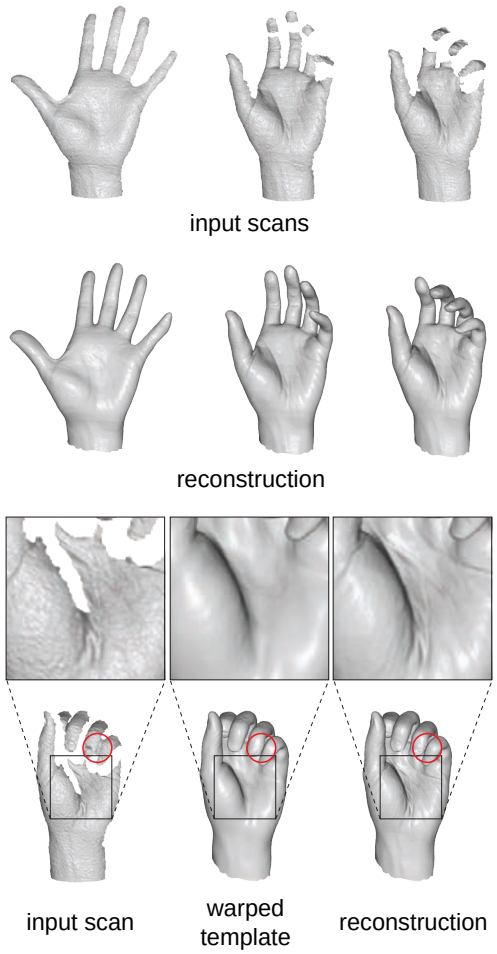


Figure 4.8: The zooms illustrate how high-frequency detail such as the skin folds is faithfully transferred to occluded regions. Even though the scan is connected at the fingertips, shape topology is correctly recovered (red circle).

Figure 4.9 shows the warped template and final reconstruction of the puppet. This example is particularly difficult due to the close proximity of multiple surface sheets when closing the puppet's hands. The reconstruction of a hand in Figure 4.8 demonstrates that our detail synthesis method is capable of capturing the intricate folds and wrinkles of human skin, even though the scans contain a large amount of measurement noise. Figure 4.10 illustrates how detail is propagated correctly into occluded regions, which leads to a plausible high-resolution reconstruction even for parts of the model that have not been observed in a particular scan. Figure 4.11 shows the reconstruction of a crumpling paper bag. Despite substantial holes caused by oversaturation in the reflections, the dynamics of the material as well as sharp geometric creases are faithfully captured.

4.2.7 Evaluation

Figure 4.12 illustrates the difference between tracking a high-resolution template versus our two-scale approach that separates global shape motion and dynamic detail reconstruction. For comparison we use the first frame of our two-scale reconstruction as the high-resolution template, which is then aligned with the input scan sequence using the registration method of Section 4.2.2. As can be seen in the zoom, dynamic detail created by the motion, in particular in the cloth, is not captured accurately. In contrast, our detail synthesis approach avoids the artifacts created by "baked-in" geometric detail

	Puppet	Head	Hand	Paper Bag	Sumo
# Scans	100	200	35	85	34
Min # Points per Scan	23k	53k	19k	82k	85k
Max # Points per Scan	37k	68k	25k	123k	86k
Input Data Size (Mb)	430	1,690	120	145	430
# Template Vertices	48k	64k	46k	64k	107k
Begin # Graph Nodes	20	152	77	37	52
End # Graph Nodes	100	458	1238	86	110
Output Data Size (Mb)	530	2,030	180	960	540
Registration Time	39	247	15	65	26
Detail Synthesis Time	26	92	8	36	23
Total Time	65	339	23	101	49

Table 4.1: Statistics for the results shown in this paper. All computations were performed on a 3.0 GHz Dual Quad-Core Intel Xeon machine with 8 GB RAM. Timings are measured in minutes and include I/O operations.

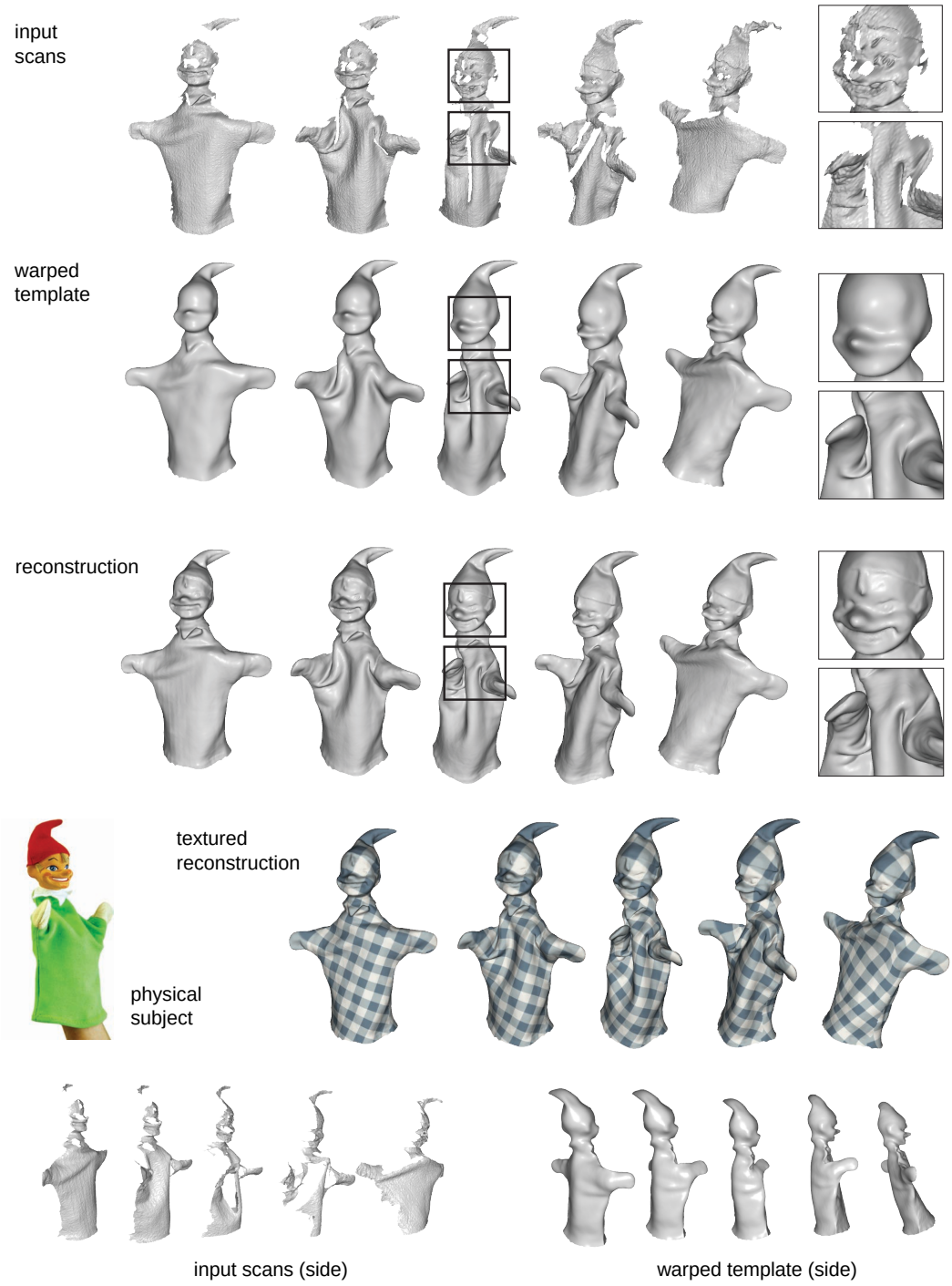


Figure 4.9: The global motion of the puppet's shape as well as fine-scale static and dynamic detail are captured accurately using the template registration and detail synthesis algorithm. The intricate folds of the cloth are handled robustly in the registration.

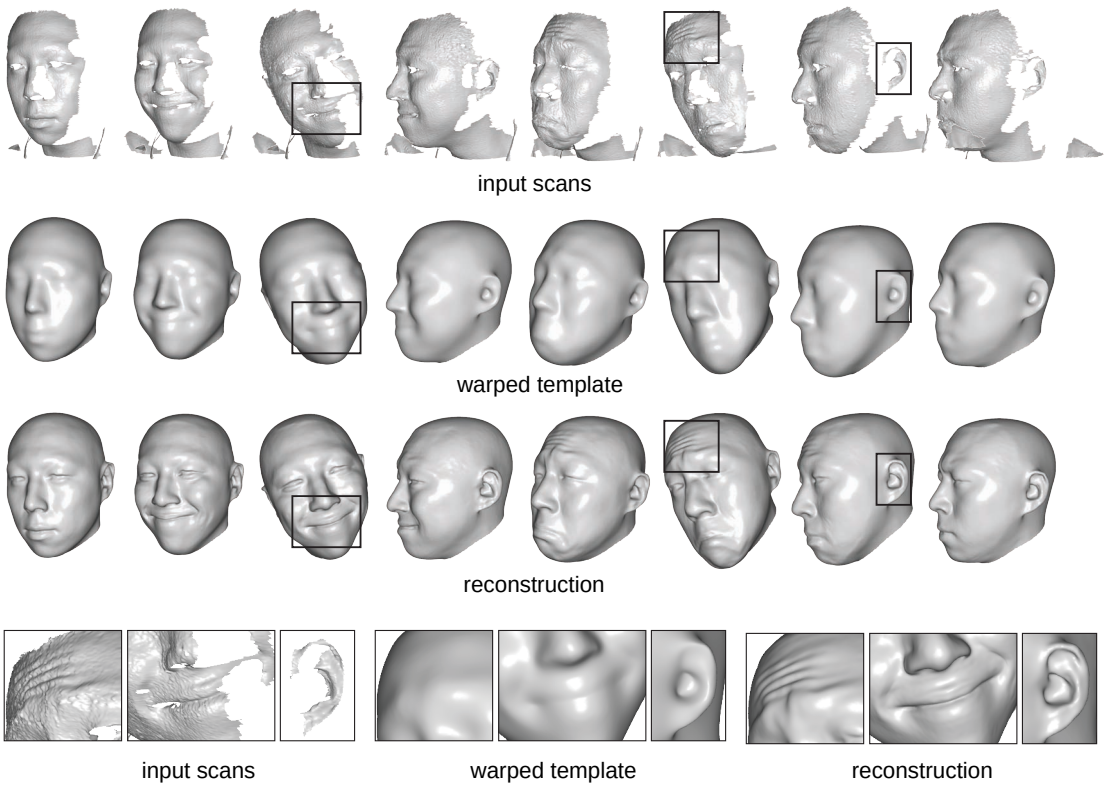


Figure 4.10: Our method faithfully recovers both the large-scale motion of the turning head, as well as the dynamic features created by the expression, such as wrinkles on the forehead or around the mouth. Intricate geometric details such as the ears are accurately captured, even though they are only observed in few frames.

and leads to a high-quality reconstruction of both static and dynamic detail. While a fairly large range of template smoothness can be tolerated, an overly coarse template can deteriorate the reconstruction as shown in Figure 4.13.

The necessity of using a template for robust reconstruction of complex deforming shape is illustrated in Figure 4.14. The method of [WAO+09] that avoids the use of a template cannot track the motion of the fingers accurately. In particular, the correspondence estimation fails when previously unseen parts of the shape, such as the back of the fingers, come into view. Figure 4.15 shows a comparison of our method to the dynamic registration approach of [SWG08] using the same template in both reconstructions.

We evaluate the robustness of the template tracking and detail synthesis method using the ground truth comparison shown in Figure 4.16. The scanning process has

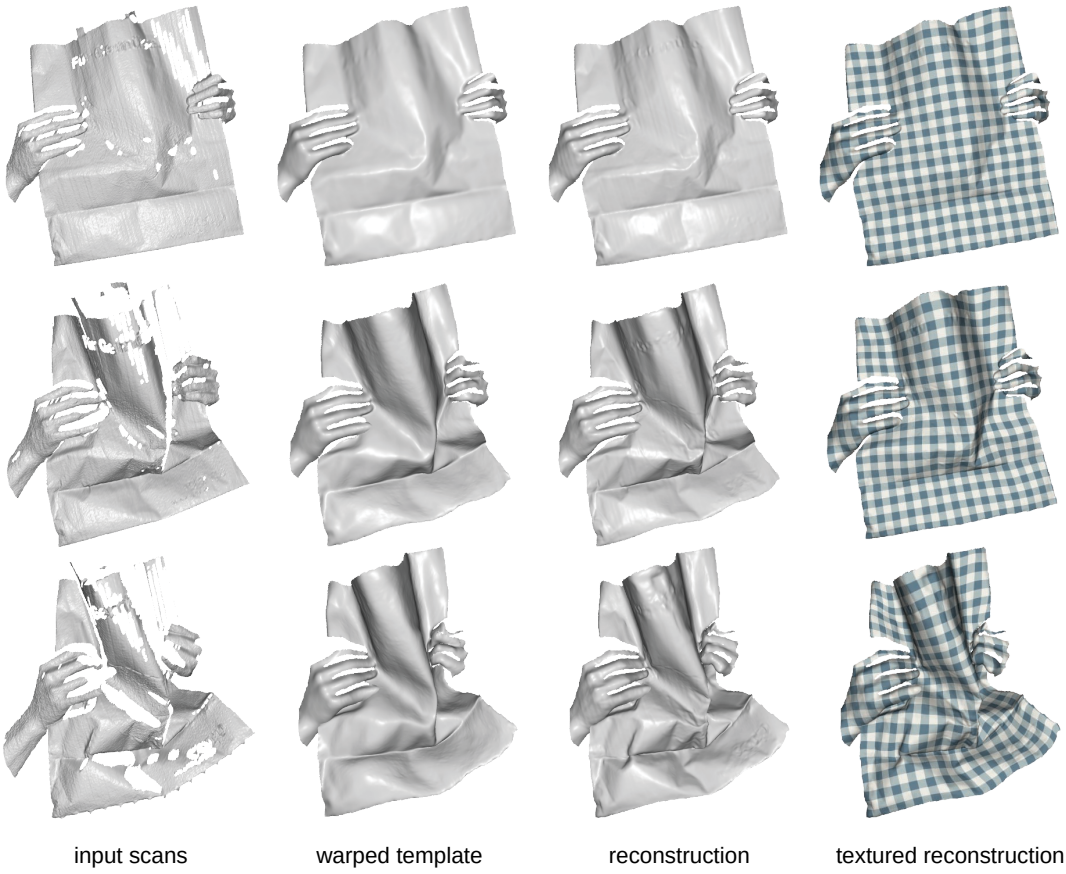


Figure 4.11: Sharp creases and intricate folds created by the complex, non-smooth deformation of a crumpling paper bag are captured accurately.

been simulated by creating a set of artificial depth maps from a fixed viewpoint. The ground-truth animation of the 3D model was obtained from dense motion capture data provided by [PH06]. In order to test the stability of the template tracking, we sampled the entire sequence at successively lower temporal resolution. The non-rigid registration robustly aligns the template with the scans for a temporally sub-sampled sequence consisting of only 34 frames. The large inter-frame motion, especially of the arms and legs, is tracked correctly, even though our correspondence computations do not make use of feature points, markers, or user assistance. Template tracking breaks down at 17 frames, where the fast motion of the arms cannot be recovered anymore (see Figure 4.17 (a)). Detail synthesis for the 34-frame sequence reliably recovers most of the fine-scale geometry correctly. Artifacts appear in the fingers and toes due to the coarse approximation of the template. In addition, drawbacks of the single-view acquisition become

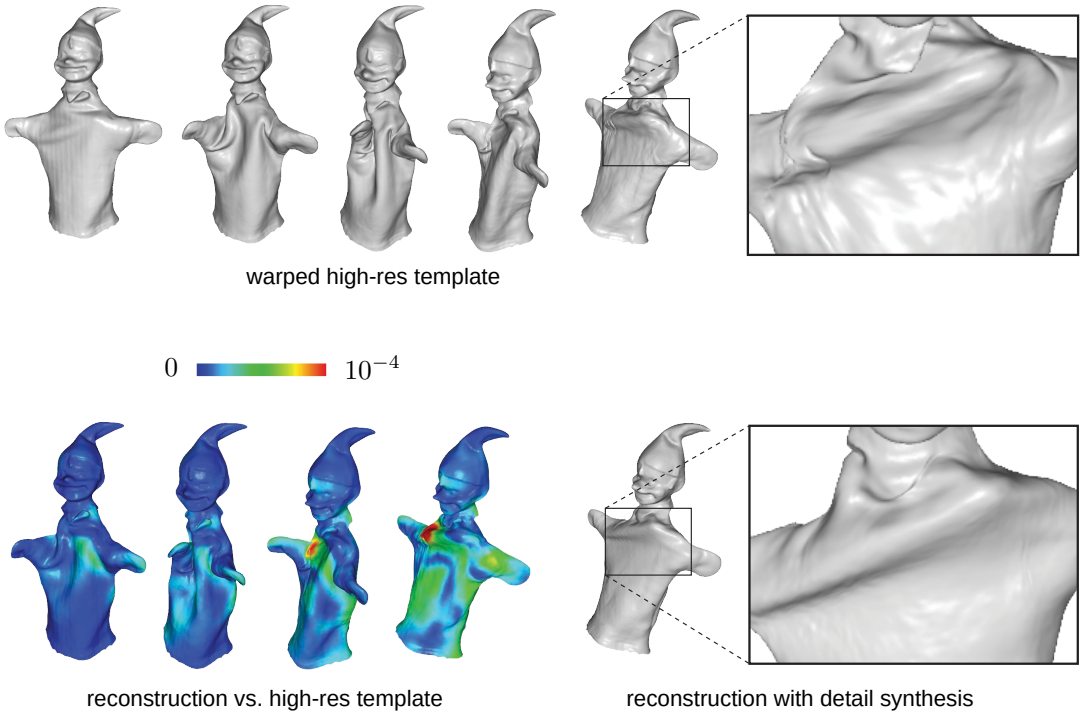


Figure 4.12: Warping a high-resolution template without detail synthesis leads to inferior results as compared to our two-scale reconstruction approach (cf. Figure 4.9). The color coding shows the distance between both results relative to the bounding box diagonal.

apparent in regions that are not observed by the scanner, such as the back of the sumo. Quantitatively, we measured the maximum of the average distance over all frames as 0.0012, the maximum of the maximum distance over all frames as 0.0283 as a fraction of the bounding box diagonal.

Limitations. We make few assumptions on the geometry and motion of the scanned objects. The correspondence estimation based on closest points, however, requires a sufficiently high acquisition frame-rate as otherwise, misalignments can occur, as shown in Figure 4.17 (a). Similarly, for parts of the shape that are out of view for an extended period of time, registration can fail if these regions have undergone deformations while not being observed by the scanner. In such a case, our system would require user interaction to re-initialize the registration. This is an inherent limitation of single-view systems where more than half of the object surface is occluded at any time instance. However, even some multi-view systems (e.g. [VBMP08]) permit user assistance to

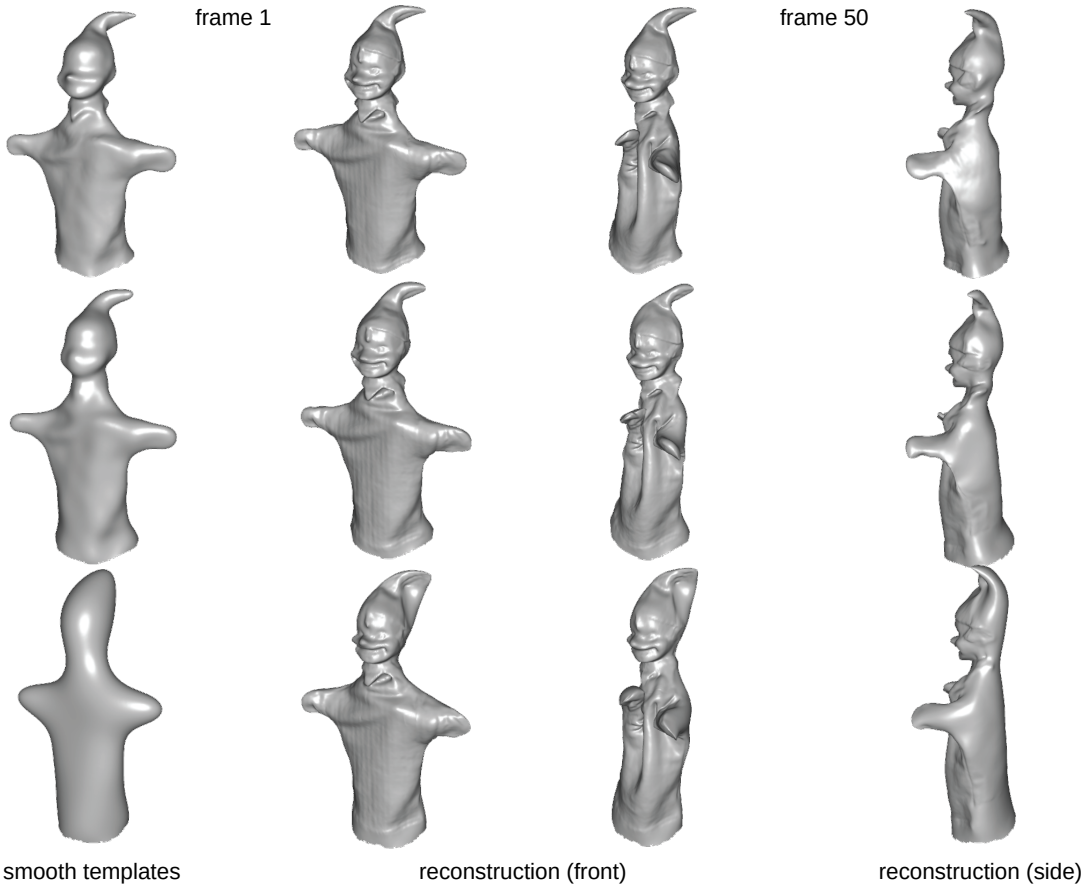


Figure 4.13: Evaluation of the reconstruction (frame 1 and 50) for three different initial templates. The upper row shows the original template. The coarser template in the second row is produced by surface reconstruction from points that are uniformly subsampled at half of the density of the original template. The last row illustrates the reconstruction using an even coarser template. This is obtained from only 25% of the initial point density.

adjust incorrect optimizations. Similar manual assistance might be required for longer sequences, where the scanner infrequently produces inferior data in certain frames. These frames need to be removed manually and the registration re-started with user assistance. While none of our sequences required such manual intervention, the acquisition of longer sequences was inhibited by this limitation of our scanning system.

Global aspects, such as the loop closure problem well-known in multi-view rigid alignment problems [Pul99] are currently not considered in our system. To address these limitations, more sophisticated feature tracking would be required in order to establish

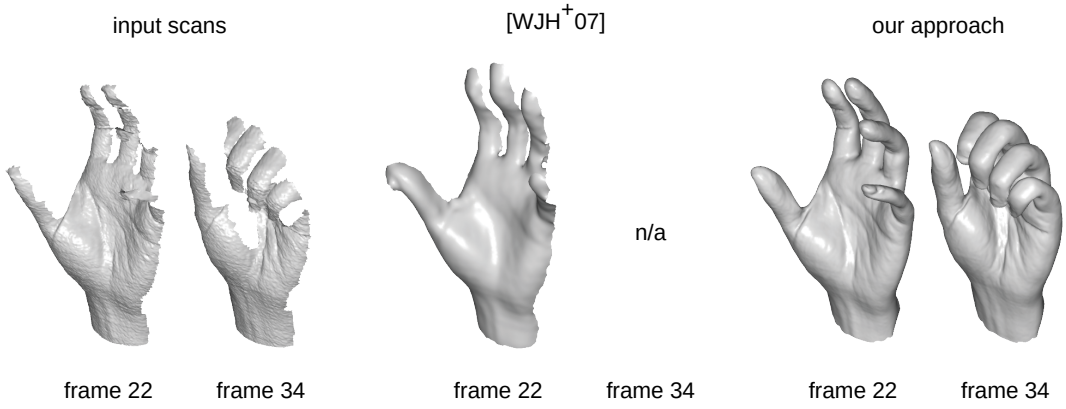


Figure 4.14: Reconstruction without a template is particularly challenging for single-view acquisition. The results in the center have been produced by the authors of [Wand et al. 2008].

reliable correspondences across larger spatial and temporal distances. We currently do not prevent global self-intersections of the reconstructed meshes. However, as shown in Figure 4.17 (b), our method robustly recovers, mainly due to the use of geodesic distances on the template mesh and the correspondence pruning strategy based on normal consistency and visibility. Avoiding self-intersections entirely would require an additional self-collision handling step in the shape deformation optimization algorithm, which would add a significant overhead to the overall reconstruction pipeline. Our method does not discover topological errors in the template, as shown in Figure 4.17 (c). In the template reconstruction the pinky has been erroneously connected to the paper bag, which leads to artifacts in the final frames of the sequences, where the finger is lifted off the bag.

4.2.8 Discussion.

We have presented a robust algorithm for geometry and motion reconstruction of dynamic shapes. One of the main benefits of this method is simplicity. Our scanning system requires no specialized hardware or complex calibration or synchronization, and can be readily deployed in different acquisition scenarios. We do not require silhouette or feature extraction, manual correction of correspondences, or the explicit construction of a shape skeleton. The framework demonstrates that even for single-view acquisition, high-quality results can be obtained for a variety of scanned objects, with a realistic reconstruction of shape dynamics and fine-scale features. Key to the success of our algorithm is the robust template tracking based on an adaptive deformation model.

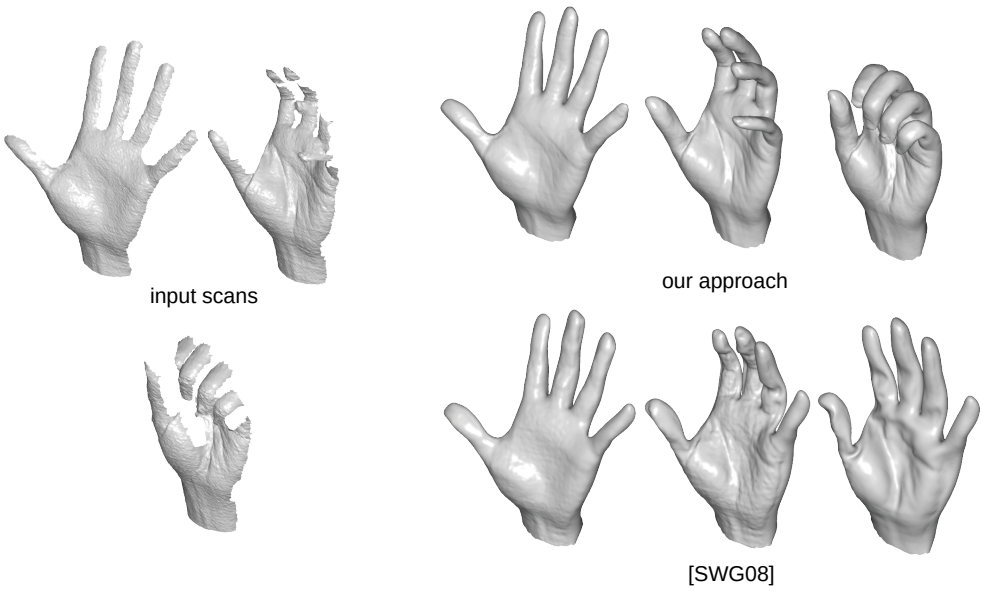


Figure 4.15: Comparison of two template based reconstruction methods. The results in the bottom right have been produced by the authors of [Süssmuth et al. 2008].

Our novel detail synthesis method exploits the accurate registration to aggregate and propagate geometric detail into occluded regions.

As future work, aforementioned limitations need to be resolved and global self-collision handling should be incorporated. Additionally, the proposed registration algorithm can be potentially used to acquire and learn material behavior (such as the crumpling of paper or folding of skin). Such information would be useful to improve the realism of physically-based simulation algorithms.

4.3 Temporally Coherent Shape Completion

Many common geometries cannot be modeled by a single mesh (e.g. gliding cloth, exposing new body parts, etc.). As a consequence, we need a dynamic shape reconstruction method that does not rely on templates which implicitly define a watertight surface.

We consider the problem of obtaining temporally coherent *watertight* 3D meshes from high-resolution scan sequences of a dynamic performance recorded from multiple views [LLV+10]. We assume that the input scans have reasonable coverage and that most noise and outliers are suppressed, either by using an improved scanning technology or by effectively post-processing the data.

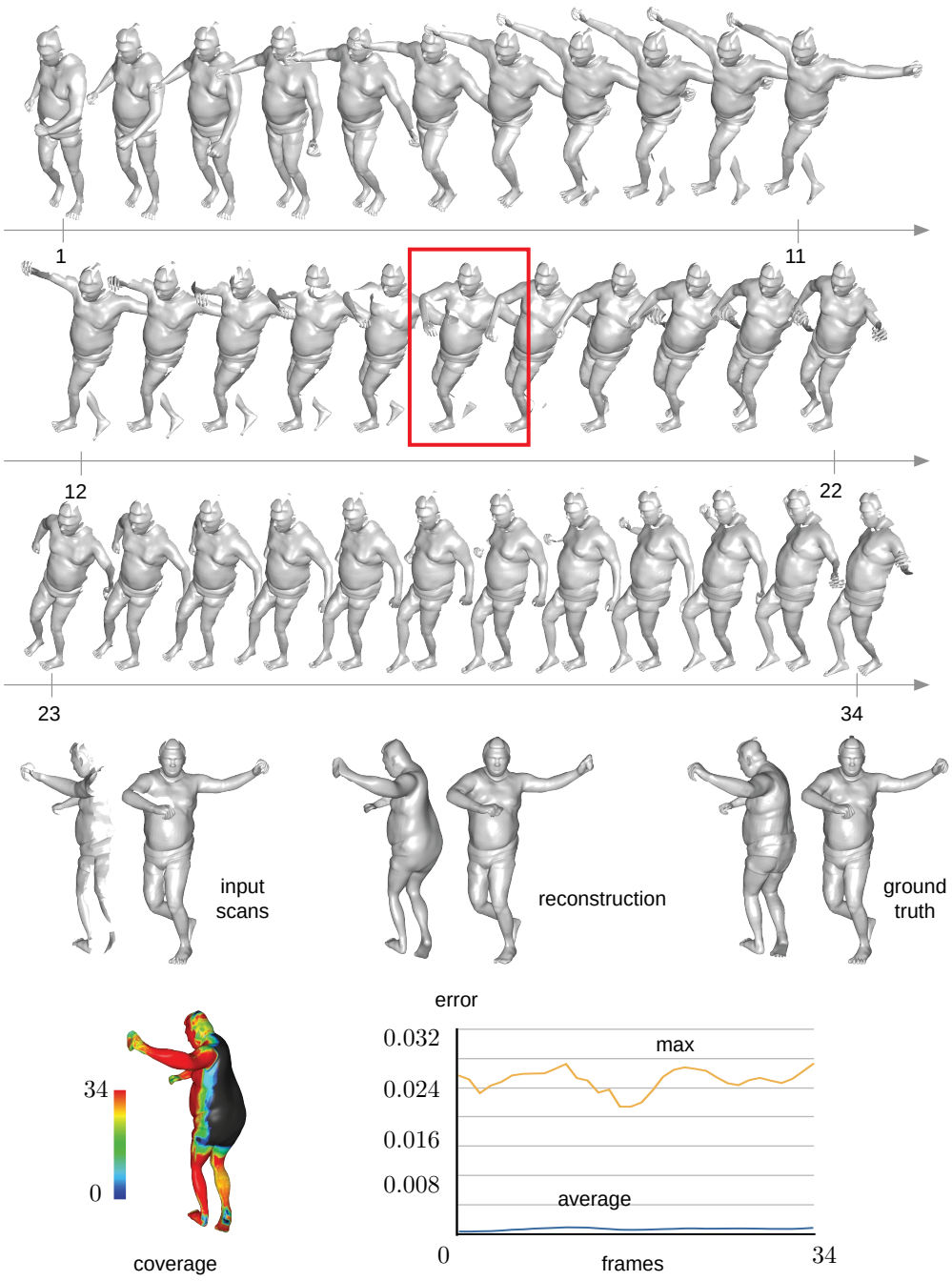


Figure 4.16: Ground truth comparison for a synthetic full-body example with fast motion. The top row shows every frame of the input sequence. The color-coded image indicates the number of frames in which a certain part of the shape is covered by the scans. The graph shows the maximum and average error distance between the ground truth and the reconstruction for each frame.

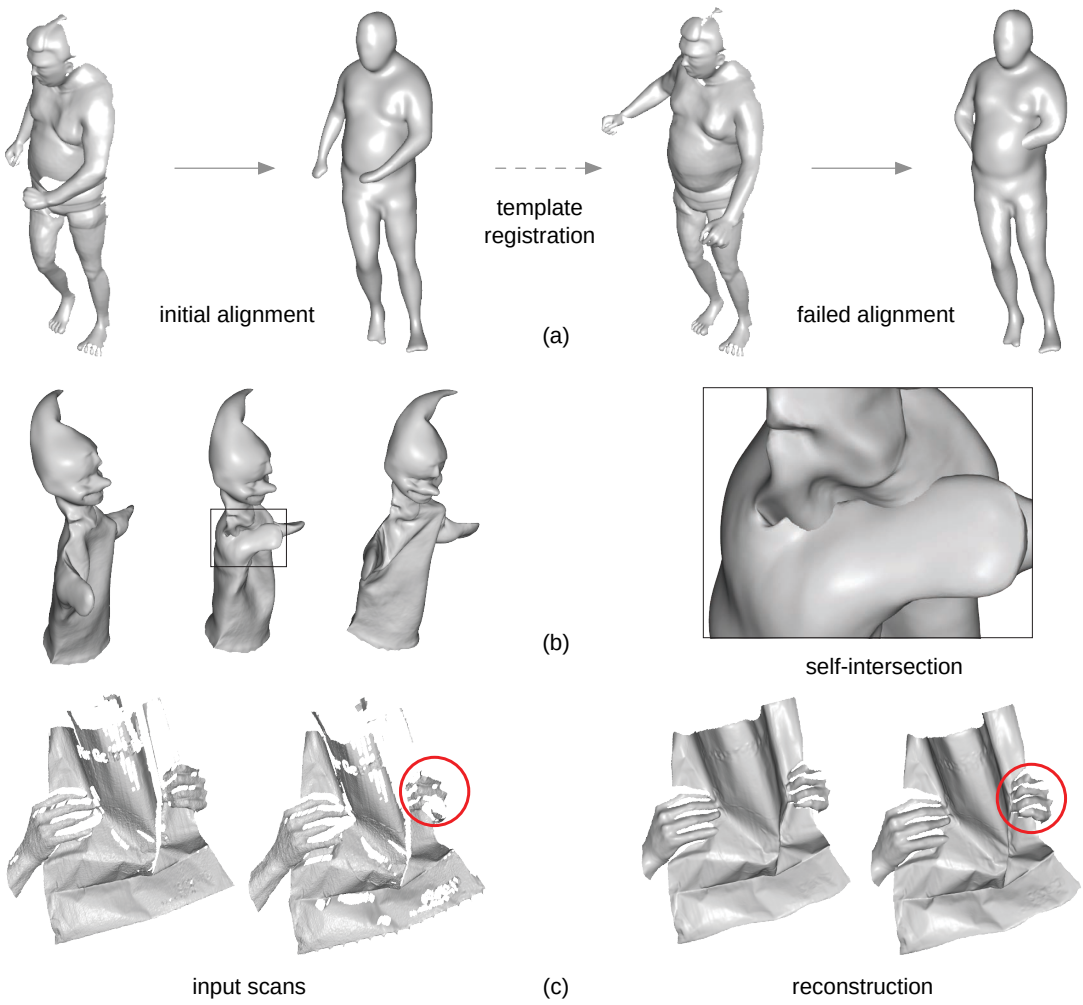


Figure 4.17: Limitations: (a) registration can fail if the frame-rate is too low relative to the motion of the scanned object; (b) self-intersections are not prevented during template alignment; (c) wrong template topology leads to artifacts when the finger is lifted off the paper bag.

In human performance capture, large holes are typically observed between legs, regions occluded by arms, and those parts exhibiting significant grazing angles to the cameras. While a deforming shape can expose newly observed regions over time, these holes are usually so large that full coverage is only possible after longer recording. Most current techniques for temporally consistent shape completion assume that the dynamic subject is represented by a single deformable surface (*template*). The template model is usually obtained by a separate rigid reconstruction step (e.g., [LAGP09, dAST+08,

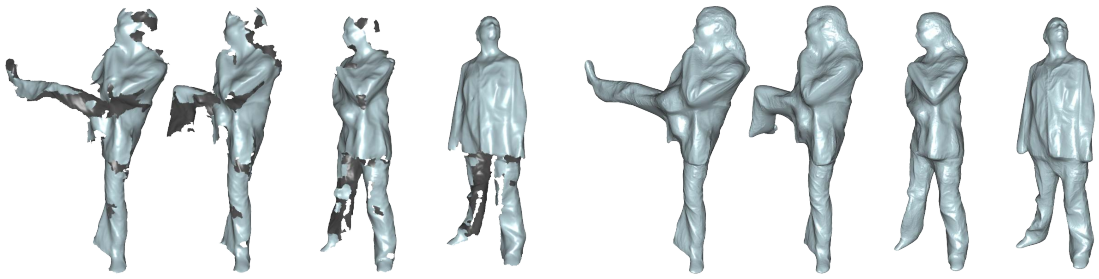


Figure 4.18: **Left:** Real-time 3D acquired dynamic performance geometry typically exhibits holes that are often temporally persistent. **Right:** Hole-filled, temporally coherent and detailed sequence of watertight surfaces reconstructed using our method.

VBMP08]) or by globally aggregating all surface samples through time (e.g., [WAO+09, MFO+07, SWG08]). Both approaches rely on establishing full inter-frame correspondences of surface points across entire recordings for the template. However, we do not wish to restrict the degree of deformation or fix the topology. Deformations that involve *topology changes* or *interactions between multiple disconnected* components cannot be accurately modeled with a single template (e.g., gliding cloth, exposing new body parts, etc.). Thus the correct shape is unlikely to be recovered by simply propagating geometry across long sequences without knowledge of full inter-frame correspondences in occluded regions. Moreover, error accumulation is likely to occur when correspondences need to be repeatedly determined between pairs of input scans. Consequently, none of these techniques can guarantee drift-free reconstruction for complex deformations and largely incomplete input data.

Our proposed method does not require globally consistent correspondences or a template model. The key insight is that only accurate pairwise correspondences are needed for temporally consistent shape completion, as the relevance of surface information decreases with time. For example, a fold on a dress observed in one frame is likely to disappear or completely change its shape at a later time. To establish dense pairwise correspondences, we employ a novel two-stage registration algorithm that (1) performs our coarse non-rigid registration algorithm [LAGP09] equipped with deformation graph prediction and sparse texture-based constraints for higher accuracy and robustness, and (2) refines this coarse correspondence computation using an improved version of a fine-scale alignment algorithm [BR07]. Because surface correspondences only reside within a subset of two consecutive pairs of incomplete scans, more coverage leads to improved

alignment quality. We maximize coverage by accumulating newly observed surfaces using an interleaved registration/merging method in a forward-and-backward fashion.

Given the original scanned surfaces and their pairwise correspondences, our shape completion approach starts by filling the holes in each frame independently using the visual hull as a topological prior. We further optimize vertex positions to satisfy spatial smoothness across hole boundaries [Lie03]. The use of the visual hull as a topological prior helps to resolve ambiguous hole filling strategies (e.g., when the arm is close to the body). To minimize temporal flicker, we unwarp all watertight shapes within a time window into the current frame using the precomputed dense pairwise correspondences. The aggregation of nearby frames forms a temporally coherent shape which we reconstruct by weighted integration of surface samples [KBH06]. We design our weighting scheme to act similarly to a temporal bilateral filter, but instead of preserving motion discontinuities, we maximize the aggregation of non-occluded regions. However, fine-scale geometrical details tend to be blurred out by the integration of the unwarped shapes. To resynthesize these fine-scale details, high-frequency details from partial input scans or user-provided normal maps are reapplied to the integrated surface using the method of Nehab and colleagues [NRDR05].

Our framework is designed to handle input data with large occlusions, topological changes, and complex deformations. Because an interleaved registration/merging scheme is employed, only a few adjacent meshes are needed simultaneously, leading to modest memory requirements. This also makes our method well suited for very high-resolution input data. We illustrate our method on the meshes of Vlasic et al. [VPB+09] and compare our results with recent work on space-time reconstruction. While the absence of globally corresponded meshes precludes certain applications, our method is the first to enable free-viewpoint video of watertight and temporally-coherent high-resolution dynamic geometries (c.f. Figure 4.22 and 4.23).

Overview. The proposed shape completion method employs a three-step algorithm to synthesize temporally coherent watertight surfaces from scanned sequences of non-rigidly deforming shapes:

1. We start by filling the holes in each frame separately, employing the visual hull as a topological prior. Furthermore, to promote temporal smoothness and avoid unnatural discontinuities across hole boundaries, we optimize the hole filled vertex

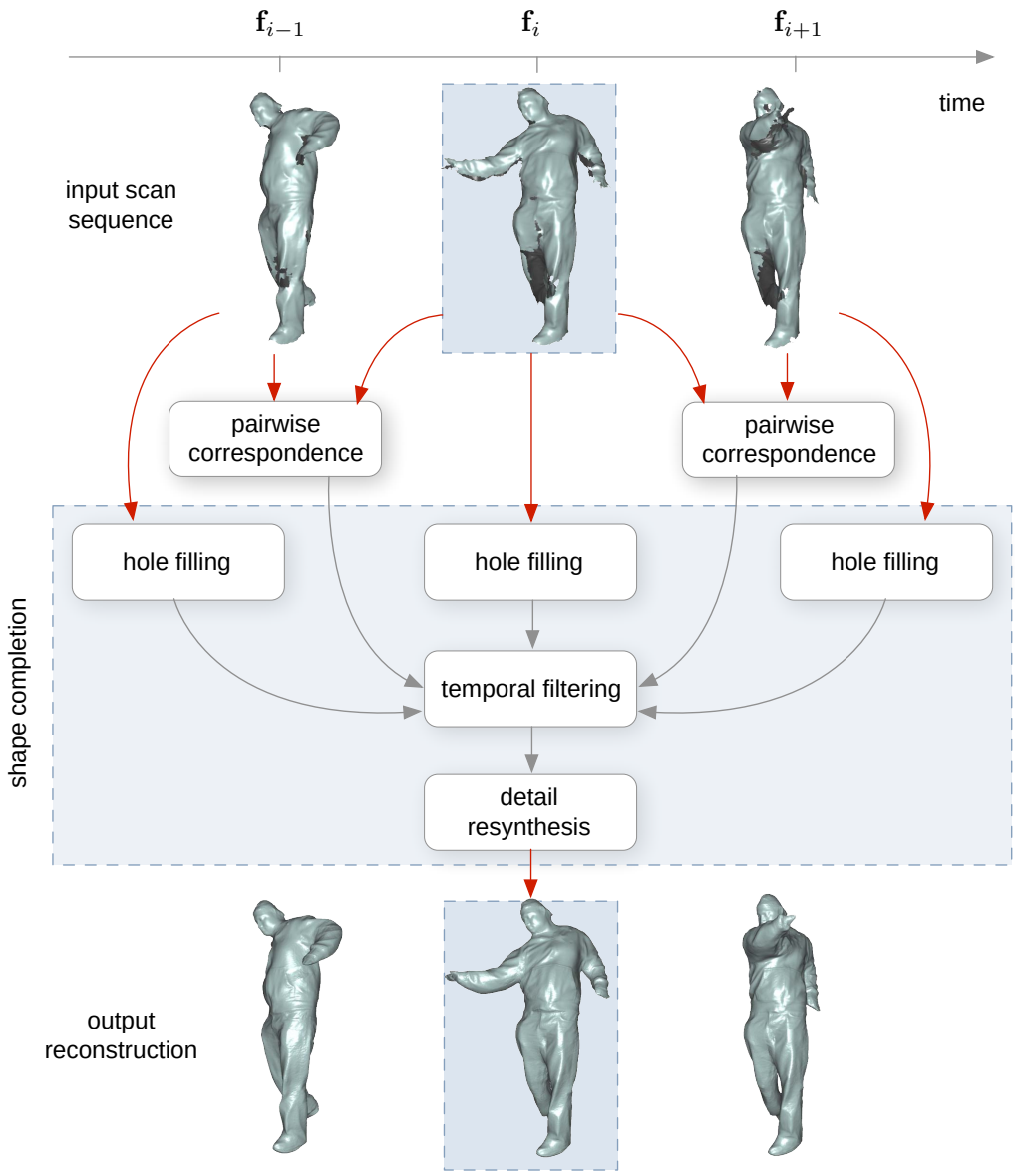


Figure 4.19: Shape completion pipeline.

positions by minimizing a bending energy fairness functional.

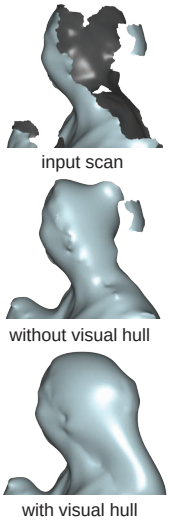
2. We proceed with a weighted surface integration scheme that reconstructs a temporally coherent watertight surface from adjacent time frames, thus minimizing temporal artifacts. We warp the resulting shapes using pairwise correspondences computed in a preprocessing step (detailed in Section 4.3.4).

3. Finally, we resynthesize the details lost during warping and integration onto the final temporally coherent watertight mesh.

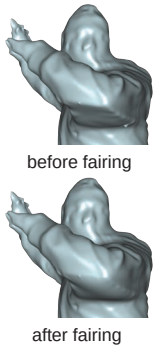
The complete three-step process is schematically depicted in Figure 4.19.

4.3.1 Single Frame Hole Filling

As illustrated in Figure 4.18, scanning human performances typically results in large holes which persist in close proximity over many frames. Filling these holes can become ambiguous when two separate incomplete surfaces get close.



Visual Hull Prior. As suggested in [VPB+09], the visual hull provides a robust estimate for obtaining watertight shapes. We therefore initialize our hole filling by combining the vertices of the original partial scans with those of the visual hull. We set a weight $w = 1$ for each surface sample located on the scans and $w = \epsilon$ for visual hull samples. A hole-free mesh is then obtained by Poisson surface reconstruction [KBH06] using the weighted oriented surface samples. As each frame is being completed independently, considerable flickering artifacts are likely to occur in hole-filled regions for dynamic input geometry.



Surface Fairing. To enforce smooth transitions with the surroundings of a hole-filled mesh region, we solve for new vertex positions by minimizing a fairness functional constrained by the hole boundaries, similar to [Lie03]. In particular, we minimize the linearized bending energy of the patched mesh's non-boundary vertices using the standard cotangent bi-Laplacian [BS08]. Since only limited views are provided for computing the visual hull, optimizing surface fairness in hole regions yields spatially smooth and more plausible reconstructions for curved surfaces such as folds in a garment. While spatial smoothness for hole regions can be directly obtained by carefully estimating sample weights at hole boundaries during Poisson reconstruction, this extra fairing step avoids the need for additional feathering parameters. While the fairing significantly reduces strong discontinuities across boundaries, flickering still persists as each frame is processed independently.

4.3.2 Temporal Filtering

Temporal flicker is present both in the original data (due to independent per-frame reconstruction) and our hole filled surfaces (due to visual-hull-based optimization). We address this with a temporal filter that combines each frame with its neighbors, and only requires knowledge of pairwise correspondences between neighboring frames in the original sequence.. The correspondences are computed in a preprocessing step (detailed in Section 4.3.4).

Our temporal filtering process starts with the incomplete reconstructed mesh (original data) and the hole filled regions at each frame. We warp the hole filled regions into the neighboring frames using a mesh deformation based on the pairwise correspondences and Laplacian coordinates [Ale03], where the reconstructed meshes define the constraints. At this point, we have the reconstructed meshes from the current and the neighboring frames, as well as the hole filled regions from those three frames, all aligned to a common pose. We combine them all using Poisson surface reconstruction [KBH06] with the following weights: 100 for the reconstructed mesh of the current frame, 10 for the reconstructed mesh of the neighboring frames (deformed to the current frame), 2 for the hole-filled regions of the current frame, and 1 for the hole-filled regions of the neighboring frames (also deformed to the current frame). This imposes a mild temporal filter on the reconstructed surfaces, and a strong filter on the hole-filled regions. This step reduces the temporal flicker, and propagates some of the reconstructed surface detail from the neighboring frames onto the current frame (this stems from the neighboring reconstructed mesh weight being larger than any hole-filled region weight).

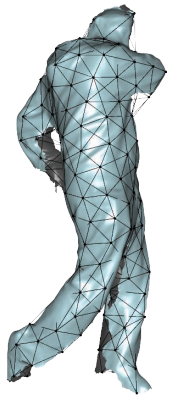
4.3.3 Detail Resynthesis

While the weighted temporal filtering approach reduces flicker between the hole filled meshes, it also tends to remove some fine geometric details mainly due to the Poisson surface reconstruction step. Since our input data is only affected by very little noise, the stability of the high frequency details in non-boundary regions allows us to reintroduce details and compensate for this loss. We employ the method of Nehab and coworkers [NRDR05] to resynthesize high frequency detail, which can either come directly from the original input scans, or alternatively from measured normal maps. In our case, stable normal information is available in the form of normal maps [VPB+09].

4.3.4 Pairwise Correspondences

A crucial component in the proposed shape completion algorithm are the accurate pairwise correspondences between consecutive frames of a dynamic performance. Several short- and long-range correspondence algorithms exist (e.g., [ZST+10, WAO+09, LAGP09, SAL+08]). However, we found that none of these methods gave the necessary accuracy to obtain high quality shape completions (see Section 4.3.5 for a qualitative comparison). In this work, we develop a novel two-scale approach. We start by computing coarse correspondences that are globally coherent and capture large-scale deformations (Section 4.3.4). Next, we refine these coarse correspondences to accurately align the fine-scale geometric details (Section 4.3.4).

Registration Based on Deformation Graphs


deformation graph

To compute the pairwise coarse-scale registration, we extend our robust non-rigid ICP algorithm from Section 3.5 with (1) a prediction-based initialization and (2) sparse positional constraints computed from input video data. The proposed improvements increase robustness to large deformations and minimize tangential drift, improving accuracy over short time windows (as validated in Section 4.3.5).

As detailed in Section 3.5, our subspace deformation technique uses a graph with nodes that are uniformly sampled on the scan surface to warp the scan mesh vertices via linear blend skinning. The optimization solves for the affine transformations on the graph nodes and is regularized with the energy terms E_{rigid} and E_{smooth}. By iteratively computing the closest points, the method minimizes the point-to-point and point-to-plane distances specified by E_{fit}.

In addition to the original energy terms, we introduce a term E_{tex} for sparse 3D positional constraints obtained from texture correspondences. At each deformation step we solve a non-linear optimization with the objective function:

$$E_{\text{tot}} = \alpha_{\text{fit}} E_{\text{fit}} + \alpha_{\text{tex}} E_{\text{tex}} + \alpha_{\text{rigid}} E_{\text{rigid}} + \alpha_{\text{smooth}} E_{\text{smooth}}, \tag{4.6}$$

where $\alpha_{\text{fit}} = 1$ and $\alpha_{\text{tex}} = 100$. As before, we ensure robustness against sub-optimal local minima by starting the registration with a high regularization ($\alpha_{\text{rigid}} = 100$ and $\alpha_{\text{smooth}} = 10$) and successively halving the weights whenever the deformation step converges. We stop the optimization when $\alpha_{\text{smooth}} = 0.01$.

Graph Prediction. While effective for a large range of deformations, the above registration technique is likely to converge to an incorrect local minimum when there is significant motion between consecutive frames (e.g., a fast kick) or in regions with few geometric features. Convergence to the *correct* deformation can be promoted by employing a prediction that provides an initial deformation close to the desired deformation. The deformation graph in frame $\mathbf{f}+2$ is predicted by linear extrapolation from frames $\mathbf{f}$ and $\mathbf{f}+1$. In short, for each edge of the deformation graph, we extract the smallest $3D$ transformation that deforms that edge from frame $\mathbf{f}$ to $\mathbf{f}+1$. We then transform each vertex of the deformation graph in frame $\mathbf{f}+1$ by the average of all the transformations corresponding to its incident edges.

Sparse Texture-Based Constraints. So far, E_{fit} is used to bring the source scan closer to the target. However, this does not preclude tangential drifts (even with the above prediction). For regions with very little detail, using only geometric constraints can yield suboptimal alignment (e.g., sliding versus stretching). Thus, we add texture constraints (obtained from image recordings that are projected onto the mesh) and use them as sparse positional constraints for the optimization.

To determine these sparse features we compute 2D feature descriptors from the video recordings of 8 different camera positions between consecutive frames. In our implementation we used SURF feature descriptors [BTVG08], though many other 2D descriptor can be employed. In the case of SURF, features tend to be concentrated at the silhouette of the subject, and do not represent true surface features. Therefore, only those features that lie away from some preset distance (8 pixels) of the silhouette are considered.

Next, we match each detected feature point to the best corresponding feature point in the subsequent frame. To speed-up detection and minimize false positive matches, we restrict the search space using an optical flow based prediction [BBPW04] and search for the best matching SURF descriptor in a small neighborhood around this predicted feature point location. We discard the pairwise match if the error on the feature descriptors exceeds a certain threshold. We search in a radius of $\min(10, d)$ around the predicted point, with d being the distance of the predicted displacement. We reject matches with a descriptor error above 0.2. To improve robustness, we only consider correspondences that can be reliably tracked for at least 3 consecutive frames.

Finally, we project every tracked 2D feature back on the original geometries to

obtain 3D positional constraints. Section 4.3.5 validates that the found texture-based correspondences (up to 1000 per frame) greatly improve the registration quality.

Fine-scale Alignment

After the coarse non-rigid alignment, we perform fine-scale registration using a non-rigid locally weighted ICP algorithm based on [BR07]. This improves the alignment of small geometric details. Our algorithm improves on [BR07] by taking the following two observations into account:

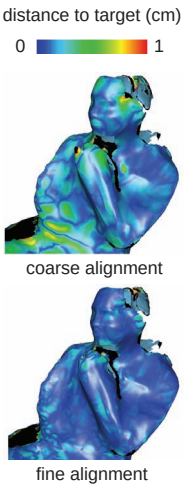
1. The main goal is to locally improve the alignment, hence the weight distribution function should have local support. Otherwise, far-away points can bias the local alignment. We use a Compactly Supported Radial Basis Function (CSRBF).

2. Gelfand and coworkers [GRIL03] showed that the stability of the ICP matching algorithm depends on the local geometry. If the matched geometry does not contain enough surface detail, drift may occur. Ideally, the size of the matched geometry should adapt to the local feature size.

These observations define the following three-step algorithm:

Sampling. We start by sampling an optimal set of feature points on the deformed mesh according to the alignment error which is defined by the distance between source mesh and nearest point on the target mesh after non-rigid ICP. This step ensures that regions with median alignment error gain average sampling weights, while the influence of large outliers is decreased.

Matching. Next, we find correspondences using a local ICP algorithm based on [BR07]. However, we differ in that we employ a CSRBF for point-selection near a feature point and iteratively select the best radius of CSRBF according to the local geometric stability. Specifically, we use a quadratically decreasing CSRBF $f(x) = max\{1 - (x/r)^2, 0\}$, where r is an adaptively selected support radius. To optimally select the support radius, we iteratively apply ICP, reducing the radius at each step as long as the alignment error decreases and the stability of the sampled points is above 0.02, a threshold that empirically prevents drifting. The iterative scheme proves to be robust since the relatively large initial CSRBF radii



avoid suboptimal local matching. We further improve robustness by rejecting correspondences whose nearest vertices are on the mesh boundaries.

Warping. We employ the RBF deformation model proposed by [KSSH02] to avoid known numerical instabilities of thin-plate splines as described in [SS91]. The resulting linear system is sparse, due to the local support of the CSRBF, and can thus be solved efficiently.

Shape Accumulation

The above two-scale registration algorithm is capable of producing accurate correspondences between mutually visible surface regions. However, in order to produce temporally consistent watertight surfaces, we also need accurate correspondences in hole regions. In order to ensure maximum accuracy in these regions, we propose an interleaved registration/merging shape accumulation approach. Pairwise correspondences between consecutive frames $\mathbf{f}'_i$ (merged) and $\mathbf{f}_{i+1}$ (original) are used to warp $\mathbf{f}'_i$ and merge it with $\mathbf{f}_{i+1}$, yielding an accumulated shape $\mathbf{f}'_{i+1}$. We repeat this process for every frame starting from the first frame going to the last frame, and vice versa.

We employ an interleaved method. First, as the scanned subject moves and deforms over time, newly visible surface regions are being exposed at each frame. To maximize the use of previously observed surfaces, we accumulate the deformed incomplete mesh $\mathbf{f}'_i$ and its target $\mathbf{f}_{i+1}$ after each pairwise alignment. Second, as we allow our subject to change topology, tracking with a single consistent mesh (as with template-based approaches) is not possible. Merging the deformed mesh $\mathbf{f}'_i$ with its target $\mathbf{f}_{i+1}$ would not only improve computational efficiency (since the vertices will not be duplicated), but it would also allow source sample positions of the correspondences to adapt to the topology of the current frame.

We employ a mesh deformation based on Laplacian coordinates [Ale03] to warp frame $\mathbf{f}'_i$ to frame $\mathbf{f}_{i+1}$, and merge them by accumulating vertices of both meshes, followed by the Poisson surface reconstruction of Kazhdan et al. [KBH06]. Note that holes from $\mathbf{f}_{i+1}$ are reintroduced in the watertight Poisson reconstruction. Because of incomplete shapes, finding correspondences in unobserved surface regions for extended periods can result in accumulation of errors. As a result, the geometry of these areas can deteriorate over time and nearby surfaces can erroneously merge into a single surface. We therefore perform visual hull based pruning by disregarding vertices that fall outside the visual hull.

Furthermore, we only use the accumulated surfaces for correspondence computations, and do not use them for hole filling due to the same error accumulation.

4.3.5 Results

We demonstrate our method on three of the sequences (Saskia, Abhijeet, and Jay) made publicly available by Vlasic and colleagues [VPB+09]. Those high resolution scans were captured from 8 cameras placed around a human body and cover, on average, approximately 75% of the entire surface. For efficiency, we operate on down-sampled meshes, and up-sample when resynthesizing the detail. The statistics of our input and output data are as follows (we measure size of holes as the ratio between hole area over the area of the completed mesh):

dataset	#frames	#input vert	#output vert	size of holes
Saskia	113	132k∼140k	353k∼380k	25%∼27%
Jay	187	95k∼119k	278k∼335k	27%∼38%
Abhijeet	112	142k∼153k	369k∼412k	20%∼29%

Figure 6.9 and the accompanying video show intermediate results from those sequences at different stages of our pipeline. In addition, our reconstructions are suitable for free viewpoint video applications and can be seamlessly integrated into a virtual scene with different illuminations as demonstrated in Figure 4.22 and 4.23.

Compared to the original data, our meshes are complete and watertight, exhibit less temporal noise, and contain an equivalent or increased amount of surface detail. Naively closing the holes with visual hulls (as mentioned in [VPB+09]) produces watertight surfaces, but introduces even more temporal noise. More sophisticated methods ([WJH+07]) attempt to accumulate surface information over time. However, they have a hard time finding correspondences over many frames of non-rigid incomplete surfaces (second row in Figure 4.25). Consensus skeleton [ZST+10] may be used to determine a consistent topology throughout the whole motion, but we observe similar issues with our data, as it assumes clearly articulated and well-sampled underlying shapes (third row in Figure 4.25). Sharf and colleagues [SAL+08] can accumulate surface over time from sparse data such as ours, but may exhibit artifacts with flowing clothes that violate their volume-preserving assumption.

Timing. Ignoring data transfer, the whole pipeline runs at about 9 minutes per frame on a modern machine. The per frame hole-filling (Section 4.3.1) takes 40 seconds, Laplacian deformation and Poisson reconstruction (Section 4.3.2) adds 50 seconds, final detail

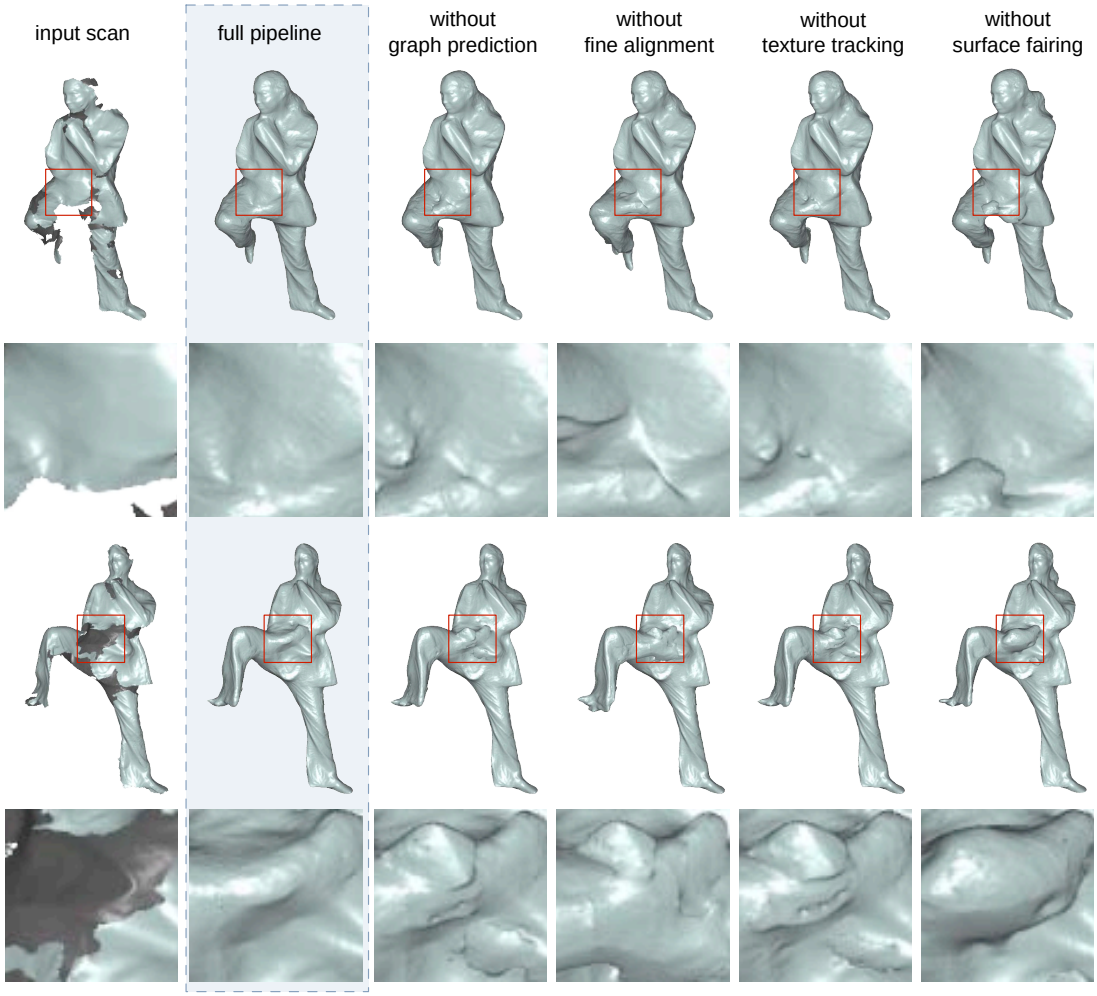


Figure 4.20: Comparison between full pipeline and leaving out individual stages of the correspondence computation. The last column clearly shows the importance of surface fairing.

resynthesis (Section 4.3.3) 90 seconds, coarse frame-to-frame alignment (Section 4.3.4) 45 seconds, and the fine-scale alignment (Section 4.3.4) an additional 320 seconds. The process can be run in parallel for each frame independently, which makes processing many frames of motion reasonable.

Limitations. Our method produces detailed watertight meshes that are smooth over time, but also lends to some limitations. First, the topology of our meshes will always match the (changing and sometimes incorrect) topology of the visual hull since we use it as the initial guess for shape completion (Figure 4.24 left). Ideally, we would like to

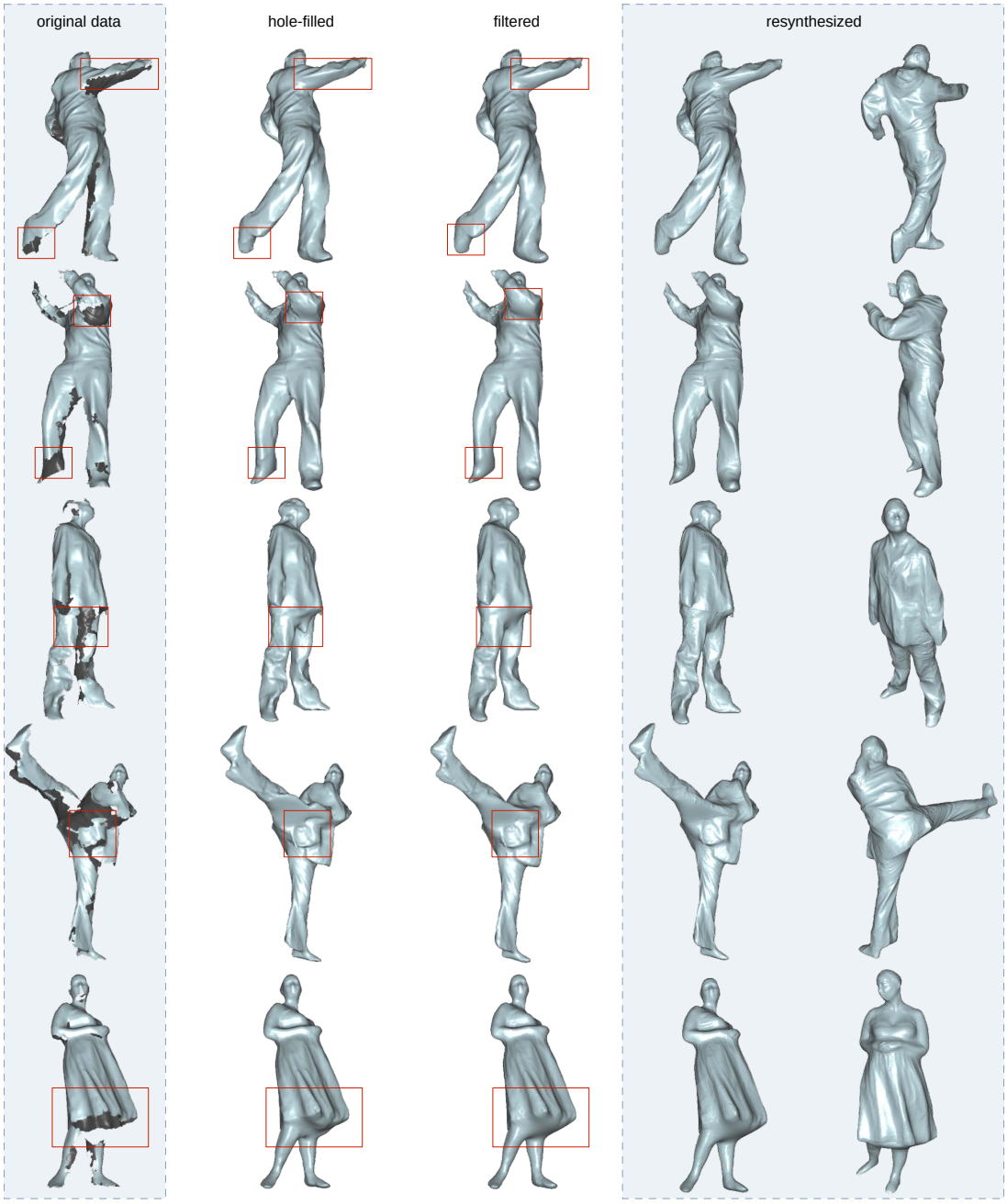


Figure 4.21: Left to right: original mesh, hole-filled mesh, temporally filtered mesh, two views of the final mesh with resynthesized detail.

extract a single consistent topology for the whole motion. Second, our temporal correspondences are valid between nearby frames, but they cannot be accurately propagated throughout the whole motion. This stands in the way of producing congruent moving

Figure 4.22: Our surfaces in conjunction with texture blending [CLB+09] are suitable for free-viewpoint video. Top: example with gliding cloth, impossible to faithfully reproduce with conventional template-based methods. Bottom: a complete digital models produce correct shadows.

meshes that are useful for analysis and editing, and should be addressed with a global approach. Third, the unobserved regions in each frame have no geometric details in them (Figure 4.24 right). With correspondences throughout the whole motion, the detail could be transferred from frames where those regions are visible. Nevertheless, we see our method as the next logical step towards the ultimate goal of dynamic shape capture, which is to acquire a single moving mesh, consistently parameterized over time, that exhibits all the observed detail and propagates it to the occluded regions throughout the whole motion.

4.3.6 Discussion

Due to the rapid advances in real-time 3D acquisition technology, the importance of obtaining temporally coherent watertight mesh sequences will be undeniable for many

Figure 4.23: Reconstructed human performance with full albedo integrated into a virtual scene with different illuminations.

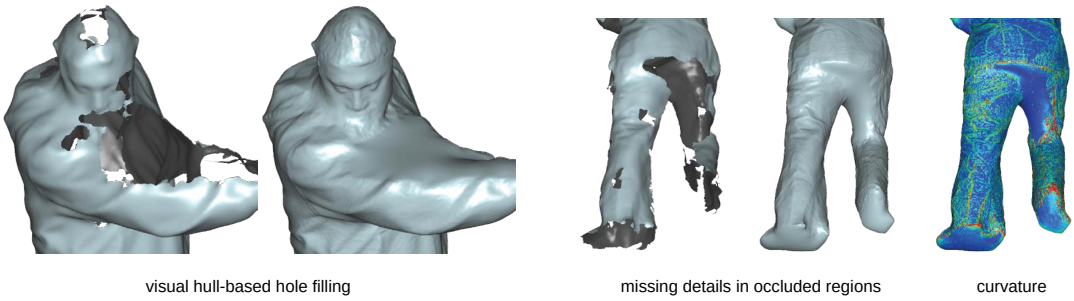


Figure 4.24: Although two recent methods (second and third row, with our resynthesized detail for fair comparison) produce single topologies over the complete motion, our method (first row) is able to recover more faithful per-frame surfaces.

applications involving digitization of dynamic objects. We present the first framework to automatically fill holes with temporal coherent patches without relying on a geometrical template. We have shown that the maturity of non-rigid registration techniques enables us to compute accurate and reliable correspondences for our purpose of filling

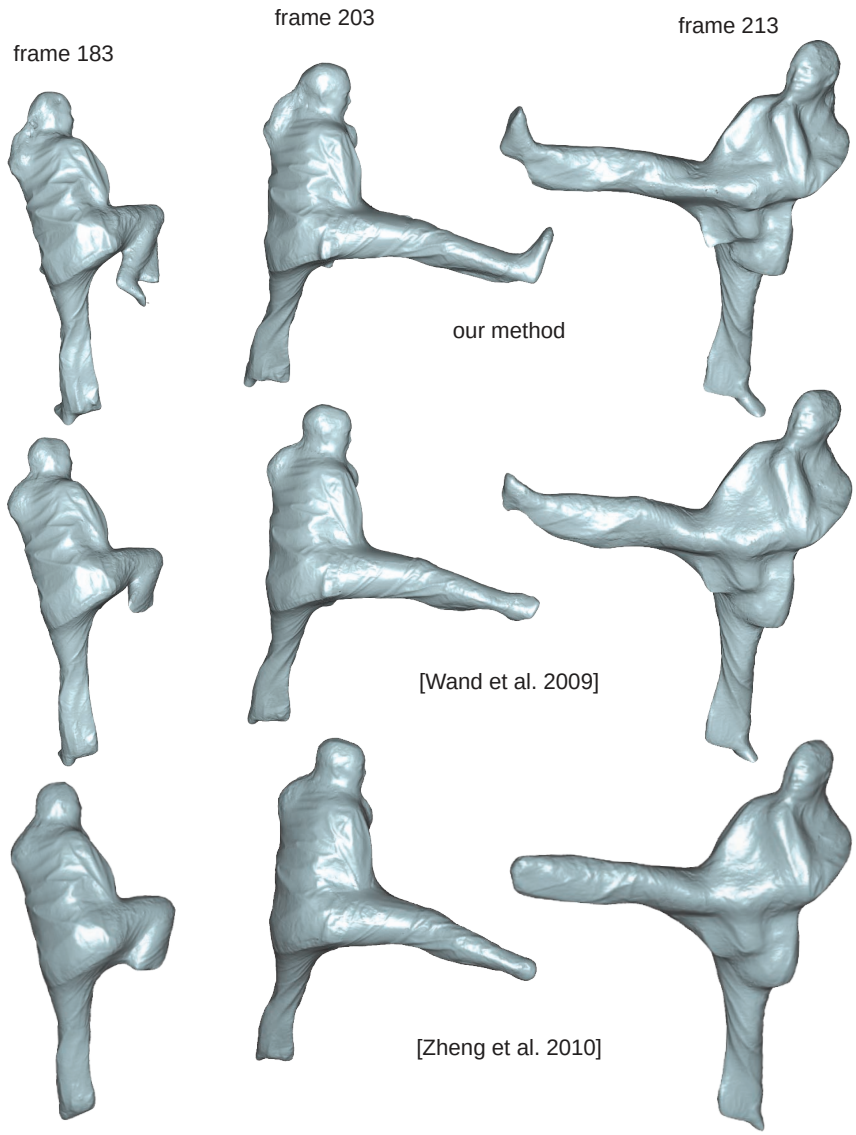


Figure 4.25: Although two recent methods (second and third row, with our resynthesized detail for fair comparison) produce single topologies over the complete motion, our method (first row) is able to recover more faithful per-frame surfaces.

holes in dynamic shapes. As opposed to other approaches, our method is specifically designed to handle changes in topology. Another advantage is that we can process scan sequences of arbitrary lengths without error accumulation because our correspondence computations are temporally localized. All presented results were produced from high resolution captured data of real-life performances that are publicly available [VPB+09]. Our key contribution is the interleaved registration/merging scheme which is propagated

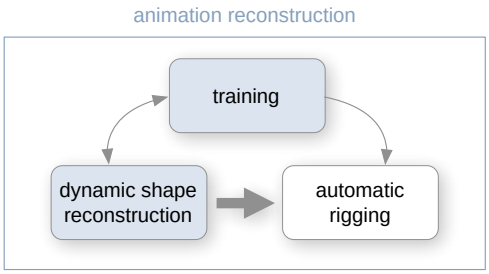
in a forward-and-backward fashion, the weighted temporal filtering of patches filled using the visual hull, and the integration of the all these components into a complete shape completion framework.

In considering shape completion of dynamic scans as a crucial step in digitization of real-world objects, we anticipate several challenges for future research. Since our proposed approach is purely geometric, a more accurate reproduction of deformations in occluded regions could possibly take into account physical properties that are either user guided or even learned from the captured data. Ultimately, we would like to address the problem of finding dense global correspondences through entire recordings and we postulate that determining them using hole-free surfaces is a simpler problem than using incomplete ones.

5

Facial Animation Reconstruction

With our feet wet, it is now time to dive into an important field of animation reconstruction, namely *facial animation*. So far, we introduced dynamic shape reconstruction as an essential tool for modeling shape and motion of arbitrarily deforming objects (such as human bodies, cloths, etc. . .). We even demonstrated that compelling facial expressions can be accurately recreated. So what motivates us to dedicate the rest of our investigation to facial animation? While many problems emphasize on high-quality reconstruction of unknown shape deformations (typically requiring several minutes of computation per frame), many others are of a different nature. For instance, in interactive applications or previsualization, the system requires a tight feedback loop in order to coordinate between user input and the resulting animation without sacrificing visual quality. Since general purpose reconstruction methods require a particularly flexible deformation model, many degrees of freedom (unknowns) are required. Hence, their computational costs present a significant bottleneck for real-time applications. In addition, the large dimensionality of their search space brings additional challenges in avoiding local minima, especially when the subject exhibits large deformations.



We consider dimensionality reduction based on a *data-driven training* process as an effective approach to lift these restrictions and thus, an integral part of the animation reconstruction pipeline (c.f., diagram). Because the space of facial expressions is considerably smaller than for example a full body performance or cloth deformation, facial animation presents an ideal playground to explore the idea of using *training* data.

This chapter introduces a complete integrated system for markerless interactive facial animation and enables the following:

- High-resolution real-time facial tracking
- Live expression transfer to another person's face.

The system utilizes the same real-time structured light scanner as in previous chapters without requiring markers or specialized tracking hardware. In addition to geometry, we also track texture information which is crucial for improved robustness and accuracy. By using a template model for tracking, we directly obtain consistent correspondences across the entire recording. As a first step, we build this template by fitting a generic facial model to a rigid surface reconstruction of an actor's face (*shrink-wrapping*).

Our objective is to achieve real-time performance by shifting complexity from online computation to an off-line training stage. Training includes robust and accurate tracking of a large set of facial performances in order to cover a maximum space of possible expressions. We then build a reduced linear subspace from this training data. In this way, we simplify the general tracking algorithm to its essentials for robust online tracking. Note that this technique dramatically reduces error accumulation for extended scan sequences. Similarly, real-time transfer of facial expressions onto a different face is achieved by a linear model based on preprocessed deformation transfer [SP04]. This allows plausible live animations of different characters, even when only a single rigid model of the target face is available.

Background and Motivation. Convincing facial expressions are essential to captivate the audience in stage performances, live-action movies, and computer-animated films. Producing compelling facial animations for digital characters is a time-consuming

and challenging task, requiring costly production setups and highly trained artists. The current industry standard in facial performance capture relies on a large number of markers to enable dense and accurate geometry tracking of facial expressions. The captured data is usually employed to animate a digitized model of the actor's own face or transfer the motion to a different one. While recently released feature films such as *The Curious Case of Benjamin Button* demonstrated that flawless retargeting of facial expressions can be achieved, film directors are often confronted with long turn-around times as mapping such a performance to a digital model is a complex process that relies heavily on manual assistance.

Figure 5.1: Real-time facial expression transfer to CG characters permits high-quality previsualization of complex facial expressions.

Markerless live puppetry enables a wide range of new applications. In movie production, our system complements existing off-line systems by providing immediate real-time feedback for studying complex face dynamics. Directors get to see a quick 3D preview of a face performance, including emotional and perceptual aspects such as the effect of the intended makeup (see Figure 5.1). In interactive settings such as TV shows or computer games, live performances of digital characters become possible with direct control by the actor.

Besides specifying a few feature points for the initial non-rigid alignment of the

template to the scanned actor's face, no manual intervention is required anywhere in our live puppetry pipeline. The automatic processing pipeline in combination with a minimal hardware setup for markerless 3D acquisition is essential for the practical relevance of our system that can easily be deployed in different application scenarios.

5.1 Related Work

Due to the great amount of research in facial modeling and animation, we only discuss previous work most relevant to our online system and refer to [PW96] and [DN07] for a broader perspective. Facial animation has been driven by different approaches, in general using parametric [Par82, CM93], physics-based [TW90, SSRMF06], and linear models [Par72, BV99, VBPP05].

Linear Face Models. Linear models represent faces by a small set of linear components. Blendshape models store a set of key facial expressions that can be combined to create a new expression [Par72, PSS99, Chu04, JTDP03]. Statistical models represent a face using a mean shape and a set of basis vectors that capture the variability of the training data [Sir87, BV99, KMG04, VBPP05, LCXS07]. This allows modeling of a full population, while blendshape models are only suitable for an individual person. Global dependencies between different face parts arising in linear models are generally handled by segmenting the face into independent subregions [BV99, JTDP03].

Facial Performance Capture. Performance-driven facial animation uses the performance of an actor to animate digital characters and has been developed since the early 80s [PB81]. Marker-based facial motion capture [Wil90, CXH03, DCFN06, LCXS07, BLB+08, MJC+08] is frequently used in commercial movie projects [Hav06] due to the high quality of the tracking. Drawbacks are substantial manual assistance and high calibration and setup overhead. Methods for offline facial expression tracking in 2D video have been proposed by several authors [PSS99, BBPV03, VBPP05, BHPS10]. The latter system uses 14 high definition video cameras in order to enable optical flow tracking at skin pore levels. Hiwada and co-workers [HMN03] developed a real-time face tracking system based on a morphable model, while Chai and colleagues [CXH03] use feature tracking combined with a motion capture database for online tracking. To the best of our knowledge, our system is the first real-time markerless facial expression tracking system using accurate 3D range data. [KMG04] developed a system to record and transfer

speech related facial dynamics using a full 3D pipeline. However, their system has no real-time capability and requires some markers for the recording. Similarly, [ZSCS04] present an automatic offline face tracking method on 3D range data. The resulting facial expression sequences are then used in an interactive face modeling and animation application [ZSCS04, FKY08]. We enhance their method for offline face tracking and use the facial expression data for online tracking and expression transfer. While all the above methods use a template model, techniques exist that do not require any prior model and are able to recover non-rigid shape models from single view 2D image sequences [BHB00]. Although only a rough shape can be reconstructed, features such as eyes and mouth can be reliably tracked.

Facial Expression Transfer. Noh and Neumann [NN01] introduced *expression cloning* to transfer the geometric deformation of a source 3D face model onto a target face. Sumner and Popovic [SP04] developed a generalization of this method suitable for any type of 3D triangle mesh. More closely related to our method, [CXH03] perform expression cloning directly on the deformation basis vectors of their linear model. Thus expression transfer is independent of the complexity of the underlying mesh. A different approach is taken by [PSS99, Chu04, ZLG+06] who explicitly apply tracked blendshape weights for expression transfer. The latter one does not require example poses of the source. [CB05] extended the method to reproduce expressive facial animation by extracting information from the expression axis of speech performance. Similarly, [DCFN06] map a set of motion capture frames to a set of manually tuned blendshape models and use radial basis function regression to map new motion capture data to the blendshape weights. In contrast, Vlasic and colleagues [VBPP05] use multi-linear models to both track faces in 2D video and transfer expression parameters between different subjects.

5.2 Real-time Markerless Facial Expression Retargeting

Our facial puppetry system [WLGP09] allows live control of an arbitrary *target* face by simply acting in front of a real-time structured light scanner projecting phase shift patterns. Geometry and texture are both captured at 25 fps. All necessary details of the scanning system can be found in [WLG07]. The actor's face is tracked online and facial expressions are transferred to the puppet in real-time.

As shown in Figure 5.2, our system consists of three main components: Person-

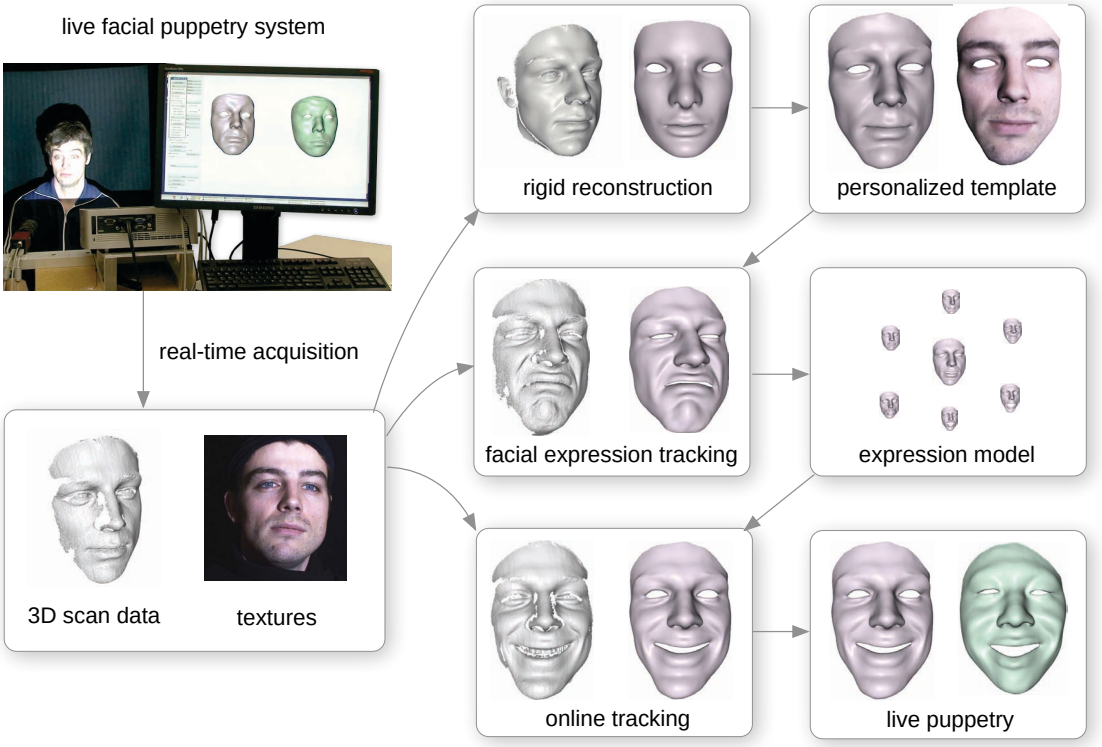


Figure 5.2: Our system is composed of three main parts: Personalized template building, facial expression recording, and live facial puppetry. All components rely on input from a real-time structured light scanner. During template building a generic template is fit to the reconstructed 3D model of the actor's face. Dynamic facial expressions of the actor are then recorded and tracked using non-rigid registration. A person-specific facial expression model is constructed from the tracked sequences. The model is used for online tracking and expression transfer, allowing the actor to enact different persons in real-time.

alized template building, facial expression recording, and live facial puppetry. These components are described in more detail in Sections 5.2.1. 5.2.2, and 5.2.3, respectively. The key to online performance is to first record a set of facial expressions of the actor that are processed offline, and then build a simplified facial expression model specific to the actor for efficient online tracking and expression transfer.

For template building, non-rigid registration is used to deform a generic template mesh to the 3D reconstruction of the actor's face. This personalized template is then tracked offline through a set of expression sequences. We take advantage of face specific

constraints to make the tracking accurate and robust. The recorded expression sequences are used to build a simplified representation of the facial expression space using principal component analysis (PCA). The reduced set of parameters of the model enables efficient online tracking of the facial expressions. We propose a simple yet effective method for real-time expression transfer onto an arbitrary target face: We build a linear face model of the target face that uses the same parameters as the actor's facial expression model, reducing expression transfer to parameter transfer. To build the linear model we use deformation transfer [SP04] (c.f. Section 3.2.3) on the facial expression sequences of the actor and then find the optimal linear facial expression space for the target.

Deformable Face Model. Building the personalized template and recording facial expressions both require a non-rigid registration method to deform a face mesh to the given input geometry. Non-rigid registration methods typically formulate deformable registration as an optimization problem consisting of a mesh smoothness term and several data fitting terms as described in Section 3.2.

We consider deformations with displacement vectors $\mathbf{d}_i = \tilde{\mathbf{v}}_i - \mathbf{v}_i$ for each mesh vertex $\mathbf{v}_i \in \mathcal{V}$ and deformed mesh vertex $\tilde{\mathbf{v}}_i \in \tilde{\mathcal{V}}$. Deformation smoothness is achieved by minimizing a bending energy term $E_{\text{bend}} = \sum_{i \in \mathcal{V}} \|\Delta \mathbf{d}_i\|^2$ on the displacement vectors, using the standard cotangent discretization of the Laplace-Beltrami operator Δ (see Section 3.2). Notice that the minimization of E_{bend} leads to the bi-Laplacian equation $\Delta^2 = 0$. The resulting linear deformation model is suitable for handling a wide range of facial deformations, while still enabling efficient processing of extended scan sequences.

We prefer the bending model with co-tangent weighted Laplacian over minimizing vertex displacement differences as in [ZSCS04] (see Section 3.2), since the latter is equivalent to a Laplace discretization with inversely-weighted edge length and results in less natural deformations as illustrated in Figure 5.3.

Our experiments showed that these differences are particularly noticeable when incorporating dense and sparse constraints simultaneously in the optimization.

When personalizing the template (Section 5.2.1) we employ dense closest-point, and point-to-plane constraints [CM92], as well as manually selected sparse geometric constraints each formulated as energy terms for data fitting. For the automated expression recording, a combination of sparse and dense optical flow texture constraints [HS81] replaces the manually selected correspondences (Section 5.2.2). In both cases, face deformations are computed by minimizing a weighted sum of the different linearized energy

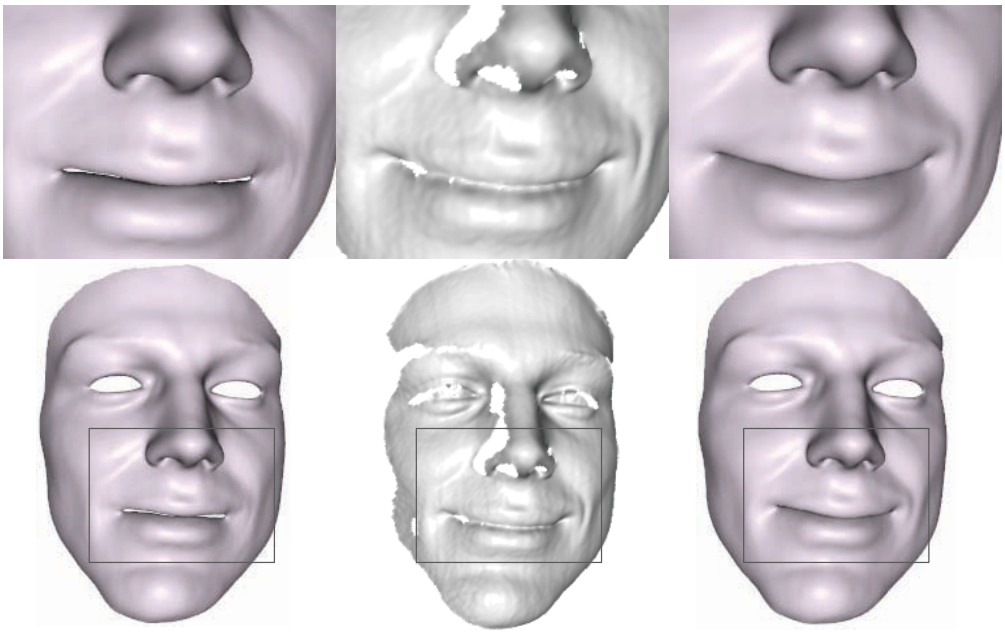


Figure 5.3: The bending model with co-tangent weighted Laplace discretization (right) allows for more natural deformations than using Laplacians with inversely-weighted edge length (left). The difference is particularly visible in the corners of the mouth.

terms described below. The resulting over-determined linear system is sparse and can be solved efficiently via Cholesky decomposition [SG04].

5.2.1 Personalized Template Building

We generate an actor-specific template $\mathcal{M}$ by deforming a generic template mesh $\mathcal{M}_{\text{neutral}}$ to the rigid reconstruction of the actor's face (see Figures 5.2 and 5.4). Besides enabling a hole-free reconstruction and a consistent parameterization, using the same generic template has the additional benefit that we obtain full correspondence between the faces of the different characters.

Rigid Reconstruction. The face model is built by having the actor turn his head in front of the scanner with a neutral expression and as rigidly as possible. The sequence of scans is combined using on-line rigid registration similar to [RHHL02] to obtain a dense point cloud $\mathcal{P}$ of the complete face model. Approximately 200 scans are registered and merged for each face.

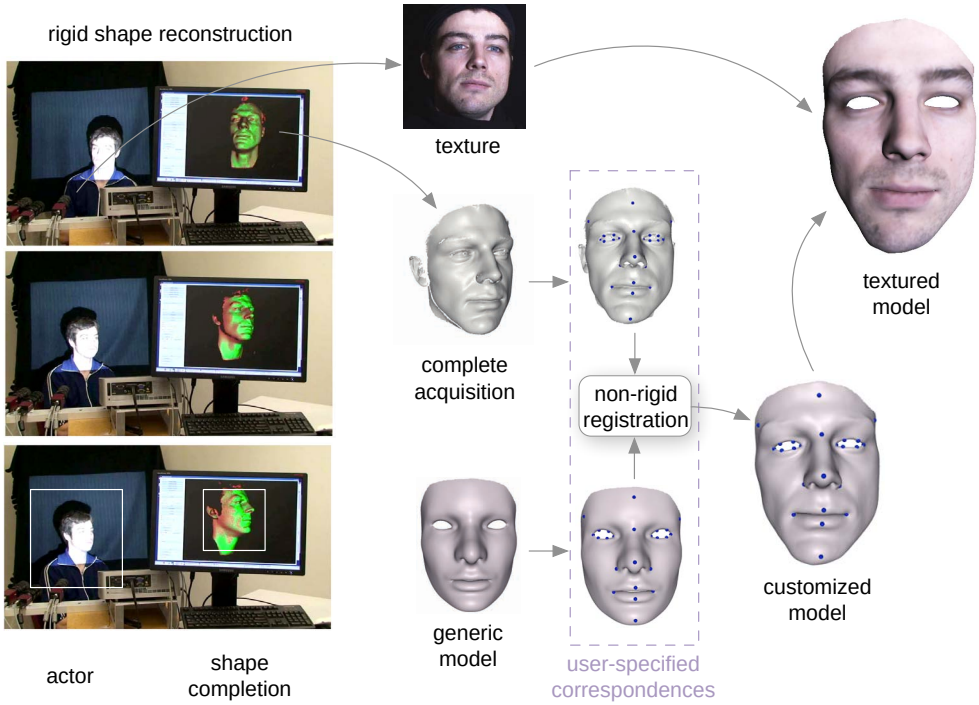


Figure 5.4: Personalized template building: 24 manually labeled reference points are used for the rigid registration and subsequent iterative non-rigid alignment. The mesh is then deformed by additionally using closest point constraints to fit the reconstructed mesh.

Template Fitting. We use manually labeled reference points $\mathbf{r}_j \in \mathcal{P}$ for initial rigid ICP registration of the generic template and the reconstructed face model (Figure 5.4). The reference points also provide sparse correspondence constraints in a subsequent non-rigid registration that deforms the template $\mathcal{M}_{\text{neutral}}$ towards $\mathcal{P}$ to obtain $\mathcal{M}$ using the sparse energy term $E_{\text{ref}} = \sum_j \|\tilde{\mathbf{v}}_j - \mathbf{r}_j\|_2^2$. Our manually determined correspondences are mostly concentrated in regions such as eyes, lips, and nose, but a few points are selected in featureless areas such as the forehead and chin to match the overall shape geometry. A total number of 24 reference points were sufficient for all our examples.

To warp the remaining vertices $\mathbf{v}_i \in \mathcal{M}_{\text{neutral}}$ toward $\mathcal{P}$, we add a dense fitting term based point-to-plane minimization with a small point-to-point regularization as described in [MGPG04]:

$$E_{\text{fit}} = \sum_{i=1}^{N} w_i (|\mathbf{n}_{\mathbf{c}_i}^{\top}(\tilde{\mathbf{v}}_i - \mathbf{c}_i)|^2 + 0.1\,\|\tilde{\mathbf{v}}_i - \mathbf{c}_i\|_2^2). \tag{5.1}$$

The closest point on the input scan from $\tilde{\mathbf{v}}_i$ is denoted by $\mathbf{c}_i \in \mathcal{P}$ with corresponding surface normal $\mathbf{n}_{\mathbf{c}_i}$. We prune all closest point pairs with incompatible normals [RL01] or distance larger than 10 mm by setting the corresponding weights to $w_i = 0$ and $w_i = 1$ otherwise. Combining correspondence term and fitting term with the bending model yields the total energy function $E_{\text{tot}} = E_{\text{fit}} + \alpha_{\text{ref}} E_{\text{ref}} + \alpha_{\text{bend}} E_{\text{bend}}$. The weights $\alpha_{\text{bend}} = 100$ and $\alpha_{\text{ref}} = 100$ are gradually reduced until $\alpha_{\text{bend}} = 5$ and $\alpha_{\text{ref}} = 1$. For all our examples we use the same scheduling for the energy weights (see also [LSP08]).

Texture Reconstruction. The diffuse texture map for the personalized face template is retrieved from the online rigid registration stage by averaging the textures of all input scans used for rigid reconstruction. The scan textures are the recorded video frames and have a resolution of 780×580 pixels. We use the projector's light source position to compensate for lighting variations assuming a dominantly diffuse reflectance model. Similarly, we remove points that are likely to be specular based on the half angle. The resulting texture map is over-smoothed, but sufficient for the tracking stage and has a resolution of 1024×768.

5.2.2 Facial Expression Recording

To generate the facial expression model we ask the actor to enact the different dynamic expressions that he plans to use for the puppetry. In the examples shown in our accompanying video, the actors perform a total of 26 facial expression sequences including the basic expressions (happy, sad, angry, surprised, disgusted, fear) with closed and open mouth as well as a few supplemental expressions (agitation, blowing, long spoken sentence, etc.). The personalized template $\mathcal{M}$ is then tracked through the entire scan sequence. For each input frame we use rigid ICP registration to compensate for global head motion yielding a rigid motion $(R, \mathbf{t})$. The generic non-rigid registration method described above then captures face deformations by adding to each rigidly aligned vertex $\bar{\mathbf{v}}_i = R\,\mathbf{v}_i + \mathbf{t}$ the displacement vector $\mathbf{d}_i = \tilde{\mathbf{v}}_i - \bar{\mathbf{v}}_i$. Note that a rigid head compensation is essential for robust tracking, since our globally elastic deformation model is a linear approximation of a non-linear shell deformation and thus cannot handle large rotations accurately [BS08]. Because of high temporal coherence between the scans, projective closest-point correspondences are used to compute $\mathbf{c}_i$ for Equation 5.1. In addition, we set w_i in E_{fit} to zero if $\mathbf{c}_i$ maps to a hole. Besides the dense geometric term E_{fit} and smoothness energy E_{bend}, we introduce a number of face specific additions, including

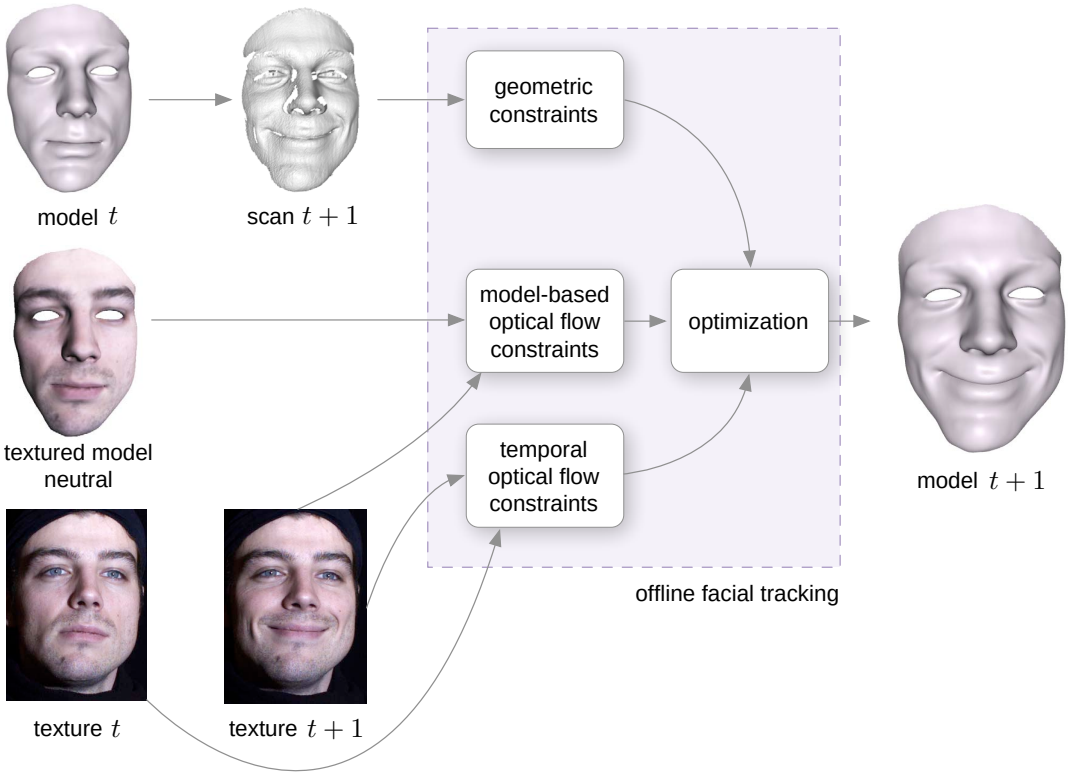


Figure 5.5: Offline facial tracking for expression recording.

dense and sparse optical flow texture constraints to improve accuracy and robustness of the tracking. Most notably, we explicitly track the mouth, chin, and eyelids.

Dense Optical Flow Constraints. Optical flow is used to enhance template tracking by establishing inter-frame correspondences from video data. Instead of using an independent optical flow procedure as in [ZSCS04], we directly include the optical flow constraints into the optimization, similar to model-based tracking methods [DM00]. We thus avoid solving the difficult 2D optical flow problem and integrate the constraints directly into the 3D optimization:

$$E_{\text{opt}} = \sum_{i=1}^{N} h_i \left(\nabla g_{t,i}^{\top} \;\; \Pi \left(\tilde{\mathbf{v}}_i^{t+1} - \tilde{\mathbf{v}}_i^{t} \right) + g_{t+1,i} - g_{t,i} \right) \tag{5.2}$$

where $g_{t,i} = g_t(\Pi(\tilde{\mathbf{v}}_i^t))$ is the image intensity at the projected image space position $\Pi(\tilde{\mathbf{v}}_i^t)$ of 3D vertex $\tilde{\mathbf{v}}_i^t$ at time t. Vertices at object boundaries and occlusions that pose problems in 2D optical flow are detected by projecting the template into both the camera and projector space and checking each vertex for visibility. We set the per vertex weight

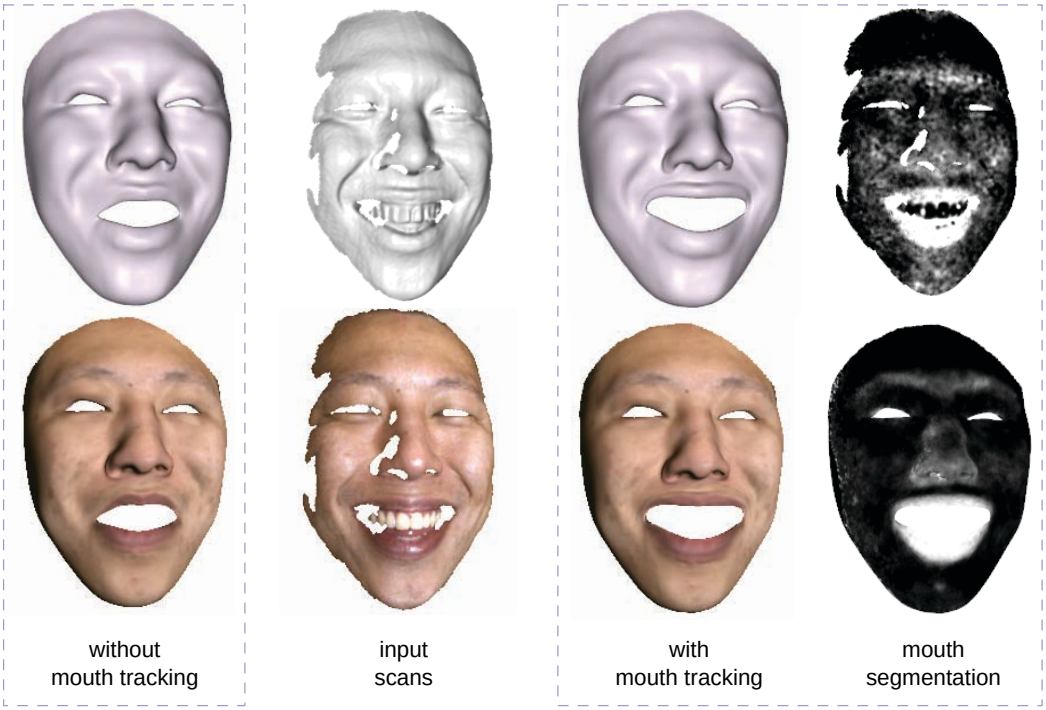


Figure 5.6: Lip segmentation considerably improves the tracking results for the mouth region. The contrast enhancement due to lip classification can be seen on the right.

to $h_i = 1$ if visible and $h_i = 0$ otherwise. To ensure linearity in the optical flow energy E_{opt} we use a weak perspective camera model that we define as

$$\Pi(\mathbf{x}_i) = \frac{f}{\bar{z}_i} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} (R_{\text{cam}}\, \mathbf{x}_i + \mathbf{t}_{\text{cam}}), \tag{5.3}$$

where $\bar{z}_i$ is the fixed depth of the current template vertex $\tilde{\mathbf{v}}_i$, f the focal length, and $(R_{\text{cam}}, \mathbf{t}_{\text{cam}})$ the extrinsic camera parameters.

Optical flow is applied in a hierarchical fashion using a 3 level Gaussian pyramid, where low resolution video frames are processed first to allow for larger deformations. In each optimization step, we re-project all visible vertices to the image plane and recalculate the spatial image gradient $\nabla g_{t,i}$ using a standard Sobel filter, and the temporal derivative of the image intensity using forward differences.

Mouth Tracking. Optical flow is calculated sequentially and assumes that vertex positions in the previous frame are correct. This inevitably leads to drift, which is particularly noticeable in the mouth region as this part of the face typically deforms the

most. We employ soft classification based on binary LDA [Fis36] to enhance the contrast between lips and skin. The normalized RGB space is used for illumination invariance. Soft classification is applied both to the scan video frame g_t and the rendering of the textured template g_t^* leading to two gray level images with strong contrast $\hat{g}_t$ and $\hat{g}_t^*$, respectively (Figure 5.6). Optical flow constraints between the template and the scan are then applied for the mouth region, in addition to the scan-to-scan optical flow constraints:

$$E_{\text{opt}}^* = \sum_{j \in \mathcal{V}_M} h_j \left(\nabla \hat{g}_{t,j}^{*\top} \;\; \Pi \left(\tilde{\mathbf{v}}_j^{t+1} - \tilde{\mathbf{v}}_j^t \right) + \hat{g}_{t+1,j} - \hat{g}_{t,j}^* \right) \tag{5.4}$$

where $\mathcal{V}_M$ is the set of vertices of manually segmented mouth and lips regions in the generic face template.

Thus mouth segmentation not only improves optical flow but also prevents drift as it is employed between scan and template texture which does not vary over time. The Fisher LDA is trained automatically on the template texture as both skin and lip vertices have been manually marked on the template mesh, which only needs to be performed once.

Chin Alignment. The chin often exhibits fast and abrupt motion, e.g., when speaking, and hence the deformable registration method can fail to track the chin correctly (Figure 5.7). However, the chin typically exhibits little deformation, which we exploit in an independent rigid registration for the chin part to better initialize the correspondence search for both geometry and texture. As a result, fast chin motion can be tracked very robustly.

Eyelid Tracking. Eyelids move very quickly and eye blinks appear often just for a single frame. Neither optical flow nor closest point search give the appropriate constraints in that case (Figure 5.10). However, the locations of the eye corners can be determined by a rigid transformation of the face. Assuming a parabolic shape of the eyelid on the eyeball, we can explicitly search for the best eyelid alignment using texture correlation. The resulting correspondences are included into the optimization using a specific fitting term E_{eye} of closest-point constraints, similar to E_{fit}. A full statistical model [HIWZ05] was not required in our experiments, but could be easily incorporated into the framework.

Border constraints. The structured light scanner observes the geometry only from a single viewpoint. The sides of the face are mostly hidden and thus underconstrained in

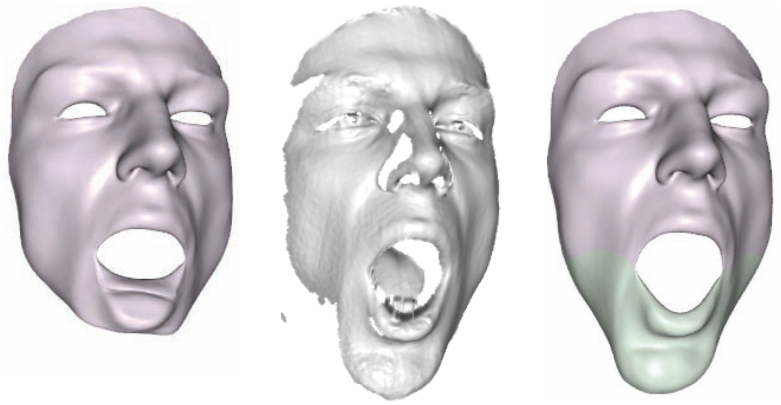


Figure 5.7: Fast motion can lead to registration failure (left). Explicit rigid tracking of the chin significantly improves the robustness and convergence of the non-rigid registration algorithm (right). The chin region is marked in green and needs only be determined once on the generic template face.

the optimization. For stability we fix the border vertices to the positions as determined by rigid registration.

Iterative Optimization. To improve convergence in the facial expression recording, we schedule $M = 5$ optimization steps for each input scan by recalculating closest points and using a coarse-to-fine video frame resolutions. After rigid alignment, we perform three steps of optimization with increasing resolution in the Gaussian pyramid for estimating image gradients and two optimization at the highest resolution. Each optimization step minimizes the total energy $E_{\text{tot}} = E_{\text{fit}} + \alpha_{\text{opt}} E_{\text{opt}} + \alpha^*_{\text{opt}} E^*_{\text{opt}} + \alpha_{\text{eye}} E_{\text{eye}} + \alpha_{\text{bend}} E_{\text{bend}}$ with constant energy weights $\alpha_{\text{opt}} = 5$, $\alpha^*_{\text{opt}} = 100$, $\alpha_{\text{eye}} = 0.5$, and $\alpha_{\text{bend}} = 10$.

5.2.3 Live Facial Puppetry

Online Face Tracking

Face tracking using the deformable face model is very accurate and robust, but computationally too expensive for online performance. Even though all constraints are linear and the resulting least-squares problem is sparse, solving the optimization requires approximately 2 seconds per iteration and 5 iterations per frame since the left hand side of a sparse but large linear system need to be updated in each step.

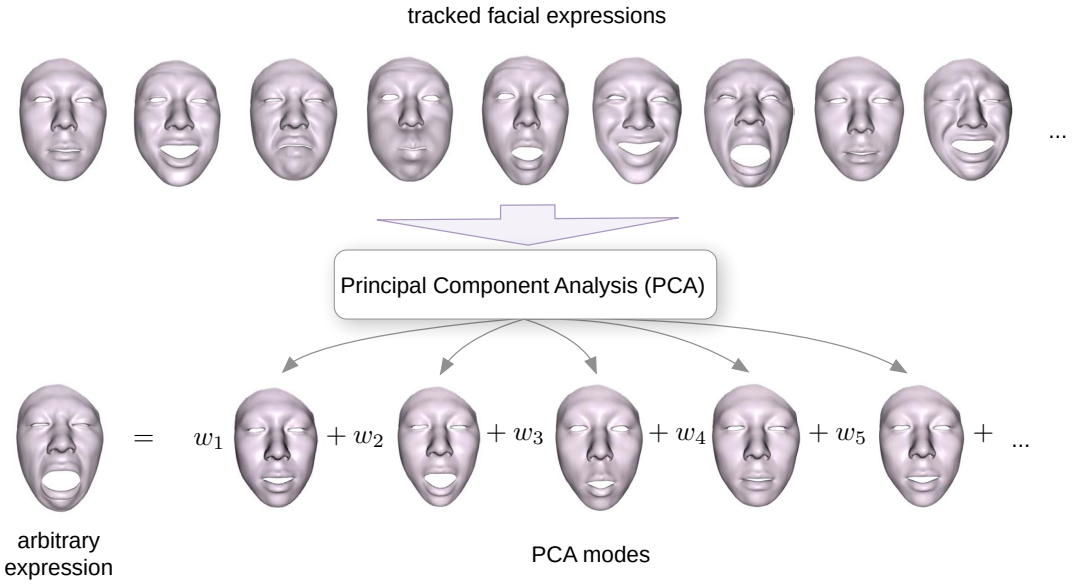


Figure 5.8: PCA dimensionality reduction. Given a large set of input facial expressions that are in correspondence, we only retain the most dominating PCA modes.

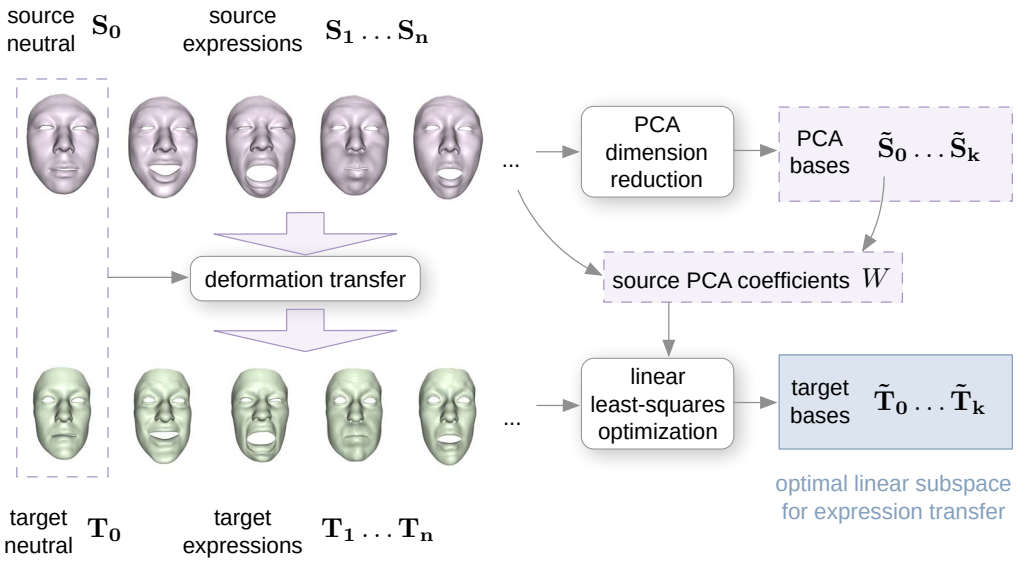
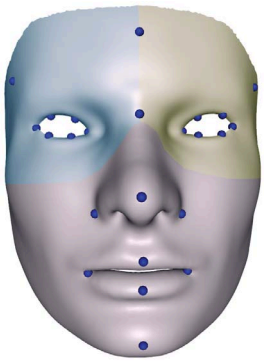


Figure 5.9: Construction of optimal linear subspace in space of deformation transfer

In order to achieve real-time performance we employ PCA dimensionality reduction in facial expression space similar to [BV99]. We also manually segment the face into

several subparts to break global dependencies. In our case this is the mouth and chin region, and symmetrically each eye and forehead (see illustration below).


three PCA segments

The effectiveness of PCA depends on the quantity, quality, and linearity of the underlying data. Linearity has been demonstrated in previous PCA-based face models [Sir87, BV99, VBPP05]. One important advantage of our system is that we can easily generate a large number of high-quality training samples by recording a continuous sequence of facial expression tracked using our offline registration method (Figure 5.11). This allows us to accurately sample the dynamic expression space of the actor, which is essential for live puppetry. As opposed to previous methods based on linear dimension reduction, our approach uses dense batches of scans for the recording of each sequence.

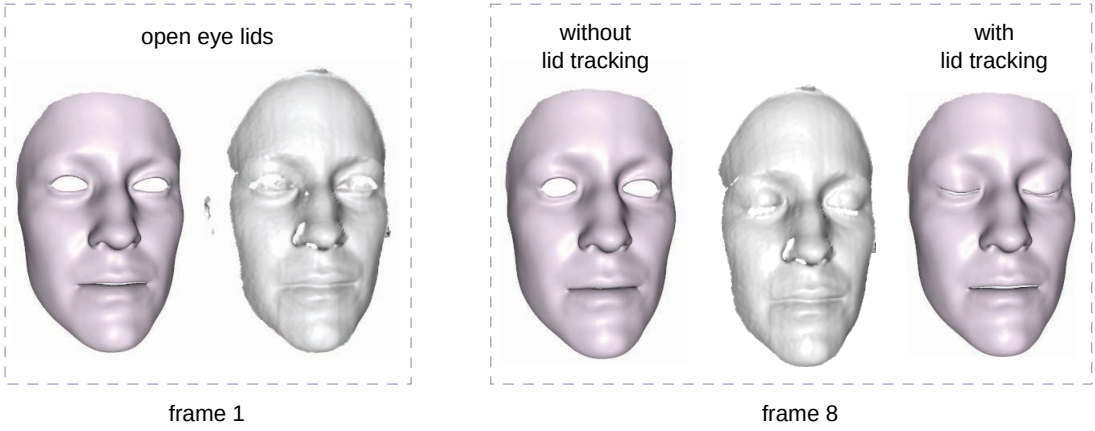


Figure 5.10: Eyelid tracking enables the system to track eye blinks correctly. Without explicit eyelid tracking the system fails to track the closing of the eyes.

Online Registration. PCA represents the expression space by a mean face and a set of K components. At run-time, the system registers the facial expression by searching for the K coefficients that best match the data.

In principle, all the constraints used in offline face tracking can be included in the optimization. We found that due to the much lower dimensionality of the problem, projective closest-point correspondence search with point-plane constraints is usually

Figure 5.11: A small subset of the roughly 250 expressions used for the generation of the PCA expression model for a specific actor.

sufficient to faithfully capture the dynamics of the face. However, we include rigid chin tracking to improve stability. We currently use $K = 32$ PCA components divided appropriately between the three face segments, which proved to be sufficient for representing more than 98% of the variability of the training data. More components did not add any significant benefit in tracking quality. We avoid discontinuities at the segment borders by pulling the solution towards the mean of the PCA model [BV99]. Online registration is achieved by optimizing $E_{\text{tot}} = E_{\text{fit}} + 0.1 \sum_{i=1}^{K} \|k_i\|_2^2$ where k_i are the PCA coefficients replacing the previous optimization variables $\mathbf{d}_i$.

All algorithms except the rigid registration are implemented on the GPU using shading languages and CUDA. With all these optimizations in place, our system achieves 15 frames per second, which includes the calculation of the structured light scanning system, rigid registration, chin registration, PCA-based deformation, and display.

Expression Transfer

Online face tracking allows the actor to control an accurate digital representation of his own face. Expression transfer additionally enables mapping expressions onto another person's face in real-time. The actor becomes a puppeteer.

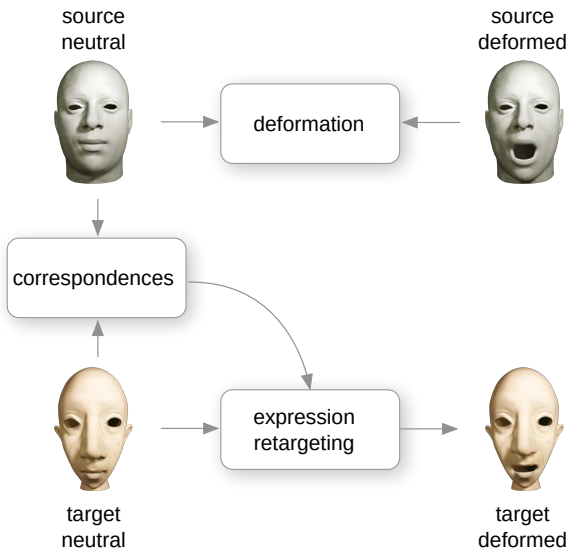


Figure 5.12: Expression retargeting. After establishing correspondences between a source and target model, we may transfer the deformation of source face onto an arbitrary target model in neutral pose.

To accomplish expression transfer, we consider the optimization in gradient space introduced by Sumner and Popović [SP04]. The goal is to map the deformation of a source mesh $\mathbf{S}$ onto an arbitrary target mesh $\mathbf{T}$. As described in Section 3.2.3, we use the per-triangle *deformation gradients* defined between a source mesh in its rest pose $\mathbf{S}$ and deformed state $\tilde{\mathbf{S}}$. The deformation gradients are then transferred to $\mathbf{T}$ by enforcing mesh connectivity by solving a Poisson equation. Since the template mesh provides correspondences, we can directly determine the deformation gradients between a face in neutral pose $\mathbf{S}_{\text{neutral}}$ and each captured expression $\mathbf{S}_i$. Thus, only a single target pose in neutral position $\mathbf{T}_{\text{neutral}}$ is required to determine all corresponding target expressions $\mathbf{T}_i$.

In our experiments we found that deformation transfer from one face to another yields very plausible face animations (Figure 5.13), giving the impression that the target face has the mimics of the actor. We note that we are not considering the problem of animating a different character with a non-human face. In that case models based on blendshapes [Chu04] seem more appropriate as deformations in the source and target face may not correlate geometrically.

Linear Deformation Basis. Unfortunately, deformation transfer on the high resolution template mesh (25 K vertices) is too inefficient for real-time performance. To enable live puppetry, we generate a linear subspace that optimally spans the space of deformation transfer. For this purpose we compute the PCA bases of all $\bar{\mathbf{S}} = [\mathbf{S}_1 \dots \mathbf{S}_n]$ and find the least squares optimal linear basis for the target face $\bar{\mathbf{T}} = [\mathbf{T}_1 \dots \mathbf{T}_n]$ that is driven by the same coefficients $\mathbf{W}$ as the actor's PCA model. Thus, expression transfer is reduced to applying the coefficients of the actor PCA model to a linear model of the target shape.

Assume the training shapes of the actor can be expressed by the linear combination of PCA basis vectors $\tilde{\mathbf{S}}_i$:

$$\begin{bmatrix} \mathbf{S}_1 \\ \dots \\ \mathbf{S}_n \end{bmatrix} = \begin{bmatrix} w_{11} & \dots & w_{1k} \\ & \dots & \\ w_{n1} & \dots & w_{nk} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{S}}_1 \\ \dots \\ \tilde{\mathbf{S}}_k \end{bmatrix} \tag{5.5}$$

We look for the linear basis $\left[\tilde{\mathbf{T}}_1 \dots \tilde{\mathbf{T}}_k\right]^\top$ that best generates the target shapes $[\mathbf{T}_1 \dots \mathbf{T}_n]^\top$ using the same weights:

$$\begin{bmatrix} \mathbf{T}_1 \\ \dots \\ \mathbf{T}_n \end{bmatrix} = \begin{bmatrix} w_{11} & \dots & w_{1k} \\ & \dots & \\ w_{n1} & \dots & w_{nk} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{T}}_1 \\ \dots \\ \tilde{\mathbf{T}}_k \end{bmatrix} \tag{5.6}$$

We solve this overdetermined linear least-squares problem using normal equations, where $\mathbf{W}$ is determined by simple projection of $\mathbf{S}_i$ onto the PCA bases $\tilde{\mathbf{S}}_i$ and $\left[\tilde{\mathbf{T}}_1 \dots \tilde{\mathbf{T}}_k\right]^\top = [\mathbf{W}^\top \mathbf{W}]^{-1}\mathbf{W}^\top \bar{\mathbf{T}}$. The resulting basis vectors of the linear model are not orthogonal, but this is irrelevant for transfer. The training samples are already available from the offline facial expression tracking, and thus all expressions that are captured by the PCA model can also be transfered to the target face. For segmented PCA, each segment is transfered independently.

5.2.4 Results

Our system achieves accurate 3D facial tracking and real-time reconstruction at 15 fps of a complete textured 3D model of the scanned actor. In addition, we can transfer expressions of the actor at the same rate onto different face geometries. All computations were performed on an Intel Core Duo 3.0 Ghz with 2 GB RAM and a GeForce 280 GTX.

We demonstrate the performance of our approach with two male actors (Caucasian and Asian) and one female actress (Caucasian) as shown in Figure 5.13. Live puppetry is conducted between each actor and with two additional target models, a 3-D scanned ancient statue of Caesar (Figure 5.14) and a digitally sculpted face of the asian actor to impersonate the Joker (Figure 5.1). For both supplemental target meshes, no dynamic models were available. Building the personalized template requires rigid reconstruction of the actor's face and interactive reference point selection in order to warp the generic template onto the reconstruction. This whole process takes approximately 5 minutes. For each actor we capture 26 different facial expressions (another 5 minutes) as described in Section 5.2.2 resulting in approximately 2000 frames. We track the deformation of the personalized template over all input frames (10 seconds per scan) and sample 200 shapes at regular intervals. These are then used to compute the reduced PCA bases which requires additional 5 minutes. The extracted 200 face meshes are also used for deformation transfer on an arbitrary target model to generate the entire set of target expressions (about 30 minutes). All results are obtained with a fixed set of parameters and no manual intervention

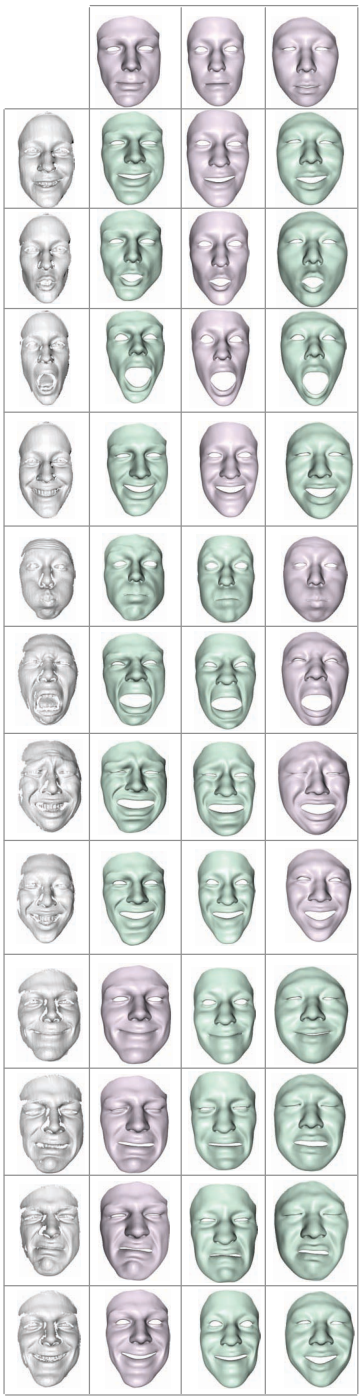


Figure 5.13: Real-time expression transfer: The tracked face is shown in magenta, the transferred expression in green.

as described in the previous sections. Once set up, the online system can run indefinitely for extended live puppetry performances. Figure 5.15 shows an evaluation of the accuracy of the online tracking algorithm for a typical sequence with considerable facial deformations. The maximum error between the online registered template and the noisy scans mostly vary between 2 and 4 mm, while the root-mean-square error lies below 0.5 mm.

Figure 5.14: Bringing an ancient Roman statue to live. The actor (magenta) can control the face of Caesar (green) that has been extracted from a laser scan of the statue.

As illustrated in Figures 5.1 and 5.13, expression transfer achieves plausible facial expressions even though the target face geometries can differ substantially. Especially the facial dynamics are convincingly captured, which is best appreciated in the accompanying video. Note that all expression transfers are created with a single 3D mesh of the target face. No physical model, animation controls, or additional example shapes are used or required to create the animations.

Limitations. Our tracking algorithm is based on the assumption that the acquisition rate is sufficiently high relative to the motion of the scanned actor. Very fast motion or large occlusions can lead to acquisition artifacts that yield inaccurate tracking results.

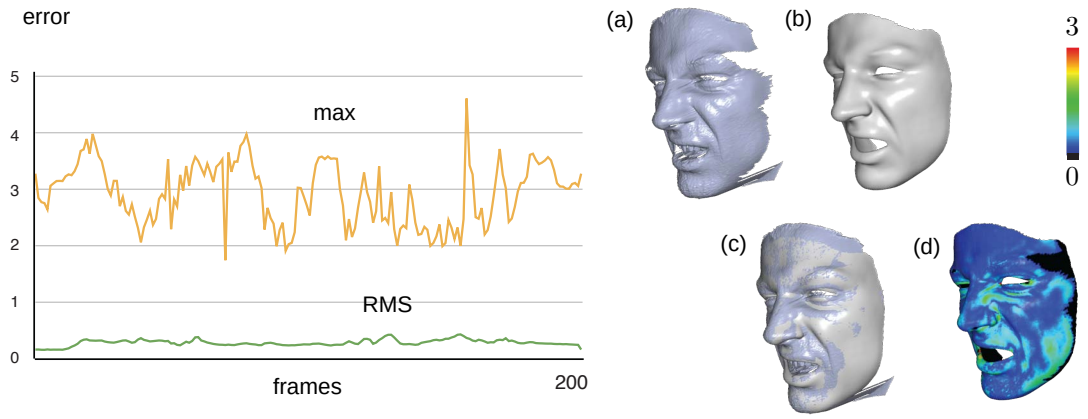


Figure 5.15: Online tracking accuracy for a sequence of 200 frames of a speaking actor. The graph shows the maximum (MAX) and root-mean-square (RMS) distance between the input scans and the warped template. On the right we show the comparison between the scan (a) and corresponding template (b) that differs the most in the entire sequence. Their overlap is shown in (c) and the distance for each vertex is visualized in (d), where black denotes a hole region. Error measurements are in mm.

However, as Figure 5.16 illustrates, our system quickly recovers from these inaccuracies. Since online tracking can be achieved real-time, slight matching inaccuracies between the input scans and the template as illustrated in Figure 5.15 are visually not apparent.

Our system does not capture all aspects of a real-live facial performance. For example, we do not explicitly track eyes, or represent the tongue or teeth of the actor. Similarly, secondary effects such as hair motion are not modeled in our system due to the substantial computation overhead that currently prevents real-time computations in the context of facial puppetry.

Facial expressions that are not recorded in the pre-processing step are in general not reproduced accurately in the online stage (Figure 5.16 right). This general limitation of our reduced linear model is mitigated to some extent by our face segmentation that can handle missing asymmetric expression. Nevertheless, high-quality results commonly require more than one hundred reconstructed scans to build an expression model that covers a wide variety of expressions suitable for online tracking. Fortunately, a 5-minute recording session per actor is typically sufficient, since the expression model can be reconstructed offline from a continuous stream of input scans.

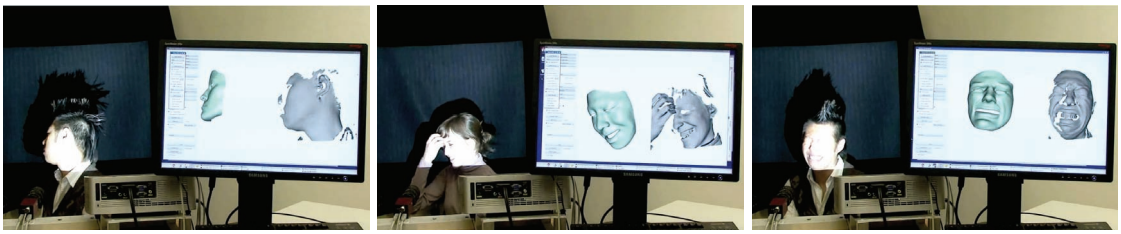


Figure 5.16: The online tracking algorithm robustly handles difficult cases such as poses where the actor faces away from the camera (left), or occlusions that invalidate parts of the scan (middle). If the actor's expression is substantially different than any of the training samples, a plausible, but not necessarily accurate reconstruction is created (right). The gray image on the screen shows the acquired depth map, the green rendering is the reconstructed expression transferred to a different face.

5.2.5 Discussion.

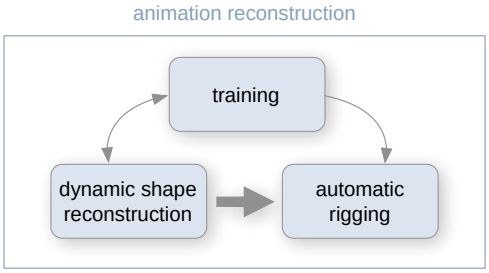
Our system is the first markerless live puppetry system using a real-time 3D scanner. We have demonstrated that high-quality real-time facial expression capture and transfer is possible without costly studio infrastructure, face markers, or extensive user assistance. Markerless acquisition, robust tracking and transfer algorithms, and the simplicity of the hardware setup, are crucial factors that make our tool readily deployable in practical applications. In future work we may wish to enrich this system with a number of components that would increase the realism of the results. Realistic modeling of eyes, tongue, teeth, and hair, are challenging future tasks, in particular in the context of real-time puppetry.

6 Directable Facial Animation

To specify how input motion deforms a surface, we have introduced non-rigid registration for general dynamic shape reconstruction. For real-time facial animation, we have proposed dimension reduction of linear face models. While we now have unsupervised ways to recreate compelling dynamic shapes, it is not immediately clear how to intuitively manipulate the resulting animation.

In character animation, an established technique for *instrumenting* realistic facial models is to use blendshape *rigs* where a set of controls are used to specify individual expressions. Since they are art-directable, blendshape parameterizations are often used for retargeting detailed recordings of facial performances to digital faces that differ strongly from the source model. For instance, an artist has maximum control over the appearance of wrinkles and folds for a particular facial pose, as opposed to physics-based muscle rigs. However, hundreds of separately sculpted shapes are typically needed to achieve realism. The ability to both efficiently generate a complete customized facial rig and automatically adjust blendshapes to match the specific look of the actor's expressions (while retaining the controller semantics) is thus an important asset for the artist.

We consider automatic rigging as an essential stage of the animation reconstruction pipeline (c.f. diagram on the right). Analogous to dynamic shape reconstruction, customizing a rig to a specific person can be achieved through training.



This chapter introduces a framework [LWP10] that automatically creates optimal blendshapes from a set of example poses of a digital face model (c.f. Figure 6.1). A predefined blendshape rig of a generic face is used as a prior to determine the semantics of each blendshape expression that we solve for. While in a traditional setting a precise pose needs to be provided for every blendshape, we only require a reduced set of example poses.

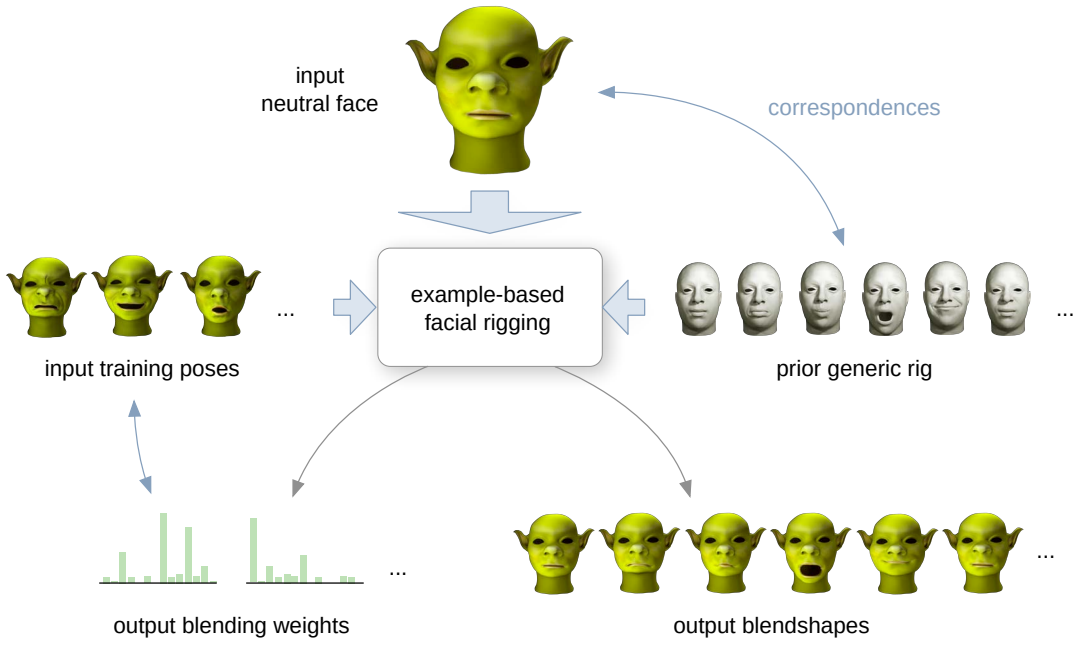


Figure 6.1: Example-based facial rigging. Given an arbitrary input model in neutral pose and a small set of training poses, our system produces a full set of output blendshapes with meaningful semantics derived from a generic prior rig. In addition the blending weights of the corresponding training poses can be determined.

We illustrate the versatility of our system with applications to art-directable rigging for sculpted virtual characters and automated rigging from 3D scans of real actors. A key aspect of our approach is that the blendshape reconstruction can be edited and adapted iteratively by either generating additional training expressions or adapting the blending weights of the example poses. This provides full control over the resulting blendshape model and facilitates easy integration into existing workflows such as facial tracking. Without our technique, an artist would have to adapt each blendshape to match all desired input examples.

6.1 Related Work

A large variety of different methods for facial rigging have been proposed in the past. Some are based on skeletons and joints [MTLT88], physically-based muscle models [Wat87], linear blendshapes [BL85], or combinations thereof. Skeleton-based rigs are most often employed for full-body animation due to their intuitive control for articulated motion. While skeletons are often used for face animation of cartoon characters, this approach is less suited to produce detailed facial expressions that exhibit wrinkles and folds. Automatic rigging using skeleton-embedding was proposed by [BP07], but with the focus on full-body animation.

Physically-based muscle models are well suited for creating realistic expression dynamics and secondary motions [SNF05]. However, artistic control can be difficult to achieve. [TW90] proposed a semi-automatic rigging of a muscle-based model to image data. Similarly, [KHpS01] fit a complex anatomical model to partial 3D scan data. [OZS08] introduced a general rigging method by transferring a generic facial rig to 3D input scans or hand-crafted models.

Linear blendshape models [BL85] provide a good compromise between realism and control. However, hundreds of blendshapes are usually necessary to capture realistic facial expression and are often used to mimic the effect of facial muscle groups as described by Ekman's *Facial Action Coding System* [EF78]. In particular, *FACS* decomposes facial behavior into 46 basic poses which are often complemented with a multitude of combined expressions and visemes. Building such a linear facial rig for highly realistic animation was recently demonstrated by [ARL+09], though each blendshape was still hand-crafted by animators. [PHL+98] build a rig automatically from photographs, and, similarly, [ZSCS04, WLGP09] from 3D scan data where all facial expressions are required as input. Automatic creation of facial models using (multi-) linear PCA models

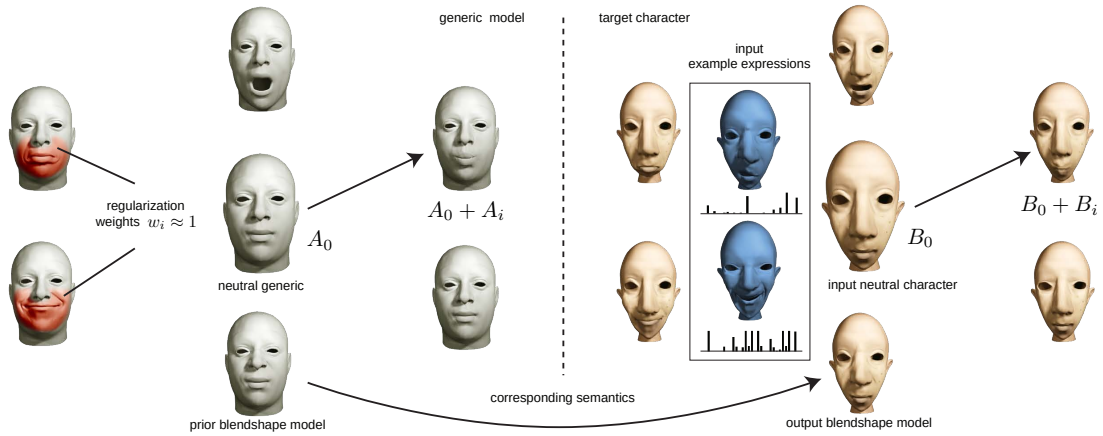


Figure 6.2: Conceptual overview of our method. The generic template, illustrated with a subset of the blendshapes (left), serves as a geometry and motion prior for an actor-specific blendshape model (right). The optimization solves for the target blendshapes such that a set of example expressions are best reproduced while maintaining the semantic correspondence between template and target models. We ensure semantically correct transfer of expressions using additional per vertex regularization weights in our optimization (shown in red).

was proposed in [BV99, BBPV03, VBPP05], though the resulting linear blendshapes are not necessarily meaningful for facial animation control. To circumvent this problem, we propose to use a predefined generic blendshape rig as a semantic prior. This has the benefit that only a subset of expressions is sufficient to build a complete model, and moreover, the resulting blendshapes match the controller semantics of the prior.

Linear blendshape models are especially suited for retargeting [Chu04]. An overview of current methods is given by [PL06]. Choe and Ko [CK05] proposed optimizing a generic predefined blendshape rig to fit sparse motion capture data of an actor. Liu et al. [LMX+08] extended this method using expression cloning [NN01] as a prior to handle under-constrained cases where less training data is available than the number of blendshapes. In this work, we focus on building a blendshape model that has the same semantics as the input model. This is achieved by formulating the optimization problem in deformation gradient space [SP04].

6.2 Example-Based Facial Rigging

We propose an interleaved optimization that refines the blending weights and solves for the optimal blendshapes in two alternating steps. We regularize the optimiza-

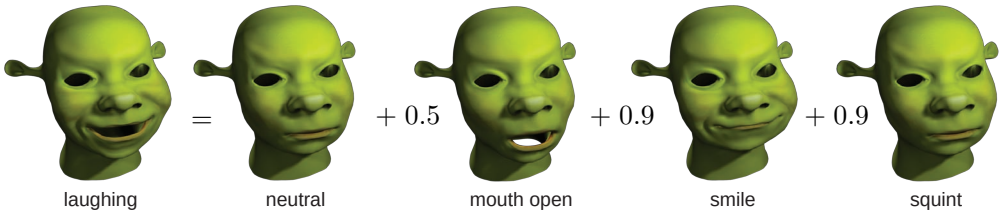
tion with meaningful blendshape expressions transferred from a generic face to accurately capture both the example poses and the semantics of the individual blendshapes. Expression transfer is achieved by mapping the deformation gradients between the neutral and blendshape pose triangles of the generic face to the target mesh triangles. Since we optimize for all blendshapes simultaneously, weighting local variations between the blendshapes in neutral and deformed pose is crucial in the regularization to prevent semantically incorrect blendshapes. We introduce an optimization that operates directly in gradient space (c.f. Section 3.2.3) in order to efficiently solve for blendshapes with semantics that corresponds to those of a generic facial rig prior.

When manipulating or fine-tuning blendshape sliders, artists often impose *activation* constraints to disallow pairs of blendshapes to simultaneously contribute to a pose. For instance, a mouth which lies exactly on the reflective symmetry plane of the face is often constrained to not squeeze to the left and to the right at the same time. The optimization for the blending weights is therefore only allowed to set a weight for either the left or right mouth squeezing blendshape. In order to prevent combinatorial explosion for the optimal solution, We design a continuous formulation of the objective function to efficiently handle the resulting non-linear constraints, while enforcing the weights to be positive.

6.2.1 Bi-Linear Optimization

We address the problem of generating a full set of blendshapes from a user-provided handcrafted character or scanned 3D model in neutral expression. The user can specify an arbitrary number of additional expressions to refine the model toward the specific geometry and motion characteristics of the actor. The algorithm then determines the optimal blendshapes that best reproduce the input examples, while preserving the controller semantics by matching the deformation gradients of a generic blendshape model rig (c.f. Figure6.2). For complex input expressions, such as an angry face, it can be difficult to determine the blending weights for a given example pose and those values can vary substantially for different characters. Thus, in addition to computing the optimal blendshapes, we also solve for blending weights given a rough initial guess provided by the user.

We consider a generic blendshape model as a set of meshes $\mathcal{A} = \{A_0, \ldots, A_n\}$, where A_0 is the rest pose and the A_i, $i > 0$ are additive displacements. Expressions can be generated as $T_j = A_0 + \sum_{i=1}^{n} \alpha_{ij} A_i$, where α_{ij} are the blending weights of pose T_j.



Our method is general in the sense that we can process input from various sources. In the case of 3D scans, we align the generic rest pose A_0 to the input shapes using the non-rigid registration method [LAGP09] presented in Section 3.5. This produces a set $\mathcal{S} = \{S_1, \ldots, S_m\}$ of complete meshes with connectivity of the prior model and shape of the respective scan. For hand-crafted models, we either perform the same registration operation or directly sculpt from the rest pose. We call these meshes *training poses*.

Our aim is to compute a new blendshape model $\mathcal{B} = \{B_0, \ldots, B_n\}$ that matches the geometry and motion of the actor. Thus we need to find blendshapes B_i and corresponding weights α_{ij} such that the training poses are faithfully reproduced, i.e., $S_j \approx B_0 + \sum_{i=1}^{n} \alpha_{ij} B_i$. To solve this *bi-linear* problem, we need to address two main challenges: Firstly, how can we compute the target blendshapes B_i, if only very few training poses are given, i.e., when the problem is under-constrained ($m < n$)? And secondly, how can we achieve the right controller semantics, i.e., ensure that similar weight settings lead to semantically similar expressions for both the template and the target blendshape models?

Our solution proceeds iteratively by alternating between two steps: step A keeps the blending weights α_{ij} fixed and optimizes for the blendshapes, while step B keeps blendshapes fixed and solves for the optimal weights (c.f. Figure 6.3). As an important means of control, the user establishes a semantic correspondence between each training pose S_j and the generic template. For this purpose, the user selects appropriate blending weights on the template to model a pose T_j that roughly corresponds to the training pose S_j. This yields (approximate) weights α^*_{ij} that provide initial values for step A of the optimization and semantic constraints for step B. We show in Section 6.2.2 that the α^*_{ij} can be intuitively determined by the user and do not need to be very accurate. Typically, the blending weights of only a few but sufficiently expressive poses (usually not more than 4) need to be manually activated in the beginning for each training pose.

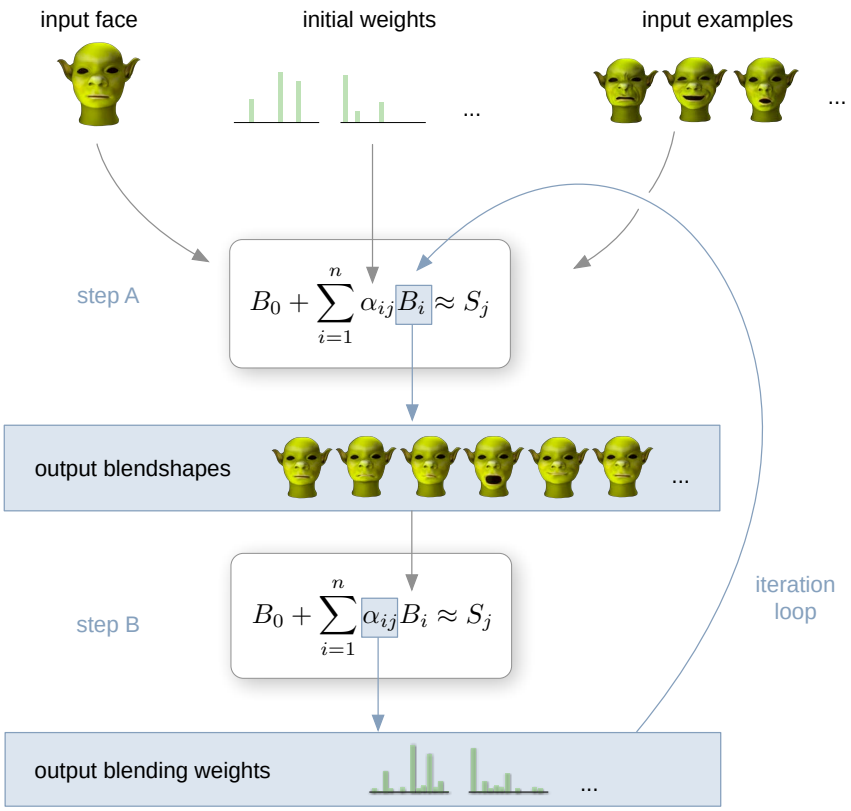


Figure 6.3: Our bilinear problem is solved using a decoupled optimization approach. We repeatedly alternate between step A and step B.

A: Optimizing Blendshapes.

To be able to reconstruct target blendshape models from few training poses, we incorporate additional constraints derived from the expression space of the template. The idea is to preserve the motion characteristics of the template by mapping the relative change between rest pose and blendshapes from the template to the target. This relative change can be encoded effectively using the deformation gradients defined in Section 3.2.3. For a triangle t with vertices $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, we define a non-orthogonal local frame as the 3×3 matrix $\mathbf{M}_t = [\mathbf{v}_3 - \mathbf{v}_1, \mathbf{v}_2 - \mathbf{v}_1, \mathbf{n}]$, where $\mathbf{n} = (\mathbf{v}_3 - \mathbf{v}_1) \times (\mathbf{v}_2 - \mathbf{v}_1)$ is the triangle normal vector. The deformation gradient that maps a source triangle s to a target triangle t is then given as $\mathbf{G}_{s \to t} = \mathbf{M}_t \cdot \mathbf{M}_s^{-1}$.

One of the key insights of this paper is that we can formulate the blendshape optimization in gradient space and reconstruct the final blendshapes from the local triangle

frames. As we show in Section 6.2.2, this leads to significant improvements compared to a direct optimization of blendshape vertex positions. Since the following optimization is performed independently for each triangle, we omit triangle indices and write e.g., $\mathbf{M}_i^B$ for the (unknown) frame of each triangle in blendshape B_i.

For the actor's rest pose B_0 and each of the training poses S_j, we can compute the frames $\mathbf{M}_0^B$ and $\mathbf{M}_j^S$, respectively. To faithfully reproduce the training poses, we define the fitting energy

$$E_{\text{fit}} = \|\mathbf{M}_j^S - (\mathbf{M}_0^B + \sum_{i=1}^{n} \alpha_{ij}\mathbf{M}_i^B)\|_F^2$$

which measures the deviation of the training poses S_j in the space of triangle frames from the best possible reconstruction in the unknown blendshape model. To account for insufficient training data we postulate that the deformation gradients of actor blendshapes B_i and template blendshapes A_i should be similar. Since the A_i and B_i for $i > 0$ are additive displacements, this means that $\mathbf{G}_{B_0 \to B_0+B_i} \approx \mathbf{G}_{A_0 \to A_0+A_i}$. We can write $\mathbf{G}_{B_0 \to B_0+B_i} = (\mathbf{M}_0^B + \mathbf{M}_i^B)(\mathbf{M}_0^B)^{-1}$ and define the regularization energy as

$$E_{\text{reg}} = \sum_{i=1}^{n} w_i \|\mathbf{M}_i^B - \mathbf{M}_i^{A*}\|_F^2$$

where the $\mathbf{M}_i^{A*} := \mathbf{G}_{A_0 \to A_0+A_i} \cdot \mathbf{M}_0^B - \mathbf{M}_0^B$ can be computed from the template blendshapes and the target rest pose. We incorporate additional regularization weights w_i as an essential means for maintaining the semantics of the generic prior. If a triangle of the template blendshape moves a little or not at all, we want to ensure that the same holds for the reconstructed target blendshape. However, if the template blendshape exhibits a strong motion, we want to allow the target deformation gradients to deviate more from the template prior to account for geometric and motion differences of the two characters. Our experiments showed that evaluating the regularization weights as $w_i = ((1 + \|\mathbf{M}_i^A\|_F)/(\kappa + \|\mathbf{M}_i^A\|_F))^\theta$ with $\kappa = 0.1$ and $\theta \geq 1$ adequately guides the optimization toward these semantics. We use $\theta = 2$ for all our results, yet similar results are obtained with other values. Note that constraining the vertices using the regularization weights does not limit the range of expressions for the target character, since other complementary blendshapes will be activated by the optimization to achieve a specific expression.

We combine both energy terms to yield the global energy $E_{\text{A}} = E_{\text{fit}} + \beta E_{\text{reg}}$, where β is a parameter that allows balancing fitting and regularization. Due to the

cross-product in the definition of the normal vector that constitutes the third column in the matrix $\mathbf{M}_i^B$, the energy E_A is non-linear in the vertex positions. Fortunately, as shown in [BSPG06], we can safely *ignore* the normal component for the reconstruction and only solve for the linear components, i.e., the first two columns of the matrices $\mathbf{M}_i^B$. Thus, minimizing E_A amounts to simply solving a linear system. Given the $\mathbf{M}_i^B$, we can reconstruct the vertex positions of each blendshape by solving a Poisson equation as described in Section 3.2.3. To prevent undesirable drifting, we constrain all vertices that are stationary in a template blendshape to remain fixed in the corresponding target blendshapes as well.

B: Optimizing Weights.

Given the computed set $\mathcal{B}$ of blendshapes, we can solve for the optimal weights α_{ij} to reconstruct the training poses S_j using least-squares fitting. We include the user-specified weights α_{ij}^* as soft constraints and define the energy E_B as a function of the unknowns α_{ij} as

$$E_\mathrm{B} = \sum_{k=1}^{N} \|\mathbf{v}_k^{S_j} - (\mathbf{v}_k^{B_0} + \sum_{i=1}^{n} \alpha_{ij}\mathbf{v}_k^{B_i})\|_2^2 + \gamma \sum_{i=1}^{n} (\alpha_{ij} - \alpha_{ij}^*)^2$$

where $\mathbf{v}_k^{S_j}$ and $\mathbf{v}_k^{B_i}$ are the vertices of the training pose S_j resp. the blendshapes B_i, and N is the total number of vertices. The parameter γ balances fitting and regularization. Note that even for $\gamma = 0$, the resulting weights are likely to match the semantics of the template controllers, since the regularization energy E_reg of the blendshape optimization couples corresponding template and target blendshapes. However, the weights α_{ij}^* allow the user to adapt the controller semantics and thus control the resulting expression space.

Since blendshape weights are typically constrained between zero and one, we use quadratic programming to solve the constrained system. Moreover, when manipulating or fine-tuning blendshape sliders, artists often impose *activation* constraints to disallow pairs of blendshapes to simultaneously contribute to a pose. For instance, a mouth which lies exactly on the reflective symmetry plane of the face is often constrained to not squeeze to the left and to the right at the same time. This can be formulated as non-linear constraints of the form $\alpha_{ij}\alpha_{kj} = 0$ for two mutually exclusive blendshapes B_i and B_k. We replace these non-linear constraints by corresponding non-linear penalty terms and apply a second optimization to update the blending weights α_{ij} using a solver for non-linear least-squares problems with linear constraints [CL96a].

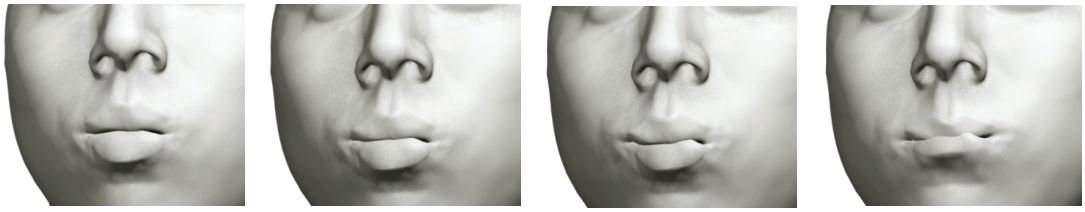


Figure 6.4: When the initial blending weights are perturbed by $\pm x$, the fitting quality using the optimized blendshape model start to decrease when $|x| > 0.3$.

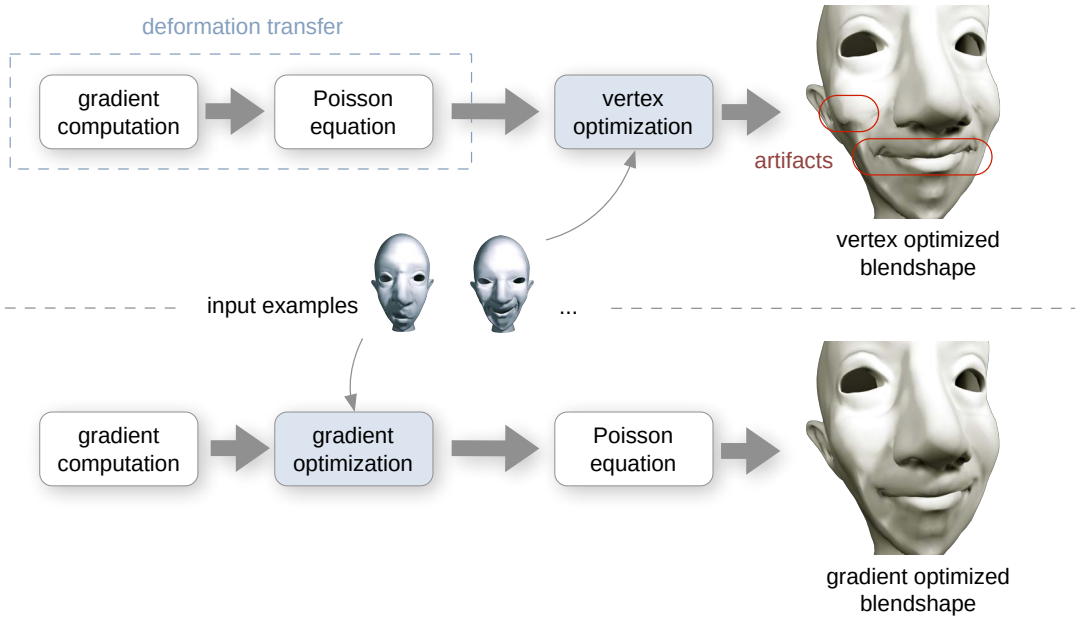


Figure 6.5: Comparison between vertex and gradient domain optimization.

6.2.2 Results

To evaluate our method, we created two sets of training poses using geometric modeling tools and two sets using a 3D scanner [WLG07]. The generic template model $\mathcal{A}$ is taken from the book *Stop Staring* [Osi07] and consists of 11K vertices, which is considered high for artists to start sculpting from. While our model is triangulated from a subdivision quad mesh, we still retain a one-to-one correspondence between our mesh and the initial model. The facial rig includes 29 different blendshapes with 6 pairs of modes that must not be activated simultaneously. For 14 training poses, our unoptimized implementation requires 45 seconds per iteration. Approximately equal computation time is

spent on blendshape optimization, reconstruction, and alpha optimization respectively.

In a typical setting, the artist mainly controls the parameters β and γ to adjust the output blendshapes. When $\beta \gg 1$ the resulting blendshape model is close to the results achieved via pure deformation transfer. In this case, even when $\gamma = 0$, no visible artifacts were observed in any of our examples. When β is closer to 0.1, the resulting blendshape is able to accurately capture the input examples, but its quality can be sensitive for $\beta \ll 0.1$. In particular, when $\gamma = 0$ some artifacts can appear for some blendshapes, but these are prevented when γ is large enough. For all our results, we simply apply 10 iterations of alternating blendshape and weight optimizations, with $\beta = 0.5$ and $\gamma = 1000$ for the first iteration. The weights are gradually decreased to $\beta = 0.1$ and $\gamma = 100$ in the last iteration. Weight scheduling ensures robustness to local minima while enabling detailed adaption to the input after optimization.

Our optimization is robust to variations in the initial selection of the blending weights α^*_{ij}. We perturbed the user-provided initial values by randomly adding a value between $-x$ and x. Up to $|x| = 0.3$, we did not obtain any noticeable differences in the reconstructed blendshapes for all examples. Figure 6.4 shows the impact of increasing variations of random α^*_{ij} when fitting the kiss expression with the optimized blendshape model.

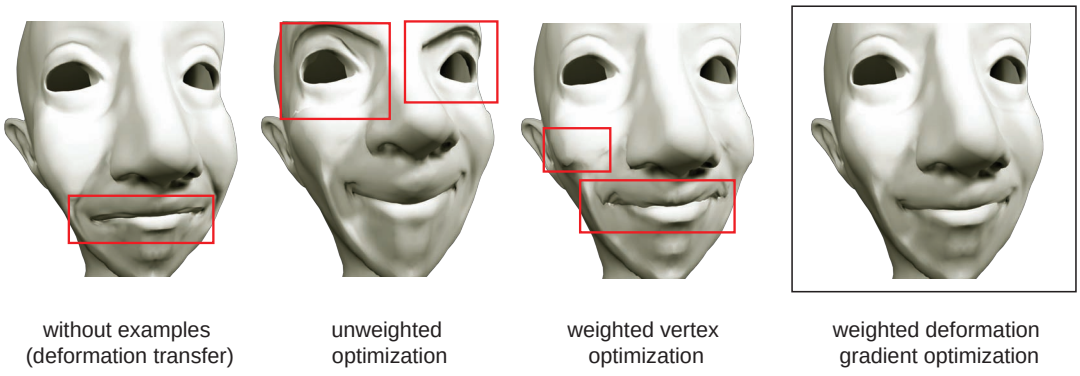


Figure 6.6: Different blendshape optimization methods. Without training data, the reconstructed blendshapes correspond to pure deformation transfer. Without weighting, undesirable mixing of blendshape modes occurs, noticeable in the motion of the eye brows in the smile blendshape. The optimization formulated in vertex space leads to visible artifacts, while our approach avoids these errors and achieves the desired semantic separation of blendshapes.

Figure 6.6 demonstrates the importance of the weights w_i in the regularization energy E_{reg}. Without weighting, the optimization creates a combination of semantically separate blendshapes, i.e., mixes undesirable eyebrow motion into the smile blendshape. However, using a weighted optimization when solving for the vertex positions directly leads to artifacts, as each vertex is considered independently. These artifacts are absent in our method as the optimization of the deformation gradient is followed by a subsequent blendshape reconstruction step.

Figure 6.8 illustrates the influence of the parameter β on the blendshape energy E_{A}. While the fitting improves with decreasing weight, at around $\beta = 0.05$ over-fitting occurs that leads to artifacts in the reconstructed blendshapes. We found that $\beta = 0.1$ is a good compromise between accuracy and robustness for both 3D scan data and hand-crafted models.

Figure 6.9 provides qualitative results for three complex expressions of two cartoon characters (see also Figure 4.18) and one model derived from 3D scans. Without training examples our method effectively performs deformation transfer on the blendshapes, which results in expressions that mimic the poses of the generic template. With more training examples, the expressions adapt closer to the characteristics of the target model while still conforming to the same controller semantics. For instance, our method automatically includes the wrinkles of the *joe* model that appear in the training examples.

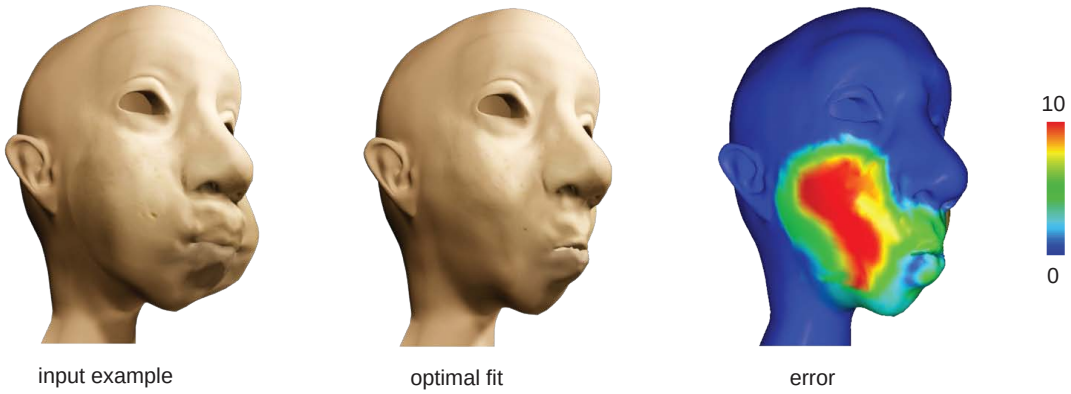


Figure 6.7: Not all training poses can be expressed by the reconstructed blendshape model for the given semantics. In such cases, additional blendshapes are required in the prior to introduce more degrees of freedom.

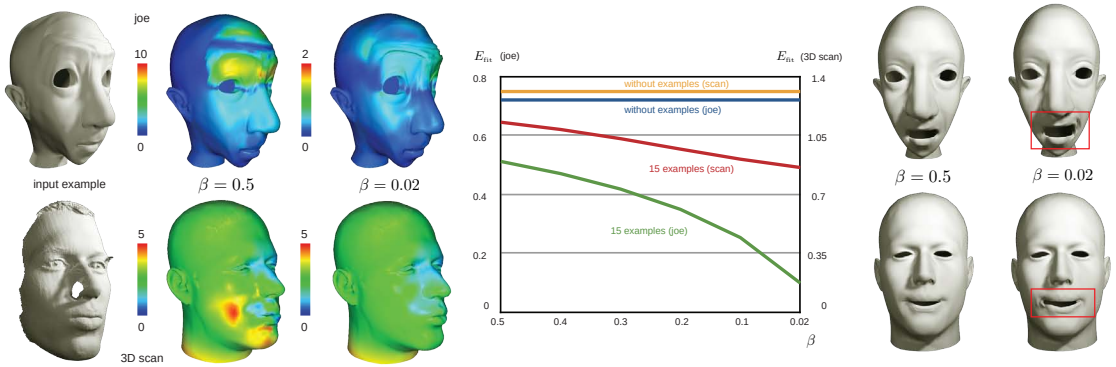


Figure 6.8: Influence of the parameter β that balances fitting term E_{fit} (in mm) and regularization term E_{reg}. Effectively, β controls the relation between deformation transfer from the generic template and example poses provided by the user. Over-fitting can occur when β is too small.

We also tested our algorithm in the context of markerless facial tracking (Figure 6.10) by using a generic model that contains all 46 FACS poses and 28 supplemental expressions [Sta10]. We used the system described in [WLGP09] and replaced the PCA model with our optimized rig. Now our approach enables artists to intuitively tweak blending weights after tracking. Also, our technique demonstrates that very few training poses (17) are sufficient to accurately express a dense facial expression space. Without examples, the blendshapes are not expressive enough.

Limitations. Our method assumes training examples to semantically correspond to valid blendshape combinations of the generic rig. The *blow* expression in Figure 6.7 cannot be represented by the prior model and therefore the optimization fails to fit this expression. However, we can easily detect the case when poses are missing in the generic model by verifying if E_{fit} exceeds a certain threshold. Semantically differing expressions would thus need to be added as additional blendshapes. Currently, the algorithm is not fast enough for interactive rates. The algorithmic complexity scales linearly with the number of training examples and mesh vertices, but the non-linear solver has cubic complexity in the number of blendshapes. More sophisticated solvers and an optimized GPU implementation may allow artists to get direct feedback on the facial rig while sculpting the example expressions. When only using a few example expressions, only those blendshapes are being optimized that influence these expressions. Every other created blendshape will look like deformation transfer, which creates plausible deformations, but may not catch the exact expressions of the character.

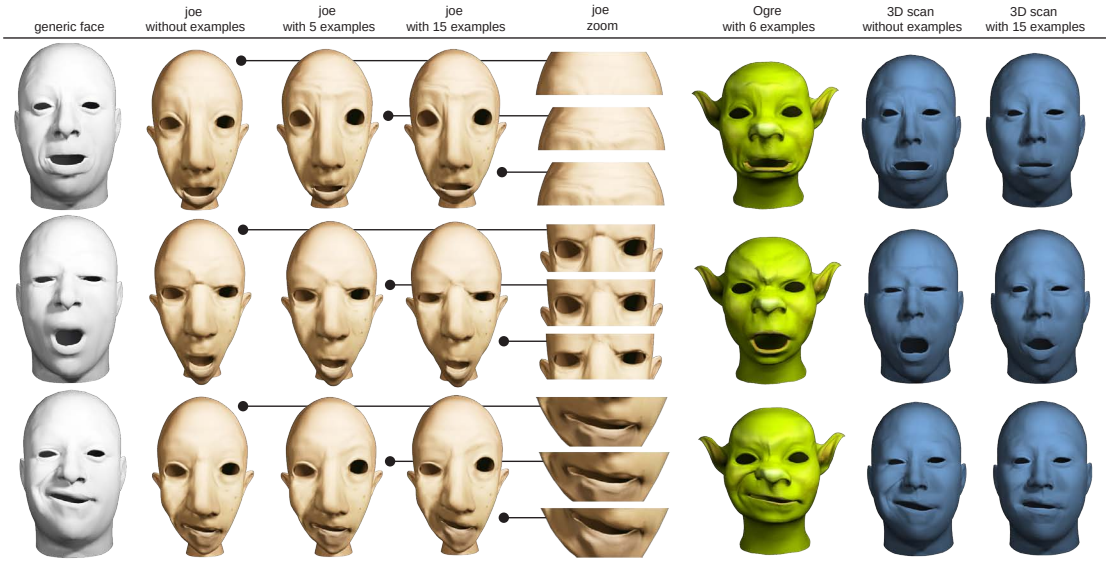


Figure 6.9: Without training examples, our method simply transfers the expression dynamics of the generic face toward the actor. With more training expressions, the reconstructed blendshape model adapts toward the geometry and motion characteristics of the actor.

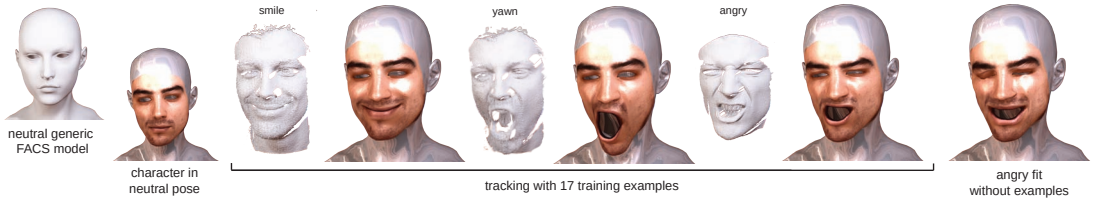


Figure 6.10: Art-directable facial tracking. Our optimized blendshapes can be directly used for facial tracking by solving for optimal blending weights for each frame of a 3D scan sequence. Compared to PCA approaches, our method allows intuitive control over the blending weights.

6.2.3 Discussion

Example-based facial blendshape rigging is intended to increase the productivity of professional artists and allow even inexperienced users to quickly generate actor-specific blendshape models. Instead of solely relying on sculpting and fine-tuning every single blendshape to match the intended expressions of an animation, our method only requires a small subset of these expression. This enables a scalable design process and effective reuse of existing rigs.

Key contribution of this algorithm is the formulation of the blendshape optimization in gradient space. In combination with appropriate weighting schemes, we obtain a consistent integration of expression transfer and reconstruction of example poses to yield high-quality customized blendshape rigs with pre-defined controller semantics.

In the future, we may wish to investigate how combining multiple template models improves expression transfer by providing a more general prior that can more closely adapt to the target model. In order to handle input examples that cannot be expressed by a given generic template model due to insufficient blendshapes, we may also consider methods that suggest supplemental expressions of the generic model that can maximize orthogonality.

The good ideas will survive!

—Quentin Jerome Tarantino

7

Conclusion and Future Directions

We have presented robust reconstruction algorithms that are aimed at facilitating the creation of compelling CG animations through the use of incomplete but densely sampled real-time data. We have focussed our investigation on general deformable surfaces which includes human performances, dynamic cloths, and faces in a markerless acquisition setting. Special emphasis was put on recurring themes such as the fundamental problems of non-rigid registration and correspondence computation.

Our distinct non-rigid registration approach unifies the computation of correspondences, the identification of common subsets, and global consistent warps through time. Earlier instances of multi-frame static surface reconstruction and shape completion were extended and generalized to the dynamic setting using a novel bi-resolution framework. We applied our algorithms to the field of facial animation where new tools for real-time expression transfer and intuitive instrumentation were further developed. This chapter summarizes our work and discoveries, then suggests several directions for future explorations.

7.1 Summary and Take-Home Messages

Fine-Scale Dynamics: No Free Lunch! Real-time markerless capture is inherently limited by the lack of explicit correspondence information. Moreover, it is impossible, like any other optical acquisition technology, to instantaneously obtain a complete digital representation of the scanned subject. These obstacles prevent us from an easy access to valuable fine-scale shape dynamics which are often difficult to model using a traditional approach such as physical simulation. Simply take as an example the challenge of accurately modeling the physical properties and complex interactions of a realistic human face. Even though existing prior models can be, in some cases, reused and adapted to specific input data, the process is still subject to a refinement procedure requiring robust correspondence and deformation computations.

Correspondence and Deformation Coupling. Most of the existing works initialize the problem of correspondence computation based on proximity heuristics or invariances w.r.t. some local transformations. Generally, this strategy is further corrected by exploiting spatial consistency constraints through deformation modeling. Several researchers have identified the effectiveness of incorporating the notion of plausible (i.e., smooth) deformation in a registration framework to eliminate outlier correspondences and equalize inaccuracy. Instead of simply performing a regularized deformation with prescribed correspondences, our approach incorporates correspondence refinement directly into the continuous deformation optimization. In this way, we forcefully break the dependency between correspondence and deformation computation and naturally achieve warps with considerably higher global spatial consistency. To this end, when no high-level shape descriptors are involved, we show that our method can handle significantly larger deformations and holes in the scans than existing non-rigid registration techniques.

Local Rigidity Maximization and Stiffness Relaxation. Fine-scale details should be preserved and not neglected in deformable registration. From a geometric standpoint, the optimal non-rigid alignment between a pair of shapes should be regarded as the one that minimizes their distances and remains locally as-rigid-as-possible. Earlier work often considered deformation smoothness as a sufficient prerequisite for deformable alignment. This is appropriate when dense correspondences can be accurately determined. In an iterative optimization context, the positions of correspondences are repeatedly updated based on their local matches. Consequently, we argue, that for considerable deformations, details are best preserved when maximizing local rigidity. We identified the

embedded deformation framework as an ideal model for accurate feature locking under large deformations in the subject.

While stiffness reduction is a common practice for scheduling a coarse-to-fine optimization problem, previous works typically proposed a succession of pre-defined regularization values. This assumption is valid for specific scenarios but any new type of input subject is subject to a careful fine-tuning procedure. We proposed a systematic stiffness relaxation approach based on the rate of energy convergence within an iterative non-rigid registration framework. Consequently, our approach can handle a larger variability in the input deformation without further manual intervention. In particular, we used the same parameters for all our pairwise non-rigid registration cases.

Adaptive Deformation Model and Multi-Frame Detail Aggregation. Flexible deformation models introduce many local minima in the energy landscape, and when they can be avoided, they should be. Several non-rigid registration frameworks consider spatial adaption of regularization weights in the course of an iterative registration process. This strategy is reasonable for the segmentation of rigid components for articulated subjects provided correspondences are sufficiently accurate and a large number of degrees of freedom are available in the deformation model. We found that spatially adapting the degrees of freedom of the deformation model creates an important advantage for solving the non-rigid registration. Our extension of embedded deformation with a dynamic deformation graph showed that suboptimal local minima can be effectively avoided when new optimization variables are only introduced in relevant regions. At the same time, we still allow the manipulation of regularization weights as they are crucial for stiffness relaxation. As an additional advantage, our method remains efficient since redundant computation in rigid regions is avoided.

The difference between pure correspondence computation and non-rigid registration is that the latter additionally estimates the shape in hole regions. Our novel bi-resolution framework reconstructs both geometry and motion, for extended scan sequences—even when the data is captured from a single-view. Through the use of a coarse template (obtained from a static reconstruction procedure), we are not only able to provide a more robust geometric prior for largely unobserved regions, but we also circumvent the highly non-trivial problem of topology extraction. As opposed to past template-based methods, we intentionally remove geometric details from the template and treat them separately during reconstruction, in order to effectively distinguish be-

tween static and dynamic ones. A forward and backward aggregation process propagates those dynamic details across the entire recording while synthesizing them consistently in unobserved regions. As a result, our proposed framework is able to fully harness the high sampling resolution of our acquired input data for improved registration accuracy and synthesis of dynamic details in occlusions.

Shape Completion of Topology Varying Data. Many subjects (e.g., gliding cloth on human body) cannot be faultlessly represented by a single connected two-manifold template model. To avoid the hassle of explicitly recovering these highly complex structures in this ill-posed setting, we took the first step of completing sequences of incomplete data while the subject may undergo complicated topology changes. We extended previous hole-filling algorithms (for static shapes) with state-of-the art non-rigid registration techniques to produce temporally coherent data which deformations are temporally coherent and compatible with the observed data. Furthermore, our system is resistant to error accumulations since correspondence computations are localized within a time window. We demonstrated a first prototype system for generating free-viewpoint video using watertight and temporally coherent geometries of human performance that are acquired using a multi-view 3D scanner.

Real-time Facial Animation. Our preceding contributions have impacted the field of facial animation. We further investigated critical components for efficient and robust tracking of compelling facial models and the expression transfer to arbitrary CG characters. In this respect, we have presented the first markerless live facial puppetry system using a real-time 3D scanner. One key factor for achieving real-time performance is the shift of costly computations to an off-line preprocessing stage. The expensive tasks include a collection of face-specific tracking techniques (rigid head alignment, separation between facial deformation and rigid chin motion, model-based optical flow using color textures and lip segmentations), deformation transfer, and the computation of linear subspaces for dimension reduction. While PCA dimension reduction is a well-known technique for achieving efficient tracking, we extended the idea of dimension reduction for expression transfer. In particular, we have shown how to construct a linear subspace for target expressions whose semantics are compatible with the source model by a simple linear least-squares optimization.

Blendshape Rigging in Gradient Domain. Rigging is a critical bottleneck in any animated content creation pipeline. For facial animations, we propose an example-based

approach for the automatic generation of blendshape rigs. Blendshapes with semantic meanings are determined through the use of a generic prior model with the corresponding expressions. We identified the task as a bi-linear problem where blendshapes and their blending weights are simultaneously solved in an iterative and decoupled fashion. With only little training data, a full blendshape rig can be generated within minutes. We have successfully produced high quality customized blendshapes from handcrafted cartoon characters as well as facial scans from real actors (using FACS poses). Furthermore, we explored the true potential of our character specific rigs in the context of art-directable facial tracking using our real-time puppetry system. By simply replacing the dimension reduced PCA model with the generated blendshapes, we are now able to instrument meaningful expressions through user-intuitive controls. From an algorithmic perspective, we learned that a per-triangle optimization in gradient domain followed by globally solving a Poisson equation was a key factor for generating artifact free blendshapes.

Closing Remarks. The core algorithms herein were mainly designed to improve robustness and accuracy over existing work on animation reconstruction. Specifically for our robust non-rigid registration approach, the reader may be misled and attribute the whole difference between related methods and ours to the fact that different optimization parameters were chosen. Besides that most comparisons were conducted with the involvement of the original authors, we support our ideas with the right choice of deformation model and, foremost, the importance of tight coupling between correspondence and deformation optimization where global spatial consistency is enforced.

In our shape completion and facial animation system, the complex interplay of a several building blocks accounts for the robust and efficient tracking. Here, we emphasize on the conceptual ideas rather than specific implementation details. The algorithmic choice of several modules stem from a cautious engineering design process but is also based on whether or not they meet the desired deeds. We believe that higher quality results can be attained by simply replacing those low-level components with more sophisticated algorithms without violating the overall architecture of the two systems.

7.2 Open Problems and Future Directions

Accurate 3D digitization of deformable surfaces, including human performances and facial expressions, remains one of the biggest challenges in computer animation. Digitally cloning an actor realistically is still not possible without the intervention of skilled artists. The work herein has covered an important aspect of animation recon-

struction which is purely based on geometric measurements and assumptions. Perhaps the greatest limitations of such an approach is that we cannot fully prevent error accumulation for very long recordings and implausible shapes (self-intersections) to occur in large hole-regions.

While current data-driven techniques (which use a statistical database containing plausible poses) can effectively sidestep the problem of drifts, they are not suitable for all types of subjects. In particular, the number of necessary input shapes would be too high to realistically span a sufficiently large space for fitting to arbitrary input data, such as all degrees of freedom of a human performance (possibly wearing cloth). For general deformations, it is still unclear how to optimally combine these approaches with captured data and also, how to easily build such a database in the first place.

In reality, many complex behaviors such as (self-)collisions and secondary motions cannot be predicted by pre-computation. For instance, without a sophisticated physical model and simulation, it would be hard to derive the accurate geometry of the palm of a grasping hand. Unfortunately, building such model is still subject to a non-trivial and time-consuming manual process. At this point, physically-based simulation is still very difficult to control and computationally too expensive. While physics is often considered as part of a post-processing stage, it is still unclear how to integrate it in the problem formulation of dynamic shape reconstruction.

Another unsolved problem is the extraction of a consistent topology from a sequence of incomplete scans. While several attempts were made to address this problem, existing algorithms are still limited to very simple examples (very slow deformations and simple topology changes). Our two dynamic shape reconstruction frameworks basically circumvent an explicit topology estimation by providing a coarse template model (geometry and motion reconstruction) or ignoring the requirement of globally consistent correspondences for the whole motion (temporally-coherent shape completion). For the latter, a large number of shape analysis and manipulation tools cannot be applied.

Although we demonstrated the capabilities of current real-time and markerless acquisition systems for creating compelling facial animations, a faithful modeling of complex skin behavior that is indistinguishable from reality has not yet been achieved. In this respect, we are convinced that several problems still need to be solved in order to effectively cross the uncanny valley. For example, recreating complex non-linear deformations that arise around eye regions when a person squints is still a challenging topic

since a significant portion of the surface is occluded and self-colliding. Additionally, in the context of real-time puppetry, several critical components for realistic facial animations were not explored in our work. Those include dedicated modeling techniques and representations of eyes, tongues, and human hair. Finally, the two addressed problems, expression retargeting and facial rigging, require accurate shrink-wrapping between a generic prior model and a custom character. For extremely dissimilar shapes (different anatomy), this is not always easy to achieve and is subject to considerable manual work. To our knowledge, there is no general method at the moment that fully automates this process.

In a nutshell. We have investigated several new dynamic shape reconstruction avenues from a purely geometric standpoint and discussed the relevant state of the art methods. However, there is still no universal solution for correspondence computations between arbitrary shapes and drift-free reconstruction of complex deforming geometries, especially when the scanned subjects exhibit (self-)contacts.

Future Work. The research conducted in this dissertation opens new doors for future explorations and improvements. Several problem specific and low-level suggestions can be found at the end of each chapter. We now highlight a few promising and more general research directions.

In terms of dynamic shape reconstruction, an immediate extension might involve a directable physical simulation module as depicted in Figure 7.1. Because our geometric approach does not consider self-contacts and secondary motions in occluded regions (e.g., palm of a grasping hand), one might imagine that incorporating a physical simulation would result in more plausible deformations in those areas and even improve the accuracy of correspondences. However, as highlighted earlier, physical simulations are generally difficult to control and computationally costly. One possible option would be to extract physical properties from (reliable) visible regions and reapply these estimated parameters on the remaining parts of the surface. Ultimately, the reconstruction problem could be phrased as a global optimization that determines a deformation field which is represented by external forces coupled with the simulation of realistic material behaviors.

Additionally, we anticipate future work in further improving the robustness w.r.t. fast input motions and the avoidance of error accumulations. While traditional data-driven approaches or methods that involve a kinematic skeleton can reach a certain degree of pose invariance during non-rigid registration, their potentials are certainly

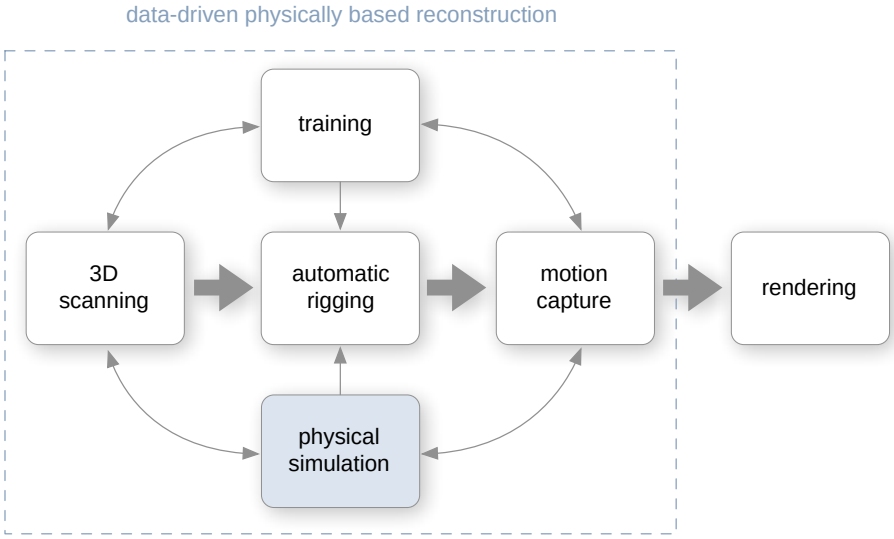


Figure 7.1: One promising direction for future research is to complement our animation reconstruction pipeline with a physically-based simulation component.

not fully unleashed. Recent advances in tracking techniques based on machine learning algorithms have demonstrated to be very agile in capturing a wide range of motions and exceptionally efficient to compute (exploiting parallelization). In this sense, one general way of improvement might be to unify the problem of correspondence estimation with a robust statistical model which should be easy to trained and adapt to specific subjects

So far, we mainly considered shrink-wrapping as a non-rigid registration problem where a source (generic) model is deformed to match a (customized) target model. When matching very dissimilar shapes, simply prescribing a smooth warp may no longer be a sufficient criterion. One could imagine extending present surface deformation models (which attempt to preserve details) to progressively adapt the shape of the source mesh to the target once the correspondences are sufficiently accurate. In particular, local geometries in the target model would be continuously transferred to the source (e.g. via differential coordinate-based representations).

On the acquisition side, we assessed our algorithms using data produced by two state-of-the-art dense real-time 3D scanning systems. For practical considerations in an everyday surrounding, these prototype systems are still limited by technological issues such as the projection of disturbing lights (structured light scanner) or costly studio setup (Light Stage 6). Nonetheless, the field of real-time 3D is likely to grow significantly in the next few years, thereby motivating further research problems including, geometry

and motion reconstruction from purely passive systems (under arbitrary illumination conditions), high-resolution facial tracking using a monocular system, and more...

Bibliography

[ACP03] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22:587–594, 2003.

[ACSD+03] Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Lévy, and Mathieu Desbrun. Anisotropic polygonal remeshing. *ACM Trans. Graph.*, 22(3):485–493, 2003.

[AHB87] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5):698–700, 1987.

[AKKS99] Mihael Ankerst, Gabi Kastenmüller, Hans-Peter Kriegel, and Thomas Seidl. 3d shape histograms for similarity search and classification in spatial databases. In *SSD '99: Proceedings of the 6th International Symposium on Advances in Spatial Databases*, pages 207–226, London, UK, 1999. Springer-Verlag.

[Ale01] Marc Alexa. Local control for mesh morphing. In *SMI '01: Proceedings of the International Conference on Shape Modeling & Applications*, page 209, Washington, DC, USA, 2001. IEEE Computer Society.

[Ale03] Marc Alexa. Differential coordinates for local mesh morphing and deformation. *The Visual Computer*, 19(2):105–114, 2003.

[AMCO08] D. Aiger, N. J. Mitra, and D. Cohen-Or. 4-points congruent sets for robust surface registration. *ACM Transactions on Graphics*, 27(3):#85, 1–10, 2008.

[AMD02] Pierre Alliez, Mark Meyer, and Mathieu Desbrun. Interactive geometry remeshing. *ACM Trans. Graph.*, 21(3):347–354, 2002.

[ARL+09] Oleg Alexander, Mike Rogers, William Lambeth, Matt Chiang, and Paul Debevec. The digital emily project: photoreal facial modeling and animation. In *SIGGRAPH '09 Courses*, 2009.

[Art] Artec. http://www.artec-group.com/.

[ARV07] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In *CVPR'07*, 2007.

[Asc] Ascension. http://ascension-tech.com/.

[ASK+05] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24:408–416, 2005.

[ASP+04] Dragomir Anguelov, Praveen Srinivasan, Hoi-Cheung Pang, Daphne Koller, Sebastian Thrun, and James Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Advances in Neural Inf. Proc. Systems 17*. 2004.

[ATD+08] Naveed Ahmed, Christian Theobalt, Petar Dobrev, Hans-Peter Seidel, and Sebastian Thrun. Robust fusion of dynamic shape and normal capture for high-quality reconstruction of time-varying geometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1–8, 2008.

[AVDI03] Pierre Alliez, Éric Colin de Verdière, Olivier Devillers, and Martin Isenburg. Isotropic surface remeshing. In *SMI '03: Proceedings of the Shape Modeling International 2003*, page 49, Washington, DC, USA, 2003. IEEE Computer Society.

[AW95] Volker Aurich and Jörg Weule. Non-linear gaussian filters performing edge preserving diffusion. In *Mustererkennung 1995, 17. DAGM-Symposium*, pages 538–545. Springer-Verlag, 1995.

[Bar84] Alan H. Barr. Global and local deformations of solid primitives. *SIGGRAPH Comput. Graph.*, 18(3):21–30, 1984.

[BBB+10] Thabo Beeler, Bernd Bickel, Paul Beardsley, Bob Sumner, and Markus Gross. High-quality single-shot capture of facial geometry. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 29(3), 2010.

[BBH03] Myron Z. Brown, Darius Burschka, and Gregory D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):993–1008, 2003.

[BBK06] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proc. National Academy of Sciences (PNAS)*, 103, 2006.

[BBPV03] Volker Blanz, Curzio Basso, Tomaso Poggio, and Thomas Vetter. Reanimating faces in images and video. In *EUROGRAPHICS '03*, 2003.

[BBPW04] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proceedings of the 8th European Conference on Computer Vision*, pages 25–36, 2004.

[BBV+10] P.-A. Blanche, A. Bablumian, R. Voorakaranam, C. Christenson, W. Lin, T. Gu, D. Flores, P. Wang, W.-Y. Hsieh, M. Kathaperumal, B. Rachwal,

O. Siddiqui, J. Thomas, R. A. Norwood, M. Yamamoto, and N. Peyghambarian. Holographic three-dimensional telepresence using large-area photorefractive polymer. *Nature*, 468:80–82, 2010.

[Bec94] Dominique Bechmann. Space deformation models survey. In *Computer Graphics Forum*, pages 571–586, 1994.

[Ben75] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.

[Ber98] R. Berthilsson. A statistical theory of shape. 1451, 1998.

[BHB00] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. *CVPR'02*, 2:2690, 2000.

[BHPS10] Derek Bradley, Wolfgang Heidrich, Tiberiu Popa, and Alla Sheffer. High resolution passive facial performance capture. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 29(3), 2010.

[BK03] Mario Botsch and Leif Koppelt. Multiresolution surface representation based on displacement volumes. *Computer Graphics Forum (Proceedings of Eurographics 2003)*, 22(3):483–491, 2003.

[BK04] Mario Botsch and Leif Kobbelt. A remeshing approach to multiresolution modeling. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 185–192, New York, NY, USA, 2004. ACM.

[BK05] Mario Botsch and Leif Kobbelt. Real-time shape editing using radial basis functions. In *Computer Graphics Forum*, pages 611–621, 2005.

[BL85] P. Bergeron and P. Lachapelle. Controlling facial expressions and body movements in the computer generated animated short 'Tony de Peltrie'. In *SIGGRAPH '85 Tutorial Notes, Advanced Computer Animation Course*. 1985.

[BLB+08] Bernd Bickel, Manuel Lang, Mario Botsch, Miguel A. Otaduy, and Markus Gross. Pose-space animation and transfer of facial details. In *Proc. of SCA'08*, 2008.

[BM92] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE PAMI*, 14:239–258, 1992.

[BMM00] Serge Belongie, Greg Mori, and Jitendra Malik. Matching with shape contexts. In *IEEE Workshop on Content-based access of Image and Video-Libraries*, page 20, 2000.

[Bot79] R. Bottema. *Theoretical Kinematics*. North Holland Publishing Company, New York, 1979.

[Bou08] J. Y. Bouguet. Camera calibration toolbox for matlab, 2008.

[BP03] Svetlana Barsky and Maria Petrou. The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1239–1252, 2003.

[BP07] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Trans. Graph.*, 26(3):72, 2007.

[BPGK06] Mario "Bierkasten" Botsch, Mark Pauly, Markus Gross, and Leif Kobbelt. Primo: coupled prisms for intuitive surface modeling. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 11–20, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[BPS+08] Derek Bradley, Tiberiu Popa, Alla Sheffer, Wolfgang Heidrich, and Tamy Boubekeur. Markerless garment capture. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–9, New York, NY, USA, 2008. ACM.

[BR04] Benedict Brown and Szymon Rusinkiewicz. Non-rigid range-scan alignment using thin-plate splines. In *Symp. on 3D Data Processing, Visualization, and Transmission*, 2004.

[BR07] Benedict J. Brown and Szymon Rusinkiewicz. Global non-rigid alignment of 3-d scans. *ACM Trans. Graph.*, 26:21, 2007.

[BS08] Mario Botsch and Olga Sorkine. On linear variational surface deformation methods. *IEEE Trans. on Visualization and Computer Graphics*, 14:213–230, 2008.

[BSPG06] Mario Botsch, Robert Sumner, Mark Pauly, and Markus Gross. Deformation transfer for detail-preserving surface editing. In *Vision, Modeling, Visualization 2006*, pages 357–364, 2006.

[BTVG08] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. *Comput. Vis. Image Underst.*, 10(3):346–359, 2008.

[BV99] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proc. SIGGRAPH '99*, 1999.

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[CB05] Erika Chuang and Christoph Bregler. Mood swings: expressive speech animation. *ACM Trans. Graph.*, 24(2):331–347, 2005.

[CEJ+06] Charles-Félix Chabert, Per Einarsson, Andrew Jones, Bruce Lamond, Wan-Chun Ma, Sebastian Sylwan, Tim Hawkins, and Paul Debevec. Relighting human locomotion with flowed reflectance fields. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches*, page 76, New York, NY, USA, 2006. ACM.

[CFB97] Jonathan C. Carr, W. Richard Fright, and Richard K. Beatson. Surface interpolation with radial basis functions for medical imaging. *IEEE Transactions on Medical Imaging*, 16:96–107, 1997.

[Chu04] E. Chuang. *Analysis, Synthesis, and Retargeting of Facial Expressions*. PhD thesis, Stanford University, 2004.

[CK01] Rodrigo L. Carceroni and Kiriakos N. Kutulakos. Multi-view scene capture by surfel sampling: From video streams to non-rigid 3d motion, shape reflectance. pages 60–67, 2001.

[CK05] Byoungwon Choe and Hyeong-Seok Ko. Analysis and synthesis of facial expressions with hand-generated muscle actuation basis. In *SIGGRAPH '05 Courses*, 2005.

[CKpS] Swen Campagna, Leif Kobbelt, and Hans peter Seidel. Directed edges - a scalable representation for triangle meshes. *Journal of Graphics Tools*, 3.

[CL96a] Thomas F. Coleman and Yuying Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2):418–445, 1996.

[CL96b] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, New York, NY, USA, 1996. ACM.

[CL96c] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, pages 303–312, August 1996.

[CLB+09] Ming Chuang, Linjie Luo, Benedict J. Brown, Szymon Rusinkiewicz, and Michael Kazhdan. Estimating the Laplace-Beltrami operator by restricting 3D functions. *Symposium on Geometry Processing*, July 2009.

[CLK01] Byoungwon Choe, Hanook Lee, and Hyeong-Seok Ko. Performance-driven muscle-based facial animation. *Journal of Visualization and Computer Animation*, 12(2):67–79, 2001.

[CLM+10] Will Chang, Hao Li, Niloy Mitra, Mark Pauly, and Michael Wand. Geometric registration for deformable shapes. In *Eurographics 2010 Tutorial Notes*. Norrköping, Sweden, Mai 2010.

[cLO05] I chen Lin and Ming Ouhyoung. Mirror mocap: Automatic and efficient capture of dense 3d facial motion parameters from video. the visual computer. *J Zhejiang Univ SCIENCE A*, 21:355–372, 2005.

[CM92] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *International Journal of Image and Vision Computing*, 10:145–155, 1992.

[CM93] Michael M. Cohen and Dominic W. Massaro. Modeling coarticulation in synthetic visual speech. In *Models and Techniques in Computer Animation*, 1993.

[Cox89] H. S. M. Coxeter. *Introduction to Geometry*. Willey, 2nd edition, 1989.

[Cur97] Brian L Curless. New methods for surface reconstruction from range images. Technical report, Stanford, CA, USA, 1997.

[CXH03] Jin Xiang Chai, Jing Xiao, and Jessica Hodgins. Vision-based control of 3d facial animation. In *Proc. of SCA '03*, 2003.

[CZ08] Will Chang and Matthias Zwicker. Automatic registration for articulated shapes. *Computer Graphics Forum (Proceedings of SGP 2008)*, 27(5):1459–1468, 2008.

[CZ09] Will Chang and Matthias Zwicker. Range scan registration using reduced deformable models. *Computer Graphics Forum (Proceedings of Eurographics 2009)*, 28(2):447–456, 2009.

[dAST+08] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. *ACM Transactions on Graphics*, 27(3):98:1–98:10, 2008.

[dC76] Manfredo P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976.

[DCFN06] Zhigang Deng, Pei-Ying Chiang, Pamela Fox, and Ulrich Neumann. Animating blendshape faces by cross-mapping motion capture data. In *I3D '06: Proc. of the Symp. on Interactive 3D graphics and games*, 2006.

[DM00] Douglas Decarlo and Dimitris Metaxas. Optical flow constraints on deformable models with applications to face tracking. *Int. J. Comput. Vision*, 38:99–127, 2000.

[DMGL02] James Davis, Steven R. Marschner, Matt Garr, and Marc Levoy. Filling holes in complex surfaces using volumetric diffusion. In *Symposium on 3D Data Processing, Visualization, and Transmission*, pages 428–438, 2002.

[DMSB99] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[DN07] Zhigang Deng and Ulrich Neumann. *Computer Facial Animation: A Survey*. Springer London, 2007.

[DRR03] James Davis, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. In *IN CVPR*, pages 359–366, 2003.

[EF78] P. Ekman and W. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, 1978.

[Far02] Gerald Farin. *Curves and surfaces for CAGD: a practical guide.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[FAT07] Hongbo Fu, Oscar K.-C. Au, and Chiew-Lan Tai. Effective derivation of similarity transformations for implicit Laplacian mesh editing. *Computer Graphics Forum*, 26(1):34–45, 2007.

[FB87] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. pages 726–740, 1987.

[FDCO03] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. Bilateral mesh denoising. *ACM Trans. Graph.*, 22(3):950–953, 2003.

[FHK+04] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bulow, and Jitendra Malik. Recognizing objects in range data using regional point descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, May 2004.

[FHM+96] Olivier Faugeras, Bernard Hotz, Herv Mathieu, Thierry Viville, Zhengyou Zhang, Pascal Fua, Eric Thron, and Projet Robotvis. Real time correlation-based stereo: Algorithm, implementations and applications, 1996.

[Fis36] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugen.*, 7:179–188, 1936.

[Fit01] A. W. Fitzgibbon. Robust registration of 2D and 3D point sets. In *British Machine Vision Conference*, pages 662–670, 2001.

[FKY08] Wei-Wen Feng, Byung-Uck Kim, and Yizhou Yu. Real-time data driven deformation using kernel canonical correlation analysis. *ACM Trans. Graph.*, 27:1–9, 2008.

[FP08] Y. Furukawa and J. Ponce. Dense 3d motion capture from synchronized video streams. pages 1 –8, jun. 2008.

[FP09a] Y. Furukawa and J. Ponce. Dense 3d motion capture for human faces. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1674–1681, 2009.

[FP09b] Yasutaka Furukawa and Jean Ponce. Accurate camera calibration from multi-view stereo and bundle adjustment. *Int. J. Comput. Vision*, 84(3):257–268, 2009.

[FP10] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1362–1376, 2010.

[Fuj95] K. Fujiwara. Eigenvalues of laplacians on a closed riemannian manifold and its nets. In *Proceedings of the AMS*, pages 2585–2594, 1995.

[GG07] Gael Guennebaud and Markus Gross. Algebraic point set surfaces. In *ACM Transactions on Graphics*, volume 26, pages 23:1–23:10, New York, NY, USA, 2007. ACM.

[GH97] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. *Computer Graphics*, 31(Annual Conference Series):209–216, 1997.

[GHDS03] Eitan Grinspun, Anil Hirani, Mathieu Desbrun, and Peter Schrder. Discrete Shells. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 62–67, Aug 2003.

[GHF+07] Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun. Efficient simulation of inextensible cloth. *ACM Trans. Graph.*, 26(3), 2007.

[GMGP05] Natasha Gelfand, Niloy J. Mitra, Leonidas J. Guibas, and Helmut Pottmann. Robust global registration. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, page 197, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[GRIL03] Natasha Gelfand, Szymon Rusinkiewicz, Leslie Ikemoto, and Marc Levoy. Geometrically stable sampling for the icp algorithm. *3D Digital Imaging and Modeling, International Conference on*, 0:260, 2003.

[Hav06] Parag Havaldar. Sony pictures imageworks. In *SIGGRAPH '06: Courses*, 2006.

[HAWG08] Qi-Xing Huang, Bart Adams, Martin Wicke, and Leonidas J. Guibas. Non-rigid registration under isometric deformations. In *SGP '08: Proceedings of the Symposium on Geometry Processing*, pages 1449–1457, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

[HB01] Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.

[Hel98] M. Held. Fist: Fast industrial-strength triangulation. Technical report, 1998.

[HIWZ05] Walter Hyneman, Hiroki Itokazu, Lance Williams, and Xinmin Zhao. Human face project. In *SIGGRAPH '05: Courses*, 2005.

[HLP93] P. Hebert, D. Laurendeau, and D. Poussart. Scene reconstruction and description: Geometric primitive extraction from multiple view scattered data. pages 286–292, 1993.

[HLS07] K. Hormann, B. Lévy, and A. Sheffer. Mesh parameterization: Theory and practice. In *SIGGRAPH 2007 Course Notes*, number 2, pages vi+115, San Diego, CA, August 2007. ACM Press.

[HMN03] Kazuhiro Hiwada, Atsuto Maki, and Akiko Nakashima. Mimicking video: real-time morphable 3d model fitting. In *VRST '03: Proc. of the Symp. on Virtual Reality Software and Technology*, 2003.

[Hor87] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[HS81] Berthold K. P. Horn and Brian G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[HS97] Janne Heikkila and Olli Silven. A four-step camera calibration procedure with implicit image correction. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 1106, Washington, DC, USA, 1997. IEEE Computer Society.

[HSZ87] R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE PAMI*, 9:532–550, 1987.

[HTB03] D. Hähnel, S. Thrun, and W. Burgard. An extension of the ICP algorithm for modeling nonrigid objects with mobile robots. In *Proceedings of IJCAI*, 2003.

[HVB+07] Carlos Hernandez, George Vogiatzis, Gabriel J. Brostow, Bjorn Stenger, and Roberto Cipolla. Non-rigid photometric stereo with colored lights. *Computer Vision, IEEE International Conference on*, 0:1–8, 2007.

[HZ06] Peisen S. Huang and Song Zhang. Fast three-step phase-shifting algorithm. *Appl. Opt.*, 45(21):5086–5091, 2006.

[IGL03] L. Ikemoto, N. Gelfand, and M. Levoy. A hierarchical method for aligning warped meshes. In *3DIM'03*, 2003.

[Ima] Dimensional Imaging. http://www.di3d.com/.

[JH97] Andrew Johnson and Martial Hebert. Surface registration by matching oriented points. In *International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pages 121–128, May 1997.

[JMD+07] Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. Harmonic coordinates for character articulation. *ACM Trans. Graph.*, 26(3):71, 2007.

[JSW05] Tao Ju, Scott Schaefer, and Joe Warren. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.*, 24(3):561–566, 2005.

[JTDP03] Pushkar Joshi, Wen C. Tien, Mathieu Desbrun, and Frédéric Pighin. Learning controls for blend shape based realistic facial animation. In *Proc. of SCA '03*, 2003.

[Ju09] Tao Ju. Fixing geometric errors on polygonal models: A survey. *Journal of Computer Science and Technology*, 24(1):19–29, 2009.

[KBH06] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Symposium on Geometry Processing*, 2006.

[KCVS98] Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans-Peter Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 105–114, New York, NY, USA, 1998. ACM.

[KF06] Renaud Keriven and Olivier Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *The International Journal of Computer Vision*, 72:2007, 2006.

[KFR03] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 156–164, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[KHpS01] Kolja Kähler, Jörg Haber, and Hans peter Seidel. Geometry-based muscle modeling for facial animation. In *In Proc. Graphics Interface 2001*, 2001.

[Kin] Kinect. www.xbox.com/kinect/.

[KMG04] G. A. Kalberer, P. Mueller, and L. Van Gool. Animation pipeline: Realistic speech based on observed 3d face dynamics. In *1st Europ. Conf. on Visual Media Prod.*, 2004.

[KS98] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. In *Proc. Natl. Acad. Sci. USA*, pages 8431–8435, 1998.

[KSK06] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 15–18, Washington, DC, USA, 2006. IEEE Computer Society.

[KSSH02] Nikita Kojekine, Vladimir Savchenko, Mikhail Senin, and Ichiro Hagiwara. Real-time 3d deformations by means of compactly supported radial basis functions. In *In Short papers proceedings of Eurographics*, pages 35–43, 2002.

[KZ02] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *in European Conference on Computer Vision*, pages 82–96, 2002.

[LAGP09] Hao Li, Bart Adams, Leonidas J. Guibas, and Mark Pauly. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2009)*, 28(5), 2009.

[LC87] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987.

[LCXS07] Manfred Lau, Jinxiang Chai, Ying-Qing Xu, and Heung-Yeung Shum. Face poser: interactive modeling of 3d facial expressions using model priors. In *Proc. of SCA '07*, 2007.

[Lee00] John M. Lee. *Introduction to Topological Manifolds (Graduate Texts in Mathematics)*. Springer, May 2000.

[LH05] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1482–1489, Washington, DC, USA, 2005. IEEE Computer Society.

[Li05] Hao Li. Rekonstruktion farbiger objekte aus strukturiert beleuchteten ansichten. Master's thesis, Universität Karlsruhe (TH), June 2005.

[Lie03] Peter Liepa. Filling holes in meshes. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 200–205, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[LKCOL07] Yaron Lipman, Johannes Kopf, Daniel Cohen-Or, and David Levin. Gpu-assisted positive mean value coordinates for mesh deformations. In *SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 117–123, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[LLV+10] Hao Li, Linjie Luo, Daniel Vlasic, Pieter Peers, Jovan Popović, Mark Pauly, and Szymon Rusinkiewicz. Temporally coherent completion of dynamic shapes. *Submitted to ACM Transaction on Graphics*, 2010.

[LMX+08] Xuecheng Liu, Tianlu Mao, Shihong Xia, Yong Yu, and Zhaoqi Wang. Facial animation by optimized blendshapes from motion capture data. *Comput. Animat. Virtual Worlds*, 19(3-4):235–245, 2008.

[Log] LogicPD. http://www.logicpd.com/.

[LP07] Hao Li and Mark Pauly. First steps toward the automatic registration of deformable scans. Technical report, ETH Zurich, June 2007.

[LPC+00] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk. The digital michelangelo project. In *SIGGRAPH'00*, 2000.

[LS08] Rui Li and Stan Sclaroff. Multi-scale 3d scene flow from binocular stereo sequences. *Comput. Vis. Image Underst.*, 110(1):75–90, 2008.

[LSBS99] Robert Lange, Peter Seitz, Alice Biber, and Rudolf Schwarte. Time-of-flight range imaging with a custom solid state image sensor. volume 3823, pages 180–191. SPIE, 1999.

[LSCO+04] Yaron Lipman, Olga Sorkine, Daniel Cohen-Or, David Levin, Christian Rössl, and Hans-Peter Seidel. Differential coordinates for interactive mesh editing. In *Proceedings of Shape Modeling International*, pages 181–190. IEEE Computer Society Press, 2004.

[LSP06] Hao Li, Raphael Straub, and Hartmut Prautzsch. Structured light based reconstruction under local spatial coherence assumption. In *Symposium on 3D Data Processing, Visualization, and Transmission*, June 2006.

[LSP08] Hao Li, Robert W. Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum (Proc. SGP'08)*, 27(5), July 2008.

[LWP10] Hao Li, Thibaut Weise, and Mark Pauly. Example-based facial rigging. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2010)*, 29(3), July 2010.

[MAB92] Kenneth Meyer, Hugh L. Applewhite, and Frank A. Biocca. A survey of position trackers. *Presence: Teleoper. Virtual Environ.*, 1(2):173–200, 1992.

[MBR+00] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J. Gortler, and Leonard McMillan. Image-based visual hulls. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 369–374, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[MDSB02] Mark Meyer, Mathieu Desbrun, Peter Schrder, and Alan H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds, 2002.

[MES] MESA. http://www.mesa-imaging.ch/.

[MFO+07] N. J. Mitra, S. Flory, M. Ovsjanikov, N. Gelfand, L. Guibas, and H. Pottmann. Dynamic geometry registration. In *SGP'07*, 2007.

[MGP06] N. J. Mitra, L. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3d geometry. In *ACM Transactions on Graphics*, volume 25, pages 560–568, 2006.

[MGPG04] Niloy J. Mitra, Natasha Gelfand, Helmut Pottmann, and Leonidas Guibas. Registration of point cloud data from a geometric optimization perspective. In *SGP '04*, 2004.

[MHP+07] Wan-Chun Ma, Tim Hawkins, Pieter Peers, Charles-Felix Chabert, Malte Weiss, and Paul Debevec. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *2007 Eurographics Symposium on Rendering*, June 2007.

[MJC+08] Wan-Chun Ma, Andrew Jones, Jen-Yuan Chiang, Tim Hawkins, Sune Frederiksen, Pieter Peers, Marko Vukovic, Ming Ouhyoung, and Paul Debevec. Facial performance synthesis using deformation-driven polynomial displacement maps. *ACM Trans. Graph.*, 27:1–10, 2008.

[MJKM04] N. Miller, O.C. Jenkins, M. Kallmann, and M.J. Mataric. Motion capture from inertial sensing for untethered humanoid teleoperation. volume 2, pages 547 – 565 Vol. 2, nov. 2004.

[MN03] Niloy J. Mitra and An Nguyen. Estimating surface normals in noisy point cloud data. In *SCG '03: Proceedings of the nineteenth annual symposium on Computational geometry*, pages 322–328, New York, NY, USA, 2003. ACM.

[MNT04] K. Madsen, H.B. Nielsen, and O. Tingleff. Methods for non-linear least squares problems. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, 2004.

[Mov] Mova. http://www.mova.com/.

[MTLT88] N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings on Graphics interface '88*, pages 26–33, 1988.

[MTSA97] Y. Matsumoto, H. Terasaki, K. Sugimoto, and T. Arakawa. A portable three-dimensional digitizer. In *NRC '97: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, page 197, Washington, DC, USA, 1997. IEEE Computer Society.

[NA02] Jan Neumann and Yiannis Aloimonos. Spatio-temporal stereo using multiresolution subdivision surfaces, 2002.

[NN01] Jun-Yong Noh and Ulrich Neumann. Expression cloning. In *Proc. SIGGRAPH '01*, 2001.

[NRDR05] Diego Nehab, Szymon Rusinkiewicz, James Davis, and Ravi Ramamoorthi. Efficiently combining positions and normals for precise 3d geometry. *ACM Transactions on Graphics*, 24(3):536–543, August 2005.

[NWN96] Shree K. Nayar, Masahiro Watanabe, and Minori Noguchi. Real-time focus range sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:1186–1198, 1996.

[Osi07] Jason Osipa. *Stop Staring: Facial Modeling and Animation Done Right.* Sybex, Second Edition, 2007.

[OZS08] Verónica Costa Teixeira Orvalho, Ernesto Zacur, and Antonio Susin. Transferring the rig and animations from a character to different face models. *Comput. Graph. Forum*, 27(8):1997–2012, 2008.

[Par72] Frederick I. Parke. Computer generated animation of faces. In *ACM'72: Proceedings of the ACM annual conference*, 1972.

[Par82] F.I. Parke. Parameterized models for facial animation. *Computer Graphics and Applications, IEEE*, 2:61–68, 1982.

[PB81] Stephen M. Platt and Norman I. Badler. Animating facial expressions. *SIGGRAPH Comput. Graph.*, 15(3):245–252, 1981.

[PBP02] Hartmut Prautzsch, Wolfgang Boehm, and Marco Paluszny. *Bezier and B-Spline Techniques*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[PG08] Yuri Pekelny and Craig Gotsman. Articulated object reconstruction and markerless motion capture from depth video. 27(2):399–408, April 2008.

[PH06] Sang Il Park and Jessica K. Hodgins. Capturing and animating skin deformation in human motion. *ACM Trans. Graph.*, 25, 2006.

[PH08] Sang Il Park and Jessica K. Hodgins. Data-driven modeling of skin and muscle deformation. *ACM Transactions on Graphics*, 27(3):96:1–96:6, 2008.

[Pha] PhaseSpace. http://www.phasespace.com/.

[PHL+98] Frédéric Pighin, Jamie Hecker, Dani Lischinski, Richard Szeliski, and David H. Salesin. Synthesizing realistic facial expressions from photographs. In *Proc. SIGGRAPH '98*, 1998.

[PHYH06] Helmut Pottmann, Qi-Xing Huang, Yong-Liang Yang, and Shi-Min Hu. Geometry and convergence analysis of algorithms for registration of 3d shapes. *Int. J. Comput. Vision*, 67(3), 2006.

[PJP93] Ulrich Pinkall, Strasse D. Juni, and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2:15–36, 1993.

[PL06] Frederic Pighin and J. P. Lewis. Facial motion retargeting. In *SIGGRAPH '06 Courses*, 2006.

[PMG+05] Mark Pauly, Niloy J. Mitra, Joachim Giesen, Markus Gross, and Leonidas J. Guibas. Example-based 3d scan completion. In *SGP'05*, 2005.

[PSS99] F. Pighin, R. Szeliski, and D.H. Salesin. Resynthesizing facial animation through 3d model-based tracking. *In Proc. 7th IEEE Int. Conf. on Computer Vision*, 1:143–150 vol.1, 1999.

[PTVF97] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1997.

[Pul99] Kari Pulli. Multiview registration for large data sets. In *Second Int. Conf. on 3D Dig. Image and Modeling*, pages 160–168, 1999.

[PW96] Frederic I. Parke and Keith Waters. *Computer facial animation*. A. K. Peters, Ltd., 1996.

[RBK05] Szymon Rusinkiewicz, Benedict Brown, and Michael Kazhdan. 3d scan matching and registration. In *ICCV 2005 Short Course*. Beijing, China, October 2005.

[RC98] Sebastien Roy and Ingemar Cox. A maximum-flow formulation of the n-camera stereo correspondence problem, 1998.

[RGB] XYZ RGB. http://www. xyzrgb.com/.

[RHHL02] Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-time 3d model acquisition. *ACM Trans. Graph.*, 21, 2002.

[RL01] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *3DIM'01*, 2001.

[Rob59] S. Roberts. Control chart tests based on geometric moving averages. *Technometrics1*, pages 239–250, 1959.

[RTG97] Holly E. Rushmeier, Gabriel Taubin, and André Guéziec. Appying shape from lighting variation to bump map capture. In *Proceedings of the Eurographics Workshop on Rendering Techniques '97*, pages 35–44, London, UK, 1997. Springer-Verlag.

[Rus01] Szymon Marek Rusinkiewicz. *Real-time acquisition and rendering of large three-dimensional models*. PhD thesis, 2001. Adviser-Levoy, Marc.

[SA07] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 109–116, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[Saa92] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Halstead Press, New York, 1992.

[SAG03] Vitaly Surazhsky, Pierre Alliez, and Craig Gotsman. Isotropic remeshing of surfaces: A local parameterization approach. In *In Proceedings of 12th International Meshing Roundtable*, pages 215–224, 2003.

[Sal79] E. Salamin. Application of quaternions to computation with rotations. Technical report, Stanford Artificial Intelligence Lab,unpublished internal memo, 1979.

[SAL+08] Andrei Sharf, Dan A. Alcantara, Thomas Lewiner, Chen Greif, Alla Sheffer, Nina Amenta, and Daniel Cohen-Or. Space-time surface reconstruction using incompressible flow. *ACM Transactions on Graphics*, 27(5):110:1–110:10, 2008.

[SCD+06a] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 519–528, Washington, DC, USA, 2006. IEEE Computer Society.

[SCD+06b] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms, 2006.

[SCOL+04] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and Hans-Peter Seidel. Laplacian surface editing. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 179–188. Eurographics Association, 2004.

[SG03] Vitaly Surazhsky and Craig Gotsman. Explicit surface remeshing. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 20–30, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[SG04] Olaf Schenk and Klaus Gärtner. Solving unsymmetric sparse systems of linear equations with pardiso. *Future Gener. Comput. Syst.*, 20:475–487, 2004.

[Sir87] M Sirovich, L.; Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4:519–524, 1987.

[SNF05] Eftychios Sifakis, Igor Neverov, and Ronald Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph.*, 24(3):417–425, 2005.

[SP86] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, 20(4):151–160, 1986.

[SP04] Robert W. Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2004)*, 23(3), 2004.

[SPIF07] M. Salzmann, J. Pilet, S. Ilic, and P. Fua. Surface deformation models for non-rigid 3–d shape recovery. *IEEE PAMI*, 29(8):1481–1487, 2007.

[SS91] Robin Sibson and G. Stone. Computation of thin-plate splines. *SIAM J. Sci. Stat. Comput.*, 12(6):1304–1313, 1991.

[SS01] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 47:7–42, 2001.

[SSP07] Robert W. Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM Trans. Graph.*, 26:80, 2007.

[SSRMF06] Eftychios Sifakis, Andrew Selle, Avram Robinson-Mosher, and Ronald Fedkiw. Simulating speech with a physics-based facial muscle model. In *Proc. of SCA '06*, 2006.

[Sta10] Steven Stahlberg. Nikita real-time character. Filmakademie Baden-Wuerttemberg / Institute of Animation's R&D Labs, 2010.

[STDT08] Sebastian Schuon, Christian Theobalt, James Davis, and Sebastian Thrun. High-quality scanning using time-of-flight depth superresolution. *CVPR Workshop on Time-of-Flight Computer Vision 2008*, 2008.

[SWG08] Jochen Süssmuth, Marco Winter, and G"unther Greiner. Reconstructing animated meshes from time-varying point clouds. *Computer Graphics Forum (Proceedings of SGP 2008)*, 27(5):1469–1476, 2008.

[SySnZ03] Jian Sun, Heung yeung Shum, and Nan ning Zheng. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:787–800, 2003.

[Tau95] Gabriel Taubin. A signal processing approach to fair surface design. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, New York, NY, USA, 1995. ACM.

[TF03] Marshall F. Tappen and William T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *In ICCV*, pages 900–907, 2003.

[TPBF87] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 205–214, New York, NY, USA, 1987. ACM.

[Tsa92] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. pages 221–244, 1992.

[Tur92] Greg Turk. Re-tiling polygonal surfaces. *SIGGRAPH Comput. Graph.*, 26(2):55–64, 1992.

[TW90] D. Terzopoulos and K. Waters. Physically-based facial modeling, analysis and animation. *Journal of Visualization and Computer Animation*, 1:73–80, 1990.

[VBK05] Sundar Vedula, Simon Baker, and Takeo Kanade. Image-based spatio-temporal modeling and view interpolation of dynamic events. *ACM Transactions on Graphics*, 24(1):240 – 261, April 2005.

[VBMP08] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. *ACM Transactions on Graphics*, 27(3):97:1–97:9, 2008.

[VBPP05] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. *ACM Trans. Graph.*, 24, 2005.

[Vic] Vicon. http://www.vicon.com/.

[VPB+09] Daniel Vlasic, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popović, Szymon Rusinkiewicz, and Wojciech Matusik. Dynamic shape capture using multi-view photometric stereo. In *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers*, pages 1–11, New York, NY, USA, 2009. ACM.

[VRpS03] J. Vorsatz, Ch. Rssl, and H. p. Seidel. Dynamic remeshing and applications. In *Department, Stony Brook University. Her*, pages 167–175, 2003.

[WAO+09] Michael Wand, Bart Adams, Maksim Ovsjanikov, Alexander Berner, Martin Bokeloh, Philipp Jenke, Leonidas Guibas, Hans-Peter Seidel, and Andreas Schilling. Efficient reconstruction of non-rigid shape and motion from real-time 3d scanner data. *ACM Transactions on Graphics*, 2009. (to appear).

[Wat87] Keith Waters. A muscle model for animation three-dimensional facial expression. In *Proc. SIGGRAPH '87*, 1987.

[WCF07] Ryan White, Keenan Crane, and David Forsyth. Capturing and animating occluded cloth. In *ACM Transactions on Graphics (SIGGRAPH)*, 2007.

[Wen05] Holger Wendland. *Scattered Data Approximation*. Cambridge University Press, 2005.

[WF02] Greg Welch and Eric Foxlin. Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Comput. Graph. Appl.*, 22(6):24–38, 2002.

[Wil90] Lance Williams. Performance-driven facial animation. In *SIGGRAPH '90*, 1990.

[WJH97] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *Personal Communications, IEEE*, 4(5):42 –47, oct. 1997.

[WJH+07] Michael Wand, Philipp Jenke, Qixing Huang, Martin Bokeloh, Leonidas Guibas, and Andreas Schilling. Reconstruction of deforming geometry from time-varying point clouds. In *SGP*, 2007.

[WLG07] T. Weise, B. Leibe, and L. Van Gool. Fast 3d scanning with automatic motion compensation. In *Proc. CVPR'07*, 2007.

[WLGP09] Thibaut Weise, Hao Li, Luc Van Gool, and Mark Pauly. Face/off: Live facial puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation (Proc. SCA'09)*, August 2009.

[Wol74] H. J. Woltring. New possibilities for human motion studies by real-time light spot position measurement. *Biotelemetry*, 1(2):132–146, 1974.

[Woo89] Robert J. Woodham. Photometric method for determining surface orientation from multiple images. pages 513–531, 1989.

[WPH+04] T. Weyrich, M. Pauly, S. Heinzle, S. Scandella, and M. Gross. Post-processing of scanned 3d surface data. In *Symposium On Point-Based Graphics*, pages 85–94, 2004.

[YFW03] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. pages 239–269, 2003.

[YZX+04] Yizhou Yu, Kun Zhou, Dong Xu, Xiaohan Shi, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Mesh editing with poisson-based gradient field manipulation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 644–651, New York, NY, USA, 2004. ACM.

[ZCHS03] Li Zhang, Brian Curless, Aaron Hertzmann, and Steven M. Seitz. Shape and motion under varying illumination: Unifying structure from motion, photometric stereo, and multi-view stereo. page 618, 2003.

[ZCS02] Li Zhang, Brian Curless, and Steven M. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *The 1st IEEE International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 24–36, June 2002.

[ZH04] Song Zhang and Peisen Huang. High-resolution, real-time 3d shape acquisition. In *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 3*, page 28, Washington, DC, USA, 2004. IEEE Computer Society.

[ZHWG08] Kun Zhou, Qiming Hou, Rui Wang, and Baining Guo. Real-time kd-tree construction on graphics hardware. In *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers*, pages 1–11, New York, NY, USA, 2008. ACM.

[ZLG+06] Qingshan Zhang, Zicheng Liu, Baining Guo, Demetri Terzopoulos, and Heung-Yeung Shum. Geometry-driven photorealistic facial expression synthesis. *IEEE Trans. on Vis. and Comp. Graph.*, 12(1):48–60, 2006.

[ZS00] Li Zhang and Steven M. Seitz. Image-based multiresolution shape recovery by surface deformation. volume 4309, pages 51–61. SPIE, 2000.

[ZSCS04] Li Zhang, Noah Snavely, Brian Curless, and Steven M. Seitz. Spacetime faces: High-resolution capture for modeling and animation. In *ACM Annual Conf. on Comp. Graphics*, 2004.

[ZST+10] Qian Zheng, Andrei Sharf, Andrea Tagliasacchi, Baoquan Chen, Hao Zhang, Alla Sheffer, and Daniel Cohen-Or. Consensus skeleton for non-rigid space-time registration. *Computer Graphcis Forum (Special Issue of Eurographics)*, 29(2):635–644, 2010.

Curriculum Vitae

PERSONAL DATA

Name	Hao Li
E-Mail	hao@inf.ethz.ch
Address	ETH Zurich – Computer Graphics Laboratory Universitätsstrasse 6, CAB G88 8092 Zurich Switzerland `http://www.hao-li.com/`
Date of Birth	January 17, 1981
Nationality	German

EDUCATION & EXPERIENCES

2010	Ph. D. in Computer Science, ETH Zurich, Switzerland
2010	Visiting Researcher and Teaching Assistant, EPFL, Switzerland
2009	Research Intern, Industrial Light & Magic, Lucasfilm Ltd., San Francisco, USA
2008	Visiting Researcher, Stanford University, USA
2006-2010	Research and Teaching Assistant, ETH Zurich, Switzerland
2006	Visiting Research Scholar, National University of Singapore, Singapore
2006	Dipl.-Inform. *Magna Cum Laude*, Universität Karlsruhe (TH), Germany

2004-2005	Undergraduate Research Assistant, Universität Karlsruhe (TH), Germany
2002-2003	ERASMUS Student Exchange, ENSIMAG, Grenoble, France
1999-2000	German Federal Armed Forces, Merzig, Germany
1999	Baccalauréat Franco-Allemand, Saarbrücken, Germany

AWARDS & SCHOLARSHIPS

2009	ACM Symposium on Computer Animation Best Paper Award '09
2006	National Science Foundation 3DPVT '06 Student Travel Stipend
2006	German Academic Exchange Service (DAAD) fellowship
2005	Karl-Steinbuch scholarship of the MFG Baden-Württemberg
2004	Thomas Gessmann-Stiftung fellowship, German Science Foundation
2004	Baden-Württemberg scholarship of the Markel Foundation
2004	Scholarship of the Richard Winter Foundation
2002	ERASMUS scholarship
2001	E-fellows scholarship

PUBLICATIONS

TEMPORALLY COHERENT COMPLETION OF DYNAMIC SHAPES
Hao Li, Linjie Luo, Daniel Vlasic, Pieter Peers, Jovan Popović, Mark Pauly, Szymon Rusinkiewicz
Submitted to ACM Transaction on Graphics, 2010.

EXAMPLE-BASED FACIAL RIGGING
Hao Li, Thibaut Weise, Mark Pauly
ACM Transaction on Graphics, Proceedings of the 37th ACM SIGGRAPH Conference and Exhibition (SIGGRAPH 2010), 07/2010.

ROBUST SINGLE VIEW GEOMETRY AND MOTION RECONSTRUCTION
Hao Li, Bart Adams, Leonidas J. Guibas, Mark Pauly
ACM Transaction on Graphics, Proceedings of the 2nd ACM SIGGRAPH Conference and Exhibition in Asia (SIGGRAPH Asia 2009), 12/2009.

FACE/OFF: LIVE FACIAL PUPPETRY (BEST PAPER AWARD)
Thibaut Weise, Hao Li, Luc Van Gool, Mark Pauly
Proceedings of the 8th ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA 2009), 08/2009.

GLOBAL CORRESPONDENCE OPTIMIZATION FOR NON-RIGID REGISTRATION OF DEPTH SCANS
Hao Li, Robert W. Sumner, Mark Pauly
Computer Graphics Forum 27(5), Proceedings of the 6th Eurographics Symposium on Geometry Processing (SGP 2008), 07/2008.

STRUCTURED LIGHT BASED RECONSTRUCTION UNDER LOCAL SPATIAL COHERENCE ASSUMPTION
Hao Li, Raphael Straub, Hartmut Prautzsch
Proceedings of the 3rd IEEE International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT 2006), 06/2006.

REKONSTRUKTION FARBIGER OBJEKTE AUS STRUKTURIERT BELEUCHTETEN ANSICHTEN
Hao Li
Diplomarbeit, Universität Karlsruhe (TH), 06/2005.

FAST SUBPIXEL ACCURATE RECONSTRUCTION USING COLOR STRUCTURED LIGHT
Hao Li, Raphael Straub, Hartmut Prautzsch
Proceedings of the Fourth IASTED International Conference on Visualization, Imaging and Image Processing (VIIP 2004), 09/2004.

RECONSTRUCTION USING STRUCTURED LIGHT
Hao Li
Studienarbeit, Universität Karlsruhe (TH), 02/2004.

TUTORIALS & TECHNICAL REPORTS

GEOMETRIC REGISTRATION FOR DEFORMABLE SHAPES
Will Chang, Hao Li, Niloy Mitra, Mark Pauly, Michael Wand
Eurographics 2010 Tutorial Notes, 05/2010.

FIRST STEPS TOWARD THE AUTOMATIC REGISTRATION OF DEFORMABLE SCANS
Hao Li, Mark Pauly
Technical Report, ETH Zurich, 06/2007.