Data Driven Analysis of Faces from Images

Dissertation zur Erlangung des Grades „Doktor der Ingenieurwissenschaften (Dr.-Ing.)"
der Naturwissenschaftlich-Technischen Fakultäten der Universität des Saarlandes

Kristina Scherbaum
28.05.2013

Universität des Saarlandes | Max-Planck-Institut für Informatik
Saarbrücken – Germany

Dekan – Dean

Prof. Dr. Mark Groves, Universität des Saarlandes, Saarbrücken

Kolloquium – Defense

Datum – Date

17. September 2013

Vorsitzender – Chairman of the Examination Board

Prof. Dr. Philipp Slusallek	Universität des Saarlandes, Saarbrücken

Prüfer – Examiners

Prof. Dr. Volker Blanz	Universität Siegen, Siegen
Prof. Dr. Hans-Peter Seidel	MPI Informatik, Saarbrücken
Prof. Dr. Thorsten Thormählen	Universität Marburg, Marburg

Protokollant – Reporter

Dr. Carsten Stoll	MPI Informatik, Saarbrücken

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Saarbrücken, den 28.05.2013

(Kristina Scherbaum)

Abstract

This thesis proposes three new data-driven approaches to detect, analyze, or modify faces in images. All presented contributions are inspired by the use of prior knowledge and they derive information about facial appearances from pre-collected databases of images or 3D face models.

First, we contribute an approach that extends a widely-used monocular face detector by an additional classifier that evaluates disparity maps of a passive stereo camera. The algorithm runs in real-time and significantly reduces the number of false positives compared to the monocular approach. Next, with a many-core implementation of the detector, we train view-dependent face detectors based on tailored views which guarantee that the statistical variability is fully covered. These detectors are superior to the state of the art on a challenging dataset and can be trained in an automated procedure. Finally, we contribute a model describing the relation of facial appearance and makeup. The approach extracts makeup from before/after images of faces and allows to modify faces in images. Applications such as machine-suggested makeup can improve perceived attractiveness as shown in a perceptual study.

In summary, the presented methods help improve the outcome of face detection algorithms, ease and automate their training procedures and the modification of faces in images. Moreover, their data-driven nature enables new and powerful applications arising from the use of prior knowledge and statistical analyses.

Kurzzusammenfassung

In der vorliegenden Arbeit werden drei neue, datengetriebene Methoden vorgestellt, die Gesichter in Abbildungen detektieren, analysieren oder modifizieren. Alle Algorithmen extrahieren dabei Vorwissen über Gesichter und deren Erscheinungsformen aus zuvor erstellten Gesichts-Datenbanken, in 2-D oder 3-D.

Zunächst wird ein weit verbreiteter monokularer Gesichtsdetektions-Algorithmus um einen zweiten Klassifikator erweitert. In Echtzeit wertet dieser stereoskopische Tiefenkarten aus und führt so zu nachweislich weniger falsch detektierten Gesichtern. Anschließend wird der Basis-Algorithmus durch Parallelisierung verbessert und mit synthetisch generierten Bilddaten trainiert. Diese garantieren die volle Nutzung des verfügbaren Varianzspektrums. So erzeugte Detektoren übertreffen bisher präsentierte Detektoren auf einem schwierigen Datensatz und können automatisch erzeugt werden. Abschließend wird ein Datenmodell für Gesichts-Make-up vorgestellt. Dieses extrahiert Make-up aus Vorher/Nachher-Fotos und kann Gesichter in Abbildungen modifizieren. In einer Studie wird gezeigt, dass vom Computer empfohlenes Make-up die wahrgenommene Attraktivität von Gesichtern steigert.

Zusammengefasst verbessern die gezeigten Methoden die Ergebnisse von Gesichtsdetektoren, erleichtern und automatisieren ihre Trainingsprozedur sowie die automatische Veränderung von Gesichtern in Abbildungen. Durch Extraktion von Vorwissen und statistische Datenanalyse entstehen zudem neuartige Anwendungsfelder.

Summary

This thesis proposes three new data-driven approaches to detect, analyze or modify faces in images. All presented contributions are inspired by the use of prior knowledge. Throughout this thesis, we explore the fundamental concept that statistical and data-driven face models are valuable tools for the analysis of faces in images. We thus gain insights about facial appearances from pre-collected databases of images or 3D face models. In particular, we explore whether the above assumption holds for two practical purposes: face detection and face modification. The resulting solutions and methods contribute to both areas, computer graphics and computer vision alike. They help improve the outcome of face detection algorithms, ease and automate their training procedures and allow for automated modification of faces in images.

Rapid Stereo-Vision Enhanced Face Detection First, we contribute an approach that extends a widely-used monocular face detector by an additional classifier that evaluates the disparity map of a passive stereo camera. We first detect faces in 2D images by applying a trained classifier on the left image of all stereo pairs to identify image patches that are potential face candidates. To evaluate these candidates in 3D, we perform a second classification step on the disparity maps, based on an offline trained PCA classifier. The algorithm runs in real-time and significantly reduces the number of false positives compared to the monocular approach.

Fast Face Detector Training using Tailored Views Secondly, with an extended many-core implementation of the detector, we train face detection classifiers based on synthetically tailored views. The method overcomes

the previously inevitable and time-consuming need to collect and label training data by hand. Using statistical insights, the generated training data guarantees full data variability and is, moreover, enriched by arbitrary facial attributes. It can automatically adapt to environmental constraints, such as illumination or viewing angle of recorded video footage from surveillance cameras. The results are efficient view-dependent and attribute-specific face detectors, superior to the state of the art on a challenging dataset.

Computer-Suggested Facial Makeup Finally, we contribute a model describing the relation of facial appearance and facial makeup. The data-driven approach extracts makeup from before/after images of faces. The presented algorithm involves 3D information for both analysis and synthesis of artificial makeup. As such, results can be relighted and inspected in arbitrary poses and under arbitrary lighting conditions. Applications such as machine-suggested, individualized makeup can improve perceived attractiveness as shown in a perceptual study. In contrast to previous approaches, our machine-suggested makeup involves 3D information for both analysis and synthesis.

In summary, all presented methods ease and automate processes which previously required time-consuming manual interaction. All approaches validate the assumption that structured knowledge, as extracted from facial databases, is a valuable benefit for a variety of applications. Moreover, their data-driven nature suggests new and powerful applications arising from the use of prior knowledge and statistical analyses.

Zusammenfassung

In der vorliegenden Arbeit werden drei neue datengetriebene Methoden vorgestellt, die Gesichter in zweidimensionalen Abbildungen detektieren, analysieren oder modifizieren. Alle Algorithmen basieren auf der Annahme, dass strukturiertes Vorwissen über Gesichter und deren Erscheinungsformen einen nützlichen Mehrwert in die Methodik einbringt. Das Vorwissen wird dabei aus zuvor erstellten Gesichts-Datenbanken extrahiert, die in zweidimensionaler oder dreidimensionaler Form vorliegen. Die daraus resultierenden Lösungen tragen mit ihren Ergebnissen zu zwei Forschungsfeldern bei, der Computergrafik und der computerbasierten Bildanalyse. Die präsentierten Algorithmen verbessern die Ergebnisse von Gesichtsdetektionsverfahren, erleichtern und beschleunigen ihre Trainingsprozedur und unterstützen die computerbasierte Modifikation von Gesichtern in Abbildungen.

Schnelle Gesichtsdetektion mit stereoskopischer Validierung Zunächst wird ein weit verbreiteter Gesichtsdetektions-Algorithmus um einen 3D-Detektor erweitert. In einem ersten Schritt wertet der ursprüngliche Detektor das jeweils linke Bild einer stereoskopischen Aufnahme aus und identifiziert damit Bildabschnitte, die Gesichter zeigen. Darauf aufbauend prüft der zweite neu vorgestellte Detektor die verbleibenden Bildabschnitte anhand eines Vergleichs mit dreidimensionalen Tiefenkarten, die zuvor gesammelt und mittels Hauptachsentransformation in eine parametrische Form überführt wurden. Bildabschnitte, die diesen Test nicht bestehen, werden aus der Menge detektierter Gesichter entfernt. Das vorgestellte Verfahren kann in Echtzeit ausgeführt werden und reduziert nachweislich die Anzahl falsch detektierter Gesichter.

Schnelles Gesichtsdetektor-Training auf synthetischen Bilddaten In einem weiteren Beitrag wird der monokulare Basis-Algorithmus durch eine parallelisierte Implementierung verbessert und mit synthetisch generierten Bilddaten trainiert. Die so erzeugten Datensätze garantieren, im Gegensatz zu gesammelten Daten, das volle Varianzspektrum möglicher Gesichter abzudecken und können um beliebige Attribute ergänzt werden, wie etwa Alter oder Körpergewicht. Auch können die synthetisch generierten Daten in einem Analyse-Synthese-Verfahren an jede beliebige Vorgabe durch eine Kamera angepasst werden. Dazu werden aus wenigen Testszenen der Kamera die typischen Lichtverhältnisse und der Blickwinkel automatisch analysiert und die Trainingsdaten daran angepasst. Hieraus resultieren effiziente Detektoren, die bisher präsentierten Detektoren auf einem schwierigen Datensatz übertreffen und in kürzerer Zeit vollautomatisch erzeugt werden können.

Computergeneriertes Make-up Abschließend wird ein neuartiges Datenmodell für Make-up vorgestellt, das die Erscheinungsform von Gesichtern in Abbildungen modelliert und deren automatisierte Veränderung ermöglicht. Die ebenfalls datengetriebene Methode findet Anwendung in der Bildmanipulation, kann computergesteuert ein passendes Make-up empfehlen und im Bild applizieren. Im Gegensatz zu bisher gezeigten Lösungen verwendet das neue Modell auch Vorwissen über die dreidimensionale Struktur individueller Gesichter, um Analyse und Synthese des Make-ups zu berechnen. Basierend auf diesem Expertenwissen wird das Make-up passend zur Gesichtsform generiert. Eine abschließende Studie zeigt, dass die computermanipulierten Bilder die wahrgenommene Attraktivität der Gesichter steigern.

Zusammenfassend lässt sich feststellen, dass alle präsentierten Algorithmen vormals langwierige oder manuell auszuführende Prozesse ersetzen. Alle gezeigten Methoden bestätigen die Annahme, dass strukturiertes Vorwissen über Gesichter einen Mehrwert für ihre Analyse und Synthese in Abbildungen bietet. Die durchweg datengetriebene Vorgehensweise schafft zudem neue Anwendungsfelder, die der zugrunde liegenden statistischen Datenanalyse zu verdanken sind.

Acknowledgements

I first want to thank all my co-authors for their valuable ideas, input and their enthusiasm which guided and motivated me throughout all projects. By collaborating with every one of you, I experienced a variety of new concepts, methods and fields. Thank you. My most profound thanks go to Prof. Dr. Thorsten Thormählen, my supervisor, for his patience and his constantly sound, friendly, motivating and helpful advice. I greatly benefit from his comprehensive knowledge and his wide area of interest. Special thanks go to him for working towards a submission deadline even when expecting the birth of his son that day. I also benefited greatly from Prof. Dr. Volker Blanz. I owe him many thanks for his very profound guidance in the first two years, the projects we shared and for his valuable feedback from afar. I am also indebted to our department chair, Prof. Dr. Hans-Peter Seidel, who gave me the possibility to work in a very professional and experienced team at MPI and greatly supported me all the time; to MPI, MMCI, IMPRS, Graduate School, Saarland University and all the administrative people keeping this exceptional research environment alive; and to IBM New York, where I spent my research internship and in particular to my co-workers there, Rogerio Feris, Lisa Brown and James Petterson. It was a pleasure to work in your team. I thank Margaret De Lap for her advises on the correct use of English and the MMCI clusteroffice staff for their great support. I am also indebted to all contributors of our user studies and acquisition projects. I remember vividly that day when many female employees at MPI, received their professional makeup. In conclusion, I thank my parents for supporting me throughout my studies. I thank my family and my friends for their understanding when I preferred working over meeting them. And finally, my very special thanks go to

Thorsten Schreiner, who always encouraged and supported me whenever my motivation seemed to drop off. Without his patience, energy and motivation this work would have never been finished.

Contents

1 Introduction 1

1.1 Motivation 1

1.2 Contributions 6

1.3 Thesis Organization 8

2 Background 9

2.1 3D Acquisition Techniques 9

2.1.1 Active Methods 10

2.1.2 Passive Methods 16

2.2 Statistical Data Analysis 19

2.2.1 Preprocessing 19

2.2.2 Dimensionality Reduction and Feature Selection 22

2.3 A 3D Morphable Face Model 31

2.3.1 Database of 3D Face Scans 32

2.3.2 Applications 37

2.4 Own Previous Work 41

2.4.1 Extended Database of Faces 41

2.4.2 Model-Based Shape Analysis — Fitting the Model to 3D Scans 42

3 Rapid Stereo-Vision Enhanced Face Detection 47

3.1 Motivation 47

3.2 Introduction 48

3.3 Related Work 49

3.4 Face Detector for Monocular Images 54

3.5 Stereo Setup and Algorithm 55

3.6 Near-Field Stereo-Enhanced Detection 57
3.7 Far-Field Stereo-Enhanced Face Detection........................ 58
3.8 Experiments and Results.. 59
3.9 Summary, Discussion and Future Aspects 60

4 Fast Face Detector Training using Tailored Views 63
4.1 Motivation.. 63
4.2 Introduction.. 65
4.3 Related Work 65
4.4 Synthetic Training Images.. 68
4.5 Enhanced AdaBoost Training 75
4.6 Evaluation and Results.. 79
4.7 Applications 80
4.8 Summary, Discussion and Future Aspects 83

5 Computer Suggested Facial-Makeup 85
5.1 Motivation.. 85
5.2 Introduction.. 86
5.3 Related Work 87
5.4 Acquisition.. 89
5.5 Data Preprocessing 90
5.6 Modeling Makeup 92
5.6.1 Appearance-to-Makeup Mapping 93
5.6.2 Re-Synthesis of Makeup...................................... 95
5.6.3 Rendering and Applications.................................. 98
5.6.4 Perceptual study 103
5.7 Summary, Discussion and Future Aspects 106

6 Conclusion 107
6.1 Summary of Contributions 108
6.2 Discussion and Future Work 110
6.3 Concluding Remarks...111

Appendix 113

1

Introduction

This thesis proposes new data-driven approaches to detect, analyze or modify faces in images. All approaches have in common that they employ useful information from previously collected databases of faces, in both 2D and 3D. In this section, we first motivate the data-driven nature of all techniques, then outline the main contributions and finally give an overview of the structure of this thesis.

1.1 Motivation

Human faces are part of our everyday life. While interacting with people, we perceive human faces in miscellaneous situations, illuminations and environments. We talk to people, view pictures of people, observe faces in movies, marvel at portrait paintings or more abstract caricature drawings. Unconsciously we are used to observing faces with a variety of appearances. Regardless of their actual representation, we inherently can recognize human faces as such whether they may be seen in a picture, video, painting or in real life. While this appears normal to us in everyday life, it may become apparent why it is not: Faces in real life are three-dimensional objects. Their equivalents in images, paintings, drawings and movies lack a substantial piece of information — the third dimension.

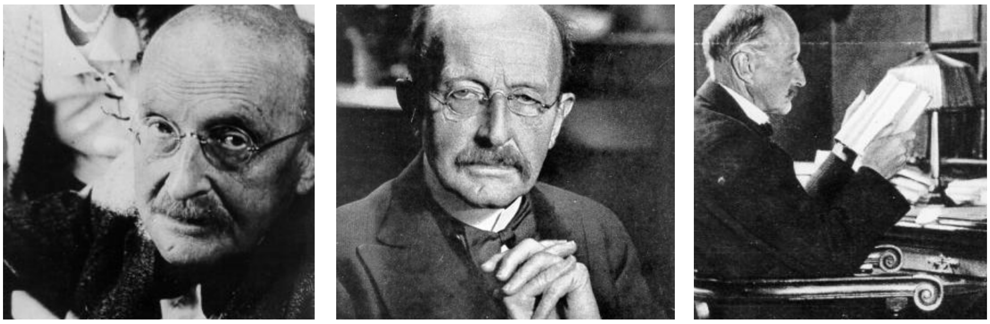


Figure credit (from left to right): [7, 8, 9]

Figure 1.1: Example of three perspective projections of the same human face (Max Planck). While humans will likely recognize the same person in all three pictures, computers will probably not. Without the help of a substantial portion of prior knowledge, they will presumably fail.

Mapping three-dimensional objects to a two-dimensional plane always involves a loss of information. Parameters such as viewing angle, position, distance and illumination gain importance and largely determine the final outcome of a projective mapping. For example, in a perspective projection, objects far distant become smaller than objects in the foreground. Consequently, recording an object from two different viewing points may yield fundamentally different projection results. Photographing, for example, a frontal and a side view (profile) of a face yields two essentially different photographs. Despite the fact that the same three-dimensional subject was observed (Figure 1.1).

Humans, however, can inherently complement the missing information [SBOR05]. Given several pictures of a person — all taken at different viewing angles — humans will likely be able to recognize the pictured faces as such and determine the person in all the pictures is the same (Figure 1.1). Besides innate abilities and genetic influences humans presumably can accomplish this task because they collect knowledge about facial appearances throughout infancy, childhood, and also adolescence [Nel01, MCNK09]. This collected information source enables humans to narrow the range of facial subtypes and thus to deduce an underlying face model that all faces have in common. Humans detect, recognize and analyze faces using a data-driven approach.

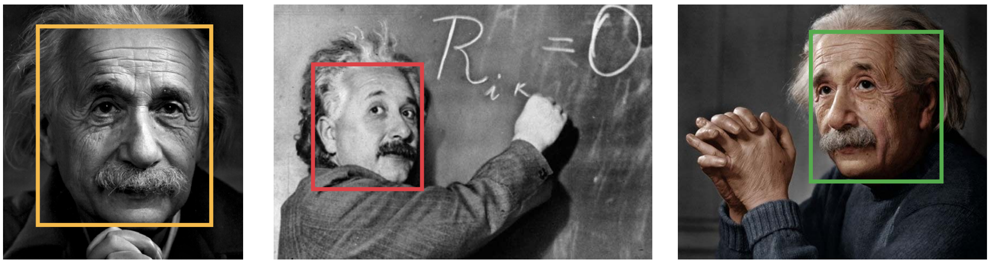


Figure credit (from left to right): [1, 2, 3]

Figure 1.2: We present two approaches that detect faces in images, as illustrated in the above photographs. Both methods face challenges such as varying viewing angles, poses, arbitrary illumination, occlusions or changing facial attributes (e.g. beards or body weight). Both algorithms are data-driven approaches and use pre-collected knowledge about facial appearances either at detection time (Chapter 3) or already at training time (Chapter 4).

Computers, in contrast, have no built-in knowledge base available. To computers, images appear as arbitrary collections of intensity values at first sight. Consequently, they are given a non-trivial task when determining whether an intensity image shows a face or not. Inferring an underlying model that all images have in common appears even more challenging. For example: Recognizing the same face in all three images in Figure 1.1 would require a computer to find significant and distinctive features in all three images which uniquely identify the face to be the same. This however might be difficult, as all images show the face from a different viewing angle. That in turn may imply that all three images have no texture patch in common. Finding a matching and distinctive feature for all images thus might work or fail by coincidence.

A more promising approach would be using prior knowledge about facial appearances to estimate the underlying three-dimensional facial model for each of the three images and to invert the perspective projection from planar space back to cartesian coordinates in 3D space. The missing third dimension may be complemented with the help of a database of known 3D faces. If the back projection of all 2D images in Figure 1.1 then leads to the same 3D model and all images may be reproduced using this 3D model, it is very likely that the faces in these images belong to the same person. However, face recognition is only one example where data-driven approaches are suitable.

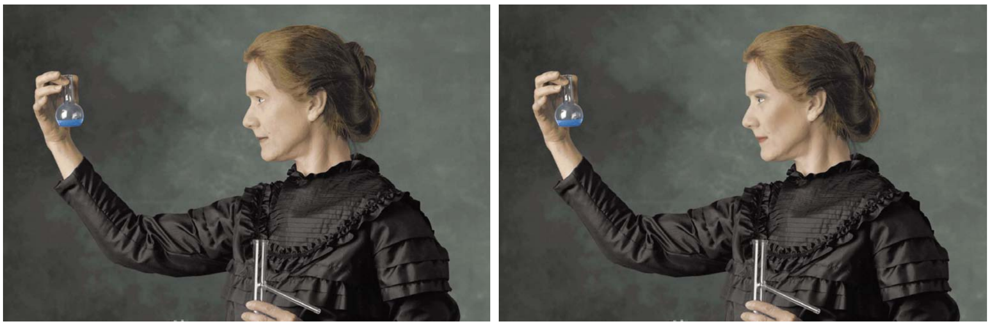


Figure credit: [6]

Figure 1.3: In this thesis, we derive a facial appearance model from a database of before/after images with and without makeup. We make use of a 3D face model to represent makeup independently from the apparent perspective of given images. The resulting makeup model allows us to modify arbitrary faces in images. The above painting of Marie Curie shows an example: before (left) and after (right) makeup application.

Generally speaking, all data-driven approaches require the computer to have a database of prior knowledge at hand. Collecting such a database, pre-processing the data and forming a statistical model of faces can be a time-consuming and non-trivial task. However, recent years have shown that, once built, prebuilt databases of 3D models are useful and promising tools for many areas. Nowadays they are used in many fields.

In this thesis, we act on the assumption that statistical and data-driven face models are valuable tools for the analysis of faces in images. In particular, we explore whether this assumption holds for two practical purposes: face detection and face modification (Figure 1.4). Throughout all projects, we either pull insights about facial appearances from planar images (Chapter 3, 4 and 5) or push them into images or detectors as gained from statistical 3D face models (Chapter 4 and 5). To bridge the gap between two-dimensional and three-dimensional representations of faces, we use a morphable face model through which we map between both coordinate systems [BV99, SSSB07]. Figure 1.4 illustrates how the presented data-driven methods gain or infer knowledge about facial appearances.

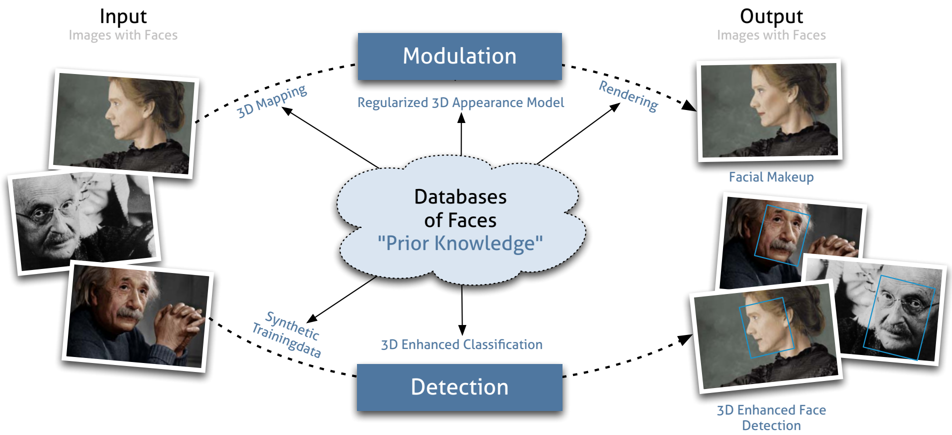


Figure credit (from left to right): [6, 7, 3]

Figure 1.4: In this thesis, we explore whether data-driven face models containing prior knowledge are suitable tools for the analysis of faces in images. In particular we test this for two practical purposes: face detection and face modification. All presented methods make use of high level knowledge about facial appearances to either detect faces in images or to modify them.

We will first introduce a method that uses a set of precomputed disparity maps of faces to improve detection accuracy. At runtime, coarse disparity maps are generated and compared to the database. Second, we present a new data-driven and parallelized training method for face detectors. The approach benefits from statistical insights about facial appearances as gained from a 3D morphable face model or apparent video scenes. Third, we introduce a newly generated appearance model of faces, which we use to apply computer suggested makeup. The model is based on a newly acquired before/after database of faces and represents makeup independently from the apparent perspective of given images.

1.2 Contributions

The new contributions in this thesis arise from three publications [KSF+09], [SFP+13][1], [SRH+11] and are listed below.

a) **Rapid Stereo-Vision Enhanced Face Detection** [KSF+09]
In Chapter 3, we investigate whether it is suitable to verify the results of state of the art face detection algorithms using a database of precomputed facial disparity maps. The results of this work have been published in [KSF+09]. The main contributions of Chapter 3 are:

- A real-time face detection algorithm that combines 2D detection with 3D evaluation
- A computational model representing three-dimensional facial disparity maps for the use of near field evaluation.
- A method for after-detection 3D evaluation reducing the number of false positives requiring affordable hardware only.

The results indicate that combining 2D detection with 3D evaluation is a suitable approach to reduce the number of false positives at low computational cost. The contributions made advance state of the art methods. In addition it could be shown, that even a small database of disparity maps is sufficient to achieve promising results.

b) **Fast Face Detector Training using Tailored Views** [SFP+13]
In Chapter 4, we generate synthetic training data covering the full spectrum of data variability and use this data to train view-dependent face detectors. The results of this work are currently under submission [SFP+13]. The main contributions of Chapter 4 are:

- A data driven method to automatically compute synthetic training views of faces using a 3D morphable face model. The resulting annotated training data guarantees full data variability, enhanced with facial attributes (such as age or body weight). Arbitrary settings for pose, viewing angle or illumination can be chosen.

[1] currently under submission

- A fast many-core Viola-Jones face detector training and evaluation method, considering multiple layers per image, such as color channels, which reduces the time for detector training and is able to handle large amounts of training data.
- Applications such as self-learning face detectors, automatically adapting to example scenes of particular surveillance cameras. The algorithm uses extracted pose and illumination parameters from a few test scenes for the generation of tailored training views.

The above contributions reduce the time for training and data collection and eliminate the need for collecting and labeling training samples manually. The presented detectors outperform previously published results on a challenging benchmark data set [2]. Also, in an almost fully automated procedure, tailored training views may be generated for particular surveillance cameras.

c) **Computer-Suggested Facial Makeup** [SRH$^+$11]
In Chapter 5, we collect a database of faces in two states, with and without makeup. Using a 3D morphable face model, we map all observations to a three-dimensional space and build a 3D facial appearance model that can be applied to any two-dimensional image in any perspective or pose. The main contributions of Chapter 5 are:

- An acquisition technique to measure changes to facial appearance caused by makeup, including a newly acquired database of 56 before/after images with and without professional makeup.
- A data-driven approach to extract makeup from before/after images, including a 2D-3D mapping using a 3D morphable face model.
- A computational model representing the relation of facial appearance and facial makeup including reflectance, gloss, surface normals and an implicit representation of the facial shape using principal components.

[2] http://vis-www.cs.umass.edu/fddb/

- A method to synthesize and transfer makeup between faces, without transferring highly individual details.
- Applications such as machine-suggested, computer-rated and individualized makeup which can improve perceived attractiveness as shown in a perceptual study.

Compared to previously presented methods, our method is the first to involve a three-dimensional statistical model of faces for the synthesis of makeup. It is also the first to recommend makeup rather than merely applying it. A variety of applications can be derived from the built model.

1.3 Thesis Organization

The present thesis is structured as follows: After the introduction, Chapter 2 gives an overview of basic principles that play a major role in the contributed methods. The background section introduces methods such as 3D scanning, statistical data analysis, data preprocessing and the morphable 3D face model used in this thesis. As the author's previous work also contributes to the presented methods, we highlight briefly its most relevant parts. Starting from Chapter 3, we describe the new contributions: First a technique for rapid stereo-vision enhanced face detection (Chapter 3), followed by Chapter 4, describing a method for fast face detector training from synthetic imagery and concluding with Chapter 5 which introduces a 3D facial appearance model describing makeup. Chapter 6 closes the main part of the thesis with a conclusion, discussion and an outlook on future work. Supplemental material can be found in the Appendix.

2 Background

All approaches presented in this thesis extract or use information as gained from three-dimensional models of faces. To collect these facial 3D models various methods were applied. In this chapter we thus briefly go through the basic methods and techniques that are commonly used to capture three-dimensional and digital information of real world objects. We further describe common techniques to process acquired data, briefly sketch the 3D morphable face model as used in this thesis and conclude with relevant own previous work.

2.1 3D Acquisition Techniques

To measure the surface of three dimensional real world objects, a variety of 3D scanners and methods are available. 3D scanners are commonly divided into two different classes: contact 3D scanners and non-contact 3D scanners. With respect to the acquisition methods used in this thesis, we focus on non-contact 3D scanners only. Among these, one may further differentiate between active and passive scanners, which will be described in the following section.

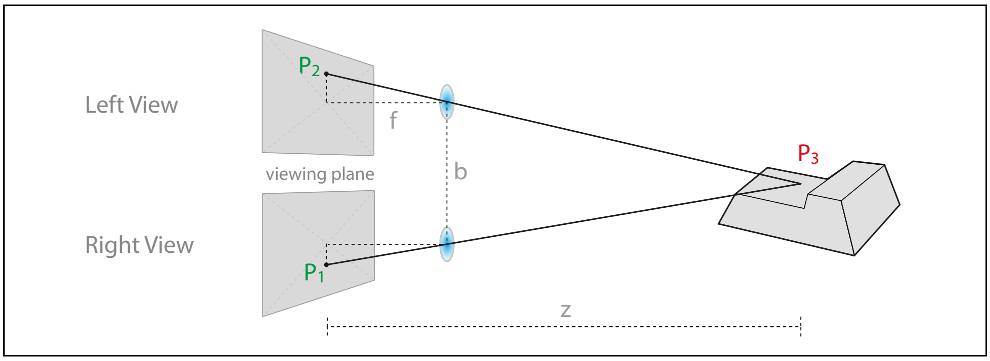


Figure 2.1: The difference between the positions of a surface point in the viewing plane is known as disparity.

2.1.1 Active Methods

In general, active 3D scanners emit light or radiation onto an object and measure the reflection with an imaging unit. From the acquired reflection and the known setup of light emitter and imaging unit, one may determine the exact position of a surface point in 3D. There exist a variety of methods to compute the distance. We focus on the most common techniques only.

3D Laser Scan Triangulation Observing an object from two different viewpoints, which are separated by a distance b, yields two different views of the same object. The different viewing angles directly result in a transformation in the viewing plane, which we refer to as ’disparity’ or ’parallax’. The disparity is always parallel to the baseline b. Thus, we always know the direction of the disparity when the exact position of the baseline b is known. From the known disparity one may compute the distance to the viewed object. The disparity is inversely proportional to the distance of the object z and directly proportional to the baseline b and the focal length of the camera lenses used. With increasing distance to the object, it becomes increasingly difficult to determine its exact position in 3D. Despite this fact, many 3D scanners — such as 3D laser scanners — rely on this principle and compute the distance between 3D object and camera plane by means of triangulation (see Figure 2.2). In contrast to only observing the object from two different viewpoints P_1 and P_2, they emit a laser light onto the

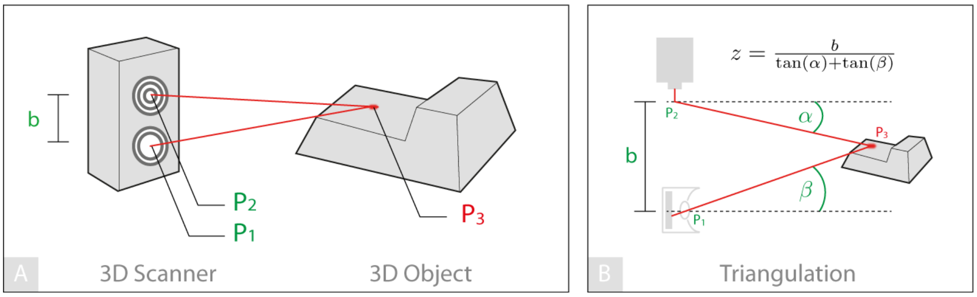


Figure 2.2: Triangulation principle of a 3D laser scanner. (A) Typical scanning setup with laser light emitter (P_1), imaging unit (P_2), and the known baseline distance (b). (B) From the known parameters (green) the position of point P_3 in 3D space may be computed.

objects surface from position P_1 and observe the reflected light dot with a built-in digital camera from a different position (P_2). The reflected laser dot, the camera and the laser light emitter form a triangle. In general, emitter and imaging unit are separated by a fixed baseline distance b, which forms one side of the triangle. To estimate the disparity and thus the position of a surface point in 3D it is necessary to identify from which direction the current pixel is illuminated and viewed. All information is known from the given setup and fully determines the distance to the observed surface point as shown in Figure 2.2–B.

Many variations of this scanning method exist. For example, the laser-pointer may be replaced by a laser-liner. The laser line is swept across the object's surface and speeds up the acquisition process tremendously. Other variations use hand-held light emitters. While camera and object remain in fixed positions, the light emitter (typically a laser-liner) is swept manually and projects a laser line onto the object's surface. This method requires a pre-calibration to align the coordinate systems of camera and object. Asymmetric calibration targets are commonly used for this purpose from which a set of reference features is calculated [WMW06].

When measuring planar and diffuse object surfaces, this scanning method works with high accuracy [BVMT03]. Measuring transparent, specular, rough or non-planar surfaces, however, may lead to a variety of inaccuracies, as shown in Figure 2.3. Inaccuracies may be further processed, or may vanish when merging several measurements of the

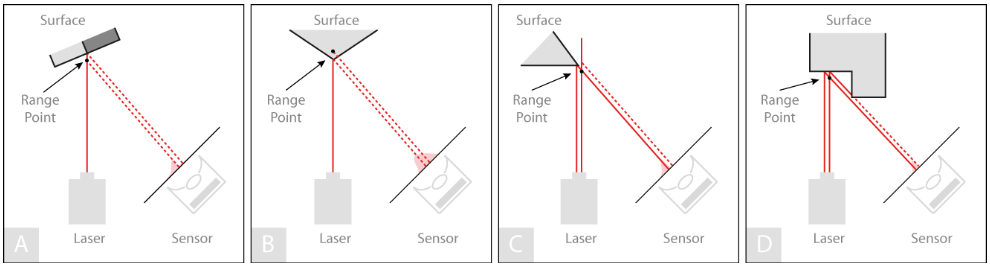


Figure 2.3: Typical range errors and inaccuracies using traditional triangulation methods: (A) Reflectance discontinuity (matte or diffuse material vs. mirrors, glass, specular materials); (B) non-planar surface geometry (e.g. at corners or bumps); (C) shape discontinuities with respect to the illumination; (D) sensor occlusion or occluded line of sight between illuminated surface and sensor, comparable to the error shown in figure (C).

same surface patch. 3D laser scanners are only applicable in the range of a few meters' distance between object and scanner, but yield very precise results (error < 1 mm).

Structured Light 3D Scanners Structured light scanners use the same triangulation principle as 3D laser scanners but determine the observed disparity by different means. In contrast to emitting a laser beam, they identify from which direction a pixel is illuminated by projecting a consecutive series of 2D stripe patterns onto the 3D object.

Common setups use a light projector to apply the light pattern onto the object's surface. There exist a variety of patterns. The simplest version is a binary stripe pattern where the stripes are perpendicular to the triangulation baseline b. Since a single stripe pattern would yield ambiguous results, a sequence of black and white stripe patterns at different wavelengths is mandatory to precisely determine the direction of light for each pixel. Assuming a pixel has been observed 5 times — each time illuminated by a different stripe pattern as shown in Figure 2.4 — the pixel may appear 5 times in either black or white. Encoding black and white with 0 and 1 respectively leads to a specific binary sequence and thus reveals the light direction and the deformation of the pattern in the image plane.

In comparison to 3D laser scanners, structured light scanners are less sensitive to varying surface properties (such as reflection coefficients) and less sensitive to long distances between object and illumination unit. On

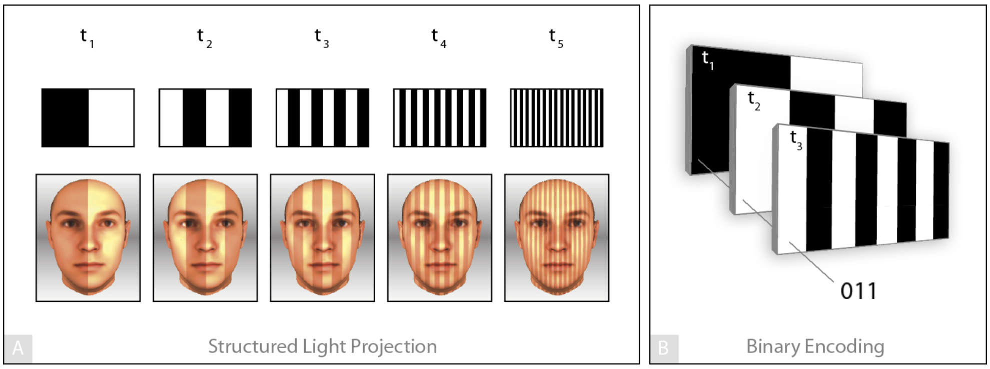


Figure 2.4: (A) Typical Stripe Patterns: From left to right the patterns are projected consecutively onto the scanned surface, e.g. faces. (B) Observing the illuminated object yields a binary encoding for each single pixel, which unambiguously determines the direction from which the pixel has been lit.

the other hand, stripe patterns allow for inaccuracies when occlusions occur. This happens for example when parts of the object are permanently shadowed and therefore yield an incorrect binary encoding. Patterns of different wavelengths may also fail for very small or tiny stripes, as the precision of the binary encoding is limited by the highest resolution.

To circumvent this problem, one may use phase-shifted periodic sine pattern series, all at the same wavelength. They allow for subpixel-exact computation of a surface point, but require precise grey levels and a photometric calibration of the camera. Also, the phase-shift is only uniquely defined within a single wavelength. Scanners often use a combination of both methods to determine the surface shape in a coarse to fine strategy (first: binary encoding; second: subpixel exact phase-shift). Compared to 3D laser scanners which acquire a single pixel or line at a time, structured-light scanners speed up the acquisition process tremendously. With each projected pattern they acquire all pixels. Depending on acquisition and projection speed, structured-light 3D scanners are thus applicable for the acquisition of dynamic scenes or deformable objects. High-speed recording systems are already available (cf. [NY08]).

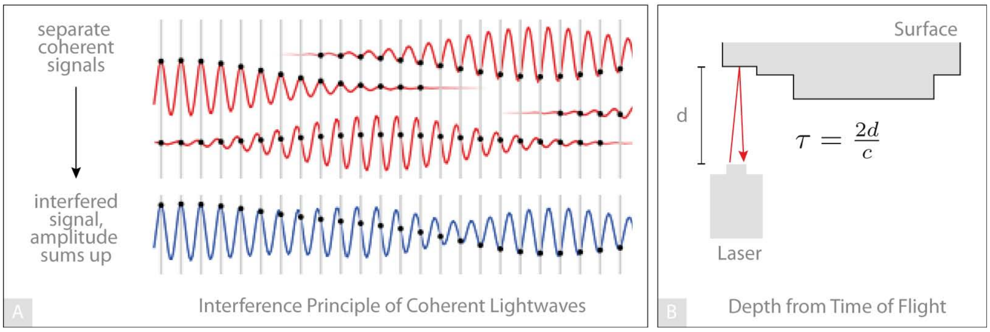

separate coherent signals
interfered signal, amplitude sums up
A
Interference Principle of Coherent Lightwaves
Surface
d
$\tau = \frac{2d}{c}$
Laser
B
Depth from Time of Flight

Figure credit: A–[10]

Figure 2.5: (A) Principle of coherent light interference. The amplitudes sum up to either high or low intensities depending on the interfering phases. (B) The distance of an object can be measured by sending a coherent signal towards a surface and measuring the time it needs for a round trip (time of flight).

Modulated Light 3D Scanners Modulated light scanners apply continuously modulated light onto the 3D surface and measure the distance by means of multiples of the emitted light wavelength (cf. [CSL08]). More precisely, they measure not only the amplitude but also the phase shift occurring between emitted and reflected light. The phase shift cannot be measured directly due to high frequencies, but may be identified by measuring the interference of the coherent light. High intensities result from interfering waves at the same phase, while low intensity values appear when the phase shift of emitted and reflected light tends towards π or $180°$ respectively. Figure 2.5(A) shows the principle of coherent light wave interference.

Time-of-Flight 3D Scanners Time-of-flight 3D scanners measure the time (TOF, τ) a light signal needs to travel from the illumination source to the object and back to the sensor (see Figure 2.5-(B), cf. [CSC+10, CST+13]). The light moves through space at a specific and characteristic speed (c) and the time between emitting and receiving the signal is proportional to twice the distance (z) between object and scanner:

$$\tau = \frac{2z}{c} \quad \Rightarrow \quad z = \frac{\tau c}{2} \tag{2.1}$$

In contrast to 3D laser scanners, the precision does not depend on the distance to the object but on the precision of the flight time measurement. Due to the very high speed of light ($299\,792\,458\,\frac{m}{s}$), measuring the round-trip time is difficult. Hence, the accuracy of these systems currently lies typically in the range of a few millimeters [CCDS09]. Taking more samples per area or applying upsampling methods [STD08] may improve the accuracy at the cost of scanning or computation time. Moreover, long acquisition times introduce new problems such as distortion from motion. Despite these issues, TOF scanners are suitable for the measurement of faraway objects such as buildings or landscapes.

Volumetric 3D Scanning Techniques The previously mentioned 3D scanning techniques differ from volumetric measurement methods in the fact that they acquire only a single depth value per surface point. Volumetric techniques, in contrast, capture a set of depth values at a time.

The most common volumetric scanning technique is the computerized X-ray tomography (CT) which is predominantly used in the medical imaging area. CT data is obtained by rotating an X-ray unit around the object and measuring the output signal with a sensor. After each full rotation, X-ray and sensor unit are shifted stepwise along the longitudinal axis of the scanned object. Variations of this setup exist, where either sensor or X-ray unit remain in a fixed position. After acquisition, the volumetric 3D model may be reconstructed by e.g. using filtered back-projection, which yields an approximate density function. From the density function one may derive a complete 3D model of the object, which can be viewed either as 2D slices or inspected using arbitrary volumetric rendering methods. In contrast to the previously described methods, computerized X-ray tomography acquires not only the surface structure but also the inner parts of the human body, in particular the bones and the major vital parts.

Another volumetric measuring method in the medical field is the magnetic resonance tomography (MRT or MRI) which uses the magnetic properties of atomic nuclei. An atomic nucleus with an odd number of nucleons behaves like a small spinning dipole in the presence of a strong magnetic field. In a homogeneous field, all atoms spin at equal frequency. By using

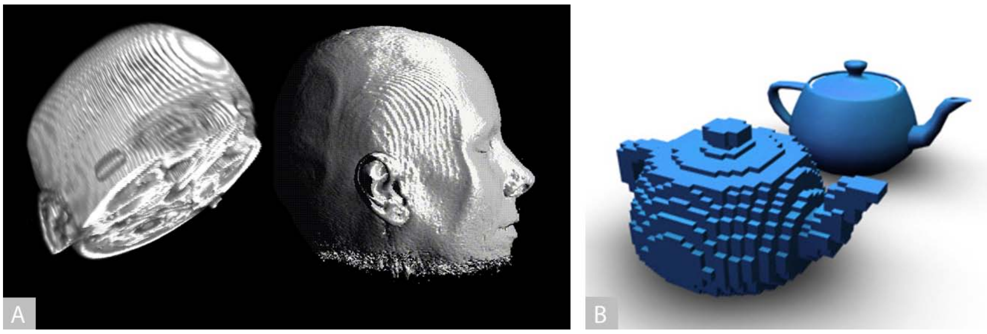


Figure credit: A–[13, 14], B–[11]

Figure 2.6: In contrast to 3D laser scans, the acquired data of volumetric 3D scanners does not lie on the same surface. (A) Volumetric CT scan of a human head which reveals the inner structure of the scanned subject. The measured depth values are usually stored and rendered as voxels. (B) shows the same object once rendered with voxels (foreground) and once rendered using a triangulated surface mesh (background).

a magnetic field with a gradient along the z-axis, each spinning frequency (Larmor frequency) becomes characteristic for one cross section. Applying different gradients during the acquisition allows for the measurement of a plane. Combining this with the CT principle (i.e. repeating the measurements for different longitudinal sections) allows for the reconstruction of a volumetric 3D model. MR imaging is most sensitive for hydrogen atoms and as such is appropriate for imaging soft tissue of human bodies.

Both methods (CT and MRT) are also used in industry, to explore the structure of materials without destroying them.

2.1.2 Passive Methods

While active 3D scanning methods change the observed scene by either applying patterns or laser beams onto the scanned objects, passive scanning methods do not involve any interaction with the scene other than observation or change of illumination.

Stereoscopy Comparable to laser scan systems, stereoscopic methods estimate the distance to the observed object by analyzing the disparity. In contrast to 3D laser scanners, stereoscopic systems passively measure the distance. Two sensors are positioned next to each other, at a distance b and both with parallel optical axes. They observe the 3D scene from different viewing angles and capture the scene simultaneously, which yields two different views or photographs of the 3D scene. Interpreting these two different views allows for the computation of disparity. Equivalently as for laser scanners, the parallax or disparity may then be computed as shown previously in Figure 2.1.

The fact that the disparity is always parallel to the baseline b makes it on the one hand easy to determine the direction of the disparity vector. On the other hand, stereoscopic systems evaluate grey value intensities to estimate the actual disparity, which is difficult when the observed object does not show any remarkable surface structure or pattern in the direction of the baseline. For these cases the estimation of depth may fail. One may circumvent this by applying special patterns or knitware onto the scanned object, which shows a sufficiently detailed surface pattern.

Stereoscopic systems are often used in the automobile industry and robotics where distances are measured to avoid collisions. In the field of topographical measurement, the use of aerial stereoscopic photographs is common, e.g., to generate 3D city models. Stereoscopic techniques are also suitable for the acquisition of face models, in particular when acquiring the facial surface of children, who should not be exposed to laser light. In the author's own previous work (Section 2.4.1), this technique was used to acquire the facial surface of infants and in Chapter 3 to measure the discriminative facial disparity maps for detection improvement.

Photometric Systems Photometric systems observe a 3D surface from a fixed viewpoint but under varying illumination conditions [TLQ08] to recover the surface orientation at each pixel.

The appearance of each lit surface point varies according to its orientation in 3D and its reflectance properties. Using calibrated illumination directions and assuming a Lambertian [Woo78, Sil80] reflectance for each observed surface point, the surface normals and albedos for each pixel may be estimated from a minimum of three acquired images [Sha92].

When recovering the surface normal from a single image, the method is also known as ’shape from shading’, as decribed by Horn in 1989 [Hor89]. Since the Lambertian model is not suitable for all types of surface reflections, there are a variety of methods which either combine Lambertian diffuse reflection models with specular models ([CJ82, SI96, DS02]) or rely on physically based models that analyze the micro surface structures ([TdF91, Geo03]).

Photometric systems are appropriate for the measurement of small surface structures such as facial pores or wrinkles. We use photometric acquisition methods in Chapter 5 to acquire the facial surface in detail.

Silhouette Techniques Silhouette techniques acquire a sequence of photographs of a 3D object. The photographs are taken from different viewpoints and in front of a high contrast background such as a green box or are extracted using chroma-keying. Using traditional computer vision techniques, the silhouette of the 3D object is computed for each single view (outline of the 3D object). Combining all silhouettes into a single model approximates the visual hull of the observed 3D object. A major drawback of the method is that cavities cannot be properly reconstructed. Comparable techniques have, for example, been used in the area of markerless motion capture [SGdA+10].

The above-described methods present a partial survey of current 3D scanning methods. Besides these methods, there exist additional 3D scanning techniques. For example, many promising variants have been presented in recent years. 3D scanners were expensive and highly specialized industry devices 10 years ago. Recent developments, however, show that processors are becoming more powerful and able to handle complex algorithms in real-time. Hence the requirements for scanning hardware have been reduced and we can observe that 3D cameras are currently becoming mass-market products. A well-known example is Microsoft’s Kinect™ camera, which was released in 2010 and has been widely used since then, including in research. For purposes of this thesis, however, we have described only the most relevant basic concepts.

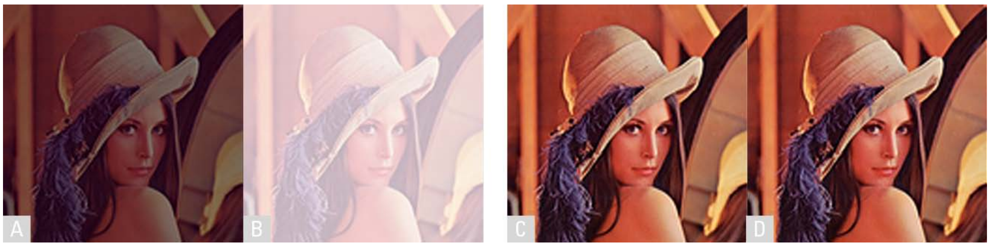


Figure credit: [5]

Figure 2.7: The two given input images show similar content but at varying intensity values (lower values (A) and higher values (B)). After normalization both images have pixel intensities that lie in the same range (C,D).

2.2 Statistical Data Analysis

Data models that have been built on the basis of acquired real-world data, such as 3D laserscans or images of faces, play a major role in this thesis. As discussed above, many data acquisition methods yield partially incomplete, incorrect or inaccurate data. To avoid building a model on the basis of erroneous data, it is thus crucial to properly process the data in advance. Two major steps are common:

2.2.1 Preprocessing

Outlier and Noise Removal An outlier is defined as a point that lies far distant from the mean of all sampled data points [TK09]. A high proportion of outliers either indicates that not enough data points have been sampled or that the acquisition method sometimes yields inaccurate or noisy data. While the first case may be solved by increasing the number of samples, the latter case exemplifies a typical problem when working with real-world data. Outliers produce large errors during training and, as such yield biased training results. It is thus crucial to remove outliers during data pre-processing to achieve proper training results in the end.

In most cases, outliers may be easily identified when assuming a Gaussian distribution. Choosing a distance threshold, one may classify the given data into samples which lie either above or below the chosen threshold. The distance threshold is usually quantified as a multiple of the standard deviation σ. A distance threshold of 2σ i.e. comprises 95.45% of all points, while 3σ covers 99.73% of all measured points. Besides this one, there are plenty of other methods to identify outliers, which we do not discuss in detail. The survey by Hodge et. al. [HA04] might serve as a good entry point for further literature inquiry.

Generally speaking, if the number of detected outliers is very small, they may be simply removed. If, however, the number is very high, removing them would reduce the dataset down to a minimum. In such cases it is more suitable to retain most outliers but to adapt all cost functions to the apparent constitution of the dataset [TK09]. Applying least squares methods, for example, would even exaggerate the influence of outliers due to the quadratic term. Choosing more appropriate methods and cost functions with respect to the given dataset is thus crucial to obtaining unbiased results.

In addition to outliers, there might be also unobtrusive noise in the dataset. To reduce its impact, one typically applies a low-pass filter, which attenuates high frequencies and retains low frequencies. Commonly, a Gaussian low-pass filter is applied. However, a variety of other techniques can be found in literature [MGMHJ04, PN12].

Data Normalization Working with real-world data often implies using datasets which have been acquired under varying conditions. The explicit values of these inhomogeneous datasets might thus vary greatly. Though describing similar content, some data samples might have higher values than others. In the simplest case, the range of values is shifted linearly (see Figure 2.7). To prevent cost functions considering higher values to be more important, it is crucial to map all data to the same range in advance. A straightforward linear method is to normalize each dataset according to its respective estimated mean and variance.

The goal is that all datasets have zero mean and unit variance [TK09]. For N available data of the k^{th} data point, we thus have

$$\overline{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_{ik}, \qquad k = 1, 2, \ldots, l \tag{2.2}$$

$$\sigma_k^2 = \frac{1}{N-1}\sum_{i=1}^{N} (x_{ik} - \overline{x}_k)^2 \tag{2.3}$$

$$\hat{x}_k = \frac{x_{ik} - \overline{x}_k}{\sigma_k} \tag{2.4}$$

Other methods aim to scale all values to a specific range. In image processing, for example, it is common to map all intensity values to a range within $[0, 1]$. This may either be performed linearly or employ non-linear mapping functions, such as the logarithmic γ-correction or sigmoid functions. In this thesis we apply comparable techniques in Chapter 3 and 5.

Missing Data Acquired real-world data may be also partially incomplete. This might happen for example when regions are occluded during 3D scanning. Straightforward ways to overcome such data gaps are to either fill in zeroes or the mean value as computed from the available data points. This procedure is also known as imputation and is sufficiently precise for many cases.

More sophisticated techniques fill in the most likely data. For such cases one may estimate a probability density function (PDF) from the available data. In the simplest case this may be, for example, a Gaussian distribution. Depending on the available data, however, a variety of approaches are suitable for PDF estimation. Common techniques are e.g. the maximum likelihood parameter estimation, Bayesian inference, maximum entropy estimation or the well-known expectation maximization algorithm [TK09]. Which method best suits a given task very much depends on the available data and the particular computational goal.

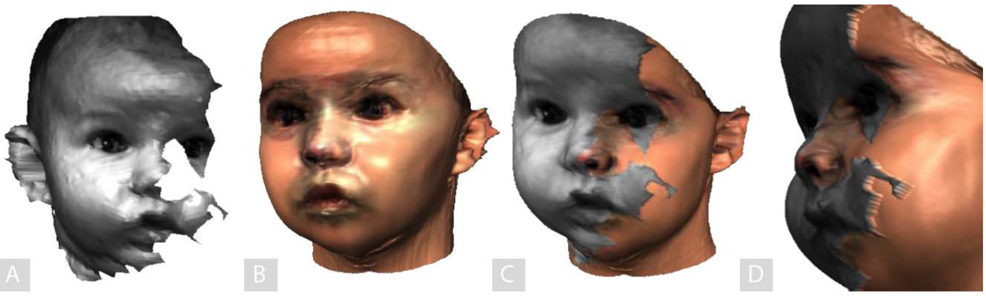


Figure 2.8: Example of the reconstruction of missing data by the use of a sample database of 3D scans. (A) Input: Incomplete 3D scan. (B) Best estimate from the the known database. (C) Merged scan and estimate. (D) Errors may occur, in particular at the boundaries, which need special treatment (such as smoothing) to appear more natural.

2.2.2 Dimensionality Reduction and Feature Selection

Dimensionality reduction and feature extraction from real-world data is one of the major tasks in machine learning. The idea is to transform a large set of data points into a new representation with fewer data points. This can be done either by selecting a few features out of many (i.e. building a subset) or by mapping the given high-dimensional space into a space of fewer dimensions. The goal is (a) to reduce the number of random variables by omitting data redundancies and (b) to find suitable data representations which implicitly reveal the underlying information and ideally offer good information packing properties. If the feature extraction method is chosen properly, the result may exhibit more relevant information using fewer data points compared to the input data. This method is also referred to as 'dimensionality reduction'.

Basis Vectors and Vector Spaces Within a vector space S, a linearly independent and (in general) finite subset of vectors $\mathbf{b} = b_0, \ldots, b_n$ with $b \in S$, is called the basis of the vector space if it spans the complete vector space and if and only if every vector $\mathbf{v}$ in the vector space may be uniquely written as a linear combination of the basis vectors

$$\mathbf{v} = v_1 b_1 + \ldots + v_n b_n = \sum_{i=1}^{n} v_i b_i \tag{2.5}$$

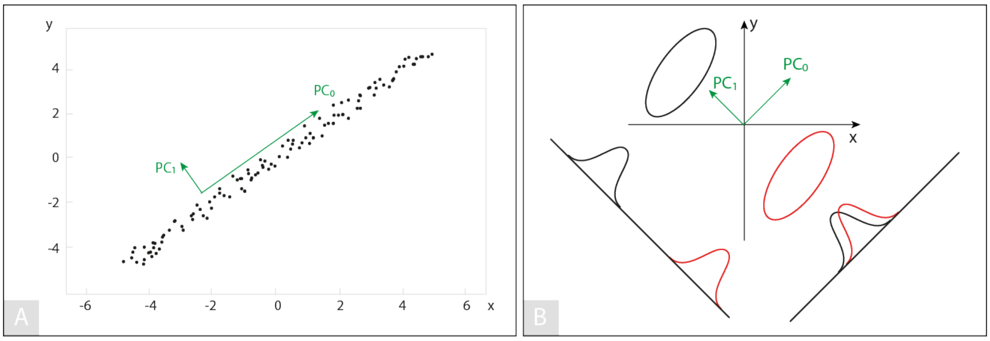


Figure credit: A–[TK09], p. 330, B–[TK09] p. 334

Figure 2.9: (A) Example data distribution (two dimensions shown, x, y). The corresponding 1st principal component (PC_0) points into the direction of maximum variance, as the 2nd (PC_1) does for the remaining subspace. (B) It might happen that classes are overlapping on one PC (PC_0), but separating properly on another PC (PC_1). It is thus crucial to analyze all subspace distributions before omitting dimensions.

A basis of a vector space is not unique. In fact, there are many different bases for any given vector space. By calculating a change of base matrix A, one may identify these and map any vector $\mathbf{v}$ to a new vector $\mathbf{y} = A\mathbf{x}$ as represented by a new basis:

$$y_i = \sum_{j=1}^{n} a_{ij} v_j \tag{2.6}$$

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} \tag{2.7}$$

The goal of the methods described in the following discussion is to identify proper vector bases, which optimally represent a given vector space with respect to a particular computational task.

Principal Component Analysis (PCA) The principal component analysis (PCA or Karhunen-Loéve transformation, KLT) transforms a given set of sample vectors $\mathbf{v}_j \in \mathbb{R}^N (j = 1, \dots, m)$ onto a new orthonormal basis, where the new basis vectors encode the variance of the dataset. The PCA is a common tool in data analysis, predominantly used in the fields of neuroscience and computer graphics. To compute a PCA on a given set of data vectors, in a first step all sample vectors $\mathbf{v}_j$ are centered with respect to their arithmetic mean $\overline{v}$ and placed in a $N \times N$ data matrix A:

$$\overline{v} = \frac{1}{M}\sum_{j=1}^{M} v_j \tag{2.8}$$

$$x_j = v_j - \overline{v} \in \mathbb{R}^n \tag{2.9}$$

$$A = (x_1, x_2, \ldots, x_m) \in \mathbb{R}^{n \times n} \tag{2.10}$$

From A one may compute the covariance matrix C which encodes the degree of linear relationship between all variables. In particular, the entries of C represent the covariance between all measured data points. C is symmetric and as such may be diagonalized.

$$\begin{aligned} C &= \frac{1}{M}AA^T && (2.11)\\ &= \frac{1}{M}\sum_{j=1}^{M} x_j x_j^T \in R^{N \times N} && (2.12)\\ C &= U\Lambda U^T && (2.13) \end{aligned}$$

There are many methods for diagonalizing C. Typically a singular value decomposition (SVD) is performed. The matrix Λ then contains the eigenvalues λ_j of C. The columns of U yield an orthonormal system of eigenvectors $\mathbf{u_j}$. They are referred to as 'principal components (PCs)' and constitute the new basis for the vector space. The eigenvalues encode the variance σ_j^2 of the data along the respective principal component $\mathbf{u_j}$. Typically, the eigenvectors and eigenvalues are sorted with descending variance magnitude: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$.

After sorting, the first principal component represents the highest variance in the dataset. Each succeeding component is linearly independent and orthogonal to the preceding one and encodes the highest variance of the remaining subspace. Thus, the first principal components contain the most important information in terms of variance, while the rear principal components may be neglected, if necessary.

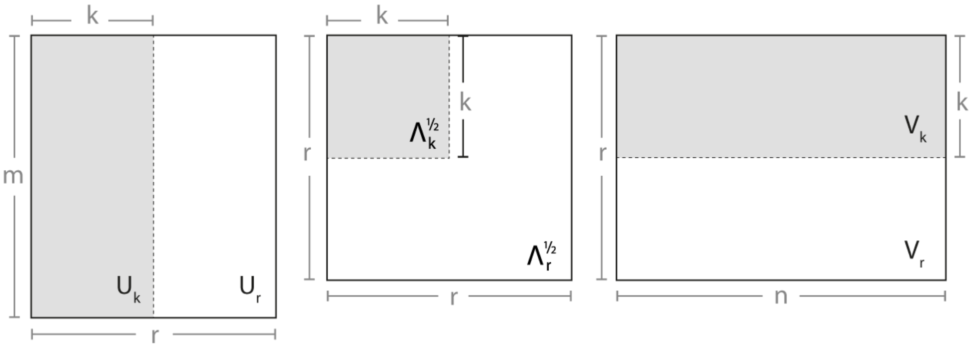


Figure credit: cf. [TK09]

Figure 2.10: Diagrammatic interpretation of the matrix products involved when performing dimensionality reduction using a SVD. X may be approximated by using only the first k columns or rows of U and V respectively.

All input data vectors may now be represented in the new basis by

$$\mathbf{x} = \sum_j \mathbf{u_j c_j} \tag{2.14}$$

$$= U\mathbf{c}, \qquad \mathbf{c_j} = \langle \mathbf{u_j}, \mathbf{x} \rangle \tag{2.15}$$

$$\mathbf{v} = \overline{v} \sum_j \mathbf{c_j u_j} \tag{2.16}$$

Note: since all vectors have been mean-shifted before, it is necessary to add the mean back at the end again.

In this thesis, we applied or used the above technique in all Chapters. In Chapter 3 for the purpose of dimensionality reduction and similarity measurement. In Chapter 4 for statistically driven data representation and modulation. And in Chapter 5 we make use of the sorted eigenvectors, when the automated makeup synthesis requires regularization.

Singular Value Decomposition (SVD) The Singular Value Decomposition is a well-known and powerful dimension reduction technique. Given a matrix $X_{m \times n}$ of rank r there exist unitary matrices $U_{m \times m}$ and $V_{n \times n}$ which decompose X such that

$$X = U \Lambda V \tag{2.17}$$

Rewriting the column vectors of U and V as $\mathbf{u}_i$ and $\mathbf{v}_i$ leads to

$$X = [\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{r-1}] \begin{bmatrix} \sqrt{\lambda_0} & & & \\ & \sqrt{\lambda_1} & & \\ & & \ddots & \\ & & & \sqrt{\lambda_{r-1}} \end{bmatrix} \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \cdots \\ \mathbf{v}_{r-1} \end{bmatrix} \tag{2.18}$$

$$X = \sum_{i=0}^{r-1} \sqrt{\lambda_i}\mathbf{u}_i\mathbf{v}_i \tag{2.19}$$

where the diagonal elements $\sqrt{\lambda_i}$ denote the nonzero eigenvalues [TK09]. One may rewrite this

$$X = U_r \Lambda^{\frac{1}{2}} V_r^H \tag{2.20}$$

such that U_r denotes the $m \times n$ matrix which consists of the first r columns of U and V_r the $r \times n$ matrix, which is formed by using the first r columns of V (see Figure 2.10).

In contrast to the PCA, the SVD is usually computed directly on the given input data. Hence the information is spread widely over all data vectors in X. It is thus more efficient to combine PCA and SVD such that the eigenvalues are computed on the covariance matrix C (which contains zero mean data). The eigenvalues thus directly relate to the covariance of the data [TK09].

Performing a SVD of large data matrices, however, is computationally expensive. In many cases it is thus more suitable to perform the SVD on X instead of the covariance matrix C. This is especially helpful, when the dimensions of the input matrix X vary strongly.

Independent Component Analysis (ICA) Comparable to the PCA, the independent component analysis is a computational method which reveals hidden structures in a given set of observed sample measurements. In contrast to the PCA, however, the ICA tries to achieve more than simple decorrelation of the data. It is based on the assumption that the given data is the result of a superposition of mutually independent signals [Com94, HKO01, TK09].

Given a set of input samples $\mathbf{x}$, the goal is to find an invertible transformation matrix $W_{N \times N}$ which maps $\mathbf{x}$ to a new vector $\mathbf{y}$ that contains independent components y_i with $i = 0, 1, \ldots, N-1$

$$\mathbf{y} = W\mathbf{x} \tag{2.21}$$

The goal of the ICA is thus to find statistical independent components rather than uncorrelated components, which is a stronger condition per se (except for the case of Gaussian random variables). The ICA may thus be seen as an extension to PCA, capable of finding independent components when common methods fail.

There are many ways to compute an ICA, which we do not discuss in detail. Originally, the ICA was developed for digital signal processing applications, but has recently been applied in many data-driven areas.

Fixed Basis Vectors vs. Flexible Basis Vectors In contrast to the previously described dimensionality reduction methods (PCA, SVD and ICA) the following methods do not map the observed sample measurements onto data-dependent basis vectors. Rather, they use fixed basis vectors for the data transformation.

On the one hand, the disadvantage of these fixed basis methods is, that they are less efficient in terms of information-packing properties than flexible basis methods. On the other hand, however, their crucial advantage is that they are computationally less expensive. Depending on the type and amount of given data, fixed basis methods thus might be more suitable.

Discrete Fourier Transform (DFT) The discrete Fourier transform (DFT) maps a discrete time-dependent signal into the frequency domain and thus reveals the underlying periodic components of the sample data (see Figure 2.11).

In the simplest case, given a one-dimensional set of observed sample measurements $\mathbf{x}^T = (x_0, x_1, \ldots, x_{N-1})$, the DFT may be computed by

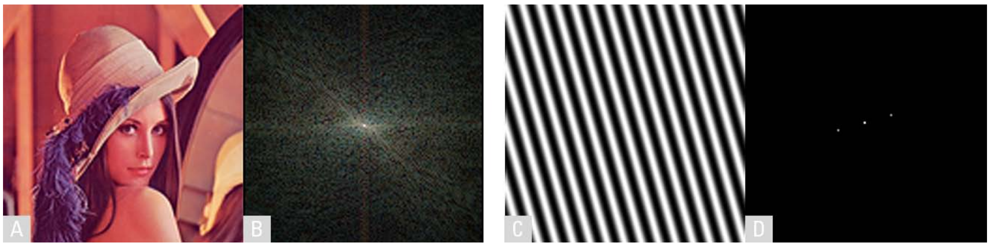


Figure credit: A–[5]

Figure 2.11: Two-dimensional DFT examples. (A) A typical photograph yields (B) widely scattered dimensions in the Fourier domain. (C) Periodical patterns however lead to (D) drastically reduced dimensions in the Fourier domain.

$$y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \, e^{\left(-\frac{2\pi ikn}{N}\right)}, \qquad k = 0, 1, \ldots, N-1, \tag{2.22}$$

where $i = \sqrt{-1}$. The inverse transformation is then defined by

$$x_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k \, e^{\left(\frac{2\pi ikn}{N}\right)}, \qquad n = 0, 1, \ldots, N-1. \tag{2.23}$$

Analogously, 2D input signals may be transformed using

$$y_{u,v} = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{m,n} \, e^{\left(-\frac{2\pi ium}{M}\right)} \, e^{\left(-\frac{2\pi ivn}{N}\right)}. \tag{2.24}$$

The DFT has found various fields of application but is predominantly used in the area of data and image compression and spectral analysis. In particular, the Fast Fourier Transform has found a wide range of applications in the domain of computer vision, especially as it offers a fast workaround to compute convolutions in the frequency domain.

Discrete Cosine Transform (DCT) The discrete cosine transform (DCT) is a suitable compression method for most real-world images and has good information packing properties. Applying DCT to a set of sample points transfers the points into a new representation in terms of sums of cosine functions at varying frequencies.

Given N one-dimensional input samples $\mathbf{x}^T = (x_0, x_1, \ldots, x_{N-1})$, their DCT is computed by

$$y_k = a_k \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi(2n+1)k}{2N}\right), \qquad k = 0, 1, \ldots, N-1, \tag{2.25}$$

while the inverse transformation is given by

$$x_n = \sum_{k=0}^{N-1} a_k y_k \cos\left(\frac{\pi(2n+1)k}{2N}\right), \qquad n = 0, 1, \ldots, N-1, \tag{2.26}$$

where

$$a_k = \begin{cases} \sqrt{\frac{1}{N}}, & k = 0 \\ \sqrt{\frac{2}{N}}, & k \neq 0 \end{cases} \tag{2.27}$$

DCTs are suitable for any task where high-frequency components may be omitted, such as for lossy audio compression (MP3) or lossy image compression (JPEG). A related transform is the discrete sine transform (DST), which is often used for the numerical solution of partial differential equations (PDE). Using either cosine or sine compression very much depends on the type of input data and the actual computational task.

Haar Transform The methods discussed so far globally transform the input signal onto a new basis. Local information is lost during transformation, since the new basis functions are applied to the complete input signal at once. The Haar transform, in contrast, partially preserves local information. The Haar basis functions form an orthonormal basis which is defined in the closed interval $[0, 1]$ (see Figure 2.12). The entries of the Haar transform matrix may be computed row-wise from the following functions

$$e_n^k(x) = \begin{cases} 2^{\frac{n}{2}} & \frac{k-1}{2^n} \leq x < \frac{k-\frac{1}{2}}{2^n} \\ -2^{\frac{n}{2}} & \frac{k-\frac{1}{2}}{2^n} \leq x \leq \frac{k}{2^n} \\ 0 & otherwise \end{cases}, \tag{2.28}$$

for integers $n = 0, 1, \ldots$, $1 \leq k \leq 2^n$ and $e_1^1(x) = 1$.

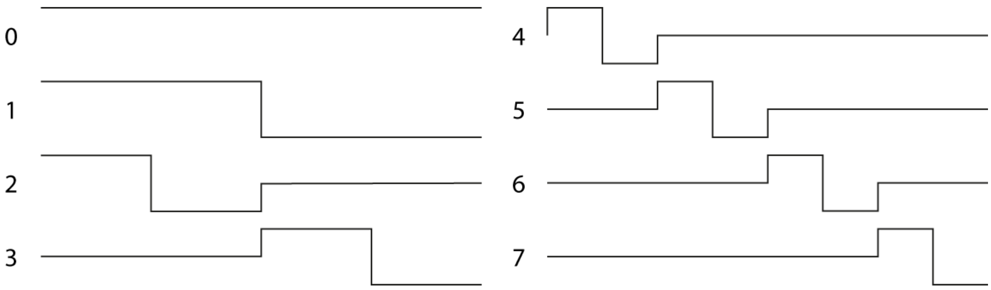


Figure 2.12: The first eight Haar basis functions. The Haar transform uses basis functions that have a kernel which involves only three values $[-1; 0; 1]$. It may thus be computed quicky and efficiently, especially compared to methods which rely on more sophisticated basis functions.

The transformed signal $\mathbf{y}$ is then achieved by applying the Haar transform matrix to the input signal $\mathbf{x}$ with $\mathbf{y} = H\mathbf{x}$. This 1D transform may be easily extended to the 2D case, where the input signal is a matrix instead of a vector. The 1D Haar transform is then consecutively applied to all rows first and columns second.

The Haar transform has found applications in fields such as image compression, computer graphics, medical imaging, signal processing, and many other fields. Compared to other fixed basis methods, its information packing properties are rather low. Nonetheless, when local structure needs to be preserved, the Haar Transform might be a good choice. For example, derived from the idea of Haar wavelets, Viola and Jones [VJ01a] introduced the so-called 'Haar features' in 2001 for the purpose of object detection. In this thesis, we employ Haar-like features in Chapter 3 and 4 to detect faces in images.

Discrete Wavelet Transform (DWT) In contrast to e.g. the Fourier Transform, which maps a signal to the frequency domain only, the Discrete Wavelet Transform (DWT) also encodes the temporal resolution. Thus, frequency and location are encoded simultaneously. The basis functions of the DWT are small wave functions which are located at different times. They are basically variations of a wavelet function. The Haar transform may be considered as the simplest case of a wavelet transform.

A crucial property of the DWT is its multiresolution property, where the input signal is analyzed into a number of different resolution levels in a hierarchical fashion. This coarse-to-fine analysis may be inverted when reconstructing the signal. At first the coarse components are reconstructed and then they are successively refined involving the higher frequencies.

Described on a highly abstract level, a given vector of N input samples $\mathbf{x}^T = (x_0, x_1, \ldots, x_{N-1})$ may be DWT-transformed by

$$y_i(k) = \sum_n x(n)\phi_{ik}(n) \tag{2.29}$$

The inverse transform is then defined by

$$x(n) = \sum_i \sum_k y_i(k)\psi_{ik}(n) \tag{2.30}$$

where $\phi_{ik}(n)$ and $\psi_{ik}(n)$ are the respective basis functions. More detailed decriptions of the DWT can be found in [VK95].

Similarly to the Haar transform, the DWT has found numerous applications in computer science, predominantly in the area of audio or image compression, in particular due to its multiresolution property, which suitably models the human perception system. The human ear, for example, perceives higher frequencies at lower resolution than low frequencies (decreasing logarithmically). Analogous phenomenons are known for the visual perception system of humans [CR68, Lev85, TK09]. The DWT addresses these properly and has proven to be suitable for most image compression tasks. It is also part of the JPEG 2000 standard.

2.3 A 3D Morphable Face Model

All above described basic concepts for 3D data acquisition and preprocessing found numerous applications in literature. One particular application example is the 3D morphable face model as introduced by Blanz and Vetter [BV99]. Since two projects described in this thesis use an extended version of this particular face model [SSSB07] we will now briefly discuss its fundamental principles.

Speaking generally, three-dimensional morphable models are powerful tools modeling specific object classes in a generic way. Due to their simplicity, in particular learning based approaches became popular and turned out to be suitable for a variety of applications in computer graphics and vision likewise. Built upon acquired databases of 3D faces, they implicitly encode semantic high-level knowledge about the facial surface topology in a parametric, yet effective, way. Their parameterization typically evolves from statistical data analysis and requires proper pre-processing of all input data in advance.

2.3.1 Database of 3D Face Scans

Data Acquisition The initial face model [BV99] is based on a set of 200 facial 3D scans of adult persons, men and women in equal shares. Each individual is represented by exactly one 3D scan. The faces were recorded using a Cyberware™ 3030PS scanner which rotates about the vertical axis of a face while recording spatial information and surface color simultaneously. The result is a set of discrete samples for shape and color. The record of a single 3D face F_i is represented by a set of parameters

$$\mathbf{I}(h,\phi) = \begin{pmatrix} r(h,\phi), \\ R(h,\phi) \\ G(h,\phi) \\ B(h,\phi) \end{pmatrix} \qquad h,\phi\,\{0,\ldots,N-1\} \tag{2.31}$$

where r gives the depth information in terms of the radius or distance from the vertical middle axis and the values R, G, B define the surface color (see Figure 2.13). Outliers were removed and all scans were trimmed and oriented uniformly in space.

Parameterization For a single face F_i, the spatial coordinates $(x_i, y_i, z_i)^T$ and the color values $(R_i, G_i, B_i)^T$ of all n surface points are collected in a shape $\mathbf{S}$ and texture $\mathbf{T}$ vector respectively:

$$\mathbf{S} = (x_1, y_1, z_1, \ldots, x_n, y_n, z_n)^T \qquad \in \mathbb{R}^{3n} \tag{2.32}$$

$$\mathbf{T} = (R_1, G_1, B_1, \ldots, R_n, G_n, B_n)^T \qquad \in \mathbb{R}^{3n} \tag{2.33}$$

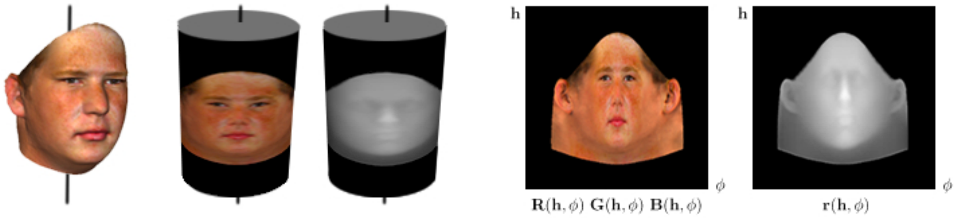

h
h
ϕ
ϕ
$\mathbf{R}(\mathbf{h},\phi)\ \mathbf{G}(\mathbf{h},\phi)\ \mathbf{B}(\mathbf{h},\phi)$
$\mathbf{r}(\mathbf{h},\phi)$

Figure 2.13: The Cyberware™ 3D scanner acquires a face while rotating about its vertical axis. It provides a cylindrical representation of each measured face, by sampling surface data for $N = 512$ height-steps h and angle-steps ϕ (both equally spaced). Depth and color values are recorded simultaneously. Apart from the black background area, faces are represented by exactly 75.972 colored vertices each.

A linear interpolation of two faces F_1 and F_2 may then be described for shape and texture separately as given below

$$\mathbf{S_{new}} = (1-\lambda)\mathbf{S_1} + \lambda\mathbf{S_2} \qquad \lambda \in [0,1] \tag{2.34}$$

$$\mathbf{T_{new}} = (1-\lambda)\mathbf{T_1} + \lambda\mathbf{T_2} \qquad \lambda \in [0,1] \tag{2.35}$$

Together, S_{new} and T_{new} describe a new face F_{new}. Introducing weight vectors $\mathbf{a}, \mathbf{b}$ for shape and texture separately, one may also combine multiple faces

$$\mathbf{S_{new}} = \sum_{j=1}^{m} a_j \mathbf{S_j} \tag{2.36}$$

$$\mathbf{T_{new}} = \sum_{j=1}^{m} b_j \mathbf{T_j} \tag{2.37}$$

Arbitrary linear combinations of the initial database of face scans may be used to blend faces (cf. Figure 2.14-A and Figure 2.14-B).

Optical Flow Registration and Correspondence To achieve natural-looking linear combinations, however, it is not sufficient to orient all faces uniformly in space. Computing a semantic vertex-to-vertex correspondence across all 3D faces is essential to construct a morphable model

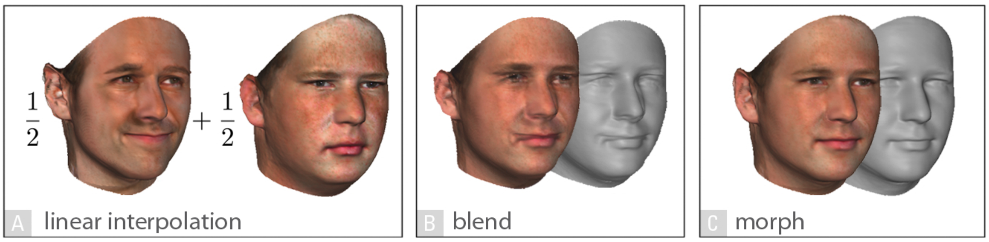


Figure 2.14: (A) Morphing two faces with incorrect or missing correspondence results in a 'blend' (B). States in between the blending process yield 'non-faces' with doubled features, such as two nose tips or four eyes. (C) If semantical correspondence has been established previously, the morphing algorithm in contrast yields a new veridical-looking human face model, with convincing facial features.

from a database of unregistered 3D scans. The difference between morphing with and without underlying semantic vertex-to-vertex correspondence is illustrated in Figure 2.14.

To compute correspondence between pairs of facial 3D scans, Blanz and Vetter applied a gradient-based optic flow algorithm. Speaking generally, optic flow algorithms aim to identify corresponding points in pairs of grey-level images. Blanz and Vetter adapted this technique for the use with cylindrically parameterized 3D scans of faces (texture and shape, [BV03]). Using this technique, they could identify the change of facial features, such as the corners of the eyes, across all faces in the database. The resulting vector flow field allowed then to find, for each vertex in a 3D sample face F_i, the semantically corresponding vertex in another, arbitrary face F_j. A detailed description of the applied algorithm can be found in [BV99] and [BV03].

Within the morphable model, the computed correspondence has then been encoded by rearranging the facial shape and texture vectors such that the same facial features of different faces are stored at the same positions in their respective shape and texture vectors. Hence, by selecting the tip of the nose in face F_i also locates the tip of the nose in any other 3D face within the data set (Figure 2.15).

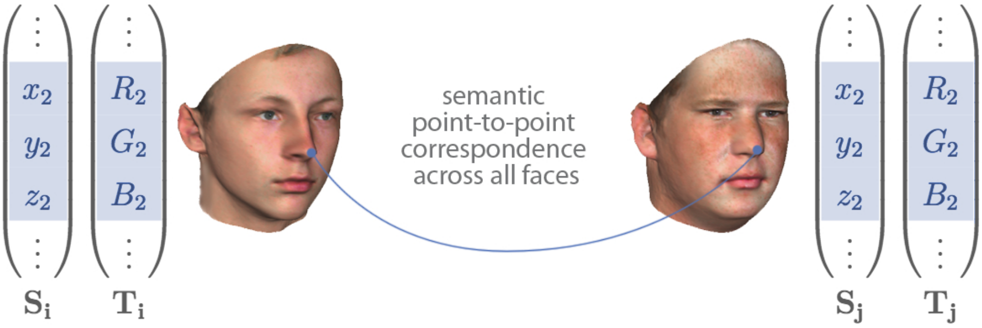


Figure 2.15: Comparable semantic features of different faces F_i and F_j are stored at the same position in their respective shape and texture vectors. This vector representation allows for straightforward and efficient face calculations. Each face consists of exactly $n = 75.972$ vertices; each vector S or T thus has $n \times 3 = 227.916$ entries.

Dimensionality Reduction Up to now, a single face scan is described by $75.972 \times 6 = 455.832$ dimensions. To reduce the dimensionality, Blanz and Vetter applied a principal component analysis (PCA; see also Section 2.2.2) to all $m = 200$ face scans. For the 3D morphable face model the PCA was applied to shape and texture separately. Formally, the result is a set of $m - 1 = 199$ principal axes s_j, t_j which vary around the averages $\bar{\mathrm{s}}$ and $\bar{\mathrm{t}}$:

$$\mathbf{S} = \bar{\mathbf{s}} + \sum_{j=1}^{m-1} \alpha_j \cdot \mathbf{s_j}, \qquad \bar{\mathbf{s}} = \frac{1}{m} \sum_{j=1}^{m} \mathbf{S_j}, \tag{2.38}$$

$$\mathbf{T} = \bar{\mathbf{t}} + \sum_{j=1}^{m-1} \beta_j \cdot \mathbf{t_j}, \qquad \bar{\mathbf{t}} = \frac{1}{m} \sum_{j=1}^{m} \mathbf{T_j} \tag{2.39}$$

Projecting the facial scans onto the new basis, one now may encode each face by a small set of $(m - 1) \times 2$ PCA coefficients. The PCA implicitly encodes the variance of the given dataset while retaining the characteristics of the input data. The probability distributions $p(\alpha)$ and $p(\beta)$ for the PCA parameters α and β (controlling shape and texture of a

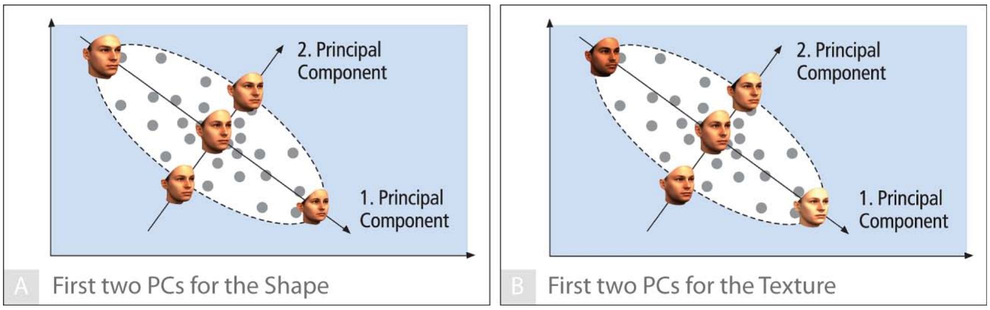


Figure 2.16: The PCA yields basis vectors (PCs) which span the face space and encode the variance of the input data. Shown are 3D face models at $\pm 3\sigma$ along the first two PCs, for both shape and texture. (A) shows the shape variation of faces along the first two principal axes while (B) shows the same for the texture.

generated face) were also estimated and may be evaluated by

$$p_S(\alpha) = \nu_S \cdot e^{-\frac{1}{2}\sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2}} \tag{2.40}$$

$$p_T(\beta) = \nu_T \cdot e^{-\frac{1}{2}\sum_i \frac{\beta_i^2}{\sigma_{T,i}^2}} \tag{2.41}$$

where ν_S and ν_T denote constant normalization factors that allow to control the face-likelihood or plausibility of synthetically generated faces. The standard deviations are given by $\sigma_{S,i}$ and $\sigma_{T,i}$.

Morphable Model Each face in the dataset may be represented as either a set of 3D coordinates and color values or as PCA coefficient vectors α, β. In either representation, the recorded sample faces may be regarded as points in face space (see Figure 2.16).

Generating new faces is straightforward. Using the PCA representation, it is necessary to vary the PC coefficient vectors α, β (Equation 2.38 and 2.39). This may be done for shape and texture separately. In the face space as spanned by the input database, these newly generated faces are located in the space between the already-existing points. Figure 2.16 illustrates the iso-surfaces of multimodal Gaussian distribution of the input data as estimated in a PCA. The Equations 2.40 and 2.41 define this distribution with respect to the PCA basis s_j and s_t. Plausible and natural-looking faces lie within the standard deviation at the same probability. In Chapter 4 we synthetically generate random faces using this approach.

In general, faces far from the average, the center of the iso-surface, appear less convincing than faces which are located in the direct neighborhood of the center. Although the generation of implausible faces is normally restrained, some synthetic faces may tend to the surface border and as such may appear non-face-like. Exaggerating the distance to the center enhances this effect and may lead to caricatures. These lie far away from the average face.

2.3.2 Applications

Fitting the Model to Images An essential application of the 3D morphable face model is to estimate the three-dimensional shape of a face starting from a single photograph of a person. The built-in a-priori knowledge about faces is crucial, since from a single picture no depth structure can be known. The resulting face reconstruction is one possible solution to an under-determined problem. In contrast to manual reconstruction techniques, which are time-consuming and require skilled artists, reconstruction using a 3D morphable face model is convenient, almost fully automated and fast.

A small set of manually clicked or automatically detected feature points serves as input for the reconstruction process which operates fully automatically (cf. Figure 2.17). Starting from the average face, rough estimates of the parameters $\alpha, \beta, \mathbf{p}$ are used, to render a first image of the reconstructed face. The set of parameters $\mathbf{p}$ denotes pose and illumination conditions while α and β describe the PCA coefficients for shape and texture.

Iteratively, a stochastic Newton optimization algorithm optimizes all parameter sets. In each iteration 40 random points are rendered and compared to the input image. To determine the optimal solution, a cost function is minimized using a stochastic gradient descent. It minimizes the difference between the input image and the estimated model while adjusting all parameters (Figure 2.17). Simplified, the cost is the sum of all pixel differences. Formally, it reads as follows:

$$E = \frac{1}{\sigma_I^2} E_I + \frac{1}{\sigma_F^2} E_F + \sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2} + \sum_i \frac{\beta_i^2}{\sigma_{T,i}^2} + \sum_i \frac{(\rho_i - \overline{\rho}_i)^2}{\sigma_{R,i}^2}. \tag{2.42}$$

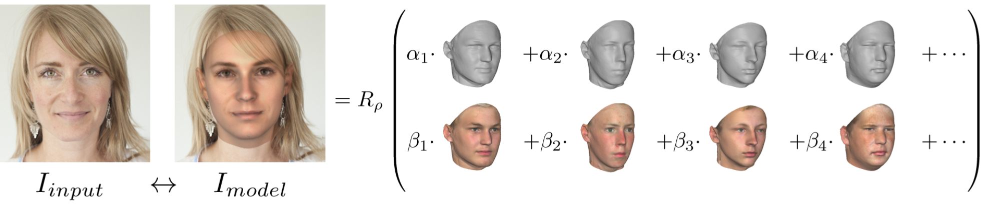


Figure 2.17: In each iteration step, a certain linear combination for shape and texture defines a texture mapped 3D model (I_{model}), which is rendered onto the input image (I_{input}). The algorithm compares I_{input} and I_{model} and iteratively refines the modeled estimate until the cost function is minimized. The parameters $\mathbf{p}$ define the rendering conditions.

$$E_I = \sum_{x,y} \|\mathbf{I}_{input}(x,y) - \mathbf{I}_{model}(x,y)\|^2 \tag{2.43}$$

where E_F measures as an additional term the 2D distance between the clicked feature points and the corresponding points of the 3D model, as projected into the image plane.

The variables σ_I and σ_F control the relative weights of E_I, E_F, and the prior probability terms, so they make the estimate more or less conservative. The parameters ρ denote the 3D orientation and position of the face, the illumination parameters of the incident light (ambient and directed light), color gain, contrast and offset, and furthermore the estimated focal length and the observed surface specularity.

At the beginning, the manually-selected feature points exert intense influence on the linear coefficients. This influence decreases gradually during the optimization algorithm. The first iterations optimize only the first ten shape and texture coefficients ($\alpha_{1,\ldots,10}, \beta_{1,\ldots,10}$) which result from the PCA, and all parameters $\mathbf{p}$. In general, during the overall fitting process, only the 99 most relevant coefficients or main axes found by the PCA are used, instead of 199 possible. This is done to prevent non-facial reconstructions or overfitting. Because of the a priori probability and the reduced number of PCA dimensions, non-facial results are almost completely excluded.

In a final step, an illumination-corrected texture is extracted. Because of the estimated rendering parameters for pose and illumination conditions, the extracted texture may be decoupled from the apparent illumination effects shown in the image. This yields a normalized texture and allows relighting the 3D model afterwards in arbitrary ways. Also, invisible surfaces may be mirrored from visible sections to obtain a complete texture. Applying slight variations to the optimization and texture extraction procedure, the algorithm is also capable of fitting the model to multiple images simultaneously. For this case the parameters α_i, β_i are the same for all images, while there are multiple parameter sets $p_{i,1}, p_{i,2}, \ldots$ that describe the respective images.

The final result is one 3D face model, generated from a single image or multiple images, which automatically establishes dense point-to-point correspondence to all faces in the database.

Fitting the Model to 3D Scans Similarly to the reconstruction of faces in images, the algorithm may be applied to fit the morphable model to 3D scans [BV99]. A possible use could be to complement fragmentary measurements.

A manual initialization, helps to find a set of initial parameters. Starting values for pose and the linear coefficients are given by clicking a set of about 7 facial features alternately on a reference face and on the scan. The fitting algorithm is based on the parameterization of the laser scans (cf. Equation 2.31 on page 32), which results from a cylindrical map with the radius r of a 3D surface

$$C : \mathbb{R}^3 \to \mathbb{R}^2 : (x, y, z) \mapsto (h, \phi) \tag{2.44}$$

Similarity to a picture which results from a perspective projection of a 3D space can be expressed formally by:

$$P : \mathbb{R}^3 \to \mathbb{R}^2 : (x, y, z) \mapsto (x, y) \tag{2.45}$$

which yields:

$$\mathbf{I}(x, y) = \begin{pmatrix} I_r(x, y) \\ I_g(x, y) \\ I_b(x, y) \end{pmatrix} \tag{2.46}$$

One may use this fact and replace the perspective projection with a cylindrical projection, and apply an adapted fitting algorithm to 3D laser scans. The linear coefficients α and β have to be optimized such that the synthetic 3D reconstruction $I_{reco}(h, \phi)$ matches the laser scan $I_{scan}(h, \phi)$ best. It is assumed that both scans are cylindrically parameterized. Unlike the rendering parameters $\mathbf{p}$ used for reconstruction from single images, the rendering parameters $\mathbf{p}$ for 3D include no illumination information. They contain values for rigid transformations and 3D translations only, and are optimized along with the linear coefficients α, β. Starting with an initial set of parameters $\alpha = \beta = \mathbf{p} = 0$, the algorithm determines iteratively the minimum of a specific cost function. A random test of 40 vertices per iteration step is used to determine the cost. A final shape and texture extraction is applied such that the original vertex coordinates, if they exist, are inherited and merged with the reconstruction result. Figure 2.8 on page 22 shows the best fit and the merged result in comparison. The final model consists partially of measured vertices (gray) and partially of estimated vertices (colored). It is fully correspondent to the face space and the 3D morphable face model.

In Section 2.4, we will discuss a new algorithm, which is part of the author's previous work, that takes perspective projection and illumination effects into account.

Further Applications Derived from the above-described methods, the 3D morphable face model has found numerous fields of application in computer graphics and computer vision alike. A variety of applications has been presented in the recent years. They comprise the change of facial attributes [BV99], extraction and application of facial expressions for facial animation [BBPV03], a technique to exchange faces in images [BSVS04], methods for cross-modal face recognition [BSS07] and for the prediction of facial growth [SSSB07].

2.4 Own Previous Work

In particular, two recently presented extensions to the 3D morphable face model have proven to be useful for the projects presented in this thesis. We describe them briefly and with respect to their applications in the main part of this thesis.

2.4.1 Extended Database of Faces

The initial face database consists of 200 3D face scans, men and women in equal shares. Most database subjects were in their mid-twenties or older when scanned; only a few persons were recorded at a younger age. For most reconstruction purposes, however, the database is sufficiently dense and properly reconstructs faces from photographs or 3D scans. Reconstructing very young faces, in contrast, often does not yield convincing results.

Hence, in our own previous work (cf. [SSSB07]) we extended the initial database of faces. Firstly we added 76 three-dimensional face scans of young infants aged between 3–12 months to the database and secondly an additional of 236 three-dimensional scans of children and young adults (aged between 7–16 years). All added scans were acquired using different scanning devices and techniques. The infant dataset consisted of a single greyscale stereoscopic picture per person. The dataset of children and young adults comprised four different 3D laser scans per subject and up to 10 separate photographs of the facial surface. A major challenge was thus to homogeneously represent this highly distinct data, before adding it to the database.

For the children dataset we thus developed a technique to merge several 3D scans of the same person for both shape and texture separately. Several texture maps were extracted from the available photographs per person, merged in a view-dependent manner and automatically freed from harsh lighting effects such as highlights and cast shadows. The result was a single diffuse texture for each single person. The greyscale textures from the infant dataset were colored automatically.

We established correspondence between all new faces and the 3D morphable face model in a single step. Very young and very small faces may now be properly reconstructed and are no longer suppressed by the optimization procedure, because of being unlikely and lying far from the mean value of the initial database. If necessary, one may also apply effects of aging to any reconstructed 3D model. We therefore presented a non-linear regression technique which applies aging effects to faces as learned from sample data [SSSB07].

We used this new and extended 3D morphable face model for all reconstruction purposes in the following chapters. In Chapter 4 we use it to generate tailored views of faces at various ages. The rendered images serve as training data for face detectors. For the projects presented in Chapter 5 the extended 3D model turned out to be helpful when reconstructing faces from images or 3D scans. In comparison to the initial model, the reconstructions using the extended model yielded more convincing reconstructions of small female faces. Furthermore, we used its texture merging technique for those cases where a single scan was partially incomplete or occluded and a second dataset was available to overcome these artifacts.

2.4.2 Model-Based Shape Analysis — Fitting the Model to 3D Scans

As described above, the initial 3D morphable face model already involved a method for fitting the model to 3D scans of faces (cf. [BV99], Section 2.3.2 on page 39). To fit the model to both, shape and texture of the 3D scan the algorithm required a cylindrically parameterized 3D scan. Most scanners, however, acquire real world objects in terms of a perspective projection with a fixed center of projection. Transferring those scanning results into a cylindrical representation would not only require an intermediate step but also ignore valuable information such as pose and illumination information, which is usually recorded by a texture image. We thus presented a new approach which is specifically designed for 3D scans that have been acquired within a perspective projection.

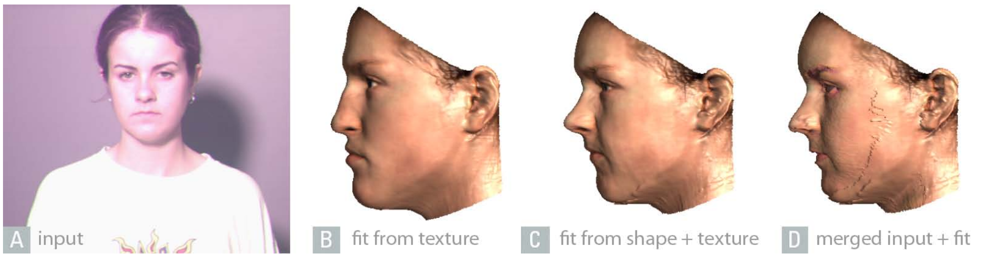


Figure credit: A–[4]

Figure 2.18: If the reconstruction is computed only from the input texture (A), the algorithm estimates the most plausible 3D shape (B), given the shading and shape of the front view. Fitting the model to both texture and shape (C) captures more characteristics of the face, which are close to the ground truth that we obtain when sampling the texture and shape values (D).

Our method builds upon the algorithm for fitting the morphable model to photographs (cf. Section 2.3.2) and generalizes the method to 3D by including range data in the cost function that is optimized during the fitting procedure to the texture image [BSS07]. In a unified framework, the algorithm optimizes shape, texture, pose and illumination simultaneously. More specifically, the algorithm synthesizes a random subset of pixels from the scan in each iteration by simulating rigid transformation, perspective projection and illumination. In our analysis-by-synthesis approach, each vertex k is mapped from the model-based coordinates $x_k = (x_k, y_k, z_k)^T$ in $\mathbf{S}$ to the screen coordinates u_k, v_k. A rigid transformation maps x_k to a position relative to the camera.

$$\begin{aligned} \mathbf{w}_k &= (w_{x,k}, w_{y,k}, w_{z,k})^T \\ &= \mathbf{R}_\gamma \mathbf{R}_\phi \mathbf{R}_\theta \mathbf{x}_k + \mathbf{t}_w. \end{aligned} \tag{2.47}$$

The angles ϕ and θ control in-depth rotations about the vertical and horizontal axis, γ defines a rotation about the camera axis, and t_w is a spatial shift. A perspective projection then maps a vertex k to image plane coordinates u_k, v_k:

$$u_k = u_0 + f \frac{w_{x,k}}{w_{z,k}} \tag{2.48}$$

$$v_k = v_0 - f \frac{w_{y,k}}{w_{z,k}} \tag{2.49}$$

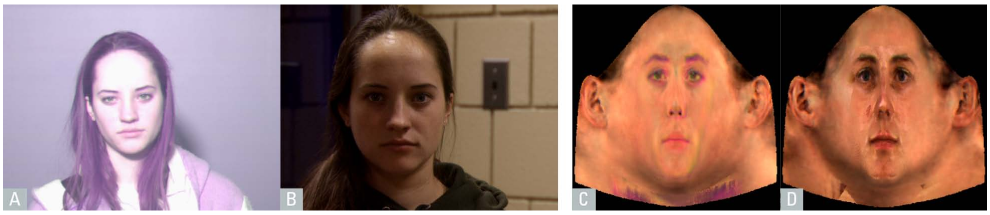


Figure credit: A–[4], B–[4]

Figure 2.19: The textures on the right (C,D) have been sampled from the scans on the left (A,B). The inversion of illumination effects has removed most of the harsh lighting from the original textures. The method compensates for the results of both overexposure and inhomogeneous shading of the face.

where f is the focal length of the camera which is located in the origin, and u_0, v_0 define the image-plane position of the optical axis (principal point). We assume that the scanning setup involves similar lighting effects as a standard photograph and thus simulate the scene lighting explicitly. The algorithm estimates ambient light with red, green, and blue intensities $L_{r,amb}$, $L_{g,amb}$, $L_{b,amb}$, and directed light with intensities $L_{r,dir}$, $L_{g,dir}$, $L_{b,dir}$ from a direction $\mathbf{l}$ defined by two angles θ_l and ϕ_l:

$$\mathbf{l} = (\cos(\theta_l)\sin(\phi_l),\ \sin(\theta_l),\ \cos(\theta_l)\cos(\phi_l))^T. \tag{2.50}$$

The illumination model of Phong (see [FvDFH96]) approximately describes the diffuse and specular reflection of a surface. In each vertex k, the red channel is

$$L_{r,k} = R_k \cdot L_{r,amb} + R_k \cdot L_{r,dir} \cdot \langle \mathbf{n}_k, \mathbf{l} \rangle + k_s \cdot L_{r,dir} \langle \mathbf{r}_k, \widehat{\mathbf{v}}_k \rangle^{\nu} \tag{2.51}$$

where R_k is the red component of the diffuse reflection coefficient stored in the texture vector $\mathbf{T}$, k_s is the specular reflectance, ν defines the angular distribution of the specular reflections, $\widehat{\mathbf{v}}_k$ is the viewing direction, and $\mathbf{r}_k = 2 \cdot \langle \mathbf{n}_k, \mathbf{l} \rangle \, \mathbf{n}_k - \mathbf{l}$ is the direction of maximum specular reflection. Depending on the scanner's camera, the textures may be color or grayscale, and they may differ in overall tone. We apply gains g_r, g_g, g_b, offsets o_r, o_g, o_b, and a color contrast c to each channel. The overall luminance L of a colored point is $L = 0.3 \cdot L_r + 0.59 \cdot L_g + 0.11 \cdot L_b$

Color contrast interpolates between the original color value and this luminance, so for the red channel we set $r = g_r \cdot (cL_r + (1-c)L) + o_r$

Green and blue channels are computed in the same way. The colors r, g and b are drawn at a position (u, v) in the final image $\mathbf{I}_{model}$.

The overall optimization procedure (for shape and texture, as introduced by [BBPV03]), optimizes shape coefficients $\alpha = (\alpha_1, \alpha_2, \ldots)^T$ and texture coefficients $\beta = (\beta_1, \beta_2, \ldots)^T$ along with 21 rendering parameters, concatenated into a vector ρ, that contains pose angles ϕ, θ and γ, 3D translation $\mathbf{t}_w$, ambient light intensities $L_{r,amb}, L_{g,amb}, L_{b,amb}$, directed light intensities $L_{r,dir}, L_{g,dir}, L_{b,dir}$, the angles θ_l and ϕ_l of the the directed light, color contrast c, and gains and offsets of color channels $g_r, g_g, g_b, o_r, o_g, o_b$. Unlike [BBPV03], we keep the focal length f fixed now.

The main part of the cost function is a least-squares difference between the transformed input scan

$$\mathbf{I}_{input}(u, v) = \begin{pmatrix} r(u, v) \\ g(u, v) \\ b(u, v) \\ w_z(u, v) \end{pmatrix} \tag{2.52}$$

and the values $\mathbf{I}_{model}$ synthesized by the model

$$E_I = \sum_{u,v} (\mathbf{I}_{input} - \mathbf{I}_{model})^T \Lambda (\mathbf{I}_{input} - \mathbf{I}_{model}) \tag{2.53}$$

with a diagonal weight matrix Λ that contains an empirical scaling value between shape and texture, which is 128 in our system (depth is in mm, texture is in $\{0, ..., 255\}$.)

For initialization, another cost function is added to E_I that measures the distances between manually-defined feature points in the image plane, and the image coordinates of the projection of the corresponding, manually-defined model vertices. This additional term pulls the face model to the approximate position in the image plane in the first iterations. Its weight is reduced to 0 during the process of optimization. To avoid overfitting, we apply a regularization by adding penalty terms that measure the PCA-based

Mahalanobis distance from the average face and the initial parameters. The cost function is optimized with a stochastic version of Newton's method [BV99, BBPV03].

As result we obtain a textured 3D model from the linear span of example faces of the 3D morphable face model. The fitting procedure establishes point-to-point correspondence of the model to the scan, so we can sample the veridical cartesian coordinates and color values of the scan, and substitute them in the face model. The result is a resampled version of the original scan that can be morphed with other faces. We estimate and remove the effect of illumination, and thus obtain the approximate diffuse reflectance at each surface point. This is important for simulating new illuminations on the scan.

In Chapter 5, we make use of the illumination-corrected texture extraction to gain diffuse textures, and also experiment with the 3D fitting technique during data pre-processing.

3

Rapid Stereo-Vision Enhanced Face Detection

3.1 Motivation

In the past decade, digital photography has become broadly available to the public and thus has led to an increased quantity of digitally stored photographs. Digital cameras have gained importance in everyday life and are often available in today's smartphones or communication devices. The number of biometric access control systems and surveillance systems has also increased remarkably in recent years. As a consequence, developing face detection algorithms that quickly and reliably locate faces in images for a variety of purposes has become an active field of research.

In general, the goal of a face detection algorithm is to locate faces in images, regardless of their actual representation in terms of 3D pose, orientation, facial expression or e.g. lighting conditions. A wide range of face detection algorithms have been presented in the literature so far, many of them involving supervised or unsupervised machine learning methods. Generally speaking, these algorithms produce a face classification function by training on a set of annotated sample pictures. Given a photograph, this function is able to determine whether a region in an image contains zero, one or multiple faces. Depending on the algorithm, partially occluded faces may also be found. If a face has been detected, the coordinates of the surrounding bounding box are returned. Each bounding box then describes the exact coordinates of a single face in the image and allows further algorithms to be applied to the defined image region. One typical application of face detection algorithms is to anonymize data which is considered to be sensitive in terms of privacy. A well-known example of this is the large-scale privacy protection in Google Street View [FCA$^+$09a], where people's faces and also license plates of cars are automatically processed to guarantee anonymity. One major advantage of such applications is that they do not require real-time speed. Pictures may be processed offline. In surveillance or biometric access control, however, real-time analysis plays a major role. Also, it is crucial that the system cannot be bypassed by e.g. presenting a printout of a face to the camera. These systems thus require additional information about faces which goes beyond standard 2D projections. A straightforward solution to this would be to add 3D scanners to the acquisition equipment. This, however, is expensive, and the increased amount of data requires higher computational effort, which makes the system slow. The goal is thus to find a suitable trade-off between processing speed and reliability (in terms of evaluating more than just pictures).

3.2 Introduction

This chapter addresses this challenge and presents a real-time face detection algorithm that combines 2D detection with 3D evaluation. The system works with standard 2D equipment (cameras) and yields real-time results.

The algorithm improves state-of-the-art 2D object detection techniques by additionally evaluating a disparity map, which is estimated for the face region using a calibrated stereo camera setup: First, faces are detected in the 2D images with a trained rapid object classifier based on Haar-like features (Section 3.4). In a second step, falsely detected faces are removed by analyzing the disparity map.

In the near field of the camera, a classifier is used, which evaluates the eigenfaces of the normalized disparity map (Section 3.6). Thereby, the transformation into eigenspace is learned off-line using a principal

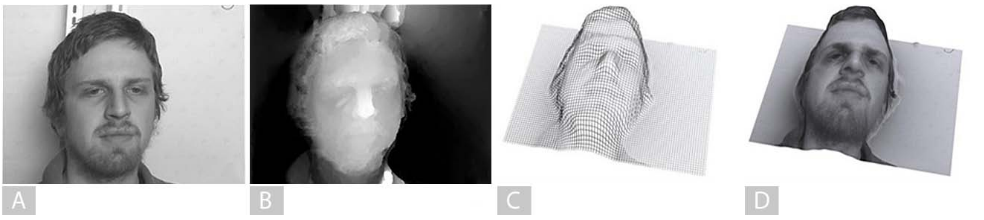


Figure 3.1: Quality of the disparity maps: (A) left frame of a stereo image, (B) reconstructed disparity map, (C) corresponding depth-map on a 3D grid, (D) reconstructed 3D head.

component analysis approach. In the far field, a much simpler approach determines false positives by evaluating the relationship between the size of the face in the image and its distance to the camera (Section 3.7). In Section 3.8 we present the results of our method, followed by a discussion.

This novel combination of algorithms runs in real time and significantly reduces the number of false positives compared to classical 2D face detection approaches.

3.3 Related Work

Face detection is often the first step in complex image processing applications, like face recognition, visual surveillance, or human-machine interaction. This explains the high level of interest of the research community in this topic. Many solutions to detecting faces in images have been presented in the last decade. A survey may be found in Yang et. al. [YKA02]. The authors differentiate between four major categories of algorithms: (1) Knowledge-based methods, (2) feature invariant approaches, (3) template matching methods and (4) appearance-based methods:

(1) Knowledge-based methods involve knowledge about faces as gained from heuristics, describing, for example, the relative positions of facial features (such as eye-to-eye distances or similar rules [Kan73, YH94, KP97]). An essential disadvantage of these methods, however, is that they are often sensitive to any kind of image deformation. They also might not work for non-frontal faces or partially occluded faces. Their crucial advantage, though, is that they are computationally inexpensive and often straightforward to implement.

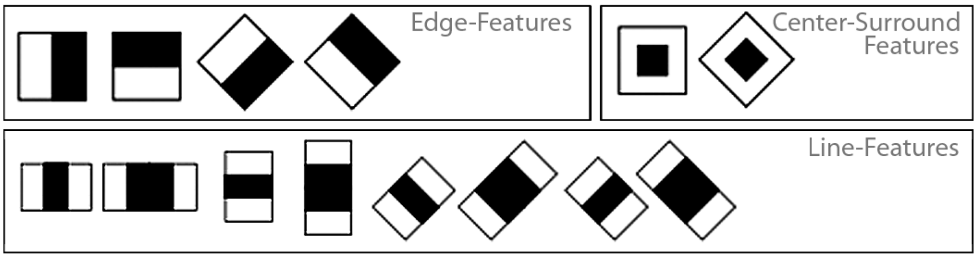


Figure 3.2: Haar-like features as used for the training of the frontal face-detector for monocular images. The features are sensitive to edges, bars, and other simple image structures and require low computational effort compared to higher-order feature sets.

(2) Feature invariant approaches describe faces in terms of selective features. The goal of the feature representation is to describe faces decoupled from pose, viewpoint or lighting changes. In contrast to the algorithms in (1), these methods are also applicable if the acquisition does not guarantee comparable pose or lighting. The type of features may vary; feasible ones can include color-based features (e.g. [YW96, MGR98, JP97]), features describing edge-structures (e.g. [YC97, LBP95, BLK10]) or comparable descriptors (see [YKA02] for a detailed overview). Feature based methods may be combined with statistical methods, such as PCA or ICA, to obtain good results.

(3) Template matching methods describe faces in terms of descriptive templates. These may be for example two-dimensional shape or feature templates ([CTB92, MP11]), or high-level representations such as Active Shape Models ([LTC94, ETC98, WAHL04]). Template matching methods often scale down to properly adjusting a set of parameters, but in general yield reliable results and are applicable to a wide variety of data.

(4) Appearance-based approaches involve high-level knowledge about faces, as learned from a set of training data. The models are learned off-line, on test data that represents the full dataset, in particular in terms of variation. Though active appearance or morphable models require relatively high computational costs in the design phase, they

have proven to generalize at a high level and are thus suitable for a wide range of input data. They yield good results for both face detection and recognition tasks. The most common approach is the eigenface approach ([TP91, RBK98, BV99, VJ01b, VJ04, KS05, BSS07]).

Most methods described above analyze intensity or color images only, not making use of any three-dimensional information. Since our goal is to enhance reliable 2D approaches with three-dimensional techniques, we focus on a widely used machine learning approach that was presented by Viola and Jones [VJ01a, VJ04] in 2004.

Their method performs visual object detection, which is capable of detecting faces in images in real time. It employs a coarse-to-fine strategy, where a classifier is trained that selects a few critical Haar-like features from a large set and then combines increasingly more complex ones in a cascade. These features are applied to detect faces in the image domain whereby consecutively smaller image patches become relevant as the complexity of the features increases. The resulting classifier detects faces in given input images only if all patches of the respective region have passed each node of the classifier cascade. Each time a single patch fails in a node, the region will be rejected and the classifier proceeds with the next patch. In order to achieve real-time performance, the core idea is to eliminate non-face patches at early stages of the cascade, to compute the more complex features only for the most relevant image patches.

Lienhart et al. [LM02] introduced a novel set of rotated Haar-like features, which are more powerful, but still easy to calculate (see Figure 3.2). In combination with a superior post-optimization procedure based on gentle AdaBoost, they were able to decrease the false alarm rate significantly at given hit rates. Followed by an empirical analysis [LKP03], they also provided an implementation of their method as part of the OpenCV library [1]. As a consequence, many other detection approaches adopted the strategy of Haar-cascade classifiers in the recent years. Most of them concentrate either on a variation of the initial feature set [Jon03, VJ03, MKH05, LZZ$^+$06],

[1] http://opencv.willowgarage.com/wiki/

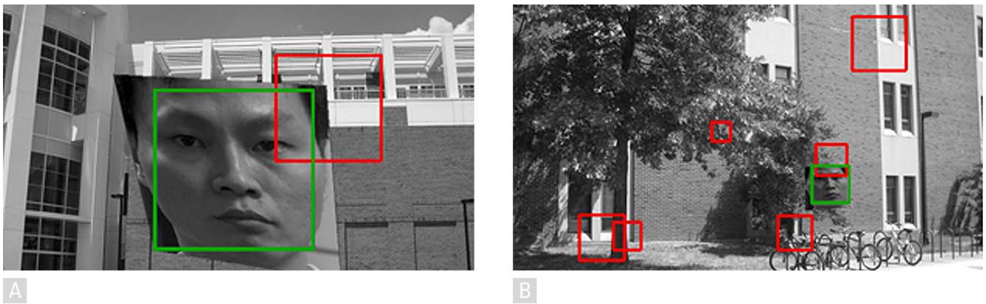


Figure 3.3: Haar-features are sensitive to detecting structures in images which are similar to borders or bars and (A,B) have proven to be a useful tool for many face detection tasks (green). However, Haar feature detectors often lead to a high number of false positives (B, here indicated in red), especially when the image background contains many bar-like structures.

on detecting faces across varying views [Jon03, LZZ$^+$06] or they use enhanced machine learning strategies, e.g., improved training algorithms or decision trees [LKP03, RRV04, WAHL04, HALL05a, MKH05, LZZ$^+$06, PC07, WBMR08, BWSM08]. Detailed overviews on recently proposed methods and feature sets for object or face detection, such as motion filters, edge based histograms or histograms of oriented gradients (HoG), may be found in [ZZ10] and [GB10].

In contrast to those methods, our approach does not work on monocular image sequences but instead processes synchronized stereo images. The approach can be divided into an off-line training phase and a real-time detection phase. For each stereo pair of the training images we train two different classifiers, one classifier on monocular frontal face images and one on the disparity maps, which we estimated from the stereo images. The classifier for monocular images follows the approach by Lienhart et al. with rotated Haar-like features, whereas the classifier that works on the disparity maps is trained by generating eigenfaces [TP91] using principal component analysis (Figure 3.7).

During real-time detection, we first apply the classifier that is based on monocular images and allows for a slightly higher rate of false positives, which is then checked by the disparity map classifier that eliminates the falsely detected positives. If the image patch containing the face is too

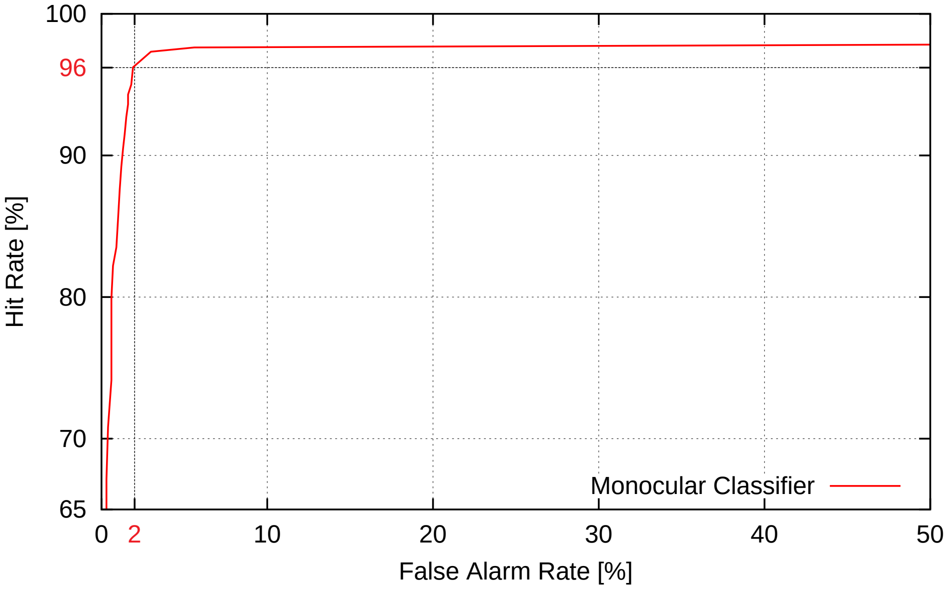


Figure 3.4: Excerpt of the receiver operator characteristic (ROC) for the monocular face detection classifier. At 2% false alarm rate, our trained classifier achieves a hit rate of 96%.

small, we revert to a simpler approach that analyzes the size of the face in the image in relation to its estimated distance to the camera.

This work is not the first to look into the combination of 2D image appearance and 3D depth maps from passive stereo. Especially for the application of face recognition, the benefit of additional 3D information has been shown before [DGHW00, TTS03, WLV04, SCLT07, BSS07]. A survey of recently presented methods in 3D and multi-modal 3D+2D face detection and recognition may be found in [BCF06]. Directly related to our approach, however, is the work by Wang et al. [WKS$^+$07, WLCV07]. They also present a real-time face detection system, which uses passive stereo and can additionally track and recognize faces. However, their approach uses a morphological filter in combination with some heuristics to detect the closest face to the camera in the depth map. As a result, only one face can be detected, whereas our multi-stage machine learning approach can detect multiple faces in the same image.

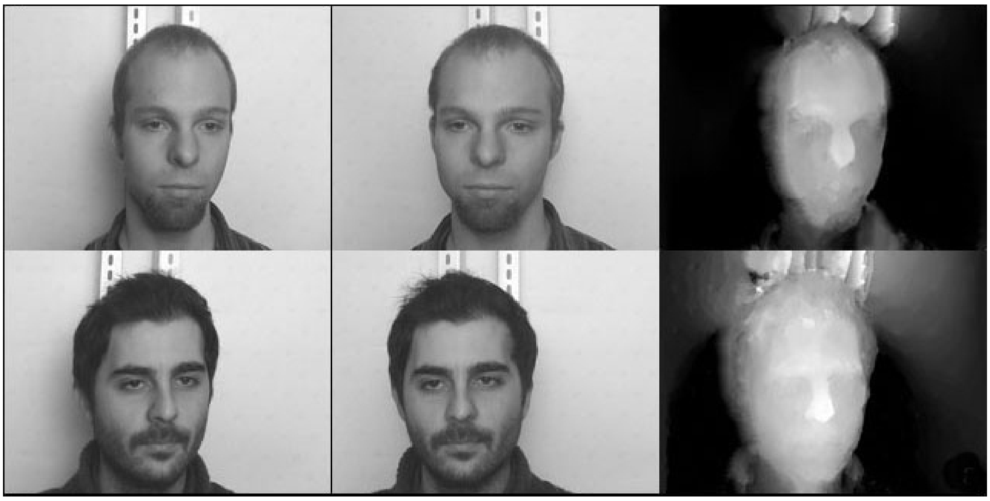


Figure 3.5: Stereo image pairs (left) and generated disparity maps from the PCA training set of faces (right).

3.4 Face Detector for Monocular Images

The first stage of our system uses a face detector that we trained on monocular images of frontal face views. Throughout training we follow the approach of Lienhart et. al. and use rotated Haar-like features (Figure 3.2).

Compared to high-level features, rectangular Haar features are relatively primitive and coarse but do not require high computational costs. Though providing only very limited orientations (vertical, horizontal and diagonal) they have their strengths in detecting structures such as borders or bars (compare Figure 3.3-B). Especially when analyzing images of faces at different scales, they have proven to reliably detect facial structures which (at small scales) appear to be similar to bar-like structures. The core idea is to build a cascade of many weak classifiers which become increasingly complex as the algorithm proceeds with the same image patch. Ideally, the weak classifiers reject non-faces at early stages in the decision tree such that only a few samples need to be tested using more complex feature sets at later stages.

During the training phase, we learned a classification function that builds the stages of the decision tree. More precisely, we trained a boosted classifier cascade, which uses rotated Haar-like features as introduced by Lienhart et al. [LM02]. Using a discrete adaptive boosting algorithm (AdaBoost), we trained on monochrome sets of 4700 positive samples (taken from [PFS$^+$05a]) and 3300 negative images (office scenes without faces). The positive samples were labeled manually and were cropped to contain 2D faces only. We combined both the negative and positive images, applying a set of random distortions, to build our training and test sets. An example of the resulting training data is shown in Figure 3.3 on page 52.

The resulting classifier cascade performs as shown in the receiver operator characteristic (ROC) in Figure 3.4.

3.5 Stereo Setup and Algorithm

Our stereo setup consists of two cameras with an image resolution of 384×288 pixels and baseline distance of 20 cm. After off-line calibration with a calibration pattern [Tsa87], we rectify the input images to standard stereo geometry and estimate a disparity map. Let the left picture be denoted by $I_l(x, y)$ and the right picture by $I_r(x, y)$. We then minimize the functional:

$$E(z(x, y)) = \iint_{\Omega} |I_l(x, y) - I_r(x - d(x, y), y)| \; dS \tag{3.1}$$

where $d(x, y)$ is the disparity at pixel $(x, y)^\top$. To find the minimum, we use a variational approach for disparity map estimation, which combines powerful tools such as regularization, automatic tracing and controlling the convergence of the iteration process with the help of Monte Carlo-based prediction techniques.

Though regularization of the disparity maps is applied, depth discontinuities are preserved. Recently, a fast implementation of this approach was presented [BWKS06]. Figure 3.1 shows an example of a disparity map estimation.

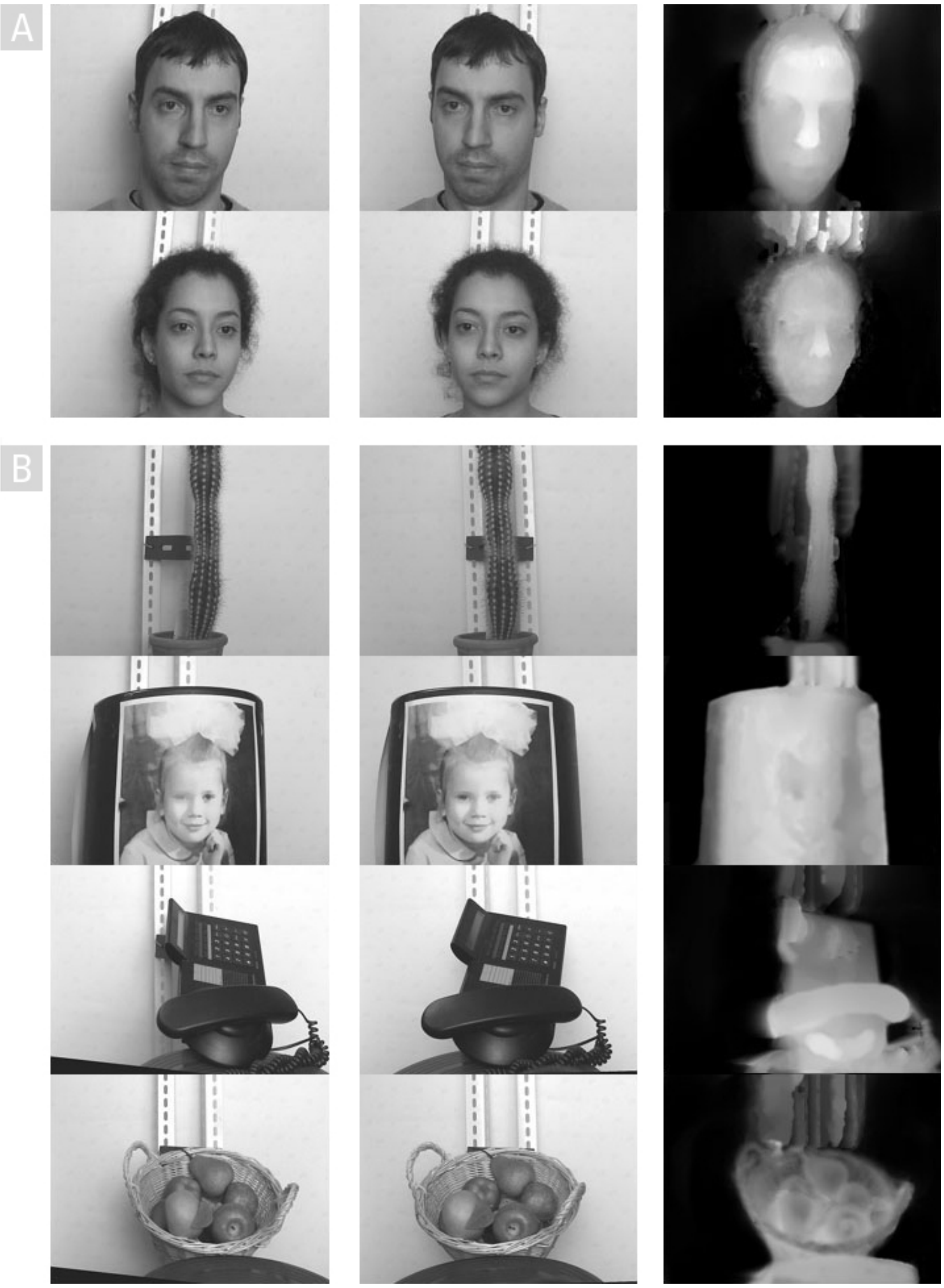


Figure 3.6: Stereo image pairs and generated disparity maps from the test set: (A) examples of faces, (B) examples of non-faces, e.g., only a picture or a sketch of a real face

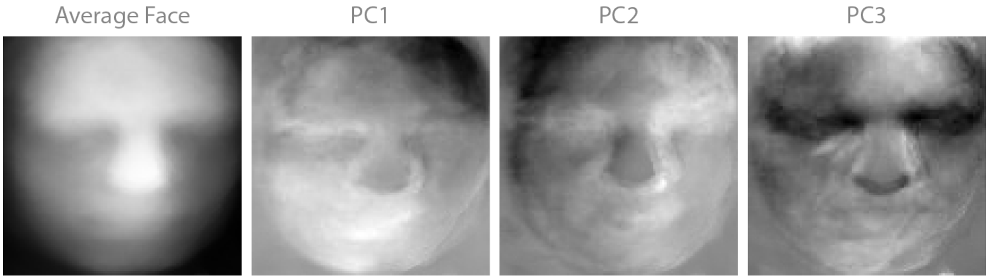


Figure 3.7: Results of the PCA on disparity maps: the average face (on the left side) and the first three eigenfaces (principal component (PC 1–3).

3.6 Near-Field Stereo-Enhanced Detection

It is known that boosted cascades of simple feature based detectors rapidly achieve high detection performances when applied to monocular images [VJ01a, LM02]. However, when applied to images that contain not only faces but also pictures of faces or simple face-like line structures, these detectors still show some false alarms (Figure 3.3 B), while on the other hand sometimes rejecting actual faces. One reason is that each stage of the cascade consists of a weak classifier that is based on Haar-like features. These are well known to be sensitive to edges, bars, and simple image structures. To overcome these problems, we employ an additional classifier that is not based on the appearance but evaluates the disparity map.

To detect faces in 2D images, we apply the trained classifier on the left image of all stereo pairs and thus identify several image patches that are potential face candidates. To evaluate these candidates in 3D, we perform a second classification step on the disparity maps. This PCA classifier is trained off-line on 30 facial regions, as shown in Figure 3.5, which were cropped out of the normalized disparity maps and coarsely adjusted according to a template disparity map applying planar rotation. The principal component analysis estimates the probability distribution of facial disparity maps around their average, as represented by the so-called eigenfaces (cf. [TP91]), which are shown in Figure 3.7.

During online detection, we evaluate whether a potential candidate, as selected by the 2D classifier, is a veridical face by projecting its signal

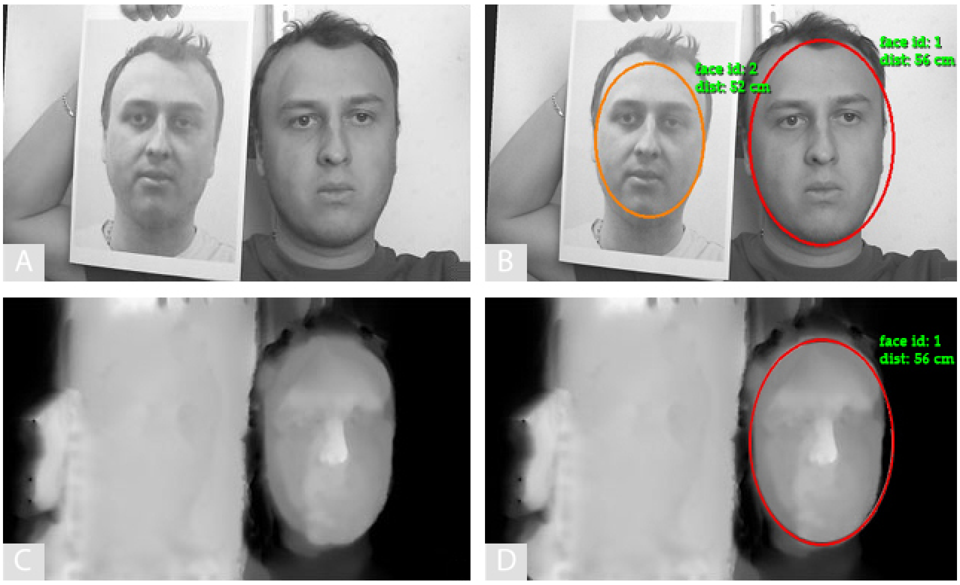


Figure 3.8: Detecting false positives with the near-field stereo-enhanced classifier: (A) input image, (B) two detected faces of the monocular classifier (true positive and one false positive), (C) calculated optical flow, (D) the stereo-enhanced classifier dismisses the false positive ($\sigma_{TP} = 3.27$, $\sigma_{FP} = 6.35$).

into PCA space. Considering the variances of the learned probability distribution, we calculate the Mahalanobis distance of the current test signal to the average face. By choosing a threshold of $\sigma = 3.29$ in terms of standard deviation σ, we accept all face candidates that lie within the range of $\approx\ 99,9\ \%$ of all training samples. Face candidates that lie above this threshold will be rejected as non-faces.

3.7 Far-Field Stereo-Enhanced Face Detection

If a face is detected in a small image patch, the face detection algorithm described above fails because a reliable disparity map cannot be estimated. Therefore, if the patch size of a face is smaller than 40×60 pixels, we revert to a simpler alternative to detect false positives generated by the classifier for monocular images. For each detected face region, we detect feature points and estimate their disparity to the right frame with the Kanade-Lucas-Tomasi approach [ST94]. Afterwards, the median disparity is calculated

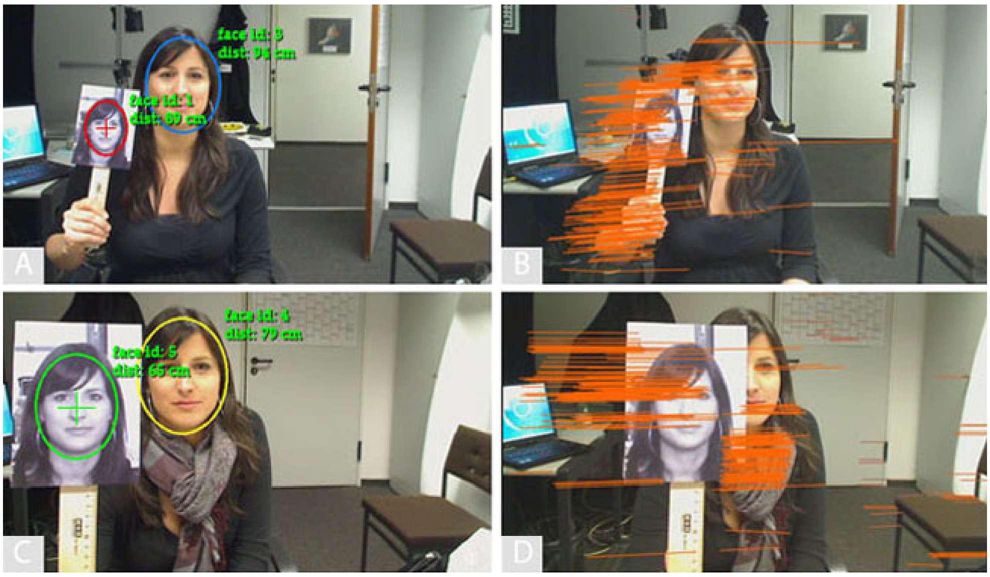


Figure 3.9: Detecting false-positives (marked by crosses) with the far-field stereo-enhanced classifier: (A) left frame with detected faces, (B) right frame with overlaid feature point disparities, (C,D) second example, where the faces have similar image sizes but different depth

within the detected face region, and the corresponding distance to the camera is calculated with the known calibration data of the cameras. From the size of the face in the image and the distance, the actual size of the face in 3D can be determined. The size is then checked against an interval of sizes for veridical faces, which is learned from a database of 3D face scans [SSSB07, PFS$^+$05a]. If the face size is smaller than 15.78 cm or larger than 24.16 cm, the face is marked as a false-positive and is dismissed.

3.8 Experiments and Results

We tested our stereo-enhanced face detector on a large number of input samples, which were different from the samples used during training. In that set, 40 samples contained one or more faces and 19 images contained some other non-face object. Figure 3.6 shows a few examples of face and non-face samples used in the test set. Table 3.1 compares the detection rates of the monocular detection approach with those obtained with our stereo-enhanced method.

	40 Faces			19 Non-Faces	
	FP	TP	FN	FP	TN
monocular	7	38	2	5	14
stereo enhanced	0	38	2	0	19

Table 3.1: Comparison of the detection rates of the monocular detection approach with our stereo-enhanced method. The two shaded columns show that the number of false positives (FP) decreases using our method, and the number of true positives (TP) and false negatives (FP) stays the same.

Step	**FF [msec]**	**NF [msec]**
Run monocular detector	27	27
Estimate disparity map	-	78
Transform into eigenspace	-	42
Estimate sparse disparity map	63	-
Total	**90**	**147**

Table 3.2: Timings for an Intel® Core™ 2 CPU with 2,66 GHz. The monocular detector must be run once per image; the other algorithms must be run once per detected face. Here FF and NF denote far-field and near-field approach, respectively.

All false-positives could be removed by taking the additional 3D information into account. The number of true-positives and false-negatives stayed the same.

Figure 3.8 and Figure 3.9 illustrate the shortcomings of the classical monocular approach because it cannot distinguish a real face from a photo print. Our improved detection approach dismisses the false detection with the near-field stereo-enhanced classifier, Figure 3.8, and the far-field stereo-enhanced classifier in Figure 3.9. In Table 3.2, timings for the different steps of our algorithms are given. For a scene with a single face, a frame rate of 11 fps for the far-field approach and 6 fps for the near-field approach can be achieved.

3.9 Summary, Discussion and Future Aspects

In this chapter a widely-used monocular face detector based on a trained Haar-feature cascade is extended by an additional classifier that evaluates the disparity map of a passive stereo camera. The algorithm runs in real time and significantly reduces the number of false positives compared to the monocular approach. In fact, as our test set is rather small (40 faces and 19 non-faces samples), all false positives could be removed in our experiment.

Though these results are already promising, a larger test set would be helpful to evaluate whether our findings generalize also for large scale databases. Also, we currently combine 2D and 3D classification only in a straightforward manner. Recent studies, however, propose more sophisticated techniques for non-maximum suppression which could further stabilize our results ([AMM10, Bla11, ZV12]). In addition, our system is currently limited to only detect frontal faces. A possible solution could be to train both, the monocular detector and the stereo extension, on different face orientations (as proposed in Chapter 4). This would result in a different detector for each orientation and could offer a wide range of applications. These improvements are left for future work.

4

Fast Face Detector Training using Tailored Views

4.1 Motivation

Face detection is an important task for a wide range of applications in computer vision. Thus, a variety of face detection algorithms have been presented in recent years, many of them involving supervised or unsupervised machine learning methods. Their goal is to learn a face classification function by training on a set of annotated sample pictures which is then applied to new, unseen pictures in order to detect faces. This, however, requires large amounts of manually labeled face pictures during the training phase, which is not only very time-consuming but also difficult to obtain. In particular as pictures of faces are considered to be sensitive in terms of privacy. Moreover, available data sets are often acquired under controlled settings, restricting, for example, scene illumination or 3D head pose to a narrow range. Or, in contrast, they may scatter widely but without a guarantee that they sample all relevant dimensions sufficiently densely. Also, manually labeled data often involves mistakes. When training multi-view classifiers for example, the manually labeled viewing angles might be inaccurate and thus lead to biased results. The following chapter addresses these shortcomings and takes a look into the automated generation of tailored training samples from a 3D morphable face model.

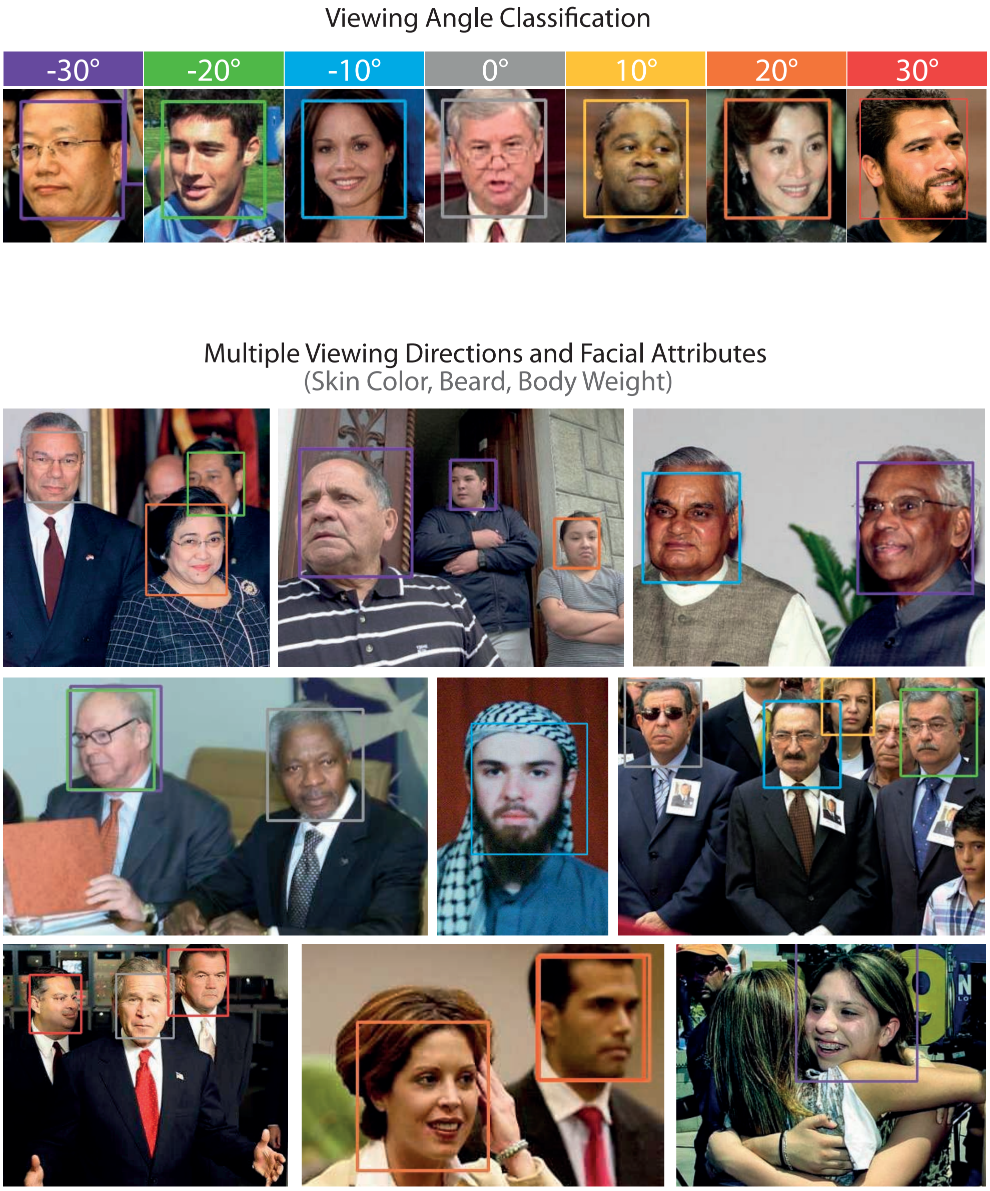

Viewing Angle Classification
-30°
-20°
-10°
0°
10°
20°
30°
Multiple Viewing Directions and Facial Attributes
(Skin Color, Beard, Body Weight)

Figure 4.1: Top row: Comparison of detection results at distinct viewing angles. The respective bounding box color denotes the automatically classified viewing angle $\phi \in [-30°; +30°]$. Bottom row: Exemplary detection results on the challenging 'FDDB' imagery [JLM10]. The algorithm robustly detects faces despite apparent variation of facial attributes such as increased body weight, beards, bright or dark skin color.

4.2 Introduction

To automatically compute training samples, we start from a few randomized facial samples which we gain from a 3D morphable face model [SSSB07, BV99]. Each 3D random face is then modulated by automatically changing a set of facial attributes such as gender, weight or age. For each modulated face, we either freely choose the settings for rendering, such as the parameters for 3D pose, position, size and illumination, or extract these parameters from given particular video sequences recorded by surveillance cameras. The latter method in particular can be helpful when training classifiers for cameras that are positioned at unusual viewing angles or in a very dark or artificially lit environment. Using this procedure, we generate seven distinct training sets, each set corresponding to a specific range of face orientations. On each training set we then train a view-specific classifier using an adaptive boosting (AdaBoost) approach which we derived from the object detection method Viola and Jones initially presented in 2001 [VJ01a]. In contrast to their approach, we implemented the algorithm to run in a many-core setup that also extends the feature plane by an arbitrary number of additional layers, which may be used either for color representation or further feature sets.

Finally, we merge our seven view-dependent classifiers to a single classifier and compare its performance to state of the art methods on the FDDB benchmark dataset [JLM10].

4.3 Related Work

Face detection is often the first step in complex image processing applications, like face recognition, visual surveillance, or human-machine interaction. This explains the high level of interest of the research community in this topic. Many solutions to detecting faces in images have been presented in the last decade. Comprehensive surveys may be found in Yang et. al. [YKA02, ZZ10]. Out of the presented approaches we focus on the widely used appearance-based machine learning approach that was presented by Viola and Jones [VJ01a, VJ04].

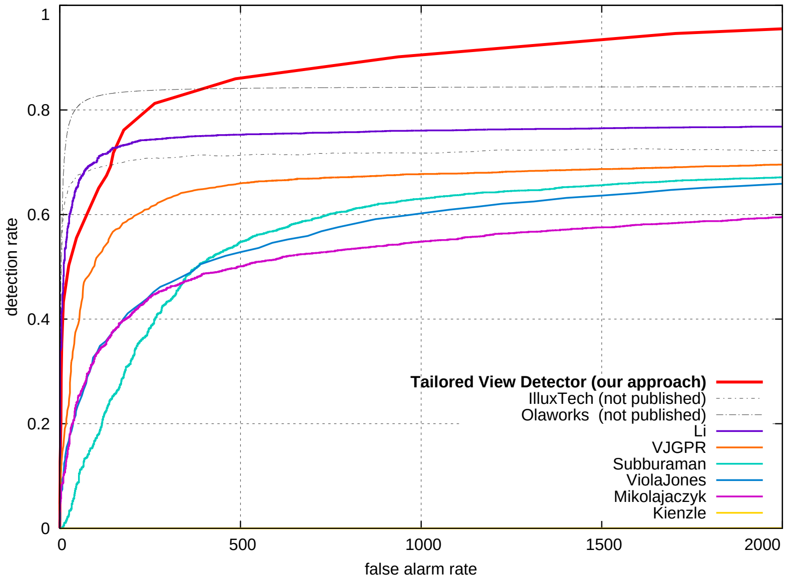


Figure 4.2: Comparison of the detections at equal error rate. Training on the synthetically tailored views of 75 distinct subjects, is sufficient to outperform previously presented face detectors on the challenging Face Detection Data Set and Benchmark (FDDB) [JLM10].

Object Detection Framework The algorithm is based on the assumption that many coarse Haar feature classifiers, connected in series, are superior to a single classifier built with high-level image descriptors. The coarse classifiers are organized hierarchically, where the number of computed Haar features tends to increase with each stage. While in the first stage only a few Haar features are computed, each following stage has stricter requirements and usually requires more features. At training time, the AdaBoost algorithm determines a constant threshold for each stage, to which the candidates can be compared at detection time. Image candidates passing all stages are considered to contain a face. Negative images exit at earlier stages. The overall number of Haar features varies and depends on the training parameters.

Detection Performance Using the integral image structure [VJ01a] Haar features can be computed quickly. On mobile devices or when applied to video sequences, however, performance may decay drastically. Consequently, a variety of algorithmic performance improvements has been presented in literature. Noticeable speedups have been achieved when running Viola and Jones' detector on several GPUs [1] [KD10, NHO$^+$13] or by combining GPU and CPU [2] [STVK09, WM11]. Chuang et. al. [CJC13] introduced an enhanced training algorithm considering sampling optima for video material. Others explored the possibilities of parallelism using many-core architectures, improved memory behavior or investigated how to optimally compute the integral image [ALT08, CMOK09, CBMK09, Zha10, PGHC10, CKLL11, WWC$^+$11, LCL11]. All above methods reported considerable computational speedup, but only at runtime. We, in contrast, parallelize the AdaBoost at training time. Though we additionally introduce new layers (and thus more features) for the use of color channels, the many-core architecture allows for fast large-scale training.

Acquired Training Data Data-driven face classifiers require appropriate training data. There are many supervised or unsupervised solutions to image annotation, such as collaborative annotation projects [vAD04, EZW$^+$05, RTMF08, TFF08, DDS$^+$09, WGHG10], or algorithmic approaches for Google's image search [FFFPZ05], web content [JLM03, SRE$^+$05, PSLS09, MPK10, TJL$^+$11] or labeled social media content [DG10, CHL11]. As a result, a large variety of manually labeled datasets has been published. Comprehensive surveys of facial datasets can be found in [Gro05, Grg13, Fri13]. Other challenging datasets can be found in [PFS$^+$05b, HRBLM07, JLM10]. Out of these, the face dataset of the MPI for Biological Cybernetics is the most related to our work. However, their dataset does not cover the full spectrum of statistical data variability and does not include facial attributes such as age or body weight.

Synthetic Training Data In general, manually collected facial datasets show insufficient data variability. To increase variability, people usually collect large amounts of images. Automatically synthesized training data, by contrast, involves high-level face models to increase variability: In 2004

[1] graphics processing unit (GPU)
[2] central processing unit (CPU)

Yue-Min et. al. [LCQ+04] relit faces in training images using harmonic images they derived from a 3D face model. Dianle et. al. [ZPDD10] use an active shape model to synthesize training data. The data is then used to find facial landmarks in different views. A variety of face recognition systems [TH95, BGPV05, RT05, TA07, CKO+08, DXQ09, WDDF09, AMAM11] use 3D models to synthesize intermediate views or viewpoint invariant reference frames for the purpose of face recognition. A more comprehensive survey on similar methods can be found in [BCF06]. Pishchulin et. al. [PTW10, PJW+11] use a morphable body model to generate training data for pedestrian detection. They could show that even a low number of synthetic training samples — with increased data variability — can outperform detectors trained on large manually collected data sets. Similar to our work, Weyrauch et. al. [WHHB04] use a morphable model for pose invariant face recognition. From three input images of each subject in the training database, a 3D model [BV99] is extracted. The model is then rendered under varying pose and illumination conditions to build a set of synthetic images, used for training a component-based face recognition system. In contrast to their method, we focus on face detection, and show that state-of-the-art results can be obtained by leveraging tailored training data from a 3D morphable model. We do not require initial facial input images, but randomly generate artificial faces while controlling the data variability. Moreover, we introduce facial attributes such as body weight or skin tone and make use of an advanced face model [SSSB07] to render the subjects' ages. We also show that our approach is suitable to adjust rendering parameters to particular illumination and pose constraints of given surveillance cameras (Figure 4.10).

4.4 Synthetic Training Images

When manually labeling and selecting training images of faces, there is no guarantee that the collected data includes all possible shape and texture variations of faces. Consequently, people usually collect large amounts of training data to at least sample as much variation as possible. Examples can be found in [Gro05, Grg13, Fri13]. When generating synthetic

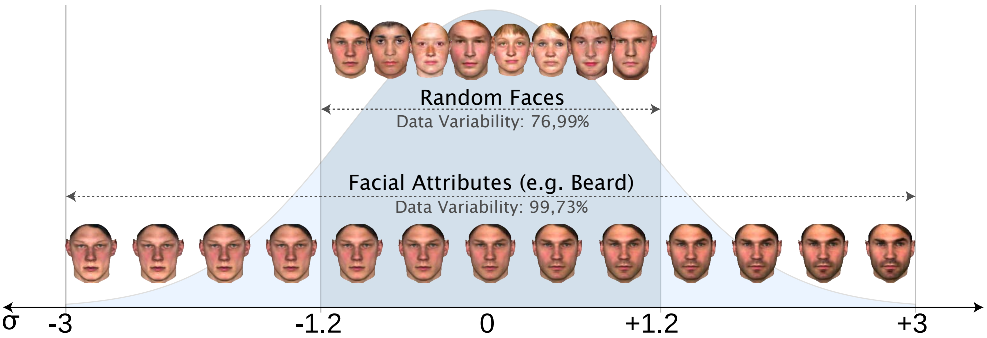


Figure 4.3: (Top row:) First we generate random faces by modulating existing faces from the database permitting 76.99% of all possible variations ($\sigma \in [-1,2;+1,2]$) which we secondly modulate applying attributes at full data variability ($\sigma \in [-3;+3]$, bottom row).

training data, in contrast, this workaround is not necessary. By deploying a statistically driven face model for data generation, one can be sure to incorporate the full data variance (with respect to the database of the face model). We use this technique in the following section to first generate randomized 3D faces using a 3D morphable face model [BV99, SSSB07]. In a second step, we modulate these three-dimensional random faces by applying facial attributes, which we then render for defined viewing angles and illumination parameters. Finally, we compose the face renderings with natural-looking background images.

Modeling To generate artificial training data we employ a 3D morphable face model [SSSB07, BV99] as described in Section 2.4. The model's database contains $m = 512$ faces ranging from the age of 3 months to ≈ 40 years with an approximately equal number of female and male individuals (200 adults, 236 children aged between 7–16 years and 76 very young children aged between 3–12 months). The 3D shape of each face F_i is stored in terms of the x, y, z coordinates of all surface vertices $k \in \{1, ..., n\}, n = 75972$ in a vector S_i. Analogously, we store the color values (red, green and blue) of the surface vertices in a texture vector T_i:

$$\mathbf{S_i} = (x_1, y_1, z_1, \ldots, x_n, y_n, z_n)^T \tag{4.1}$$

$$\mathbf{T_i} = (R_1, G_1, B_1, \ldots, R_n, G_n, B_n)^T \tag{4.2}$$

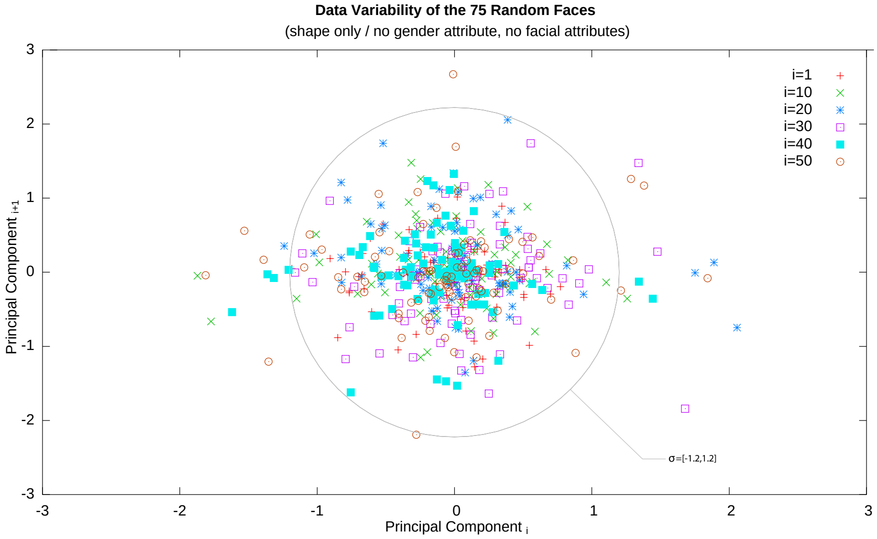


Figure 4.4: First we generate 75 randomized 3D faces by manipulating faces from the 3D morphable model face database deploying a varying factor to the first 50 eigenvectors ($\sigma \in [-1.2; 1.2]$). The result corresponds to 76.99% of all possible variations within the morphable model's database. The distribution of the respective principal components is shown above.

Performing a Principal Component Analysis on all shape and texture vectors we estimate the probability distributions of faces around their averages $\bar{s}$ and $\bar{t}$. The result is a small set of $(m-1) = 511$ orthogonal principal axes (eigenvectors) s_j, t_j which vary around the averages $\bar{s}$ and $\bar{t}$:

$$\mathbf{S} = \bar{\mathbf{s}} + \sum_{j=1}^{m-1} \alpha_j \cdot \mathbf{s_j}, \qquad \mathbf{T} = \bar{\mathbf{t}} + \sum_{j=1}^{m-1} \beta_j \cdot \mathbf{t_j} \tag{4.3}$$

The eigenvectors of the PCA represent the variation across all faces in the database. Most eigenvectors do not explicitly represent semantically meaningful facial features. By definition of the principal component analysis, however, the eigenvectors are sorted according to their corresponding eigenvalues in decreasing order. Thus the highest variation between all faces in the database is represented by the frontmost eigenvectors. In the following analysis, we therefore only consider the first eigenvectors $(j <= 50)$ for shape s_j and texture t_j to control the variance of the com-

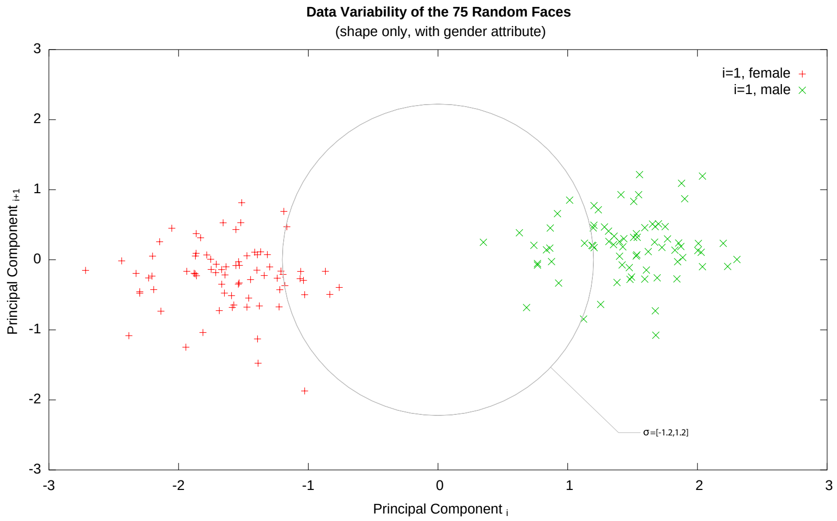


Figure 4.5: For each random face computed we generate a female and male version, resulting in a total of 150 faces. The Figure above shows the clustered distribution of the respective first principal components.

puted training data. The rear eigenvectors $(j > 50)$ contain highly individual details which are not relevant for our purpose and thus may be ignored.

In a first step, we generate a set of randomized 3D faces by manipulating available 3D faces from the 3D morphable model face database. We then apply shape and texture variation to the selected samples by deploying a varying factor σ to the first 50 eigenvectors s and t ($\sigma \in [-1.2; 1.2]$). We initially keep the scaling factor σ relatively low with respect to the available variability to prevent producing unwanted artifacts which would require a manual quality check (cf. Figure 4.3). Modifying the first 50 eigenvectors only serves as regularization and prevents enhancing highly individual details. The results are randomly generated faces (or 'random faces') which serve as the basis for all following steps (Figure 4.4). We additionally require a large angle (Mahalanobis Distance) between computed sample face vectors to ensure low similarity in-between the random faces.

Modulation In a first step we generate a female and male version for each random face (Figure 4.5). To these we subsequently apply a set of facial attributes such as increased or decreased body weight or, for example, light or dark skin color (Figure 4.7).

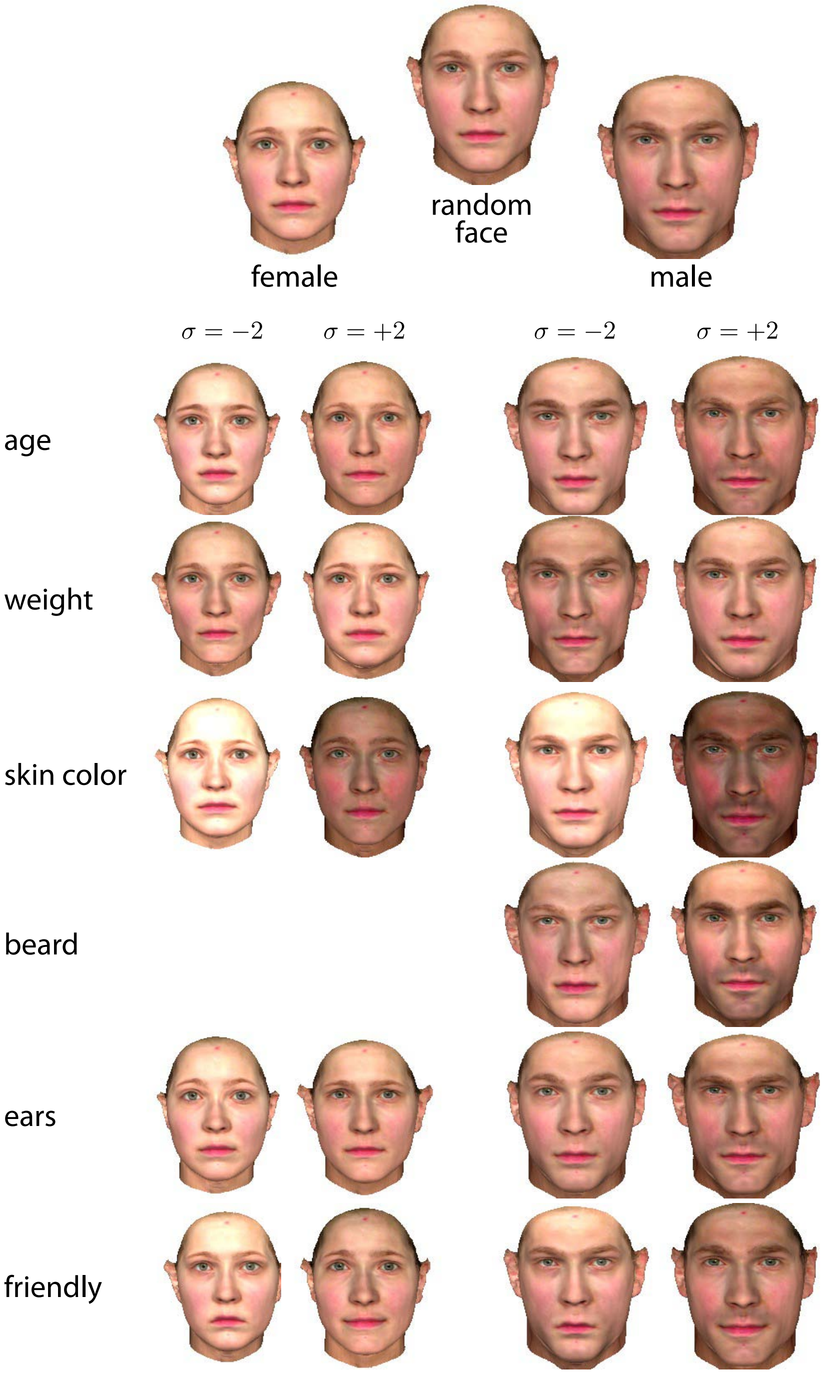


Figure 4.6: Generation of synthetic face data: from an initially generated random face (top middle), we first derive a female and male version (second row) and then apply attribute-vectors to those to modulate the age, weight, skin color, beard shadow, ear size and the facial expression. The above figure shows the modulation results for $\sigma = \pm 2$.

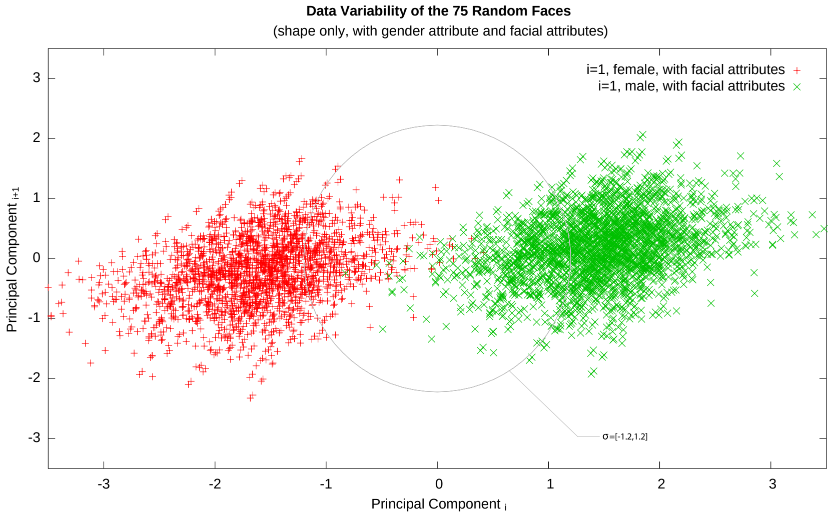


Figure 4.7: In a third step we apply a set of facial attributes to all female and male random faces. We apply these attributes using a scaling factor $\sigma \in [-3.0; 3.0]$, corresponding to 99.73% of all possible variations within the morphable model's database (full data variability).

We learn these attributes in a step prior to the stimulus generation procedure, so they can be applied automatically to each face. The learning procedure ([BV99]) involves manual labeling of each database face according to the perceived strength of the attribute in each face. Each face F_i is attributed a scalar value μ_i. Then, we fit a linear function f to these data that reproduces the labels μ_i. Following the gradient of f in PCA space will then produce a perceived change of the attribute strength in a given face, while all other individual characteristics of the face remain unchanged. Please note, that this has to be done only once for each attribute, regardless of how many data sets for training are required. We performed this method for the following facial attributes:

- age (young / old)
- body weight (obese / skinny)
- beard shadow (dark shadow / no shadow)
- skin color (light / dark)
- ears (big / small)
- one facial expression (friendly / unfriendly)

We apply these attributes to all random faces using a scaling factor $\sigma \in [-3.0; 3.0]$, corresponding to 99.73% of all possible variations (cf. Figure 4.3). Results of the modulation are shown in Figure 4.6, where we rendered all facial attributes for the values $\sigma = \pm 2$.

Rendering In the next step, we transform all previously generated 3D faces into 2D representations. To obtain an image $I_i(x, y)$ from a given 3D face F_i, we apply standard computer graphics procedures:

$$R_\rho(F_i) = I_i(x, y) \tag{4.4}$$

The resulting images depend on a set of rendering parameters ρ, where the respective number of parameters is given in brackets:

- 3D rotation (3)
- 3D translation (3)
- focal length of the camera (1)
- angle of directed light (2)
- intensity of directed light (3)
- intensity of ambient light (3)
- color contrast (1)
- color gain in each color channel (3)
- color offset in each color channel (3)

All faces are rendered at seven predefined viewing angles. We thus generate a total of seven distinct training sets that all differ in the chosen face viewing angle ϕ, where

$$\phi \in \{-30°; -20°; -10°; 0°; +10°; +20°; +30°\} \tag{4.5}$$

In addition, to achieve naturally varying results, we apply slight random variations while rendering. Within each set, we modulate for example the left-right viewing direction of the face ($[-3°; 3°]$), the up-down angle ('nodding', $\theta \in [-15°; 15°]$) and the in-plane rotation ($\gamma \in [-5; 5]$). All above values are motivated by heuristics. To simulate various environments, we additionally modulate color and intensity of the ambient light and apply random variations to color contrast, color gain and offset. Finally, we render a segmentation mask for each face. For each of the resulting training sets we later train a single face classifier which we combine to a multi-view face detector in the end.

Compositing In a final step we blend the rendered faces with background scenes that do not contain any faces ('negatives'). We use background images that are randomly sampled from the 'PASCAL2 Visual Object Classes Challenge 2012 (VOC2012)'[3] image collection. To exclude apparent faces from the selected images, we initially remove all images that are labeled to contain humans or human faces. The remaining images typically show outdoor scenes, urban scenes or any other human environment (cf. Figure 4.8, top row). We blend each rendered face with a random background and apply a smooth contour blend (Gaussian blur) using the rendered contour mask at the facial contour. In addition, we randomly vary the size and position of the face in the background image and then store its rectangular coordinates as ground truth. In a final step we randomly scale the composed images ($> 20 \times 20$ pixels). Each composed image contains exactly one individual face in the end. Images containing at least one face are also referred to as 'positives'. Figure 4.8 illustrates examples of the composed positive training images.

4.5 Enhanced AdaBoost Training

Using the above-described synthetic training data, we train a face classification system. Given an image patch, the system should determine whether the patch contains faces or not and should locate potential faces in the image. Moreover, the system should work across varying image sizes, depths and resolutions.

The above requirements are satisfied by the OpenCV[4] implementation

[3] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2012/

[4] http://opencv.willowgarage.com/wiki/

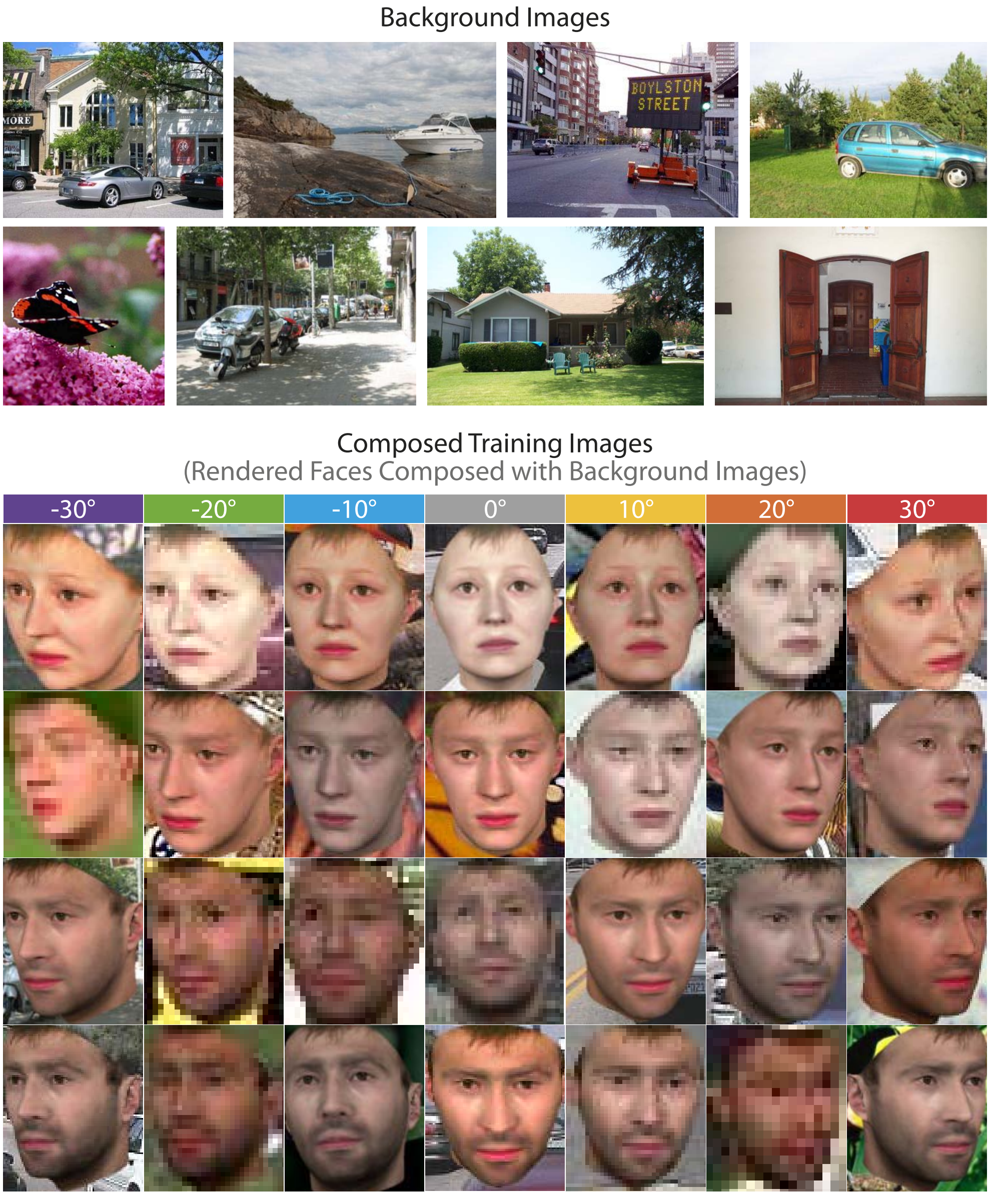


Figure 4.8: (Top:) The background scenes used for image compositing are randomly sampled from the PASCAL2 Visual Object Classes Challenge 2012. The scenes typically show outdoor scenes, urban scenes or any other human environment. (Bottom:) Each row shows the various renderings of a single manipulated random face. In a final compositing step we randomly scale the size of the rendered faces to simulate typical surveillance recordings and blend the rendered views with background scenes.

of Viola and Jones' AdaBoost framework [VJ01a, VJ04]. We thus mainly base our system on this algorithm. The core idea of the method is to concatenate a series of weak Haar filter classifiers rather than finding a single high-level descriptor of faces in images. The boosting algorithm proceeds in stages, where at each stage a new weak classifier is determined. Concatetenating all stages to a cascade yields the final detector. While the algorithm performs very well during testing phase (real-time), the training phase can quickly turn slow and tedious, especially when training on numerous images ($>$3000). One reason is that per stage, a very high number of features has to be computed, evaluated, selected or finally discarded. This makes the training procedure very slow and can become very impractical when it comes to determine proper training parameters. One can easily spend days to weeks tweaking parameters. Another drawback of Viola and Jones' method is that only greyscale images are processed as opposed to recent approaches that have shown that color information may improve face detection results ([FCA$^+$09b]). To overcome these drawbacks we present two major adaptions to the OpenCV system: Firstly, we introduce new feature layers that can be used either for color channels or arbitrary descriptors and secondly, we parallelize the complete training procedure to run on many-core architectures. Despite using more feature layers (or colors) we could thus tremendously speed up the training procedure.

Parallelization Sharing the load among multiple CPUs allows for very fast training procedures and makes it possible to train on thousands of images, rather than a few hundred, in a very short time. In our tests this decreased the training time by a factor of 5.3 using eight cores versus a single core. Parallelizing the AdaBoost training procedure [VJ01a, VJ04] is not straightforward, however, since the algorithm is sequential by nature. Among the non-sequential parts, the most expensive step is the computation of features on every image patch, per step. Each time a massive number of features has to be computed and among these, the most descriptive ones have to be selected. We speed up these processes by introducing the following improvements:

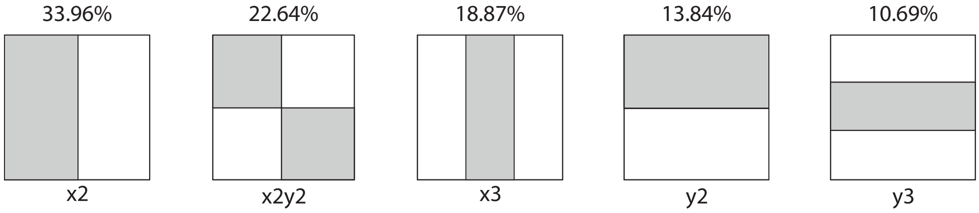


Figure 4.9: Histogram of the selected Haar features at training time. Haar feature $x2$ is selected in $\frac{1}{3}$ of all cases.

a) We precompute a subset of the features in each CPU and keep them in memory (no synchronization necessary).

b) We parallelize the feature selection at each AdaBoost iteration (involving a lightweight synchronization with the master CPU).

c) We parallelize the negative patch selection (for which the classifier fails) as this procedure massively slows the algorithm down at the later stages (on the order of hours). Each CPU now searches for negative patches in different subsets of the negative images, and the result is synchronized with the master CPU.

All modifications have been implemented using the OpenMP[5] API.

New Features Layers OpenCV's Viola-Jones implementation was initially designed to train on intensity images only. We, however, expect the color channels to contain relevant information for the detection of faces. Evaluation showed that by introducing additional layers the algorithm now finds the most descriptive features in the red (57.86%) channel, followed by the $blue$ (25.16%) and $green$ (16.98%) channels. Out of these, Haar feature $x2$ is the most frequently selected feature (cf. Figure 4.9). In future work, we are planning to exploit other color spaces suitable for skin-color modeling, as for example proposed by [TFAS00].

[5] http://openmp.org/wp/

View-Classification We moreover alter the training procedure: In contrast to common practice, where people use manually labeled and probably also biased data, we know the exact viewing angle of each face in our training sets, allowing us to train view-dependent detectors rather than a generalized single classifier. However, during the detection phase one might want to detect all faces in an image, regardless of their viewing angles. For this purpose we later recombine the view-dependent cascades to a single multi-view classifier that is capable of finding faces at any viewing direction in the image (cf. Figure 4.1).

4.6 Evaluation and Results

Overall we trained seven distinct classifiers, one for each of our seven synthetic training sets. For each classifier we trained on 5000 positive and negative samples. After training, we evaluated the resulting classifiers according to the Face Detection Data Set and Benchmark (FDDB) as recommended by Jain and Learned-Miller in 2010 [JLM10].

The FDDB benchmark dataset comprises 2845 images with a total of 5171 annotated faces. Within the dataset a considerable number of challenging pictures can be found. Examples are challenges such as low resolution faces, out-of-focus faces, occlusions or difficult and unusual face poses. Please also note that the FDDB-benchmark framework requires evaluation in terms of a tenfold cross validation per definition.

Before evaluating our classifiers according to the FDDB standard, we first align all seven classifiers. We therefore apply a stage threshold bias shift to all seven detectors such that they all start with zero false alarms at the same point in the receiver operator characteristic (ROC). After threshold adjustment, we build a cascade of all seven classifiers, which — for each image patch — searches for faces at the respective viewing angles. We currently combine the view-based detectors naively, more sophisticated methods, however, (e. g., vector boosting [HALL05b]) could lead to improved accuracy. The resulting cross-validated ROC curve is shown in Figure 4.2 on page 66.

The results indicate that training on statistically well distributed synthetic training data seems to be a promising concept: Though the method of Li et al. still appears partially superior to ours, our classifier could outperform most previously published methods, such as Kienzle, Mikolajaczyk, Subburaman, VJGPR and the standard Viola-Jones approach reported on the FDDB benchmark homepage [6].

4.7 Applications

Most presented face detection algorithms so far do involve a rather time-comsuming initial step: the collection and labeling of training images. This step is inevitable to achieve optimal detection results for a specific camera type or environmental setting. Detection accuracy is directly related to the quality of training data. Synthetic training data, in contrast, might overcome these drawbacks, and additionally offers a wide range of new applications:

Self-Learning Surveillance Cameras Surveillance cameras, as for example in large cities, are usually installed over the course of years and thus vary widely in terms of their intrinsic parameters (such as focal length or resolution). They are located all over the city, at varying positions, viewing perspectives, illuminations and environments. Regardless of this fact, surveillance systems often use the same detector for all cameras. Camera-specific properties are ignored and detection might fail, for example at unusual viewing perspectives like a bird's-eye view. Camera-specific detectors could be a solution to this. However, generalized detectors are still standard, since it would be too time-consuming to train hundreds of camera-specific face detectors.

Using our system, however, taking a few snapshots per surveillance camera is sufficient to train camera-specific detectors. With the help of little manual interaction (about seven clicks per sample face) we can extract the parameters ρ (cf. Page 74) from the camera snapshots . All parameters are estimated automatically in an analysis-by-synthesis loop which finds

[6] http://vis-www.cs.umass.edu/fddb/results.html

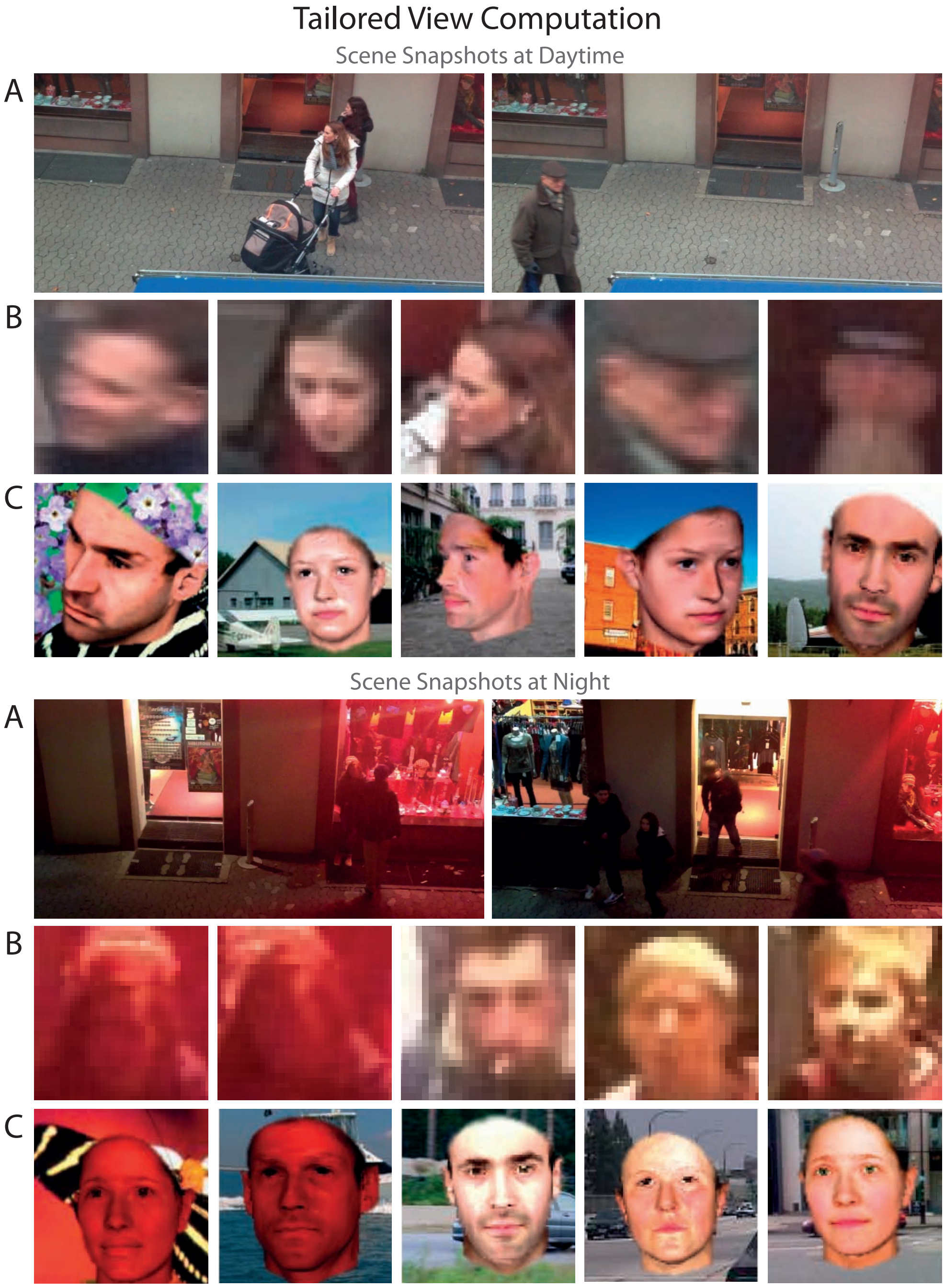


Figure 4.10: Shown is the tailored view generation for two different scenarios (scenery at day (top) and night (bottom). (A) still images recorded by a single surveillance camera. (B) From the recorded still images we manually extract a few cropped face images. We use these to automatically estimate facial illumination and pose parameters by fitting the 3D morphable face model to them. (C) Using the estimated light and pose parameters we render tailored training imagery showing arbitrarily modulated random faces.

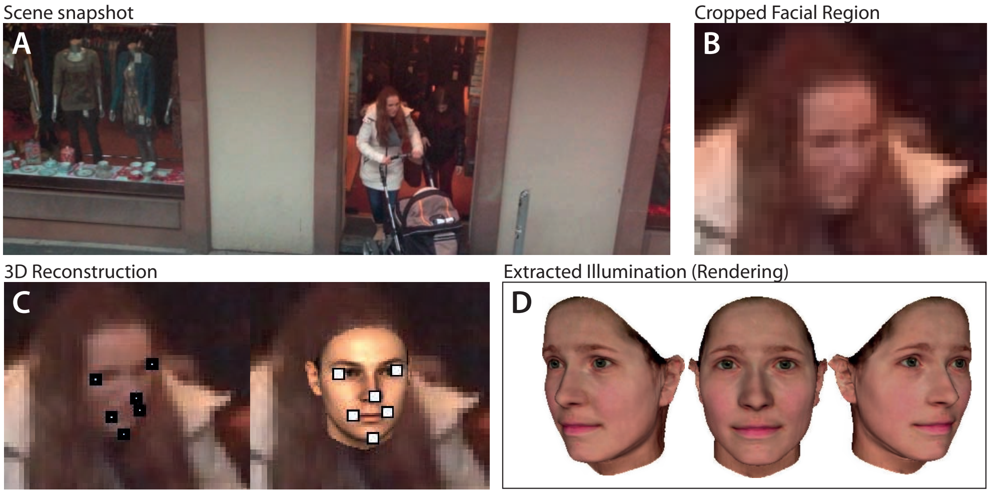


Figure 4.11: To extract illumination parameters we use scene snapshots recorded by surveillance cameras (A) and crop apparent facial regions (B). To these we fit the facemodel which involves little manual interaction as defining a few feature points on the image and the generic 3D model (C). Along with parameters for shape and texture, the fitting procedure estimates a set of parameters defining pose and illumination of the face in the scene. We use these to render arbitrary faces reproducing the facial illumination (D).

the parameters α, β, ρ that make the synthetic image $\mathbf{I}_{model}$ as similar as possible to the original image $\mathbf{I}_{input}$ in terms of pixelwise difference

$$E_I = \sum_x \sum_y \sum_{c \in \{r,g,b\}} (I_{c,input}(x,y) - I_{c,model}(x,y))^2 \tag{4.6}$$

Inferring from the extracted parameters ρ, it is straightforward to automatically generate thousands of labeled training images, all considering the extracted camera specific settings. The generated files guarantee coverage of the full spectrum of statistical data variability and may be equipped with tailored facial attributes. Using the parallelized training procedure on top, one may quickly compute effective detectors for each single camera, at the cost of a few clicks.

Selective Training Data Regarding surveillance systems, it might also be the case that one wants to train a detector for specific target groups. One example could be a surveillance camera at a primary school. For these cases, specific training data might be helpful but difficult to obtain. To overcome these difficulties we can use our system to generate training samples following predefined constraints. We can do this by either taking a few samples and generating many variants (bootstrapping) of them or by labeling our data with respect to the wanted attribute (young vs. old) and manipulating existing faces from our datasets.

Full Control of Arbitrary Attributes Depending on the use of the detector, it might be useful to train detectors for specific accessories or attributes, such as glasses, beards or hats. While common methods would require observing enough sample data in the real world, our system allows us to design arbitrary attributes in 3D and to place them on any rendered face. This way, one may produce a large amount of training images for any specific purpose.

4.8 Summary, Discussion and Future Aspects

We presented a face detection system that is trained only on synthetic training data. The results indicate that using synthetic training data is meaningful and offers a variety of useful applications. The time consuming process of manually labeling faces can be replaced by a fully automated procedure.

However, the generation of synthetic training data highly depends on the availability of a suitable 3D face model (such as morphable 3D face model or active shape model). Constructing a morphable model from scratch is very time consuming and also requires available 3D data of faces. But once a model is all set, training data comes at almost no computational cost and scales easily to larger quantities. In addition, it is easy to adapt the artificial training data to any specific requirement. Facial attributes may be designed, modified or extracted in arbitrary ways.

Though the presented results are already very promising, one could explore whether combining real-world data with synthetic training data could further improve the reported results. Also, when combining many view-dependent classifiers, it would be suitable to perform an additional post-processing step as, for example, vector boosting. For the detection of facial skin, advanced color models could be helpful to enhance our results. These improvements will be part of future work.

5 Computer Suggested Facial-Makeup

5.1 Motivation

Finding the best makeup for a given human face is an art in its own right. Experienced makeup artists train for years to be skilled enough to propose best-fit makeup for an individual. For many centuries, people have changed facial reflectance by different means to achieve a pleasant facial appearance. In current everyday life, facial makeup is mostly used to mimic healthy and attractive facial geometry and reflectance. Finding the best makeup for a given human face can be considered an art form, and nowadays many people trust the advice of professional makeup artists. In this thesis we propose a system that automates this task. We acquired the appearance of 56 human faces, both without and with professional makeup. To this end, we use a controlled-light setup, which allows capturing detailed facial appearance information, such as diffuse reflectance, normals, subsurface scattering, specularity, or glossiness. A 3D morphable face model is used to obtain 3D positional information and to register all faces into a common parameterization. We then define makeup to be the change of facial appearance and use the acquired database to find a mapping from the space of human facial appearance to makeup. Our main application is to use this mapping to suggest the best-fit makeup for novel faces that are not in the database. Further applications are makeup transfer, automatic rating of makeup, makeup training, or makeup exaggeration.

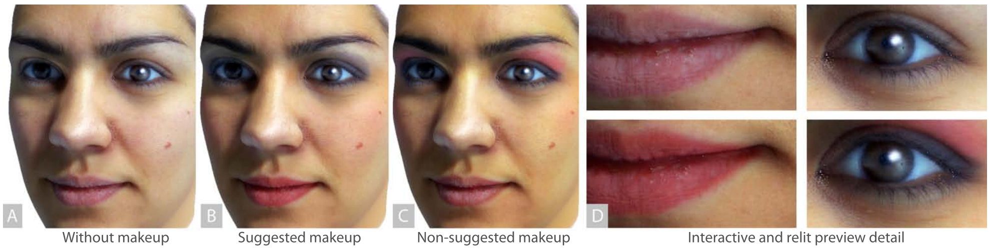


Figure 5.1: (A) Given a photograph of a novel face without makeup, we first reconstruct the facial surface in 3D. (B) Our system then suggests makeup that fits the face best. In contrast, (C) shows non-suggested makeup for the given face (farthest neighbor). (D) The new makeup may be rendered, relit, and inspected in 3D.

As our makeup representation captures a change in reflectance and scattering, it allows us to synthesize faces with makeup in novel 3D views and novel lighting with high realism. The effectiveness of our approach is further validated in a user study.

5.2 Introduction

This Chapter introduces a model of facial makeup using a database of example faces with and without makeup. The makeup model makes it possible, for example, to provide computer-suggested makeup for new subjects, who are not part of the database. It is also possible to rate applied makeup in order to give objective feedback. For all these applications, we make the assumption that the choice of human facial makeup is based on the more or less conscious process of mapping facial features to reflectance and scattering changes. This is, of course, just an approximation of reality, as there are some widely accepted standards but no strict rules for makeup [Tho05]. Furthermore, the mapping from facial appearance to makeup might change depending on culture, ethnicity, and history. For the sake of simplicity, we limit our investigations to western, 21st century, female facial makeup.

In contrast to existing approaches [GS09, TTBX07], our approach allows not only the transfer of makeup between subjects, but also provides computer-suggested makeup. Furthermore, we perform our makeup analysis and synthesis in 3D, whereas related approaches worked in the 2D domain. Nevertheless, our 3D analysis can provide makeup suggestions, even if only a 2D photograph of the subject is available. The proposed 3D synthesis allows simulating the suggested makeup under different lighting conditions.

The chapter is structured as follows: After reviewing previous work in section 5.3, we describe our approach to building a makeup model in section 5.6, which leads to a number of example applications that are then validated in a perceptual study. We conclude with a discussion and an outlook on prospective work.

5.3 Related Work

Facial Appearance Convincingly modeling the appearance of the human body, and especially the human face, has been a long-standing goal in computer graphics. Modeling facial appearance, which consists of both geometry as well as reflectance and scattering, is challenging, because human observers are well-tuned to perceive faces [HHG00] and even subtle imperfections may easily lead to an unpleasant impression [SN07]. To capture the essence of facial geometry, Blanz and Vetter [BV99] introduced a morphable face model. Besides geometry, the appearance (i.e. reflectance and scattering) of human skin was captured [MWL+99, DHT+00, WJG+06, WLL+07, DWD+08]. The effects of aging, alcohol consumption and also foundation cosmetics can be predicted by a model based on the separation of hemoglobin and melanin [TOS+03].

Makeup The main purpose of makeup is to temporarily change the facial appearance. Most people apply makeup to either cover unwanted facial structures, such as wrinkles or pores, or to emphasize specific features. Others apply makeup to express themselves and to play around with different styles and looks. However, makeup is a very subjective matter. The common understanding of makeup varies between different cultures and has changed over the course of history [Cor72].

Further, cosmetic makeup is, and has been historically, much more

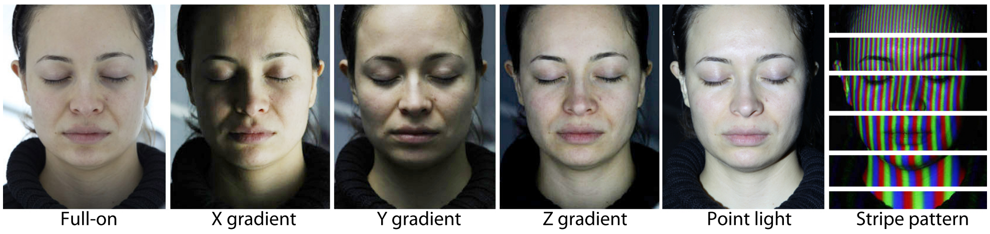


Figure 5.2: The controlled-light setup used for facial appearance capture. Each face was acquired once without and once with professional makeup.

common among women [Cor72], and we will hence limit our investigation to female, present-day makeup for simplicity. Russel [RR03] studies the relation of contrast between different facial parts, indicating that luminance changes as found in cosmetics are more effective on female faces.

Given a certain facial appearance, makeup is used to emphasize certain features or deemphasize others. Therefore, makeup is a change of facial appearance that depends on the initial appearance. This change is not limited to color; gloss and scattering are also affected. To our knowledge, makeup was neither rendered nor captured in previous work and was only considered in an image-based context [TTBX07, GS09, DS12]. In this work, we will argue as to how makeup is a change of facial appearance as a function of facial appearance, i. e. it is a mapping that can be modeled given some examples.

Adding or emphasizing existing makeup can be achieved approximately for live footage using simple image filters [NRK99]. Both Numata et al. [NRK99] and Tsumura et al. [TOS$^+$03] change the melanin and hemoglobin mixture to indirectly mimic the effect of foundation cosmetics, which in reality do not change the melanin and hemoglobin mixture. The system of Tong et al. [TTBX07], allows the transfer of cosmetics when given a before/after image pair. Guo and Sim [GS09] transfer makeup and other facial details from a single image to another image, but without an explicit notion of makeup itself. Finding a suitable 'decoration' of a face is similar to automated caricature as done by Chen and colleagues [CXS$^+$01].

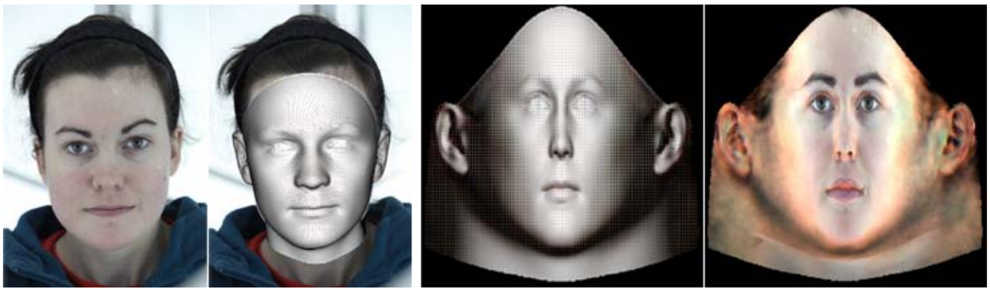


Figure 5.3: (Left:) We fit a morphable face model to all 'full-on' images, both with and without makeup. (Right:) We thus gain dense correspondence across all images and can transfer all acquired images into a uniform cylindrical parameterization. We use the new parameter space to build the makeup model.

Facial Attractiveness Facial attractiveness is perceived consistently between observers of different age, sex and cultural background [LR90] and is believed to relate to averageness [LR90], symmetry [PBPV^{+}99] and sexual dimorphism [PLPV^{+}98].

Mappings from human shape to simple attributes were investigated by [EDR06] (attractiveness) or [WV09] (social judgments). The model of Eisenthal et al. [EDR06] was later used for facial image beautification by Leyvand and co-workers [LCODL08]. While all previous work has analyzed real attractiveness (and synthesized virtual beauty in the case of Leyvand's [LCODL08] work), we conduct an analysis, complemented by a synthesis step that improves real physical attractiveness when applying our suggested makeup.

5.4 Acquisition

To extract the essence of makeup, we model it as a mapping from a domain of facial appearance to a range of appearance changes, i.e. makeup. We use a data-driven approach where we acquire facial appearances including makeup, then extract the makeup, construct the mapping, and finally generate suggested makeup. We acquired a database of $N = 56$ female faces in two states: without makeup and with professional makeup.

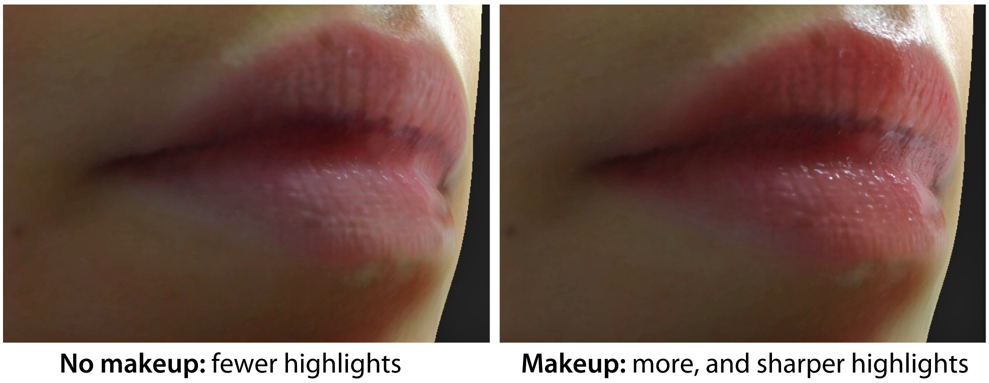

No makeup: fewer highlights
Makeup: more, and sharper highlights

Figure 5.4: The measured specularity and glossiness changes after applying makeup. This is most noticeable on the lips: (Left) rendered lips without makeup, (Right) rendered lips with makeup.

Subjects were placed in a light tent surrounded by six light projectors. To capture advanced facial properties, we project a number of structured light patterns in each state: Gradients along X, Y and Z and vertical stripe patterns of increasing frequencies, 8 octaves in particular (cf. Figure 5.2). The patterns were successively projected in less than a second and each was captured using a 'Canon EOS 5D Mark II' camera.

5.5 Data Preprocessing

First, we used the stripe patterns to perform a coarse point-based 3D reconstruction of the subject's facial surface. Next, we fit a morphable face model [SSSB07] to all 'full-on' (cf. Figure 5.2) images, both with and without makeup. Using the registered face model, we then converted all images into a common parameterization, i. e. we make sure that the tip of the nose is in the same location in all images and perform the same registration for all acquired images. An example of the uniform parameterization is shown in Figure 5.3.

Finally, we use manual 2D image deformation to make the registration pixel-accurate. Remaining holes, i.e. parts of the face not seen in the image, are filled using texture inpainting [Tel04]. Please note that these steps require little manual interaction and are performed while building the model only; no manual intervention is required when using the model.

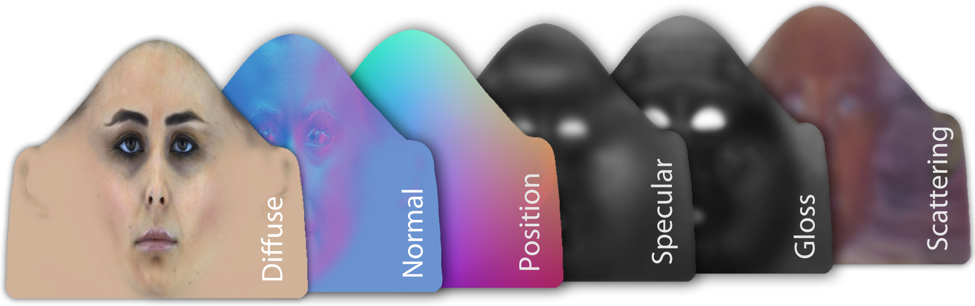


Figure 5.5: Appearance images (here: a single subject) have a common parametrization for all subjects and all bands.

From the gradient images we acquire photometric normals [MHP+07]. We did not use polarized light to separate specular and diffuse normals. Finally, we remove low-frequency normal bias using Nehab et al.'s [NRDR05] method. We use the 3D surface model to perform inverse lighting, i.e. using the 'full-on' image (cf. Figure 5.2) to compute spatially varying diffuse reflectance [WJG+06]. An individual specularity and glossiness term for each subject is computed from the 'single light source' image and for different segments, such as the lips. The change of specularity and glossiness after applying makeup is most noticeable on the lips. Figure 5.4 shows a typical example, where the lips show stronger specularity and gloss after the makeup is applied.

Finally, we compute spatially varying subsurface scattering strength from the stripe patterns by direct-indirect separation [NKGR06]. We approximate scattering using a sum of Gaussians, which has been shown to be perceptually plausible [JSG09]. Comparing the subsurface scattering with and without makeup, it can be observed that makeup reduces subsurface scattering of the skin. The makeup on the lips (typically lipstick) mostly eliminates the subsurface scattering.

In summary, for subject i in state X we acquired an *appearance image* $A_{i,X}$ that stores the following information per-pixel:

– diffuse color RGB (3)
– 3D position XYZ (3)

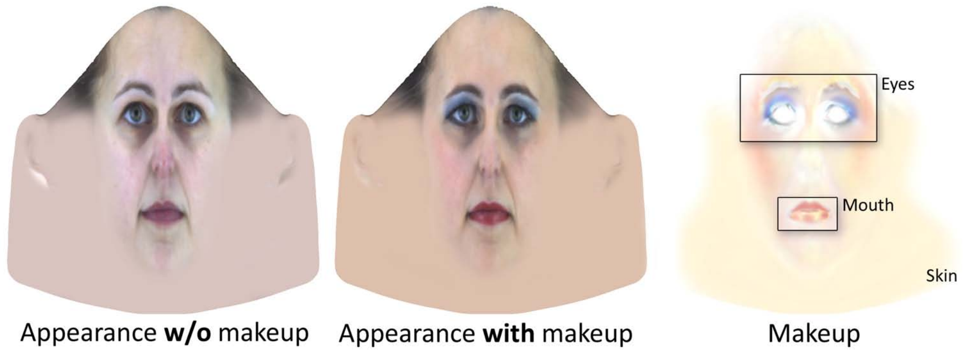


Figure 5.6: Registered diffuse appearance and makeup. All images have been transferred to a uniform cylindrical parameterization.

- 3D normal XYZ (3)
- specularity (1)
- glossiness (1)
- scattering strength RGB (3)
- scattering size RGB (3)

in a common parameterization for all faces (figure 5.5). The values in brackets state the number of required bands. Thus, an appearance image has a total of 17 bands.

5.6 Modeling Makeup

Once acquired, we compute the change of facial appearance and call this change the 'makeup'. The makeup of subject i from state X (without makeup) to state Y (with makeup) is denoted as $M_{i,X\rightarrow Y}$. Further, we define the ratio of appearance with and without makeup to produce makeup

$$M_{i,X\rightarrow Y} = \frac{A_{i,Y}}{A_{i,X}} \tag{5.1}$$

Consequently, the multiplication of appearance without makeup $A_{i,X}$ and makeup $M_{i,X\rightarrow Y}$ produces appearance with makeup

$$A_{i,Y} = A_{i,X} \cdot M_{i,X\rightarrow Y} \tag{5.2}$$

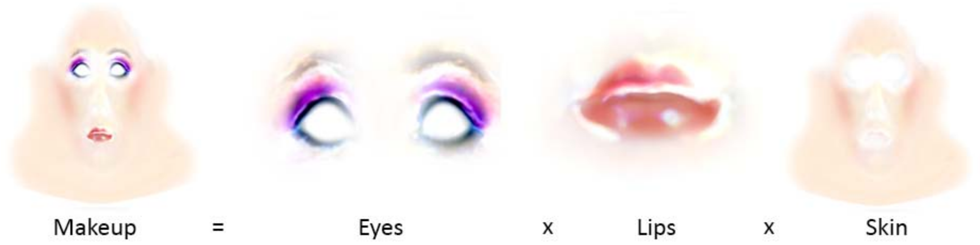


Figure 5.7: We compose the makeup by merging three regions (eyes, lips, skin). Each of the regions is reconstructed from its own eigenspace using 35,15 and 10 eigenvectors respectively. Note that the makeup shown is squared for better visibility.

Note, that M_i and A_i are images and multiplication or division must be done per pixel and band. We assume that makeup has no geometric effects, i.e. normals and positions are not affected by makeup and are omitted. The change in diffuse color is most important and we use the ratio in RGB space [LSZ01] to express the effect. Other color spaces like LAB performed worse. Glossiness and specularity is modeled as scalar addition, and all changes to scattering as monochromatic multiplication.

5.6.1 Appearance-to-Makeup Mapping

Let $\mathcal{A}$ be the space of all possible facial appearances and $\mathcal{M}$ the space of all possible makeup looks. We want to find the best mapping from a facial appearance $A \in \mathcal{A}$ to a makeup $M \in \mathcal{M}$. For each of our 56 examples in the database, this mapping is given, because for the particular makeup M_i to go from a no-makeup-state X into a makeup-state Y is only a function of the subject's appearance A_i. If we want to determine a makeup M_{query} for a new subject A_{query}, who is not in our database, we can perform nearest neighbor matching in the space of facial appearance, i. e.

$$M_{\text{query}} = M_j \tag{5.3}$$

where

$$j = \operatorname*{argmin}_i d(A_{\text{query}}, A_i). \tag{5.4}$$

A naïve distance function $d(A_{\text{query}}, A_i)$ for nearest neighbor matching would be the sum of absolute differences of all pixels and bands of the two appearance images. However, this approach would assume that each pixel and each band contains the same amount of information. Instead, we perform a PCA of all facial appearances A_i using the classical eigenface approach [TP91]. The only difference is that compared to the classical approach our appearance images contain a larger number of bands, as for example the 3D coordinates describing the 3D shape of the face.

The eigenface approach allows use of the Mahalanobis distance of the corresponding facial appearance PCA coefficients as the distance function $d(A_{\text{query}}, A_i)$. By determining the nearest neighbor in PCA space, it is ensured that the most descriptive pieces of information are used to differentiate between the subjects and the subject with the most similar appearance is selected.

However, we found during our experiments that the nearest neighbor matching using the Mahalanobis distance of the PCA coefficients can only to some extent find the makeup that is most aesthetic for a given face. A human observer focuses strongly on certain features of the face, in particular on the color of the eyes, hair, or skin. These inter-subject differences can also be found in the PCA coefficients, but are not weighted as high in the Mahalanobis distance as they are weighted by a human observer. Consequently, we extend our distance function for nearest neighbor matching with the following heuristic:

$$d(A_{\text{query}}, A_i) = w_1\, d_{\text{pca}} + w_2\, d_{\text{eye}} + w_3\, d_{\text{skin}} + w_4\, d_{\text{hair}} \tag{5.5}$$

where d_{pca} is the Mahalanobis distance between the PCA coefficients of A_{query} and A_i, and d_{eye}, d_{skin}, and d_{hair} are the distances between the eye, skin, and hair color vectors of A_{query} and A_i in RGB space, respectively. The eye, skin, and hair color for each subject can easily be extracted from the rectified appearance images. We set the weights to

$$w = \begin{bmatrix} 0.6_1 \\ 0.2_2 \\ 0.1_3 \\ 0.1_4 \end{bmatrix} \tag{5.6}$$

in all our experiments. Experimenting with several combinations, we found these heuristic weight parameters to yield the most convincing results.

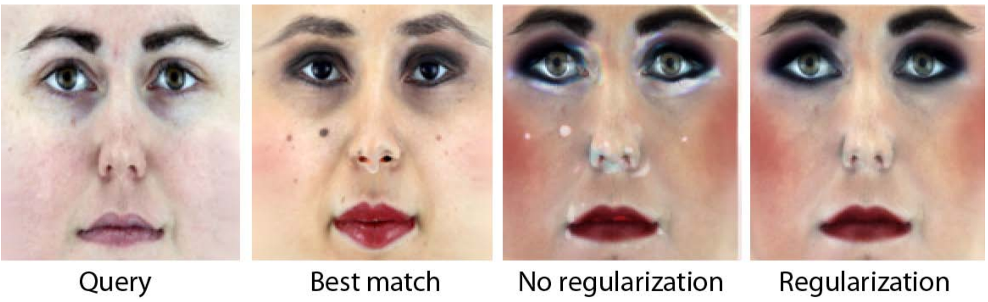


Figure 5.8: The two images on the right side show the query face after makeup transfer of the best match. By applying PCA compression, our approach allows the re-synthesis of makeup that automatically omits unwanted transfer of personal details, like freckles, wrinkles or moles. Note: In the two images on the right side, the makeup is exaggerated and applied 5 times for better visibility.

5.6.2 Re-Synthesis of Makeup

Now that we have determined by nearest neighbor matching which makeup we want to copy, we can apply this makeup to the query appearance with

$$A_{\text{query},Y} = A_{\text{query},X} \cdot M_{\text{query},X\to Y} \tag{5.7}$$

However, as shown in Figure 5.8 this would result in a transferal of personal details, like freckles, wrinkles or moles. If the nearest neighbor subject has freckles or moles and these are covered by makeup, these details appear as 'inverted' freckles or moles in the transfered makeup of the query subject. Thus, we need to find a way to regularize our makeup. In other words, we want to transfer only the essence of that particular makeup compared to other makeup, excluding the personal details.

Regularization This can be achieved by applying a PCA on all makeup styles M_i. We use the eigenface approach and calculate the corresponding 'eigenmakeups'. The resulting $N = 56$ eigenmakeups are sorted according to their eigenvalues in descending order (see Section 2.2.2). The eigenmakeups with large eigenvalues are likely to contain variations that can be observed between many makeup styles, whereas the eigenmakeups with small eigenvalues contain the personal details. We then re-synthesize

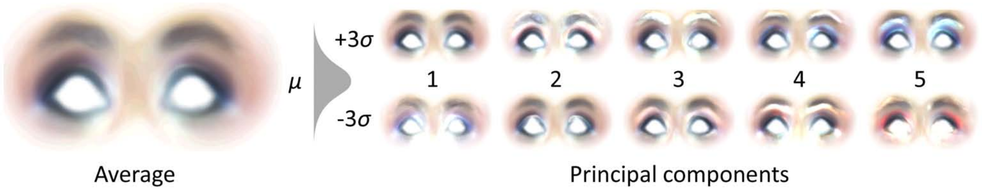


Figure 5.9: The first five eigenmakeups for the eye segment (right) and the respective average (left). The contrast was enhanced for better visibility.

each makeup style $\tilde{M}_i$ by using the 10 eigenmakeups with the 10 largest eigenvalues. As shown in Figure 5.8, these re-synthesized makeup looks are regularized to omit personal details and can be applied to the the query appearance with

$$A_{\text{query},Y} = A_{\text{query},X} \cdot \tilde{M}_{\text{query},X\to Y} \tag{5.8}$$

We can improve upon this approach by performing individual PCAs for different parts of the face. This allows the PCA to identify those differences between makeups that are typical for these different parts.

We segment the face into three regions, for the eyes, lips and skin. For each of the three regions, we create individual makeup eigenspaces (cf. Figure 5.7) which we recombine later to the compose complete makeup. The eye and lip regions have soft borders to fade into white, i.e. the identity makeup. We then re-synthesize a complete makeup style $\tilde{M}_i$ by re-synthesizing each part individually, and compose the individual regions to the complete makeup style. As a result, the regularization is performed on the individual parts rather than the whole makeup style, which we found to perform better in practice.

For the makeup synthesis we state that for a proper reconstruction of skin and lip makeup, only 10–15 principal components (PCs) are sufficient, whilst for the reconstruction of eye makeup 30–35 PCs are needed. The above parameters are motivated by the apparent data distribution. Figure 5.10 shows the distribution of the respective principal components and justifies that:

1. it is meaningful to separate the measured makeup into three different eigenspaces, as they show a sufficiently different distribution

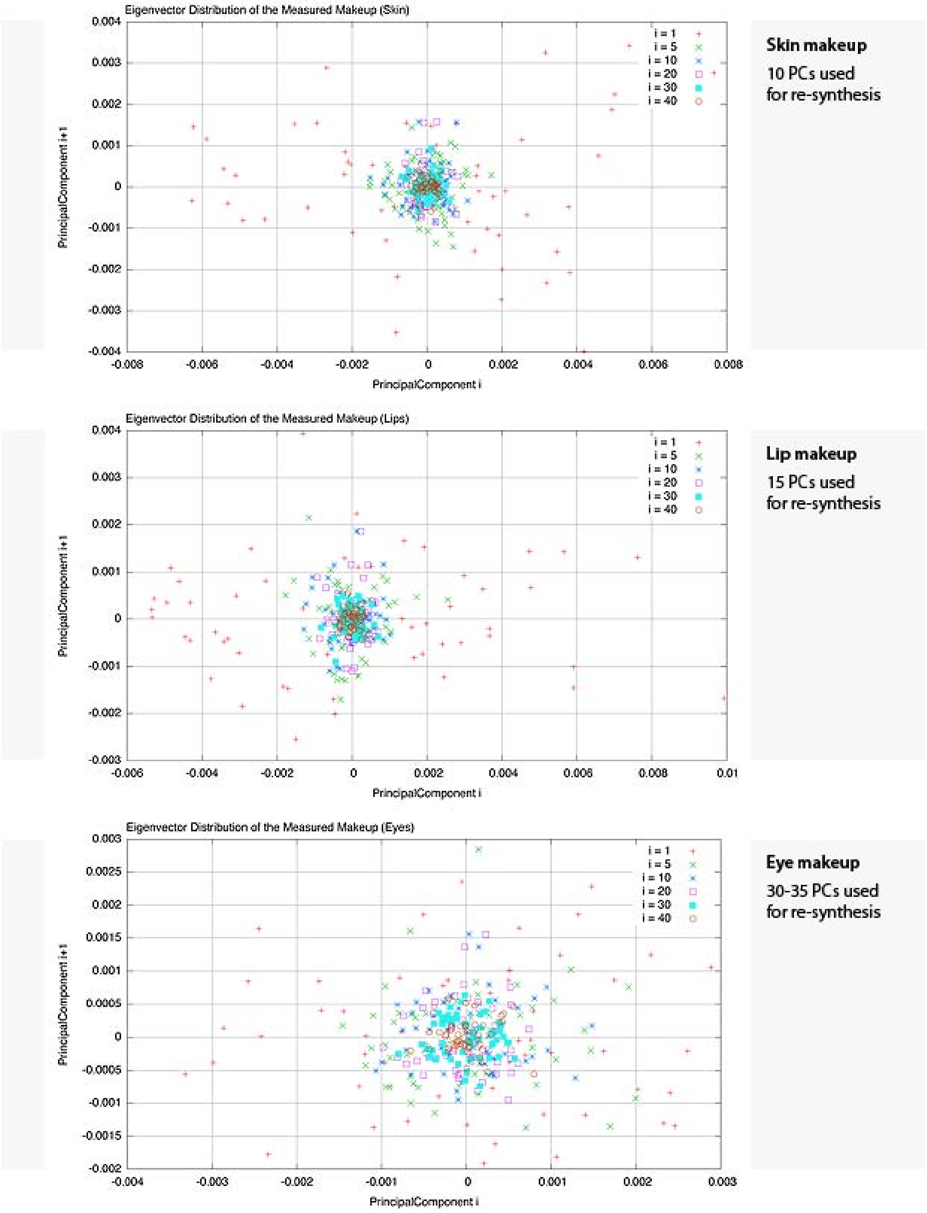


Figure 5.10: The above distributions illustrate the acquired makeup data in terms of variance. For makeup re-synthesis we derived different settings from the apparent data distribution for each individual region. Skin and lip eigenspace are sampled sufficiently densely, while the eye region would profit from more acquired sample images in the database. Examples of the respective eigenmakeups for skin, lips and eyes are shown in the appendix (page116 et sqq.).

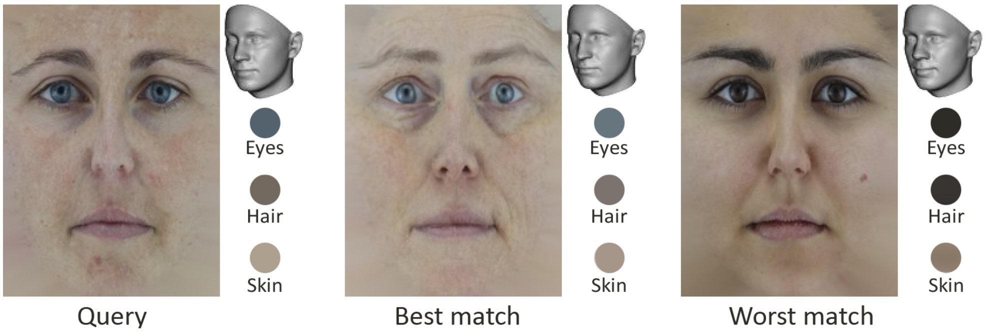


Figure 5.11: The computer-suggested makeup is found by a nearest neighbor search that involves the 3D-shape PCA coefficients of the facial appearance A_i as well as skin, eye, and hair color.

2. for the reconstruction of skin and lip makeup, 10–15 eigenvectors are sufficient, as the last 35 eigenvectors all cluster around one centroid
3. for the reconstruction of the eyes, a higher number of eigenvectors is suitable, because the data scatters widely, even in the rear eigenvectors (30–40).

Note that the data distribution of the eye makeup indicates, that it would be meaningful to acquire more data samples, in particular to achieve more variety when re-synthesizing eye makeups. Using the new approach with the apparent data, however, we can for example automatically remove wrinkles all over the face while preserving the full variance of possible eye makeup.

5.6.3 Rendering and Applications

Having acquired the facial appearance from the subject in question, we can show a visualization of novel synthesized makeup from novel viewpoints under novel lighting, including effects such as subsurface scattering [JSG09] and specular lighting in real time.

Computer-Suggested Makeup The key application of our model is computer-suggested makeup. Given a 2D image (e.g., a photograph) of a

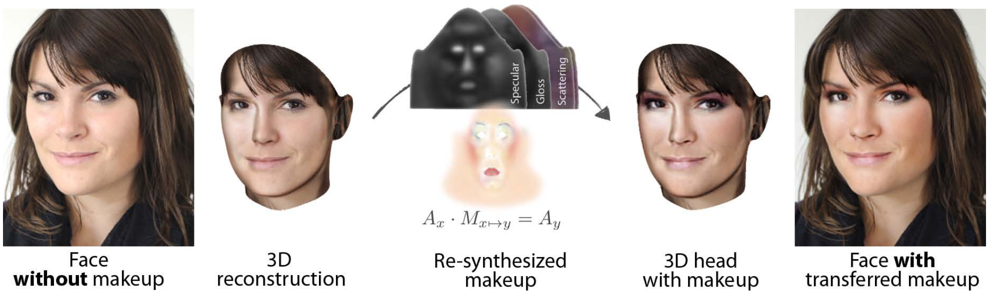


Figure 5.12: Given a 2D photograph of a subject, we first compute its facial appearance without makeup in 3D (left). To apply makeup, we apply any re-synthesize makeup from our database and copy the bands for specularity, gloss and scattering to the query appearance (center). The result may be relit and rendered into the input image (right).

subject as a query, we fit the morphable face model [SSSB07] to the face in the image. This gives us the facial appearance without makeup $A_{X,\text{query}}$.

However, this appearance image has only 6 bands (diffuse color RGB and position XYZ) in contrast to the 17 bands we have acquired for the appearance images A_i in the database, which were captured in the light tent. Consequently, we can use only the first 6 bands to perform nearest neighbor matching, as described in section 5.6.1 (cf. Figure 5.11). Once the nearest neighbor is found, we just copy the missing bands from the nearest neighbor to the query appearance, except for the normals. This allows us to perform convincing real-time 3D renderings of the subject without makeup under varying view position and lighting setups. We then re-synthesize the suggested makeup from the nearest neighbor makeup (cf. Section 5.6.2) and generate $A_{Y,\text{query}}$. The user can now inspect the query subject with suggested makeup in real time under varying viewpoints and different illuminations (Figure 5.13).

Makeup Transfer Makeup transfer (as done in an image-based way by Guo and Sim [GS09]) can be done more robustly in 3D. The 3D information allows for performing robust inverse lighting, and thus we can render the subject under different viewpoints and with novel lighting (Figure 5.12). The disadvantage of our approach is that we require a with-and-without-makeup image pair to capture the makeup. The advantage of our approach

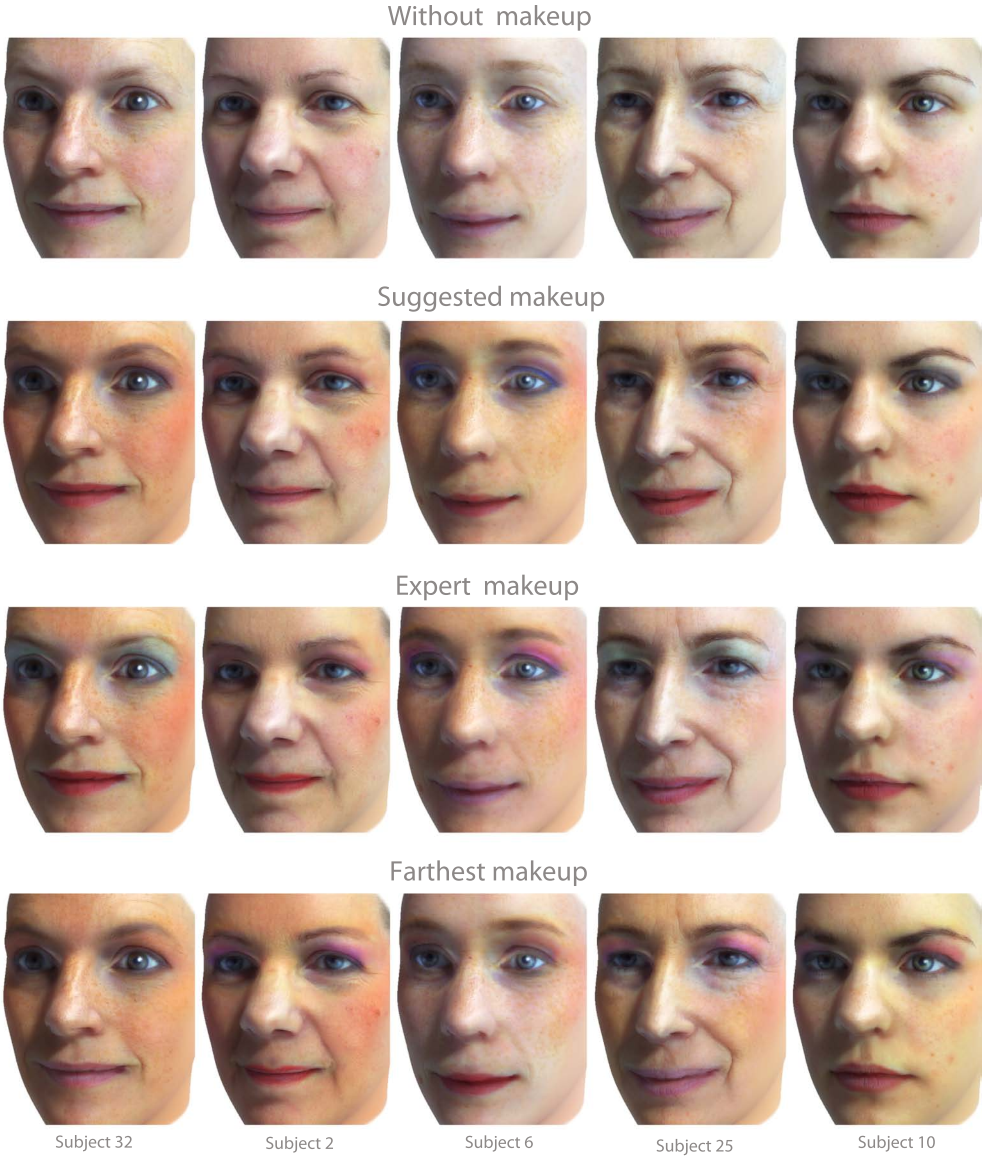


Figure 5.13: Our main application is computer-suggested makeup. Given a subject (one in each column), starting from a face without makeup (first row), we suggest makeup (second row), according to a mapping acquired from a makeup artist (his result shown in the third row). We include the makeup that fits the least (fourth row) for comparison.

is that we use the PCA compression as a regularizer, whereas Guo and Sim's approach relies on low-pass filters to remove personal details that should not be transferred.

Our PCA approach allows for transferring finer details that are shared by all subjects, i. e. in proximity of the eyes, which are blurred away even for advanced filters (bilateral, non-local means).
To perform a makeup transfer, the user provides a facial appearance of one subject without makeup $A_{\text{source},\mathrm{X}}$, the same subject with makeup $A_{\text{source},\mathrm{Y}}$, and a second subject without makeup $A_{\text{target},\mathrm{X}}$. As we already know which makeup we need to transfer, we can skip the nearest neighbor search and just need to perform the re-synthesis step from section 5.6.2 to generate $A_{\text{target},\mathrm{Y}}$. A user can then inspect the effect of the first subject's makeup on the second subject.

Automatic Rating of Makeup A variant of computer-suggested makeup is the automatic rating of existing makeup. Here, a user provides a with-and-without-makeup appearance pair $(A_{\mathrm{X},\text{query}}, A_{\mathrm{Y},\text{query}})$ to the system. Given this appearance pair, the query makeup is given by

$$M_{\text{query}} = \frac{A_{\text{query},\mathrm{Y}}}{A_{\text{query},\mathrm{X}}} \tag{5.9}$$

In contrast to the computer-suggested makeup, the nearest neighbor search is now performed in the makeup space $\mathcal{M}$ and not in the appearance space $\mathcal{A}$. To be more specific, we use the Mahalanobis distance of the makeup PCA coefficients as the distance function $d(M_{\text{query}}, M_i)$ to find the nearest neighbor in makeup space (instead of $d(A_{\text{query}}, A_i)$ in appearance space as before). Once we found the nearest neighbor makeup M_{nearest}, we know the corresponding appearance$A(M_{\text{nearest}})$ in the database.

We can then return the Mahalanobis distance $d(A(M_{\text{nearest}}), A_{\mathrm{X},\text{query}})$ in appearance space as a rating for the makeup. Thereby, a small distance is considered a better makeup.

Inverse Computer-Suggested Makeup In inverse computer-suggested makeup, our system generates a facial appearance A that would be best

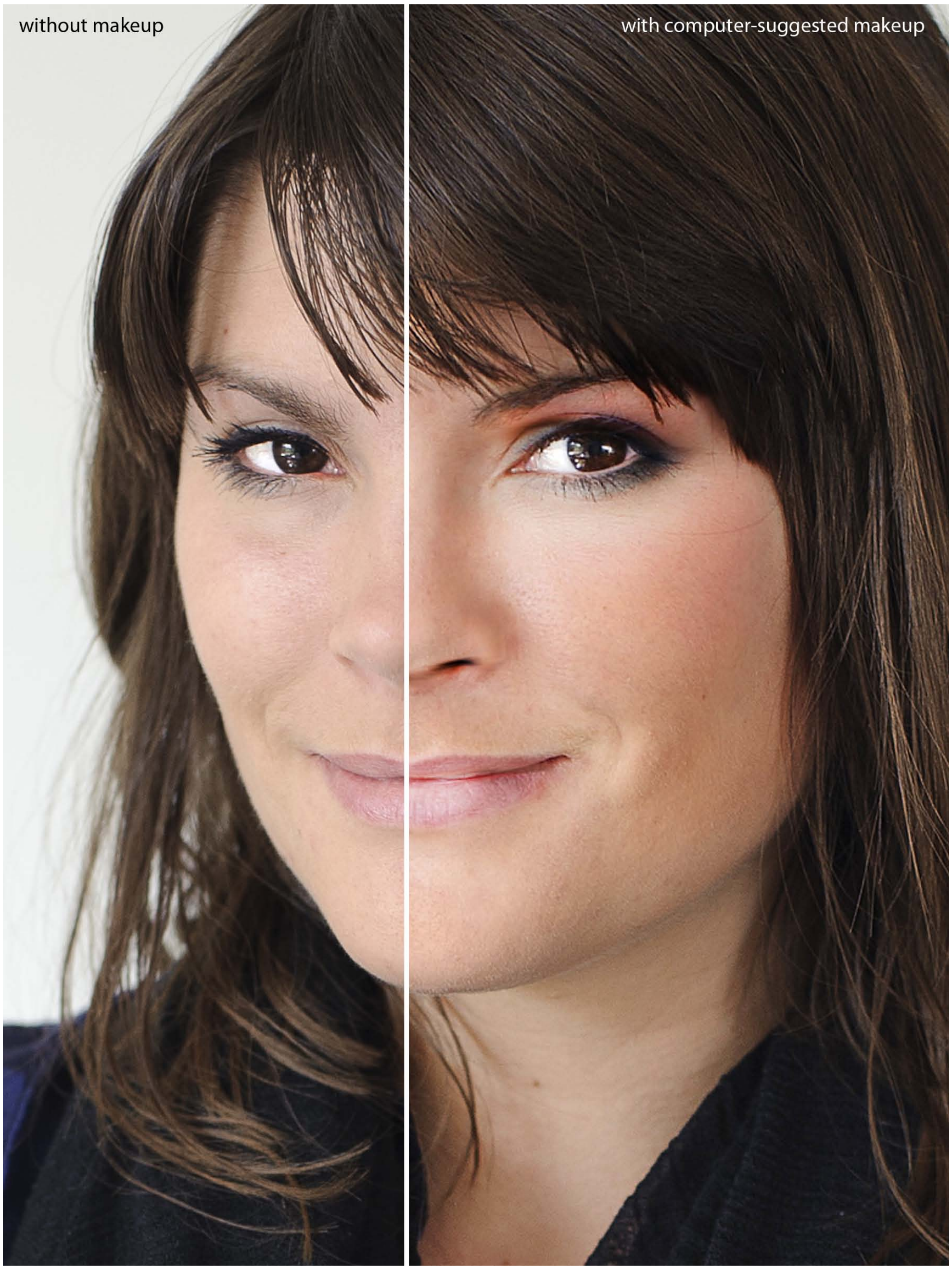


Figure 5.14: Example of computer suggested makeup. (Left:) Input photograph of a face without makeup. (Right:) The same face after computer suggested makeup was applied, relit and rendered back into the photograph. Note: soft borders fade the 3D model into the background.

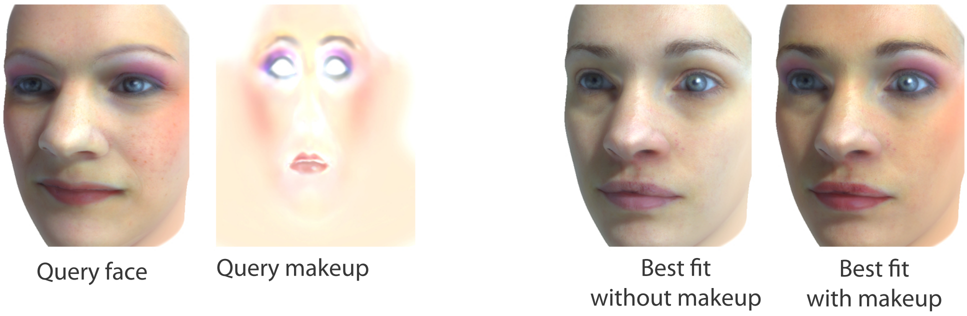


Figure 5.15: Given a query Makeup (left), the user can now search for the facial appearance (right) that would fit best to its makeup. This could be useful in training makeup artists.

for a given makeup query M_{query}. The approach is similar to the automatic rating of makeup, except that $A(M_{\text{nearest}})$ is returned by the system instead of a rating. This can be used for didactic purposes, to study what facial appearance corresponds to which makeup (cf. Figure 5.15).

Makeup Exaggeration Given a facial appearance without makeup A and its makeup M, our model can be used for (de)-exaggeration. The naïve approach to this problem is to multiply M by a constant, which indeed increases the effect of M. However, best results are achieved when moving away from the intra-subject average (Figure 5.16).

5.6.4 Perceptual study

We evaluated the effectiveness of our model for the 'computer-suggested makeup' application in a perceptual study. Participants compared three screenshots of rendered faces that were placed next to each other in one row: I_{q} (query subject without makeup), I_{a} (this subject with makeup), and I_{b} (this subject with different makeup). The query image of the subject without makeup was placed in between both images with makeup.

We asked the same question at all times: "What makeup fits best for the query appearance in I_{q}: The makeup in image I_{a} or the makeup in image I_{b}?". In each trial we randomized the position of the query image I_{a} and I_{b}. We ran three experiments, which used

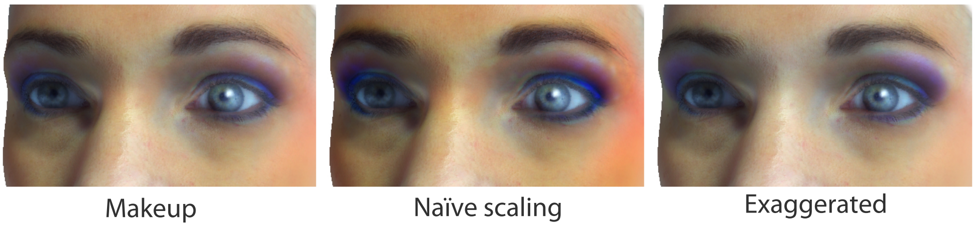


Figure 5.16: Starting from a facial appearance with makeup (left), naïve makeup scaling (middle) leads to overall scaled makeup, e.g. on skin and eyes — which is found in many makeup looks — is enhanced. When scaling the eigenmakeup instead ($\sigma_{eyes} = 2.0$), features typical for this individual makeup are exaggerated (right).

1. the re-synthesized makeup applied by the professional makeup artist for I_a and a randomly selected makeup for I_b,
2. our suggested makeup for I_a and the farthest neighbor makeup for I_b. (The farthest neighbor makeup is the makeup with the largest distance $d(A_{\text{query}}, A_i)$),
3. our suggested makeup for I_a and a randomly selected makeup for I_b.

For each of these three experiments, the participants were shown 17 image triplets. Thus, in total, each participant had to rate 51 image triplets.

We ran the three experiments on the Amazon Mechanical Turk platform. Though Mechanical Turk allows recruiting hundreds of user study participants at low cost and in a short amount of time, it is important to carefully design the posted micro-tasks to achieve reliable results [KCS08]. We decided on two mechanisms for avoiding random answers. First, within each set of 17 images, we showed three images twice. Consequently, each user rating these three images differently was excluded from the final results. Secondly, we evaluated the time users needed to answer the three experiment tasks. Heuristics showed 30 seconds to be the minimal time for answering a single task. Thus, users taking less than 30 seconds were excluded as well. Overall, these restrictions reduced the number of valuable results from an initial 210 to 146, which is 69%. Another side effect of the filtering is that the distribution of male and female participants differs in each experiment.

	Experiment 1		Experiment 2		Experiment 3	
	prof.	rand.	NN	FN	NN	rand.
All	62 %	38 %	67 %	33 %	58 %	42 %
Female	64 %	36 %	66 %	34 %	62 %	38 %
Male	58 %	42 %	67 %	33 %	55 %	45 %
Signif.	$p < 10^{-8}$		$p < 10^{-15}$		$p < 10^{-4}$	

Table 5.1: Results for the three experiments of the study are shown. The upper 3 rows state the percentage of participants that voted for a particular makeup (prof. = professional makeup, rand. = random makeup, NN = nearest neighbor makeup, FN = farthest neighbor makeup). The last row shows the p-values of the corresponding Pearson's chi-square test.

Table 5.1 summarizes the results of the three experiments. It can be observed that 62 % of all participants preferred the professional makeup over the random makeup. If we formulate the null hypothesis that professional and random makeup are equal, this null hypothesis can be rejected with a Pearson's chi-square test. The probability of the observed rating under the null hypothesis is $< 10^{-8}$ (p-value). Consequently, it is extremely likely that professional and random makeup are not equal. In the second experiment, 67 % of all participants preferred our suggested nearest neighbor makeup over the farthest neighbor makeup. Again, it can be shown with Pearson's chi-square test that this difference is statistically significant (p-value $< 10^{-15}$). Finally, in the third experiment, 58 % of the participants liked our suggested makeup better than a randomly selected makeup (p-value $< 10^{-4}$).

These results have some interesting implications. The small advantage by which a professional makeup expert can improve upon randomness is at best $62\,\% - 50\,\% = 12\,\%$ and serves as a baseline of what can be achieved. Overall, the computer-suggested makeup does perform only slightly worse than the professional makeup (58 % vs. 62 %). This can be considered a very good result. Much higher percentages cannot be expected, as we have modeled our mapping from the professional makeup, which constitutes an upper bound. Examples of the stimuli used as well as the full ratings are found in the appendix (pages 119–121).

5.7 Summary, Discussion and Future Aspects

We presented a model describing the relation of facial appearance and facial makeup using a database of example faces with and without makeup. describing. The model involves 3D information for both analysis and synthesis of artificial makeup. The results can be rendered, relighted and inspected in arbitrary poses or lighting conditions and may be used for applications such as machine-suggested, individualized makeup. In all results presented, we use nearest-neighbor sampling over $\mathcal{M}$, though more advanced reconstructions can improve the results. To extract facial features (eyes, hair and skin) we employ straightforward color descriptors though more sophisticated feature models could be used ([JR02, FMG$^+$12]). Further, we use multiplication [LSZ01] of diffuse colors to simulate the effect of makeup, while the more advanced Kubelka-Munk theory could be used. Linear blending of multiple components using the first three eigenvectors that are sufficiently orthogonal creates in-between makeup, but becomes implausible for others. Here, more than 56 samples would be required. Assuming uniform distribution, we can infer from the given eigenvector distribution that our makeup space is sufficiently dense for lips and skin, while for the eyes more samples would be needed. Within our current setup we expect 100 samples to be suitable. In our approach, we currently assume that there is no bad example in the database and that the expert was always right. To generalize to arbitrary non-expert input (e. g. from community photo databases), a rating of the individual makeup would complement the approach. We base our model on makeup applied by a single professional expert makeup artist. This introduces the artist's personal taste as a bias which could be reduced by using more artists. Consequently, this would require even more samples in order to guarantee a densely sampled makeup space. We always re-synthesize new makeups from a single best-match makeup. On the one hand this is suitable, as makeup artists focus on creating styles where eye, skin and lip makeup homogeneously match. However, combining different makeup choices could lead to new and interesting makeup styles, such as mixing the facial regions of different makeups. In this case, more sophisticated blending algorithms would be mandatory. These improvements will be part of future work.

6 Conclusion

This thesis presents three new data-driven approaches which aim to detect, analyze or modify faces in images. All presented methods bridge the gap between two-dimensional and three-dimensional representations of faces. In 2D or 3D respectively — they make use of prior knowledge about facial appearances and extract underlying statistical information from pre-collected databases of faces. The resulting solutions and methods contribute to both areas, computer graphics and computer vision alike. They help improve the outcome of face detection algorithms, ease and automate their training procedures and allow for automated modification of faces in images.

The following three scenarios may illustrate possible applications for the presented contributions in this thesis:

Scenario 1: If a face detector has successfully been trained but the resulting detections are not convincing and contain too many falsely detected faces, one might want to eliminate these incorrect results at detection time. A straightforward solution would be to tweak detector parameters until the false alarm rate approaches zero, unfortunately at the cost of missing faces. Another solution would be to collect new and more specific training data and to retrain the classifier. Both workarounds are either time consuming or lead to suboptimal results. Our method, however, involves depth information and detects false positives at runtime without significant loss of performance (Chapter 3).

Scenario 2: If the goal is, for example, to automatically detect faces in images, the first step to solve this problem is training a face classification function. One of the first questions to arise is that of which training data to use. In particular, in some cases it is very important to collect training data which properly represents the environmental constraints in which, for example, a surveillance camera is located. This may be a specific illumination, or an unusual position or viewing angle, as for example in a dark subway station. So far, the solution to this has required collecting an enormous amount of video footage and manually labeling faces in images, a very cumbersome and tedious process. Our presented approach makes use of synthetic training data and automates this task (Chapter 4).

Scenario 3: In another case, the location of a face in an image may be already known but the goal is to vary its facial appearance, for example to apply or enhance facial makeup, a very common task in the advertising industry. This usually requires a skilled artist who manually retouches the face in the image. Though the outcome might look convincing, the process is very time-consuming and there is no guarantee that the applied makeup fits the person's facial shape best. Additionally, the applied changes could not be transferred to another image, where the face is shown from a different viewing angle or in another illumination. We present an automated solution to this task which not only automatically applies makeup to arbitrary faces in images but also guarantees that the optimal makeup for the given face is selected (Chapter 5).

6.1 Summary of Contributions

All presented contributions are inspired by the use of prior knowledge. As such they derive information about facial appearances from pre-collected databases of facial images or 3D face models. As a side effect of their data-driven nature, all methods involve the extraction and application of statistical information about faces. In summary, the presented methods ease and automate processes which previously required time-consuming manual interaction. Moreover, new and powerful applications arise through the use of statistical insights.

Rapid Stereo-Vision Enhanced Face Detection In Chapter 3 we contribute an approach that extends a widely-used monocular face detector by an additional classifier that evaluates the disparity map of a passive stereo camera. A small set of pre-collected disparity maps is used for classification. The algorithm runs in real time and significantly reduces the number of false positives compared to the monocular approach. On a small test set, all false positives could be removed in our experiment.

Fast Face Detector Training Using Tailored Views In Chapter 4 we contribute a method which quickly trains face detection classifiers based on synthetic training data. The method overcomes the previously inevitable and time-consuming need to collect and label training data by hand. Using statistical insights, the generated training data is guaranteed to mirror the full variability of human faces and is, moreover, designed to expose arbitrary facial attributes. It automatically adapts to environmental constraints, such as illumination or viewing angle of recorded video footage from surveillance cameras. A fast many-core and multilayer implementation of Viola-Jones' state-of-the-art AdaBoost training additionally reduces the training time. At the cost of a few manual clicks, one may now train view-dependent, attribute- and illumination-specific classifiers which previously always involved significant manual interaction. Self-training and self-adjusting surveillance cameras is one possible application of the presented contributions.

Computer Suggested Facial-Makeup In Chapter 5 we contribute a model describing the relation of facial appearance and facial makeup. The data-driven approach extracts makeup from before/after images of faces, with and without makeup. The presented algorithm involves 3D information for both analysis and synthesis of artificial makeup. As such, results can be relighted and inspected in arbitrary poses and under arbitrary lighting conditions. Applications such as machine-suggested, individualized makeup can improve perceived attractiveness as shown in a perceptual study. In contrast to previous approaches, our machine-suggested makeup involves 3D information for both analysis and synthesis.

6.2 Discussion and Future Work

The presented data-driven approaches have shown that using structured prior knowledge can be a very powerful technique for the analysis of faces from images.

Thanks to pre-collected facial disparity maps we could eliminate all false alarms generated by state-of-the-art monocular face detectors (Chapter 3). By exploiting a database of 3D faces and making use of its statistical representation we could enhance, automate and accelerate the previously tedious and manual training procedure of face detectors (Chapter 4). With the help of before/after images of faces we could automate the modification of facial appearances in images, regardless of facial pose and illumination in the image (Chapter 5).

For each of the presented techniques and approaches we previously discussed possible improvements and proposed a variety of next steps for future work. However, more generally speaking, all findings from the presented approaches in this thesis could also be transferred to a wider range of applications. For example, face detectors are often only the first step for a variety of applications. In Chapter 3 and Chapter 4 we concentrated on this first step only. Nonetheless, the subsequent steps would also profit from the findings presented in this thesis: Applications such as face recognition would profit, for example, from knowledge about particular facial attributes. Face detectors could presumably also improve performance when automatically removing makeup from faces. The techniques presented in Chapter 5 could be helpful to solve this task. Furthermore, all applications shown have been evaluated for faces only. One aspect of future work should be to explore whether comparable principles also generalize to the full variety of objects in images. In industry, object detection is an important issue, for example, when monitoring automated production processes. Synthetic and self-adaptive training data, and improved and fail-save object detectors, are of interest for a variety of industrial applications. Similarly, the basic idea from Chapter 5 could be transferred to other application scenarios, for example, to the automated computation of best-fit clothing, with the help of a morphable body model.

In summary, the presented methods suggest a variety of new and exciting applications. The findings in thesis contribute a set of valuable tools that could be helpful addressing these future challenges.

6.3 Concluding Remarks

This thesis endorses and recommends the use of structured prior knowledge for the analysis of faces from images. In all presented approaches we were able to show that structured knowledge as extracted from facial databases turned out to be suitable for both analysis and synthesis of faces in images.

Prior knowledge is the key element that connects classical computer vision — such as object detection — to the field of computer graphics, where morphable models are already known as powerful tools. By combining both fields, analysis and synthesis of faces gear into each other and yield very robust and powerful tools to understand, analyze or modify faces in images.

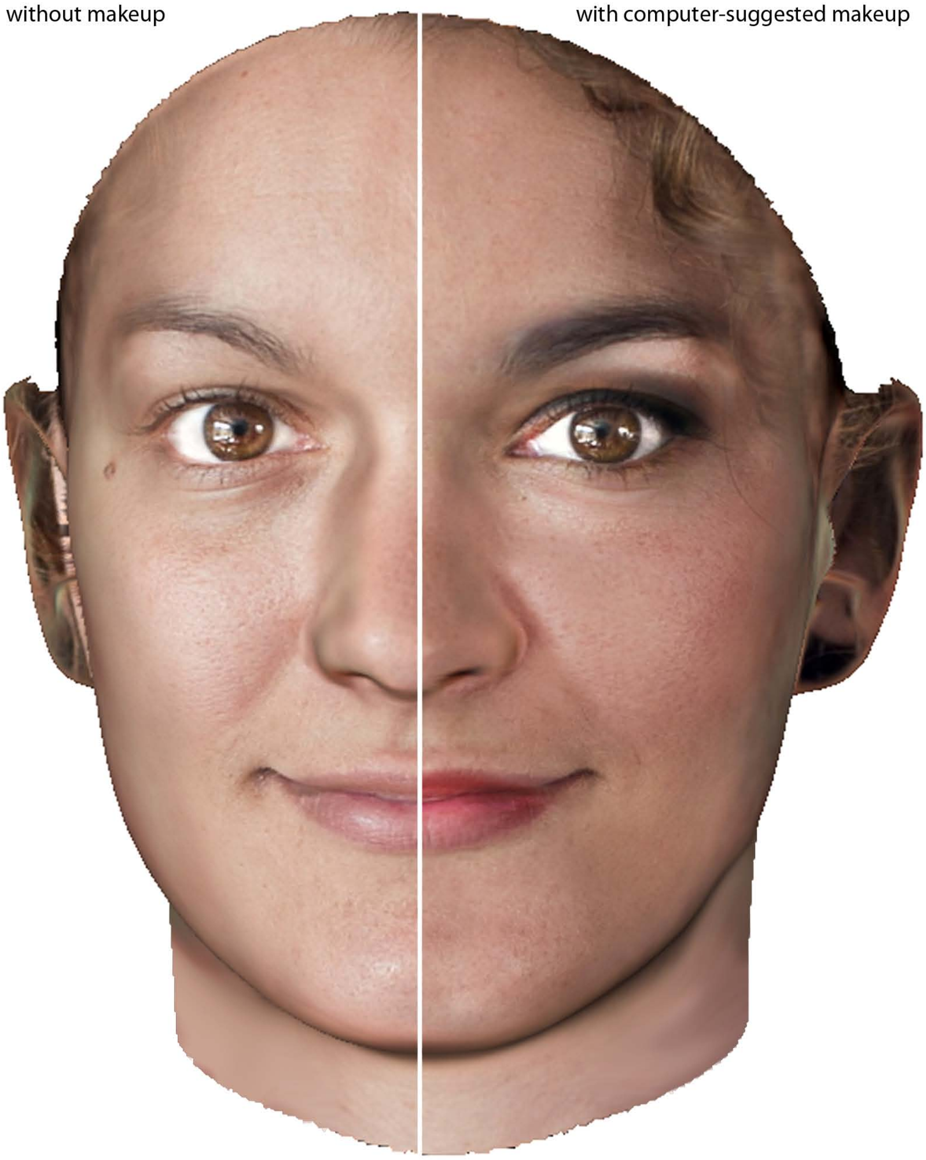
Appendix

Facial Appearance

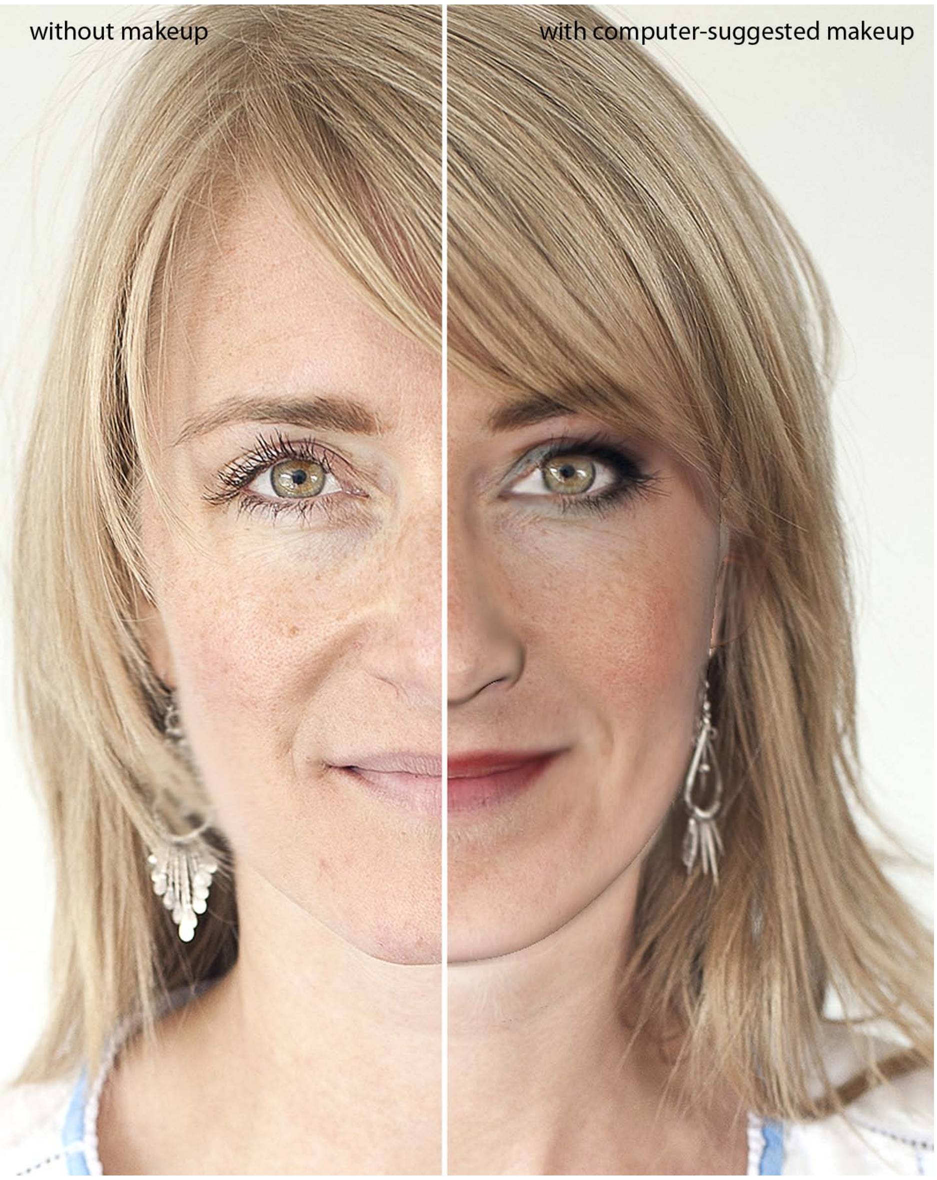
The following sections show detailed results of the user study conducted in Chapter 5, eigenmakeup renderings for the lip, skin and eye segments and example renderings of computer suggested makeup.

Computer Suggested Makeup Examples

Example 1:

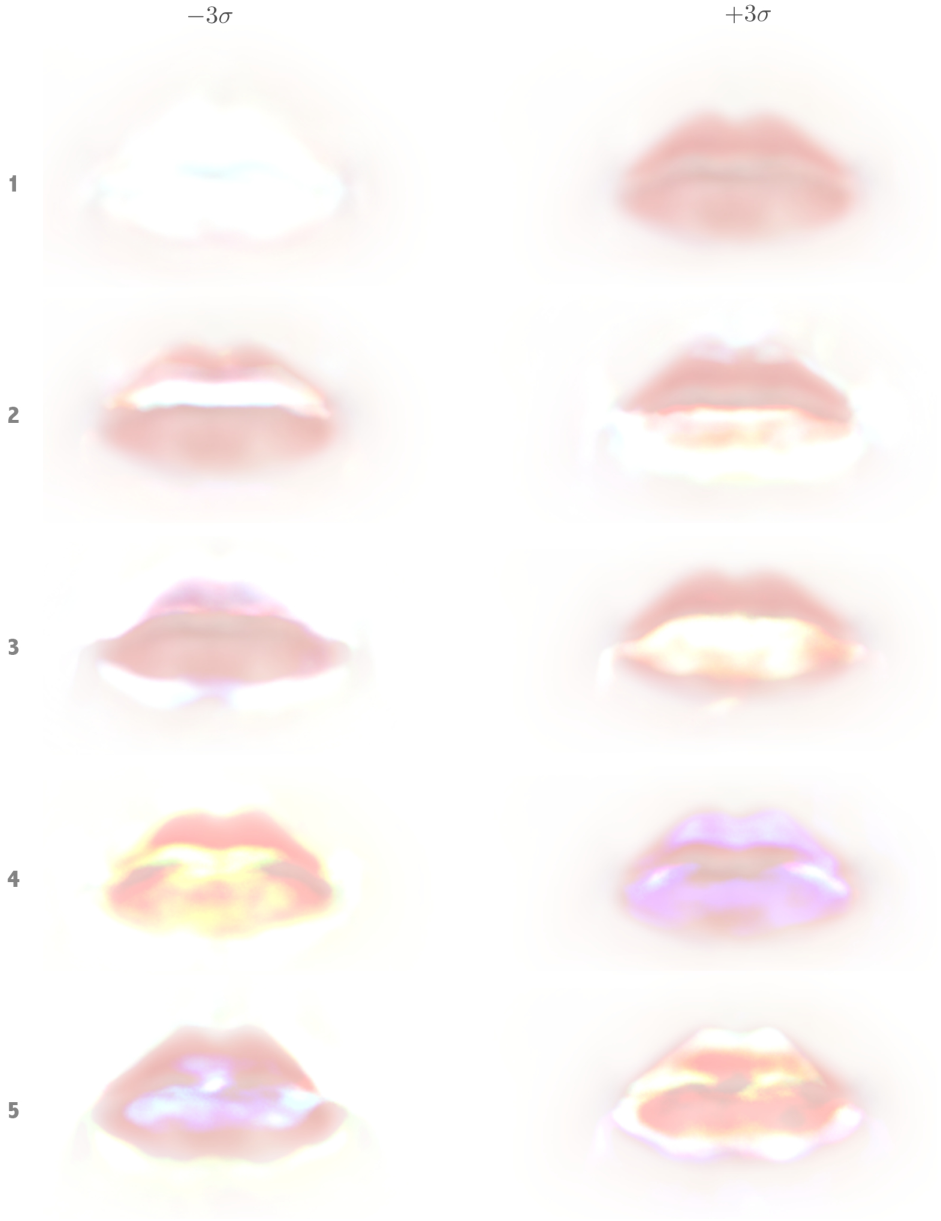


Example 2:

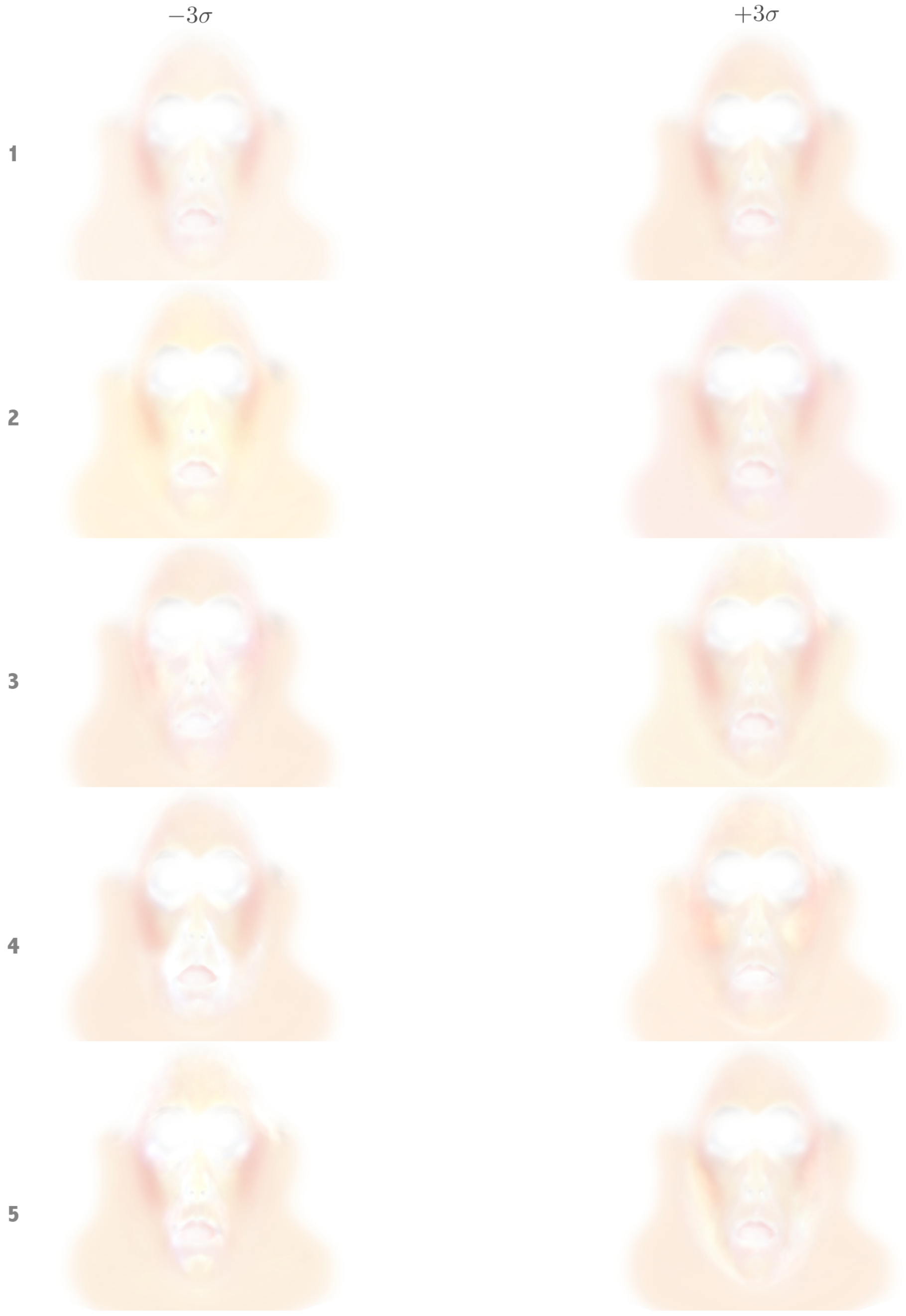


Eigenmakeup Renderings

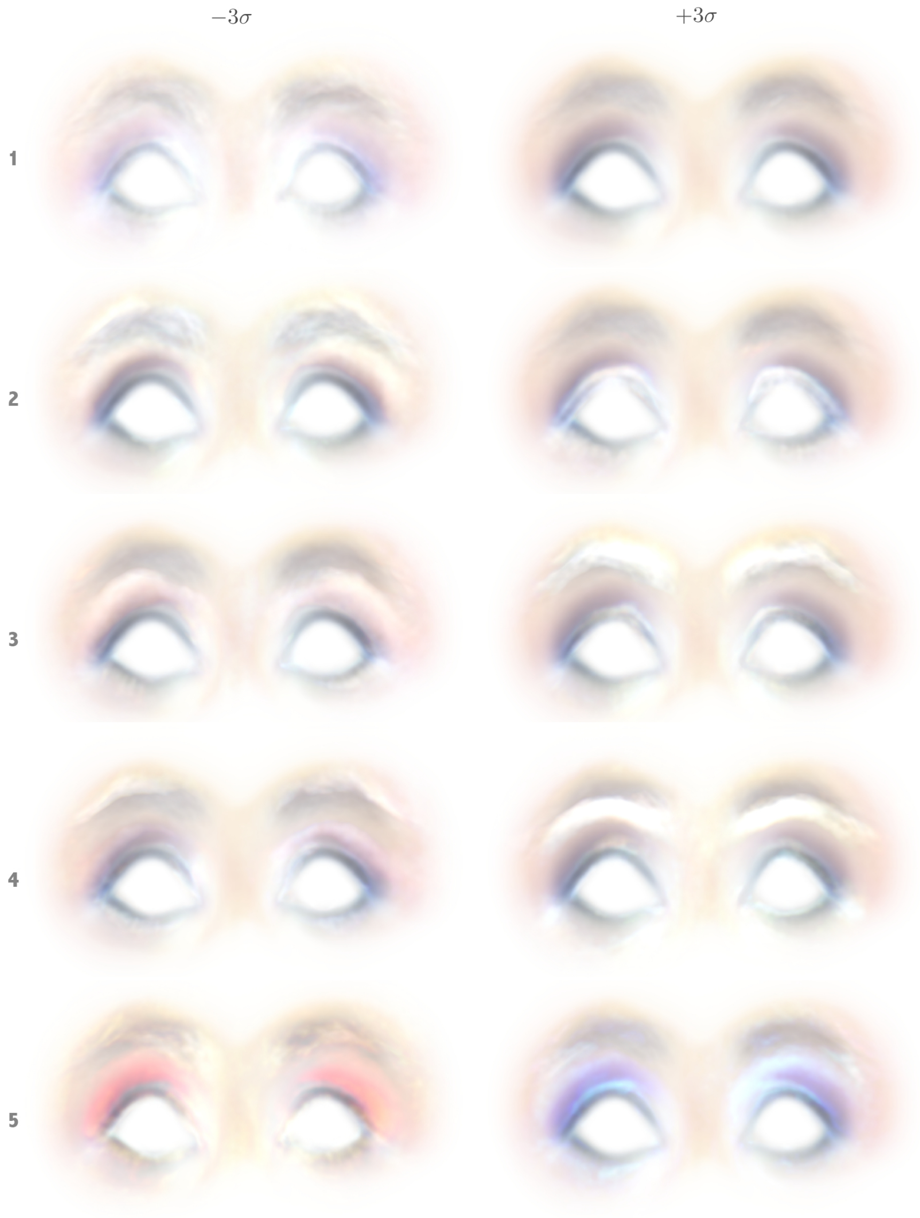
The first five eigenvectors of the lip segments:



The first five eigenvectors of the skin segments:



The first five eigenvectors of the eye segments:



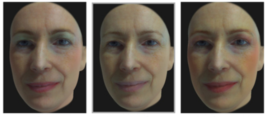
User Study Evaluation Results

Experiment 1:

	g	t	6	9	10	17	21	24	25	29	30	31	35	38	45	47	24	9	45
172	m	T1	a	b	a	a	a	b	b	a	b	a	b	a	a	b	b	A	B
287	M	T2	b	a	b	a	a	a	a	b	b	a	a	b	b	a	a	a	a
282	M	T2	a	a	b	a	b	b	a	a	b	b	a	b	a	b	a	b	a
89	M	T2	a	a	a	a	a	b	a	a	b	a	a	a	a	a	a	a	a
122	M	T2	b	b	a	b	a	a	a	a	a	a	a	b	a	a	a	b	a
394	M	T2	b	a	b	b	a	a	b	a	a	a	a	b	a	a	a	a	b
193	M	T2	b	a	b	a	a	b	a	b	b	b	a	a	a	b	a	a	a
145	M	T2	b	b	b	a	a	a	a	a	a	a	b	a	b	b	b	b	a
724	M	T2	a	a	b	b	a	a	b	b	b	b	a	a	b	a	b	a	a
320	M	T2	a	b	b	b	a	a	b	a	a	b	a	b	a	b	b	a	a
398	M	T2	a	b	b	b	a	a	a	b	a	a	a	a	b	a	a	a	a
490	M	T2	a	a	b	a	b	a	a	a	b	a	a	b	a	b	a	a	b
77	M	T2	a	b	a	a	a	b	a	b	a	b	b	a	b	a	a	b	a
213	M	T2	B	A	B	B	B	A	A	A	A	B	B	A	A	A	A	A	A
530	M	T2	a	b	b	a	b	b	a	b	b	b	b	b	a	a	b	a	b
106	m	T2	a	a	b	a	b	a	a	b	a	b	a	b	b	b	a	b	b
320	M	T2	b	b	b	a	a	b	b	a	b	a	b	b	b	b	a	b	b
268	m	T2	a	a	b	b	a	b	b	b	a	a	b	a	a	b	b	a	b
114	M	T2	a	a	b	a	a	b	a	a	b	a	a	a	a	b	a	b	a
372	F	T2	a	a	b	b	a	b	a	a	b	a	a	b	a	a	a	a	a
189	F	T2	b	a	a	a	b	b	a	a	a	a	a	b	b	a	b	b	b
121	F	T2	B	A	B	B	A	B	A	B	A	B	A	B	A	A	B	A	B
389	F	T2	b	a	b	a	a	b	a	a	a	a	b	a	a	a	a	b	a
74	F	T2	b	a	a	b	a	a	b	a	b	b	b	a	a	b	a	b	a
418	F	T2	b	a	b	a	a	a	a	b	b	a	a	a	a	b	a	b	b
32	F	T2	b	a	b	a	a	b	a	a	a	a	b	b	b	b	a	a	b
924	F	T2	a	b	b	a	a	a	a	a	b	b	a	a	a	a	a	b	a
69	F	T2	a	b	a	a	b	a	a	b	a	a	a	b	a	a	b	b	a
3074	F	T2	b	a	b	a	b	b	a	b	a	a	a	b	a	a	b	b	a
446	F	T2	b	a	a	a	a	a	b	b	b	b	a	b	a	b	a	a	a
552	F	T2	b	b	a	b	a	a	a	a	a	a	b	b	b	a	a	b	b
212	F	T2	b	a	b	a	a	a	a	b	a	b	a	a	b	a	a	b	b
172	F	T2	a	b	b	b	a	a	a	a	a	a	b	b	a	a	a	b	b
284	F	T2	a	b	b	a	a	a	a	a	b	a	b	a	a	a	a	a	b
110	F	T2	b	a	a	a	a	b	a	a	b	a	a	b	a	b	a	a	a
217	F	T2	b	a	b	b	a	b	a	a	b	b	a	b	a	b	b	a	b
88	F	T2	a	a	b	b	b	b	b	a	a	b	a	b	b	a	a	a	a
116	F	T2	a	a	a	a	a	b	a	b	b	a	a	a	a	b	b	a	a
68	F	T2	B	A	A	B	B	A	B	A	B	B	B	B	A	B	A	B	B
867	F	T2	b	b	a	b	a	b	b	a	b	a	a	a	a	b	a	b	b
1946	F	T2	a	a	b	a	b	a	a	b	b	b	b	a	a	a	b	a	a
111	F	T2	b	b	b	b	a	b	a	a	b	a	a	b	a	b	b	b	b
288	F	T2	a	b	b	b	a	a	a	a	a	a	a	a	a	a	b	b	a
51	F	T2	a	b	a	a	a	b	a	a	b	a	a	b	a	b	a	a	a
224	F	T2	b	a	b	a	b	a	a	b	b	a	a	a	a	b	b	a	a
479	F	T2	a	b	b	b	a	b	a	b	b	a	a	b	a	b	a	b	a
1072	F	T2	a	b	a	b	a	a	a	a	b	a	a	a	a	a	a	a	a

removed all samples below 30 sec
and with 3 checksum errors

Example Stimulus


Makeup A
Query face (no makeup)
Makeup B

#FEMALE	28
#MALE	19
#TOTAL	47

	6	9	10	17	21	24	25	29	30	31	35	38	45	47
groundtruth	B	A	B	A	A	B	A	A	B	A	A	B	A	A
total for A	24	28	15	27	35	24	36	29	20	30	32	22	35	24
total for B	23	19	32	20	12	23	11	18	27	17	15	25	12	23
	51%	60%	32%	57%	74%	51%	77%	62%	43%	64%	68%	47%	74%	51%
	49%	40%	68%	43%	26%	49%	23%	38%	57%	36%	32%	53%	26%	49%
female for A	12	17	11	15	21	14	23	18	11	19	20	12	23	15
female for B	16	11	17	13	7	14	5	10	17	9	8	16	5	13
	43%	61%	39%	54%	75%	50%	82%	64%	39%	68%	71%	43%	82%	54%
	57%	39%	61%	46%	25%	50%	18%	36%	61%	32%	29%	57%	18%	46%
male for A	12	11	4	12	14	10	13	11	9	11	12	10	12	9
male for B	7	8	15	7	5	9	6	8	10	8	7	9	7	10
	63%	58%	21%	63%	74%	53%	68%	58%	47%	58%	63%	53%	63%	47%
	37%	42%	79%	37%	26%	47%	32%	42%	53%	42%	37%	47%	37%	53%

#TASK 1
pro vs random

ALL
47

"pro is better"
62%

"random is better"
38%

FEMALE
28

"pro is better"
64%

"random is better"
36%

MALE
19

"pro is better"
58%

"random is better"
42%

chi square test		
Total votes:	*658*	*329*
Votes for Pro:	*406*	
Votes for Rand:	*252*	
Chi-Sqr:	*36,0425532*	
P-Value:	*1,9306E-09*	

Experiment 2:

seconds	*g*	*t*	1	2	3	9	10	17	21	31	35	38	40	45	48	49	3	31	48
209	m	T3	a	b	a	a	b	a	a	a	a	a	b	a	b	b	a	b	b
298	M	T3	a	a	a	a	b	a	a	a	a	b	b	a	b	a	b	a	a
425	M	T3	a	b	a	a	b	a	a	a	a	b	a	a	b	a	a	a	b
79	M	T3	b	b	a	a	b	a	a	a	a	b	a	a	b	a	a	a	b
309	M	T3	a	b	b	a	a	a	b	a	a	b	b	a	b	a	a	a	b
81	m	T3	b	b	a	a	b	b	a	a	a	a	b	b	b	a	a	a	a
229	M	T3	b	b	b	a	a	b	b	a	b	b	a	a	a	b	b	a	a
134	M	T3	a	a	a	a	b	a	b	a	a	a	a	b	b	a	a	a	b
160	m	T3	b	b	b	a	b	a	a	a	a	b	a	b	b	b	a	a	a
162	M	T3	a	b	a	b	b	b	a	a	b	b	a	b	a	b	a	a	b
171	M	T3	B	B	A	B	A	A	A	A	A	A	B	A	B	B	A	A	A
131	M	T3	a	a	b	a	b	a	a	a	a	a	b	a	a	a	b	b	a
399	M	T3	b	b	b	a	a	b	b	a	a	b	b	a	b	a	b	a	b
344	m	T3	a	a	b	b	b	b	b	a	a	b	a	b	a	b	a	a	b
92	M	T3	a	a	b	a	a	b	a	b	b	a	b	a	b	a	a	a	b
68	M	T3	a	a	a	a	b	b	a	b	a	b	a	a	b	b	a	a	b
279	M	T3	a	b	a	a	a	b	a	a	b	a	b	a	a	b	a	a	b
127	m	T3	a	a	a	b	b	b	a	a	b	a	a	a	b	b	a	a	b
239	M	T3	b	a	a	a	a	b	a	b	a	b	b	a	a	b	b	b	a
159	M	T3	a	b	a	a	b	a	a	a	b	a	a	a	b	a	b	a	b
319	m	T3	b	b	a	a	b	a	b	a	a	b	a	a	b	b	b	a	a
146	M	T3	b	b	b	a	a	a	a	a	b	b	b	b	a	a	b	a	b
85	M	T3	a	b	a	a	a	b	a	a	b	a	b	a	b	a	a	a	b
598	m	T3	a	b	a	b	a	a	a	b	a	b	a	a	b	a	a	b	a
538	M	T3	b	b	a	a	a	b	b	a	a	b	a	a	a	a	b	b	a
70	M	T3	a	b	b	a	b	a	a	a	a	b	b	a	b	a	a	a	b
278	M	T3	a	b	a	a	b	a	a	a	a	a	a	a	b	b	a	a	b
89	F	T3	B	B	B	A	B	B	A	A	A	B	B	A	B	B	B	B	B
351	f	T3	a	a	b	a	a	a	b	a	a	b	b	a	b	a	b	a	a
276	F	T3	a	b	a	a	a	b	b	a	a	a	b	b	b	a	b	b	b
49	F	T3	a	b	b	b	a	b	a	b	b	a	a	a	b	b	a	b	a
198	F	T3	a	b	a	a	a	a	b	a	a	b	a	a	a	a	a	a	b
77	F	T3	a	b	b	a	a	b	b	a	a	a	a	a	b	a	a	a	b
414	F	T3	b	b	a	b	b	b	b	a	a	b	a	a	b	a	b	b	b
293	F	T3	a	b	a	a	a	b	b	a	a	a	b	a	a	b	a	a	b
174	F	T3	a	a	a	a	b	a	a	a	b	a	a	b	a	a	b	a	b
552	F	T3	a	b	a	a	b	a	b	a	a	b	b	a	b	b	b	a	b
135	F	T3	a	b	b	a	b	a	a	b	a	b	b	a	b	a	b	a	a
82	F	T3	a	b	a	a	a	b	a	a	b	b	b	a	b	a	a	a	b
383	F	T3	A	A	B	A	A	B	A	A	A	B	A	B	B	A	A	A	A
92	f	T3	b	a	b	a	a	b	b	b	a	b	b	a	b	a	b	a	a
56	F	T3	a	b	a	a	b	b	a	a	b	b	b	a	b	a	b	a	b
178	F	T3	a	b	b	a	a	b	a	a	a	a	a	a	a	a	b	b	a
171	F	T3	b	b	a	a	b	a	a	a	a	b	a	a	b	b	a	a	a
329	F	T3	a	a	a	b	a	b	a	a	a	b	a	a	a	a	b	a	a

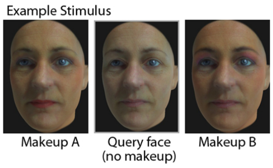
#FEMALE	18
#MALE	27
#TOTAL	45

	1	2	3	9	10	17	21	31	35	38	40	45	48	49
groundtruth	A	B	A	A	B	A	A	A	A	B	B	A	B	A
total for A	31	13	28	37	22	21	30	38	33	17	23	36	13	28
total for B	14	32	17	8	23	24	15	7	12	28	22	9	32	17
	69%	29%	62%	82%	49%	47%	67%	84%	73%	38%	51%	80%	29%	62%
	31%	71%	38%	18%	51%	53%	33%	16%	27%	62%	49%	20%	71%	38%
female for A	14	5	10	15	11	6	10	15	14	6	9	15	5	13
female for B	4	13	8	3	7	12	8	3	4	12	9	3	13	5
	78%	28%	56%	83%	61%	33%	56%	83%	78%	33%	50%	83%	28%	72%
	22%	72%	44%	17%	39%	67%	44%	17%	22%	67%	50%	17%	72%	28%
male for A	17	8	18	22	11	15	20	23	19	11	14	21	8	15
male for B	10	19	9	5	16	12	7	4	8	16	13	6	19	12
	63%	30%	67%	81%	41%	56%	74%	85%	70%	41%	52%	78%	30%	56%
	37%	70%	33%	19%	59%	44%	26%	15%	30%	59%	48%	22%	70%	44%

chi square test		
Total votes:	*630*	*315*
Votes for NN:	*419*	
Votes for FN:	*211*	
Chi-Sqr:	*68,6730159*	
P-Value:	*1,1622E-16*	

removed all samples below 30 sec
and with 3 checksum errors

Example Stimulus



#TASK 2
NN vs FN

"NN is better"
67%

"FN is better"
33%

ALL
45

"NN is better"
66%

"FN is better"
34%

FEMALE
18

"NN is better"
67%

"FN is better"
33%

MALE
27

Experiment 3:

removed all samples below 30 sec
and with 3 checksum errors

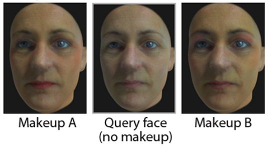
	g	t	4	6	8	9	17	19	21	24	27	31	33	35	52	54	17	24	31
124	M	T4	b	b	b	a	a	a	a	b	b	a	a	b	b	b	a	b	a
91	M	T4	b	b	b	a	b	b	b	a	a	a	a	a	a	b	b	a	b
92	M	T4	b	b	a	a	a	b	a	b	a	a	b	b	a	b	a	b	a
167	M	T4	a	b	b	a	a	a	b	b	b	a	a	a	b	b	a	a	b
182	M	T4	a	a	a	a	a	a	a	b	a	a	a	a	b	b	b	b	a
258	M	T4	a	a	b	b	b	a	b	b	a	a	a	b	b	b	a	b	a
139	M	T4	B	A	B	B	B	B	A	B	A	B	B	B	B	B	B	B	A
956	M	T4	b	b	a	b	b	b	b	a	a	b	a	b	b	b	a	a	b
246	M	T4	a	a	b	b	a	a	b	a	b	b	b	b	b	a	a	a	a
342	M	T4	a	a	a	b	a	a	b	b	a	a	b	b	b	a	a	b	b
721	M	T4	a	a	b	a	b	a	a	b	a	b	a	a	b	b	a	a	b
124	m	T4	a	a	b	b	a	a	b	b	b	a	b	a	a	a	a	b	a
837	m	T4	A	B	A	A	B	A	B	B	B	A	B	A	B	B	B	A	A
144	m	T4	a	b	a	a	a	b	b	a	b	b	a	b	b	a	b	a	a
208	M	T4	b	b	a	b	a	a	b	a	b	a	a	a	b	a	b	a	a
57	M	T4	a	b	a	a	a	b	a	b	a	b	a	b	b	a	b	a	b
153	M	T4	a	b	b	a	b	b	a	a	b	b	b	a	b	b	a	a	a
95	m	T4	a	b	b	a	b	a	a	a	b	b	a	b	a	b	a	a	b
103	M	T4	b	b	a	a	b	a	a	b	b	a	a	a	b	a	b	a	b
554	M	T4	a	a	b	b	b	a	a	b	b	a	b	a	b	b	a	b	a
265	M	T4	a	a	a	b	a	b	b	a	a	a	a	b	a	b	b	a	a
167	m	T4	a	a	b	a	b	b	a	a	a	b	a	b	b	a	a	b	b
40	M	T4	b	b	a	a	b	a	a	a	b	a	a	a	b	a	a	a	b
130	M	T4	b	a	b	b	a	a	b	b	b	a	b	b	b	a	a	a	a
208	M	T4	a	b	a	b	a	a	a	b	b	a	b	b	b	a	b	a	a
437	M	T4	a	b	a	b	b	a	a	a	b	a	b	b	b	b	b	a	a
810	M	T4	A	A	B	A	B	B	A	B	B	A	B	A	B	B	B	A	A
102	M	T4	a	a	a	a	b	a	b	b	b	a	a	a	a	b	b	a	a
908	M	T4	b	b	a	b	a	a	b	a	a	b	a	b	a	b	a	a	b
162	M	T4	b	a	b	a	a	b	b	b	a	a	b	a	b	a	b	b	a
348	M	T4	b	b	a	a	a	b	b	a	a	a	a	b	b	a	b	a	a
526	F	T4	a	a	b	a	b	b	a	a	a	a	a	a	b	b	b	a	a
263	F	T4	b	a	b	a	a	b	a	b	a	a	a	a	b	b	a	b	a
357	F	T4	b	b	a	a	a	a	b	b	a	a	b	a	b	a	a	b	a
149	f	T4	b	a	a	a	a	b	b	a	a	b	b	b	a	a	b	a	b
76	F	T4	b	a	a	a	a	a	b	a	a	b	b	a	a	b	a	a	a
268	F	T4	b	b	a	a	a	a	b	a	a	a	a	a	b	b	b	a	a
448	F	T4	a	a	b	b	a	a	a	b	a	a	a	b	b	a	a	b	b
244	F	T4	a	a	b	a	b	a	b	b	a	a	b	b	a	b	a	a	a
195	F	T4	a	b	a	a	b	a	a	a	a	a	a	a	b	b	b	b	a
92	f	T4	a	a	a	a	b	b	a	b	a	a	a	a	b	b	b	a	a
140	F	T4	a	b	b	a	b	a	a	b	b	a	b	b	a	b	b	a	b
217	F	T4	a	a	a	a	b	a	b	a	a	a	a	a	a	a	b	a	a
432	F	T4	a	b	b	b	a	a	b	b	a	a	b	a	b	b	b	a	a
262	F	T4	a	b	b	a	a	b	b	b	b	a	b	a	b	b	a	b	a
196	F	T4	a	b	b	b	a	a	a	a	b	a	b	a	b	b	b	b	a
193	F	T4	b	b	a	a	b	a	b	a	b	a	a	a	b	b	a	a	a
418	F	T4	b	a	b	a	b	a	b	b	b	a	a	a	b	a	a	b	b
439	F	T4	b	a	a	b	a	b	b	a	b	b	a	b	b	b	a	a	b
149	F	T4	a	b	a	a	b	a	b	a	b	a	a	a	b	b	b	b	a
241	F	T4	b	a	b	b	a	a	a	a	b	a	a	a	b	a	a	b	a
981	F	T4	B	B	A	A	A	A	A	B	A	A	A	A	B	B	A	B	A
812	F	T4	b	b	a	a	a	b	a	a	a	b	a	b	a	b	b	b	b
219	F	T4	b	b	a	a	b	a	b	a	a	a	a	b	b	a	a	a	b

#FEMALE	23
#MALE	31
#TOTAL	54

	4	6	8	9	17	19	21	24	27	31	33	35	52	54
groundtruth	B	B	B	A	A	A	A	B	A	A	A	A	B	B
total for A	30	25	29	36	29	35	25	26	29	40	33	30	13	20
total for B	24	29	25	18	25	19	29	28	25	14	21	24	41	34
	56%	46%	54%	67%	54%	65%	46%	48%	54%	74%	61%	56%	24%	37%
	44%	54%	46%	33%	46%	35%	54%	52%	46%	26%	39%	44%	76%	63%
female for A	11	11	13	18	13	16	10	13	15	19	15	16	6	7
female for B	12	12	10	5	10	7	13	10	8	4	8	7	17	16
	48%	48%	57%	78%	57%	70%	43%	57%	65%	83%	65%	70%	26%	30%
	52%	52%	43%	22%	43%	30%	57%	43%	35%	17%	35%	30%	74%	70%
male for A	19	14	16	18	16	19	15	13	14	21	18	14	7	13
male for B	12	17	15	13	15	12	16	18	17	10	13	17	24	18
	61%	45%	52%	58%	52%	61%	48%	42%	45%	68%	58%	45%	23%	42%
	39%	55%	48%	42%	48%	39%	52%	58%	55%	32%	42%	55%	77%	58%

chi square test		
Total votes:	*756*	*378*
Votes for NN:	*438*	
Votes for Rand:	*318*	
Chi-Sqr:	*19,047619*	
P-Value:	*1,275E-05*	

Example Stimulus



#TASK 3
NN vs Random

ALL
54

"NN is better"
58%

"Random is better"
42%

FEMALE
23

"NN is better"
62%

"Random is better"
54%

MALE
31

"NN is better"
55%

"Random is better"
47%

Figure Credits

[1] Portrait: "Albert Einstein". http://pixelicia.com/files/2012/08/Albert-Einstein.jpg. last checked: 01/2013.

[2] Portrait: "Albert Einstein". http://energysolar.org.uk/energy_solar_pics/einstein.jpg, last checked: 01/2013.

[3] Portrait: "Albert Einstein". http://deskarati.com/wp-content/uploads/2012/12/albert-einstein-colored-photo.jpg, last checked: 01/2013.

[4] Drawing: "Face Recognition Grand Challenge Database". http://www.nist.gov/itl/iad/ig/frgc.cfm, last checked: 01/2013.

[5] Portrait: "Lena Söderberg". http://en.wikipedia.org/wiki/File:Lenna.png, last checked: 01/2013.

[6] Portrait: "Marie Curie". http://bit.ly/19nIk34, last checked: 01/2013.

[7] Portrait: "Max Planck mit seiner Ehefrau, Photographie, Zentralbild, 1947". http://www.dhm.de/lemo/objekte/pict/plancbio/index.html, last checked: 01/2013. Deutsches Historisches Museum, Berlin.

[8] Portrait: "Max Planck, Photographie, F 54/94". http://www.dhm.de/lemo/objekte/pict/f54_94/index.html, last checked: 01/2013. Deutsches Historisches Museum, Berlin.

[9] Portrait: "Max Planck, Photographie, um 1932, F 78/328". http://www.dhm.de/lemo/objekte/pict/f78_328/index.html, last checked: 01/2013. Deutsches Historisches Museum, Berlin.

[10] Christian Schirm. http://commons.wikimedia.org/wiki/File:Wellenpakete_Summe.svg, last checked: 04/2013. Wikipedia.

[11] Rendering: "Teapot". http://www.cs.montana.edu/~halla/cs525/teapot-voxel.jpg, last checked: 01/2013.

[12] Drawing: "Theodoridis, Sergios and Koutroumbas, Konstantinos". *Pattern Recognition*. Academic Press, Inc., Orlando, FL, USA, 4 edition, 2009. 19, 20, 21, 26, 31

[13] Rendering: "Volumetric CT scan of a human head". http://www.csee.umbc.edu/~ebert/693/THu/GIF/h600wd5.gif, last checked: 01/2013.

[14] Rendering: "Volumetric CT scan of a human head". http://www.mathworks.com/matlabcentral/fx_files/32344/5/MRI_rendering.png, last checked: 01/2013.

Bibliography

[ALT08] R. Ach, N. Luth, and A. Techmer. Real-time detection of traffic signs on a multi-core processor. In *IEEE Intelligent Vehicles Symposium (IV 2008)*, pages 307–312, 2008. 67

[AMAM11] A. Ansari, M. Mahoor, and M. Abdel-Mottaleb. Normalized 3D to 2D model-based facial image synthesis for 2D model-based face recognition. In *IEEE GCC Conference and Exhibition (GCC 2011)*, pages 178–181, 2011. 68

[AMM10] C. Atanasoaei, C. McCool, and S. Marcel. A principled approach to remove false alarms by modelling the context of a face detector. In *British Machine Vision Conference (BMVC 2010)*, pages 17.1–17.11. BMVA Press, 2010. doi:10.5244/C.24.17. 61

[BBPV03] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating Faces in Images and Video. *Computer Graphics Forum (EUROGRAPHICS 2003)*, 22(3):641–650, 2003. 40, 45, 46

[BCF06] K. Bowyer, K. Chang, and P. Flynn. A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition. *Computer Vision and Image Understanding*, 101(1):1–15, 2006. 53, 68, 54

[BGPV05] V. Blanz, P. Grother, P. Phillips, and T. Vetter. Face recognition based on frontal views generated from non-frontal images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, volume 2, pages 454–461, 2005. 68

[Bla11] M. B. Blaschko. Branch and Bound Strategies for Non-maximal Suppression in Object Detection. In *Lecture Notes in Computer Science*, volume 6819, pages 385–398, 2011. 61

[BLK10] O. Barinova, V. Lempitsky, and P. Kohli. On Detection of Multiple Object Instances using Hough Transforms. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, 2010. 50

[BSS07] V. Blanz, K. Scherbaum, and H.-P. Seidel. Fitting a Morphable Model to 3D Scans of Faces. In *IEEE International Conference on Computer Vision (ICCV 2007)*, pages 1–8. Ieee, 2007. 40, 43, 51, 53, 42, 54, 59, 90

[BSVS04] V. Blanz, K. Scherbaum, T. Vetter, and H.-P. Seidel. Exchanging Faces in Images. *Computer Graphics Forum (EUROGRAPHICS 2004)*, 23(3):669–676, 2004. 40

[BV99] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. *ACM Transactions on Graphics (SIGGRAPH 1999)*, pages 187–194, 1999. 4, 31, 32, 34, 39, 40, 42, 46, 51, 65, 68, 69, 73, 87, 41, 45, 99

[BV03] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI 2003)*, 25(9):1063–1074, 2003. 34

[BVMT03] W. Boehler, M. B. Vicent, A. Marbs, and S. Technology. Investigating Laser Scanner Accuracy. *19th CIPA symposium (CIPA 2003)*, 27 Suppl 1(October), 2003. 11

[BWKS06] A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. A Multigrid Platform for Real-Time Motion Computation with Discontinuity-Preserving Variational Methods. *International Journal of Computer Vision (IJCV 2006)*, 70(3):257–277, 2006. 55, 57

[BWSM08] S. Brubaker, J. Wu, J. Sun, and M. Mullin. On the design of cascades of boosted ensembles for face detection. *International Journal of Computer Vision (IJCV 2008)*, 77(1-3):65–86, 2008. 52

[CBMK09] J. Cho, B. Benson, S. Mirzaei, and R. Kastner. Parallelized architecture of multiple classifiers for face detection. In *IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP 2009)*, pages 75–82, 2009. 67

[CCDS09] M. Callieri, P. Cignoni, M. Dellepiane, and R. Scopigno. Pushing Time-of-Flight Scanners to the Limit. In *International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST 2009)*, pages 85–92, 2009. 15

[CHL11] Y.-Y. Chen, W. H. Hsu, and H.-Y. M. Liao. Learning facial attributes by crowdsourcing in social media. In *International Conference Companion on World Wide Web (WWW 2011)*, pages 25–26, New York, NY, USA, 2011. ACM. 67

[CJ82] E. N. Coleman and R. Jain. Obtaining 3-Dimensional Shape of Textured and Specular Surfaces Using Four-Source Photometry. *Computer Graphics and Image Processing*, 18:309–328, 1982. 18

[CJC13] S. L. H. Chuang Jan Chang. Lso-adaboost based face detection for ip-cam video. In W.-H. Hsieh, editor, *Applied Mechanics and Materials*, pages 3543–3548, 2013. 67

[CKLL11] C.-H. Chiang, C.-H. Kao, G.-R. Li, and B.-C. Lai. Multi-level parallelism analysis of face detection on a shared memory multi-core system. In *International Symposium on VLSI Design, Automation and Test (VLSI-DAT 2011)*, pages 1–4, 2011. 67

[CKO$^+$08] H.-C. Choi, S.-Y. Kim, S.-H. Oh, S.-Y. Oh, and S.-Y. Cho. Pose invariant face recognition with 3d morphable model and neural network. In *International Joint Conference on Neural Networks (IJCNN 2008)*, pages 4131–4136, 2008. 68

[CMOK09] J. Cho, S. Mirzaei, J. Oberg, and R. Kastner. Fpga-based face detection system using haar classifiers. In *ACM/SIGDA international Symposium on Field Programmable Gate Arrays (FPGA 2009)*, pages 103–112, New York, NY, USA, 2009. ACM. 67

[Com94] P. Comon. Independent component analysis, A new concept? *Signal Processing*, 36(3):287–314, 1994. 26

[Cor72] R. Corson. *Fashion in makeup: From Ancient to Modern Times*. Peter Owen Publishers, 1972. 87, 88

[CR68] F. W. Campbell and J. G. Robson. Application of Fourier analysis to the visibility of gratings. *J. Physiol. Paris*, 197(3):551–566, 1968. 31

[CSC$^+$10] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt. 3d shape scanning with a time-of-flight camera. In *CVPR*, pages 1173–1180, 2010. 14

[CSL08] T. Chen, H.-P. Seidel, and H. P. A. Lensch. Modulated phase-shifting for 3d scanning. In *CVPR*, 2008. 14

[CST$^+$13] Y. Cui, S. Schuon, S. Thrun, D. Stricker, and C. Theobalt. Algorithms for 3d shape scanning with a depth camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(5):1039–1050, 2013. 14

[CTB92] I. Craw, D. Tock, and A. Bennett. Finding Face Features. *IEEE European Conference on Computer Vision (ECCV 1992)*, pages 92–96, 1992. 50

[CXS$^+$01] H. Chen, Y.-q. Xu, H.-y. Shum, S.-c. Zhu, and N.-n. Zheng. Example-based facial sketch generation with non-parametric sampling. *IEEE International Conference on Computer Vision (ICCV 2001)*, pages 433–438, 2001. 88

[DDS$^+$09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 248–255, 2009. 67

[DG10] L. Denoyer and P. Gallinari. A ranking based model for automatic image annotation in a social network. In *International AAAI Conference on Weblogs and Social Media (ICWSM 2010)*, 2010. 67

[DGHW00] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated Person Tracking Using Stereo, Color, and Pattern Detection. *International Journal of Computer Vision (IJCV 2000)*, 37(2):175–185, 2000. 53, 54

[DHT$^+$00] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. *27th annual conference on Computer graphics and interactive techniques (SIGRAPH 2000)*, pages 145–156, 2000. 87

[DS02] O. Drbohlav and R. Sara. Specularities Reduce Ambiguity of Uncalibrated Photometric Stereo. *IEEE European Conference on Computer Vision (ECCV 2002)*, pages 46–62, 2002. 18

[DS12] H. Du and L. Shu. Makeup transfer using multi-example. In *Proceedings of the 2012 International Conference on Information Technology and Software Engineering*, 2012. 88

[DWD$^+$08] C. Donner, T. Weyrich, E. D'Eon, R. Ramamoorthi, and S. Rusinkiewicz. A layered, heterogeneous reflectance model for acquiring and rendering human skin. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 27(5), 2008. 87

[DXQ09] Y. Dai, G. Xiao, and K. Qiu. Efficient face recognition with variant pose and illumination in video. In *International Conference on Computer Science Education (ICCSE 2009)*, pages 18–22, 2009. 68

[EDR06] Y. Eisenthal, G. Dror, and E. Ruppin. Facial attractiveness: beauty and the machine. *Neural computation*, 18(1):119–42, 2006. 89

[ETC98] G. Edwards, C. Taylor, and T. Cootes. Learning to Identify and Track Faces in Image Sequences. *IEEE International Conference on Computer Vision (ICCV 1998)*, pages 317–322, 1998. 50

[EZW$^+$05] M. Everingham, A. Zisserman, C. Williams, L. V. Gool, M. Allan, C. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorko, S. Duffner, J. Eichhorn, J. Farquhar, M. Fritz, C. Garcia, T. Griffiths, F. Jurie, D. Keysers, M. Koskela, J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. Shawe-Taylor, A. Storkey, S. Szedmak, B. Triggs, I. Ulusoy, V. Viitaniemi, and J. Zhang. The 2005 pascal visual object classes challenge. In *in First PASCAL Challenges Workshop*. Springer-Verlag, 2005. 67

[FCA$^+$09a] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent. Large-scale privacy protection in Google Street View. *International Conference on Computer Vision (ICCV 2009)*, pages 2373–2380, 2009. 48

[FCA$^+$09b] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent. Large-scale privacy protection in google street view. *IEEE International Conference on Computer Vision (ICCV 2009)*, pages 2373–2380, 2009. 77

[FFFPZ05] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google"s image search. In *IEEE International Conference on Computer Vision (ICCV 2005)*, pages 1816–1823, Washington, DC, USA, 2005. IEEE Computer Society. 67

[FMG$^+$12] C. Florea, M. Moldovan, M. Gordan, A. Vlaicu, and R. Orghidan. Eye color classification for makeup improvement. In *Federated Conference on Computer Science and Information Systems (FedCSIS 2012),*, pages 55–62, 2012. 106

[Fri13] R. Frischholz. `http://www.facedetection.com/facedetection/datasets.htm`, last checked: 02/2013. The Face Detection Homepage. 67, 68

[FvDFH96] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, Ma, 2. edition, 1996. 44

[GB10] C. Galleguillos and S. Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722, 2010. 52

[Geo03] A. S. Georghiades. Incorporating the Torrance and Sparrow Model of Reflectance in Uncalibrated Photometric Stereo. *IEEE International Conference on Computer Vision (ICCV 2003)*, 2:816–823, 2003. 18

[Grg13] M. Grgic. `http://www.face-rec.org/databases/`, last checked: 02/2013. Face Recognition Homepage. 67, 68

[Gro05] R. Gross. Face databases. In A. S.Li, editor, *Handbook of Face Recognition*. Springer, New York, 2005. 67, 68

[GS09] D. Guo and T. Sim. Digital face makeup by example. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 1:73–79, 2009. 86, 88, 99

[HA04] V. J. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004. 20, 19

[HALL05a] C. Huang, H. Ai, Y. Li, and S. Lao. Vector boosting for rotation invariant multi-view face detection. *IEEE International Conference on Computer Vision (ICCV 2005)*, 1:446–453 Vol. 1, 2005. 52

[HALL05b] C. Huang, H. Ai, Y. Li, and S. Lao. Vector boosting for rotation invariant multi-view face detection. *IEEE International Conference on Computer Vision (ICCV 2005)*, 2005. 79

[HHG00] J. V. Haxby, E. A. Hoffman, and M. I. Gobbini. The distributed human neural system for face perception. *Trends in cognitive sciences*, 4(6):223–233, 2000. 87

[HKO01] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*. Adaptive and learning systems for signal processing, communications, and control. J. Wiley, 2001. 26

[Hor89] B. K. P. Horn. Shape from Shading. In *Obtaining shape from shading information*, chapter Obtaining, pages 121–171. MIT Press, Cambridge, MA, USA, 1989. 18

[HRBLM07] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007. 67

[JLM03] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *ACM International Conference on Research and Development in Information Retrieval (SIGIR 2003)*, pages 119–126, New York, NY, USA, 2003. ACM. 67

[JLM10] V. Jain and E. Learned-Miller. FDDB: A benchmark for face detection in unconstrained settings. Technical report, University of Massachusetts, Amherst, 2010. 64, 65, 66, 67, 79

[Jon03] M. Jones. Fast multi-view face detection. *Mitsubishi Electric Research Lab TR-20003-96*, 2003. 51, 52

[JP97] T. Jebara and A. Pentland. Parameterized Structure from Motion for 3D Adaptive Feedback Tracking of Faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI 1997)*, pages 144–150, 1997. 50

[JR02] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *Int. J. Comput. Vision*, 46(1):81–96, January 2002. 106

[JSG09] J. Jimenez, V. Sundstedt, and D. Gutierrez. Screen-space perceptual rendering of human skin. *ACM Transactions on Applied Perception (TAP 2009)*, 6(4):1–15, 2009. 91, 98

[Kan73] T. Kanade. *Picture processing system by computer complex and recognition of human faces*. PhD thesis, Kyoto University, 1973. 49

[KCS08] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with Mechanical Turk. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 453–456, New York, NY, USA, 2008. ACM. 104

[KD10] J. Kong and Y. Deng. GPU accelerated face detection. In *International Conference on Intelligent Control and Information Processing (ICICIP 2010)*, pages 584–588, 2010. 67

[KP97] C. Kotropoulos and I. Pitas. Rule-based face detection in frontal views. In *IEEE International Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1997)*, volume 4, pages 2537–2540. IEEE, 1997. 49

[KS05] H. Kemalekenel and B. Sankur. Multiresolution face recognition. *Image and Vision Computing*, 23(5):469–477, 2005. 51

[KSF$^+$09] S. Kosov, K. Scherbaum, K. Faber, T. Thormählen, and H.-P. Seidel. Rapid stereo-vision enhanced face detection. In *ICIP*, pages 1221–1224, 2009. 6

[LBP95] T. K. Leung, M. C. Burl, and P. Perona. Finding Faces in Cluttered Scenes Using Random Labeled Graph Matching. *IEEE International Conference on Computer Vision (ICCV 1995)*, pages 637–644, 1995. 50

[LCL11] B.-C. Lai, C.-H. Chiang, and G.-R. Li. Data locality optimization for a parallel object detection on embedded multi-core systems. In *IEEE International Conference on Software Engineering and Service Science (ICSESS 2011)*, pages 576–579, 2011. 67

[LCODL08] T. Leyvand, D. Cohen-Or, G. Dror, and D. Lischinski. Data-driven enhancement of facial attractiveness. *ACM Transactions on Graphics (SIGGRAPH 2008)*, 27(3):1, 2008. 89

[LCQ$^+$04] Y.-M. Li, J. Chen, L.-Y. Qing, B.-C. Yin, and W. Gao. Face detection under variable lighting based on resample by face relighting. In *International Conference on Machine Learning and Cybernetics (ICMLC 2004)*, volume 6, pages 3775–3780, 2004. 68

[Lev85] M. D. Levine. *Vision in Man and Machine*. McGraw–Hill, 1985. 31

[LKP03] R. Lienhart, E. Kuranov, and V. Pisarevsky. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. In *Annual Symposium of the Association for Pattern Recognition (DAGM 2003)*, pages 297–304, 2003. 51, 52

[LM02] R. Lienhart and J. Maydt. An Extended Set of Haar-Like Features for Rapid Object Detection. In *IEEE International Conference on Image Processing (ICIP 2002)*, pages 900–903, 2002. 51, 55, 57, 52

[LR90] J. H. Langlois and L. A. Roggman. Attractive faces are only average. *Psychological science*, 1990. 89

[LSZ01] Z. Liu, Y. Shan, and Z. Zhang. Expressive expression mapping with ratio images. *ACM Transactions on Graphics (SIGGRAPH 2001)*, pages 271–276, 2001. 93, 106

[LTC94] A. Lanitis, C. J. Taylor, and T. F. Cootes. An automatic face identification system using flexible appearance models. In *Image and Vision Computing (Journal)*, pages 393–401, BMVA Press, 1994. 50

[LZZ+06] S. Li, L. Zhu, L. Zhang, A. Blake, H. Zhang, and H. Shum. Statistical learning of multi-view face detection. *IEEE European Conference on Computer Vision (ECCV 2006)*, 2006. 51, 52

[MCNK09] E. McKone, K. Crookes, and N. Nancy Kanwisher. The Cognitive and Neural Development of Face Recognition in Humans. *The Cognitive Neurosciences*, pages 467–482, 2009. 2

[MGMHJ04] M. C. Motwani, M. C. Gadiya, R. C. Motwani, and F. C. Harris Jr. Survey of image denoising techniques. In *Proceedings of GSPX*, pages 27–30. Citeseer, 2004. 20

[MGR98] S. McKenna, S. Gong, and Y. Raja. Modelling Facial Colour and Identity with Gaussian Mixtures. *Pattern Recognition*, 31(12):1883–1892, 1998. 50

[MHP+07] W.-C. Ma, T. Hawkins, P. Peers, C.-F. Chabert, Malte Weiss, P. E. Debevec, and M. Weiss. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. *Eurographics Symposium on Rendering (EGSR 2007)*, pages 183–194, 2007. 91

[MKH05] T. Mita, T. Kaneko, and O. Hori. Joint Haar-like features for face detection. *IEEE International Conference on Computer Vision (ICCV 2005)*, 2:1619–1626 Vol. 2, 2005. 51, 52

[MP11] M. G. Moazzam and M. R. Parveen. Human Face Detection under Complex Lighting Conditions. *International Journal of Advanced Computer Science and Applications (IJACSA 2011)*, pages 85–90, 2011. 50

[MPK10] A. Makadia, V. Pavlovic, and S. Kumar. Baselines for image annotation. *International Journal of Computer Vision (IJCV 2010)*, 90(1):88–105, 2010. 67

[MWL+99] S. R. Marschner, S. H. Westin, E. P. F. Lafortune, K. E. Torrance, and D. P. Greenberg. Image-Based BRDF Measurement Including Human Skin. In *Proc. EGWR*, pages 139–152, 1999. 87

[Nel01] C. A. Nelson. The development and neural bases of face recognition. *Infant and Child Development*, 10(1-2):3–18, 2001. 2

[NHO+13] T. Nguyen, D. Hefenbrock, J. Oberg, R. Kastner, and S. Baden. A software-based dynamic-warp scheduling approach for load-balancing the viola-jones face detection algorithm on gpus. *Journal of Parallel and Distributed Computing*, 2013. 67

[NKGR06] S. K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar. Fast separation of direct and global components of a scene using high frequency illumination. *ACM Transactions on Graphics (SIGGRAPH 2006)*, 25(3):935–944, 2006. 91

[NRDR05] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi. Efficiently combining positions and normals for precise 3D geometry. *ACM Transactions on Graphics (SIGGRAPH 2005)*, 24(3):536, 2005. 91

[NRK99] K. Numata, K. Ri, and K. Kira. HD "E-make": A real-time HD skin-make-up machine. *Eizo Joho Media Gakkai Gijutsu Hokoku*, 24:45–50, 1999. 88

[NY08] S. G. Narasimhan and S. Yamazaki. Temporal dithering of illumination for fast active vision. In *In European Conference on Computer Vision*, pages 830–844, 2008. 13

[PBPV+99] D. I. Perrett, D. M. Burt, I. S. Penton-Voak, K. J. Lee, D. A. Rowland, and R. Edwards. Symmetry and human facial attractiveness. *Evolution and human behavior*, 20(5):295–307, 1999. 89

[PC07] M.-T. Pham and T.-J. Cham. Fast training and selection of Haar features using statistics in boosting-based face detection. *IEEE International Conference on Computer Vision (ICCV 2007)*, pages 1–7, 2007. 52

[PFS+05a] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the Face Recognition Grand Challenge. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 947–954, 2005. 55, 59

[PFS+05b] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 947–954, Washington, DC, USA, 2005. IEEE Computer Society. 67

[PGHC10] M.-T. Pham, Y. Gao, V. Hoang, and T.-J. Cham. Fast polygonal integration and its application in extending haar-like features to improve object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pages 942–949, 2010. 67

[PJW+11] L. Pishchulin, A. Jain, C. Wojek, T. Thormählen, and B. Schiele. In Good Shape: Robust People Detection based on Appearance and Shape. *British Machine Vision Conference (BMVC 2011)*, 2011. 68

[PLPV+98] D. I. Perrett, K. J. Lee, I. S. Penton-Voak, D. A. Rowland, S. Yoshikawa, D. M. Burt, S. P. Henzik, D. L. Castles, and S. Akamatsu. Effects of sexual dimorphism on facial attractiveness. *Nature*, 394(394):884–887, 1998. 89

[PN12] S. Preethi and D. Narmadha. Article: A survey on image denoising techniques. *International Journal of Computer Applications*, 58(6):27–30, 2012. Published by Foundation of Computer Science, New York, USA. 20

[PSLS09] R. Pereira, L. Seabra Lopes, and A. Silva. Semantic image search and subset selection for classifier training in object recognition. In *Portuguese Conference on Artificial Intelligence: Progress in Artificial Intelligence (EPIA 2009)*, pages 338–349. Springer-Verlag, 2009. 67

[PTW10] L. Pishchulin, T. Thormählen, and C. Wojek. Learning People Detection Models from Few Training Samples. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pages 1–8, 2010. 68

[RBK98] H. Rowley, S. Baluja, and T. Kanade. Neural Network-Based Face Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI 1998)*, 20(1):23–38, 1998. 51

[RR03] R. Russel and R. Russell. Sex, beauty, and the relative luminance of facial features. *Perception*, 32(9):1093–1107, 2003. 88

[RRV04] M. Rätsch, S. Romdhani, and T. Vetter. Efficient Face Detection by a Cascaded Support Vector Machine Using Haar-Like Features. *Pattern Recognition*, pages 62–70, 2004. 52

[RT05] A. Rama and F. Tarres. P2CA: a new face recognition scheme combining 2D and 3D information. In *IEEE International Conference onImage Processing (ICIP 2005)*, volume 3, pages III–776–9, 2005. 68

[RTMF08] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: A Database and Web-Based Tool for Image Annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008. 67

[SBOR05] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell. Face recognition by humans: 20 results all computer vision researchers should know about. *Department of Brain and Cognitive Sciences Massachusetts Institute of Technology Cambridge, MA*, 2005. 2

[SCLT07] T.-H. Sun, M. Chen, S. Lo, and F.-C. Tien. Face Recognition Using 2D and Disparity Eigenface. *Expert Systems with Applications Journal*, 33(2):265–273, 2007. 53, 54

[SFP$^+$13] K. Scherbaum, R. S. Feris, J. Petterson, V. Blanz, and H.-P. Seidel. Fast Face Detector Training Using Tailored Views. In *IEEE 14th International Conference on Computer Vision (ICCV 2013)*, Sydney, Australia, December 3-6, 2013. to appear. 6

[SGdA$^+$10] C. Stoll, J. Gall, E. de Aguiar, S. Thrun, and C. Theobalt. Video-based reconstruction of animatable human characters. *ACM Transactions on Graphics (TOG 2010)*, 29(6):139:1–139:10, 2010. 18

[Sha92] A. Shashua. Geometry and Photometry in 3D Visual Recognition. In *Neural Information Processing Systems 4*, pages 404–411. Massachusetts Inst. of Technology, 1992. 17

[SI96] F. Solomon and K. Ikeuchi. Extracting the Shape and Roughness of Specular Lobe Objects Using Four Light Photometric Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI 1996)*, 18:449–454, 1996. 18

[Sil80] W. Silver. *Determining Shape and Reflectance Using Multiple Images*. PhD thesis, Massachusetts Inst. of Technology, 1980. 17

[SN07] J. Seyama and R. S. Nagayama. The Uncanny Valley: Effect of Realism on the Impression of Artificial Human Faces. *Presence: Teleoperators & Virtual Environments*, 16(4):337–351, 2007. 87

[SRE$^+$05] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. *IEEE International Conference on Computer Vision (ICCV 2005)*, 1:370–377, 2005. 67

[SRH$^+$11] K. Scherbaum, T. Ritschel, M. B. Hullin, T. Thormählen, V. Blanz, and H.-P. Seidel. Computer-Suggested Facial Makeup. *Computer Graphics Forum (EUROGRAPHICS 2011)*, 30(2):485–492, 2011. 6, 7

[SSSB07] K. Scherbaum, M. Sunkel, H.-P. Seidel, and V. Blanz. Prediction of Individual Non-Linear Aging Trajectories of Faces. *Computer Graphics Forum (EUROGRAPHICS 2007)*, 26(3):285–294, 2007. 4, 31, 40, 41, 42, 59, 65, 68, 69, 90, 99, 32

[ST94] J. Shi and C. Tomasi. Good Features to Track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1994)*, pages 593–600, 1994. 58

[STD08] S. Schuon, C. Theobalt, and J. Davis. High-quality scanning using time-of-flight depth superresolution. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2008)*, pages 1–7, 2008. 15

[STVK09] B. Sharma, R. Thota, N. Vydyanathan, and A. Kale. Towards a robust, real-time face processing system using CUDA-enabled GPUs. In *International Conference on High Performance Computing (HiPC 2009*, pages 368–377, 2009. 67

[TA07] M. Toews and T. Arbel. Detecting and Localizing 3D Object Classes using Viewpoint Invariant Reference Frames. In *IEEE International Conference on Computer Vision (ICCV 2007)*, pages 1–8, 2007. 68

[TdF91] H. Tagare and R. de Figueiredo. A Theory of Photometric Stereo for a Class of Diffuse Non-Lambertian Surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI 1991)*, 13:133–152, 1991. 18

[Tel04] A. Telea. An Image Inpainting Technique Based on the Fast Marching Method. *J. Graphics Tools*, 9(1):23–34, 2004. 90

[TFAS00] J.-C. Terrillon, H. Fukamachi, S. Akamatsu, and M. N. Shirazi. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. *IEEE International Conference on Automatic Face and Gesture Recognition (FG 2000)*, pages 54–63, 2000. 78

[TFF08] A. Torralba, R. Fergus, and W. Freeman. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI 2008)*, 30(11):1958–1970, 2008. 67

[TH95] L.-A. Tang and T. Huang. Face recognition using synthesized intermediate views. In *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS 1995)*, volume 2, pages 1066–1069, 1995. 68

[Tho05] K. Thomson. *Sometimes Less Is Not More: A Fabulous Guide to Fabulous Makeup*. Lipstick Press, 2005. 86

[TJL$^+$11] D. Tsai, Y. Jing, Y. Liu, H. Rowley, S. Ioffe, and J. Rehg. Large-scale image annotation using visual synset. In *IEEE International Conference on Computer Vision (ICCV 2011)*, pages 611–618, 2011. 67

[TK09] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, Inc., Orlando, FL, USA, 4 edition, 2009. 19, 20, 21, 26, 31, 22, 27

[TLQ08] P. Tan, S. Lin, and L. Quan. Subpixel photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI 2008)*, 30(8):1460–1471, 2008. 17

[TOS$^+$03] N. Tsumura, N. Ojima, K. Sato, M. Shiraishi, H. Shimizu, H. Nabeshima, S. Akazaki, K. Hori, and Y. Miyake. Image-based skin color and texture analysis/synthesis by extracting hemoglobin and melanin information in the skin. *ACM Transactions on Graphics (SIGGRAPH 2003)*, 22(3):770, 2003. 87, 88

[TP91] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1991)*, pages 586–591, 1991. 51, 52, 57, 94, 53

[Tsa87] R. Y. Tsai. A Versatile Camera Calibration Technique For High-accuracy 3-D Machine Vision Metrology Using Off-the-Shelf Cameras and Lenses. *IEEE Transaction on Robotics and Automation*, 3(4):323–344, 1987. 55

[TTBX07] W.-S. Tong, C.-K. Tang, M. Brown, and Y.-Q. Xu. Example-based cosmetic transfer. In *Pacific Conference on Computer Graphics and Applications (PG 2007)*, pages 211–218, 2007. 86, 88

[TTS03] F. Tsalakanidou, D. Tzovaras, and M. G. Strintzis. Use of depth and colour eigenfaces for face recognition. *Pattern Recognition Letters*, 24(9-10):1427–1435, 2003. 53, 54

[vAD04] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (SIGCHI 2004)*, CHI 2004, pages 319–326, New York, NY, USA, 2004. ACM. 67

[VJ01a] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, page 511, 2001. 30, 51, 57, 65, 66, 77

[VJ01b] P. Viola and M. Jones. Robust real-time face detection. *IEEE International Conference on Computer Vision (ICCV 2001)*, 2:747, 2001. 51

[VJ03] P. Viola and M. Jones. Detecting pedestrians using patterns of motion and appearance. *IEEE International Conference on Computer Vision (ICCV 2003)*, 2003. 51, 52

[VJ04] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision (IJCV 2004)*, 57(2):137–154, 2004. 51, 65, 77

[VK95] M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding (Prentice Hall Signal Processing Series)*. Prentice Hall PTR, 1995. 31

[WAHL04] B. Wu, H. Ai, C. Huang, and S. Lao. Fast rotation invariant multi-view face detection based on real adaboost. *IEEE Automatic Face and Gesture Recognition (FG 2004)*, 2004. 50, 52

[WBMR08] J. Wu, S. C. Brubaker, M. D. Mullin, and J. M. Rehg. Fast Asymmetric Learning for Cascade Face Detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI 2008)*, 30(3):369–382, 2008. 52

[WDDF09] L. Wang, L. Ding, X. Ding, and C. Fang. Improved 3D assisted pose-invariant face recognition. In *IEEE International Conference onAcoustics, Speech and Signal Processing (ICASSP 2009)*, pages 889–892, 2009. 68

[WGHG10] Z. Wang, O. Gnawali, K. Heath, and L. J. Guibas. Collaborative image annotation using image webs. *Army Science Conference (ASC 2010)*, 2010. 67

[WHHB04] B. Weyrauch, B. Heisele, J. Huang, and V. Blanz. Component-based face recognition with 3d morphable models. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop, 2004. (CVPRW '04)*, page 85, 2004. 68

[WJG+06] T. Weyrich, H. W. Jensen, M. Gross, W. Matusik, H. Pfister, B. Bickel, C. Donner, C. Tu, J. McAndless, J. Lee, A. Ngan, and J. L. Janet McAndless. Analysis of human faces using a measurement-based skin reflectance model. *ACM Transactions on Graphics (SIGGRAPH 2006)*, 25(3):1013, 2006. 87, 91

[WKS+07] J.-G. Wang, H. Kong, E. Sung, W.-Y. Yau, and E. K. Teoh. Fusion of appearance image and passive stereo depth map for face recognition based on the bilateral 2DLDA. *Journal of Image Video Processing*, 2007(2):6, 2007. 53, 54

[WLCV07] J.-G. Wang, E. T. Lim, X. Chen, and R. Venkateswarlu. Real-time Stereo Face Recognition by Fusing Appearance and Depth Fisherfaces. *Journal of VLSI Signal Processing*, 49(3):409–423, 2007. 53, 54

[WLL+07] T. Weyrich, J. Lawrence, H. P. a. Lensch, S. Rusinkiewicz, and T. Zickler. Principles of Appearance Acquisition and Representation. *Foundations and Trends® in Computer Graphics and Vision*, 4(2):75–191, 2007. 87

[WLV04] J.-G. Wang, E. T. Lim, and R. Venkateswarlu. Stereo head/face detection and tracking. In *IEEE International Conference on Image Processing (ICIP 2004)*, pages 605–608, 2004. 53, 54

[WM11] G. Wei and C. Ming. The face detection system based on GPU+CPU desktop cluster. In *International Conference on Multimedia Technology (ICMT 2011)*, pages 3735–3738, 2011. 67

[WMW06] S. Winkelbach, S. Molkenstruck, and F. Wahl. Low-cost laser range scanner and fast surface registration approach. *Pattern Recognition*, pages 718–728, 2006. 11

[Woo78] R. Woodham. Photometric stereo: A reflectance map technique for determining surface orientation from image intensity. *SPIE Annual Technical Symposium (SPIE 1978)*, pages 136–143, 1978. 17

[WV09] M. Walker and T. Vetter. Portraits made to measure: Manipulating social judgments about individuals with a statistical face model. *Journal of Vision*, 9(11):1–13, 2009. 89

[WWC$^+$11] Y.-T. Wu, Y.-T. Wu, C.-Y. Cho, S.-Y. Tseng, C.-N. Liu, and C.-T. King. Parallel integral image generation algorithm on multi-core system. In *IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA 2011)*, pages 31–35, 2011. 67

[YC97] K. Yow and R. Cipolla. Feature-Based Human Face Detection. *Image and Vision Computing*, 15(9):713–735, 1997. 50

[YH94] G. Yang and T. S. Huang. Human face detection in a complex background. *Pattern Recognition*, 27(1):53–63, 1994. 49

[YKA02] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: a survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI 2002)*, 24(1):34–58, 2002. 49, 50, 65

[YW96] J. Yang and A. Waibel. A Real-Time Face Tracker. *IEEE Workshop on Applications of Computer Vision (WACV 1996)*, pages 142–147, 1996. 50

[Zha10] N. Zhang. Working towards efficient parallel computing of integral images on multi-core processors. In *Computer Engineering and Technology (ICCET 2010)*, volume 2, pages V2–30–V2–34, 2010. 67

[ZPDD10] D. Zhou, D. Petrovska-Delacrétaz, and B. Dorizzi. 3D Active Shape Model for Automatic Facial Landmark Location Trained with Automatically Generated Landmark Points. In *International Conference on Pattern Recognition (ICPR 2010)*, pages 3801–3805, 2010. 68

[ZV12] E. Zaytseva and J. Vitrià. A search based approach to non maximum suppression in face detection. In *ICIP*, pages 1469–1472, 2012. 61

[ZZ10] C. Zhang and Z. Zhengyou. A survey of recent advances in face detection. *Microsoft Research, Technical Report MSR-TR-2010-66*, 2010. 52, 65