DOI: 10.1111/cgf.14087

COMPUTER GRAPHICS *forum*
Volume 39 (2020), number 6 pp. 480–495

Progressive Acquisition of SVBRDF and Shape in Motion

Hyunho Ha, Seung-Hwan Baek, Giljoo Nam and Min H. Kim

School of Computing, KAIST, Daejeon, Korea
{hhha, shwbaek, gjnam}@vclab.kaist.ac.kr, minhkim@kaist.ac.kr

Abstract

To estimate appearance parameters, traditional SVBRDF acquisition methods require multiple input images to be captured with various angles of light and camera, followed by a post-processing step. For this reason, subjects have been limited to static scenes, or a multiview system is required to capture dynamic objects. In this paper, we propose a simultaneous acquisition method of SVBRDF and shape allowing us to capture the material appearance of deformable objects in motion using a single RGBD camera. To do so, we progressively integrate photometric samples of surfaces in motion in a volumetric data structure with a deformation graph. Then, building upon recent advances of fusion-based methods, we estimate SVBRDF parameters in motion. We make use of a conventional RGBD camera that consists of the colour and infrared cameras with active infrared illumination. The colour camera is used for capturing diffuse properties, and the infrared camera-illumination module is employed for estimating specular properties by means of active illumination. Our joint optimization yields complete material appearance parameters. We demonstrate the effectiveness of our method with extensive evaluation on both synthetic and real data that include various deformable objects of specular and diffuse appearance.

Keywords: Inverse rendering, SVBRDF, 3D reconstruction

ACM CCS: • Computing methodologies → Reflectance modelling

1. Introduction

Capturing material appearance has been a long-lasting research problem in computer graphics. Many specialized hardware systems and software have been proposed to capture appearance parameters, which can be used for photorealistic rendering of real-world objects [DRS10]. Besides the cost of building a specialized hardware setup, a long process of material acquisition is required. It begins with capturing various photometric observation samples with diverse angles of light and camera, resulting in hundreds of images [GHP*08, HLZ10, NLW*16, TFG*13, SSWK13, FHW*11, TAL*07, LWS*13]. The process is then followed by heavy computational processes that include calibration, registration, inverse rendering and so on, often resulting in computation taking numerous hours.

In addition to the system-building cost and the long hours of processing, the entire input images with different angles of light and camera should be captured in advance to be processed all together for the optimization process of inverse rendering. This setup forces the captured object to be static. If the object moves with motion or is deformed into a different shape, the registration and geometric relationship of input images are broken so that the entire input images should be recaptured from scratch to estimate appearance parameters. The state-of-the-art material appearance acquisition techniques assume that a target object is both *rigid* and *static*. Neither deformation nor motion has been allowed in traditional acquisition methods. To address the aforementioned drawbacks of the existing solutions, we were motivated to capture the material appearance of a dynamic object in motion like a person or any deformable object such as cloth.

Acquiring the appearance of dynamic objects has been achieved by developing a specialized multiview video system [TAL*07, FHW*11, LWS*13]. However, these systems are limited to capturing subjects placed within the multiple light-camera stage. Also, these systems are significantly more expensive than practical acquisition methods. In contrast, we were motivated to devise a practical acquisition solution without requiring any specialized hardware setup, such as a mechanical gantry with two robotic arms or a multiview camera-light stage. To this end, we decided to make use of a conventional RGBD camera for our acquisition setup, following

Figure 1: *(a) We provide the first-ever method to simultaneously estimate the SVBRDF, shape and motion of dynamic objects using a single RGBD camera. (b)–(e) We obtain both diffuse and specular appearance with our novel joint optimization scheme, based on our hierarchical data structure, which allows us to render captured scenes under novel view and light conditions. Refer to the supplemental video for more results.*

the trend of state-of-the-art practical techniques [AWL15, RPG16, HSL*17, RRFG17, WZ15, WWZ16, PNS18, NLGK18].

The conventional RGBD camera that we used in this work is a Kinect 2 sensor that consists of two camera modules: a colour imaging module is an ordinary colour camera to capture red, green and blue colours of objects, and a time-of-flight (TOF) imaging module is an infrared camera to capture the depth information with active illumination of an infrared light module. We utilize the colour camera for capturing diffuse properties and use the pair of the infrared camera and the infrared illumination module to estimate specular properties.

In this work, we propose a progressive estimation of the spatially varying bidirectional reflectance distribution function (SVBRDF) and the shape of a deformable object in motion using a single RGBD camera. As we are using a depth camera, we can estimate the shape and motion vectors of the target object simultaneously while estimating appearance. We introduce a novel architecture to progressively integrate photometric observation samples in motion in a volumetric structure through a deformation graph. Existing works using a single camera can capture SVBRDFs of static objects based on a hierarchical data structure that consists of multiple clusters of similar appearance. To the best of our knowledge, none of these methods can acquire SVBRDF and surface geometry with motion simultaneously. Our method estimates not only geometry with motion but also *SVBRDFs*.

In addition, the traditional material acquisition methods [GHP*08, HLZ10, NLW*16, TFG*13, SSWK13, FHW*11, TAL*07, LWS*13] require several hours to capture input images of rigid objects. Our novel inverse rendering framework allows us to estimate SVBRDF parameters and shape information *progressively* in interactive time as we build our framework by combining the recent advances of fusion-based methods [NFS15, IZN*16, GXY*17] and the practical inverse rendering technique that captures SVBRDF with active illumination [NLGK18, WZ15]. Our progressive acquisition approach does not need to wait for several hours to capture input images. From an application perspective, it does not force the target object to be static until all of the input images are captured. Our method can progressively update both appearance and shape parameters simultaneously. Processing each frame takes less than a half second with a single GPU to estimate every parameter from photometric samples accumulated through motion vectors.

In summary, our method is the first to bridge the gap between SVBRDF acquisition of rigid objects and fusion-based dynamic scanning of diffuse colours, allowing for simultaneous acquisition of SVBRDF and shape in motion. Our main contributions are summarized as follows:

- an architecture to accumulate photometric samples of a dynamic object in a volumetric structure through a deformation graph of motion,
- a joint optimization framework that can estimate SVBRDF, shape and motion simultaneously, and
- a progressive appearance computation framework for inverse rendering.

2. Related Work

Appearance Acquisition of Static Objects. Traditionally, material appearance of static objects has been effectively acquired with specialized hardware systems that consist of multiple lights or cameras [GHP*08, HLZ10, TFG*13, SSWK13, GCHS10, NLW*16, RRFG17, BJTK18]. However, the building cost of such systems is too high to make the acquisition process not available for casual users to have access to this acquisition process. To resolve this issue, practical methods using a single camera have been introduced [AWL15, RPG16, HSL*17, RRFG17, WZ15, WWZ16, SWK19, PNS18, NLGK18]. These methods can capture material appearance by inferring diffuse and specular appearance parameters from multiple observations with different view/light angles. Although being effective for appearance and shape estimation, these methods are limited to capturing *static* objects, meaning objects without any motion. In contrast, we extend the target objects of appearance acquisition to *dynamic* objects through a joint estimation of appearance, shape and motion.

Multi-Camera Acquisition of Dynamic Objects. To capture the geometry and appearance of dynamic objects, various specialized multi-camera systems have been proposed. Most previous systems only target diffuse appearance, neglecting specular appearance [WVT12, DKD*16, DDF*17, XSH*19]. There have been few attempts to estimate the complete appearance of diffuse and specular

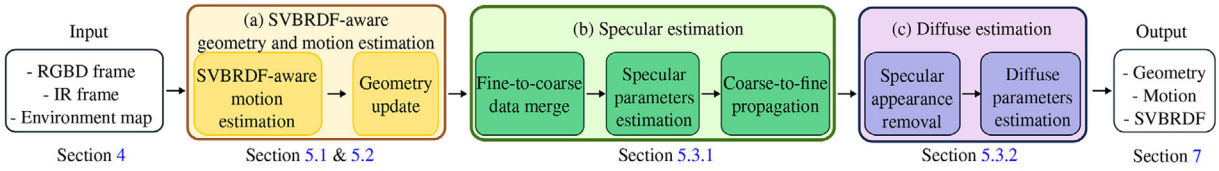


Figure 2: *For each frame, our method takes inputs of RGB, IR and depth images from a conventional RGBD sensor (Kinect 2), in addition to the static environment map (only captured once at the calibration stage). (a) We first estimate motion fields and scene geometry with consideration of SVBRDF. (b) Specular parameters are then estimated by exploiting the hierarchical data structure. (c) Given the specular estimates, residual observation is fitted to the diffuse component, resulting in the diffuse albedo estimates. This framework runs in an online manner, producing geometry, motion and SVBRDF per frame as output.*

components simultaneously [FHW*11, TAL*07, LWS*13]. However, these multiview methods require very expensive acquisition systems with multiple cameras and lights. They are also offline methods with high computational costs. In contrast, our method *progressively* estimates diffuse/specular parameters, geometry and motion in an *online* manner using a *single* RGBD camera, which makes them more practical.

Single-Camera Acquisition of Dynamic Objects. Although estimating the shape and motion of dynamic objects from a single camera has been extensively studied, estimating appearance simultaneously is known to be challenging [NFS15, SBCI17, LZG18, YGX*17, YZG*18, ZYL*18, YGX*17, YZG*18, ZYL*18, YZZ*19]. Only a few studies have been attempted to capture diffuse components either in diffuse albedo [GXY*17] or shaded diffuse colours [DDF*17, IZN*16, SBI18]. In addition, there are practical acquisition methods that allows users to capture appearance in a simple setup. Lin *et al*. [LPG19] estimate appearance parameters by simply capturing HDR images of an object and a light probe. Dong *et al*. [DCP*14] capture SVBRDFs with known geometry from an input video with motion. The main technical challenge of simultaneously estimating *specular appearance* is that the number of light/view samples in each frame is not sufficient for appearance estimation. In order to overcome this, we utilize the active infrared illumination in the TOF camera for estimating specular parameters and integrate photometric samples into a hierarchical data structure. This enables us to reconstruct all appearance parameters, geometry and motion of dynamic objects simultaneously.

3. Overview

Our method progressively estimates SVBRDF, geometry and motion of a deformable object in a frame-by-frame manner. Based on the traditional fusion framework [NFS15, GXY*17], using an RGBD camera, our method accumulates photometric samples of the target object in our hierarchical data structure. The data structure allows us to estimate the full appearance of the object using a small number of frames. Using the estimated appearance parameters of the object, our algorithm progressively updates the appearance information more accurately over time through the object's motion vectors. Figure 2 describes the overview of our method.

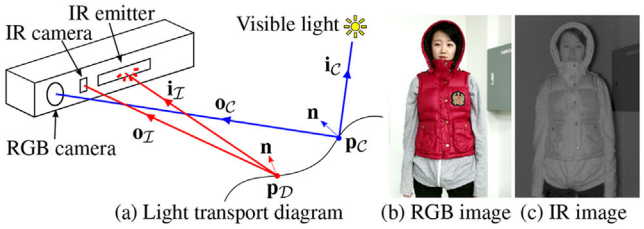


Figure 3: *(a) Environmental scene illumination in visible RGB channels is reflected at object surfaces and captured by the colour camera on the RGBD sensor. Estimating the specular component from the RGB image (b) is challenging due to lack of the view/light direction information. In contrast, the point IR illumination of the depth camera enables effective reconstruction of specular appearance from the IR image (c).*

4. Acquisition Setup

In order to make our acquisition system practical, we make use of two off-the-shelf imaging devices: a conventional RGBD camera (Kinect 2) and a 360° camera (Ricoh Theta). The RGBD camera is the main device to capture SVBRDF and shape in motion, and the spherical camera is used to capture the environment illumination of the scene.

We chose the RGBD camera because the camera consists of the colour and infrared cameras with active infrared illumination. First, the colour camera can be used for capturing diffuse colour properties under the scene ambient illumination. Second, unlike the previous generation of RGBD cameras (Kinect 1 or PrimeSense), the second generation of the Kinect sensor includes the TOF camera module to estimate depth. The camera API allows us to access to raw infrared image data, time-modulated phase images under active infrared (IR) illumination, without having spatial modulation artefacts shown in the previous generation. The clear infrared image data under the known active illumination can be utilized to estimate view-light-dependent reflectance property, that is, specular albedo and surface roughness. As the angle between active infrared illumination and infrared TOF sensor in the RGB-D camera is approximately 5° at a distance of around 1 m, this could be sufficiently wide to capture most SVBRDF except the Fresnel effect, as discussed in [NJR15, NLGK18].

Table 1: *Symbols and notations used in the paper.*

	Symbol	Description
Image	t	Frame number
	u	Image pixel
	$\mathcal{P}$	Pixel domain
	$\mathcal{P}^t_{\mathcal{D}}$, $\mathcal{P}^t_{\mathcal{C}}$	Set of visible pixels at the depth camera space and the colour camera space at t
	$\tilde{u}_{\mathcal{D}}$, $\tilde{u}_{\mathcal{C}}$	Corresponding pixel of a rendered image pixel u in the depth and the colour camera space
	$\tilde{u}_{\mathbf{x}_{\mathcal{D}}}$	Corresponding depth pixel of a voxel $\mathbf{x}$ in the depth camera space
	$\mathbf{D}^t$, $\mathbf{C}^t$, $\mathbf{I}^t$	Depth, colour and IR image at the frame t
	Γ, Y	Chromacity and luminance of the colour image
	$\tilde{\mathbf{V}}^t_{\mathcal{D}}$, $\tilde{\mathbf{N}}^t_{\mathcal{D}}$	Vertex map and normal map of the warped mesh at the depth camera space at t
	$\tilde{\mathbf{V}}^t_{\mathcal{C}}$, $\tilde{\mathbf{N}}^t_{\mathcal{C}}$	Vertex map and normal map of the warped mesh at the colour camera space at t
	$\mathbf{V}^t_{\mathcal{D}}$, $\mathbf{N}^t_{\mathcal{D}}$	Vertex map and normal map of the depth image at t
	$\mathbf{O}^t_{\mathcal{C}}$	View direction of $\tilde{\mathbf{V}}^t_{\mathcal{C}}$ to the colour camera at t
Transformation	$\mathcal{K}$, $\mathcal{D}$, $\mathcal{C}$	Canonical, depth (IR) and colour camera space
	P	Perspective projection
	$\mathbf{T}_i$	Deformation graph transformation matrix at the node i
	$\mathbf{K}_{\mathcal{D}}$, $\mathbf{K}_{\mathcal{C}}$	Depth (IR) camera, Colour camera intrinsic matrix
	$\mathbf{T}^t_{\mathcal{K}\to\mathcal{D}}$	Canonical space to depth camera space transformation matrix at t
	$\mathbf{T}^t_{\mathcal{D}\to\mathcal{K}}$	Depth camera space to canonical space transformation matrix at t
	$\mathbf{T}^t_{\mathcal{K}\to\mathcal{C}}$	Canonical space to colour camera space transformation matrix at t
	$\mathbf{T}^t_{\mathcal{D}\to\mathcal{C}}$	Depth camera space to colour camera space transformation matrix at t
Geometry	$\mathbf{p}^t_{\mathcal{K}}$, $\tilde{\mathbf{p}}^t_{\mathcal{K}}$, $\mathbf{n}^t_{\mathcal{K}}$, $\tilde{\mathbf{n}}^t_{\mathcal{K}}$	Point in the canonical space and its warped point and its normal at t
	$\mathbf{x}$, $\mathbf{x}_{\mathcal{K}}$	Voxel and its canonical position
	$\tilde{\mathbf{x}}^t_{\mathcal{D}}$, $\tilde{\mathbf{x}}^t_{\mathcal{C}}$, $\tilde{\mathbf{n}}^t_{\mathcal{D}}$, $\tilde{\mathbf{n}}^t_{\mathcal{C}}$	Position of a voxel $\mathbf{x}$ at the depth camera and the colour camera space and its normal at t
	$\mathcal{T}$	TSDF structure
	$d_{\mathcal{T}}$, $\omega_{\mathcal{T}}$	Signed distance value and its weight
	$\mathcal{T}^t_{\mathcal{D}}$, $\mathcal{T}^t_{\mathcal{C}}$	Set of visible voxel at the depth space and the colour space at t
	$\mathcal{V}_{\mathcal{K}}$, $\tilde{\mathcal{V}}_{\mathcal{K}}$	Canonical frame and warped mesh
Motion	$\mathcal{G}$	Deformation graph
	$\mathcal{W}^t$	Motion field at frame t
	$\mathbf{q}_i$, σ_i, w_i	Position, radius and the weight of the ith deformation graph node
	Φ	Kernel funcion
Parameters	λ	Regularize parameter
	κ	IR emmiter illumination
	$\gamma_{\mathcal{C}}$, $\gamma_{\mathcal{I}}$	Gamma value of the colour and the IR camera
	$v^t_{\mathbf{x}}$, $\omega^t_{\mathbf{x}}$	Half angle buffer value and its weight of a voxel $\mathbf{x}$ at t
	v^t_m, ω^t_m	Half angle buffer value and its weight of a cluster m at t
Reflectance	$\mathcal{A}$	Appearance of the canonical space model
	$\boldsymbol{\rho}_d$, ρ_s, α	Diffuse albedo, specular albedo and specular roughness
	$\tilde{\mathbf{o}}^t_{\mathcal{D}}$, $\tilde{\mathbf{o}}^t_{\mathcal{C}}$	View direction of $\tilde{\mathbf{x}}^t_{\mathcal{D}}$ to the depth camera at t and $\tilde{\mathbf{x}}^t_{\mathcal{C}}$ to the colour camera at t
	$\tilde{\mathbf{i}}^t_{\mathcal{I},\mathcal{D}}$	Light direction of $\tilde{\mathbf{x}}^t_{\mathcal{D}}$ to the IR emitter at t
	$\theta_{\mathbf{i}}$, $\theta_{\mathbf{o}}$, $\theta_{\mathbf{h}}$	Zenith angle between the normal plane and the light, view and half vector direction
	f_r, f_d, f_s	Reflectance, diffuse reflectance, and specular reflectance function
	H_k, l_k	Spherical harmonics basis function and coefficient
	B, S	Diffuse and specular reflection
	$\mathcal{M}$	Cluster

Figure 1(a) presents our acquisition setup. Figure 3 depicts light transport in our imaging setup. First, to estimate the incident illumination of the scene, we capture an environment map using a 360° camera. Second, for each frame, an RGB colour frame is captured by the colour camera module in Kinect 2 under the ambient scene illumination. An infrared frame is captured by the TOF camera module under active illumination of the TOF camera module in the RGBD camera. We use both colour and infrared frames in the video stream to estimate the diffuse and specular parameters of SVBRDF.

5. Capturing SVBRDF and Shape in Motion

To capture SVBRDF on a non-rigid object using a single RGBD camera, we introduce a two-step framework of dynamic inverse rendering. First, we calculate motion fields by comparing both the appearance and geometry of the current frame with those properties of the static model continuously accumulated from previous frames. Second, using the estimated motion fields, we update three properties sequentially in each frame: geometry, specular reflectance

parameters and diffuse albedo, sequentially in each frame. Refined geometry is used for estimating the parameters of specular albedo and surface roughness from photometric samples under active infrared illumination of the TOF camera. We cluster these parameters in the deformation graph structure to estimate SVBRDFs per cluster. Our SVBRDF acquisition progressively captures diffuse albedo, specular coefficient, specular roughness, geometry and motion frame by frame under visible environment illumination.

5.1. Background

5.1.1. *Voxel Grid and SVBRDF Model*

Voxel Grid. We make use of a truncated signed distance function (TSDF) volume $\mathcal{T}$ [CL96] to store the estimated appearance and shape information. We reconstruct actual geometry and appearance properties in the 3D voxel grid of TSDF, which is a set of voxel $\mathbf{x} \in \mathbb{N}^3$ that consists of two sets of properties: $\mathcal{T}=\{\mathcal{V}, \mathcal{A}\}$. First, geometry $\mathcal{V}=\{[d_\mathcal{T}, \omega_\mathcal{T}]\}^t$ at frame t is defined as a signed distance value $d_\mathcal{T}$ and its weight $\omega_\mathcal{T}$. Second, appearance $\mathcal{A}=\{[\boldsymbol{\rho}_d, \rho_s, \alpha]\}^t$ is a set of diffuse albedo $\boldsymbol{\rho}_d$, specular albedo ρ_s and roughness parameter α. As appearance parameters are surface properties, we only accumulate the appearance parameters to the voxels near to surface: $|d_\mathcal{T}| < 0.5\tau$, where τ is the truncate value.

Reflectance. We formulate appearance as SVBRDF, where the reflectance function f_r represents the isotropic Ward model [War92] at vertex point $\mathbf{p}$ in the voxel grid. The diffuse term f_d represents individual diffuse albedo $\boldsymbol{\rho}_d$ per point, and the specular term f_s shares specular albedo (a.k.a. specular coefficient) ρ_s and roughness parameter α per cluster $\mathcal{M}$ as follows:

$$\begin{aligned} f_r(\mathbf{i}, \mathbf{o}; \boldsymbol{\rho}_d, \rho_s, \alpha, \mathbf{n}, \mathbf{p}) &= f_d(\boldsymbol{\rho}_d, \mathbf{p}) + f_s(\mathbf{i}, \mathbf{o}; \rho_s, \alpha, \mathbf{n}, \mathbf{p}), \\ &= \frac{\boldsymbol{\rho}_d}{\pi} + \frac{\rho_s}{4\pi\alpha^2\sqrt{\cos\theta_\mathbf{i}\cos\theta_\mathbf{o}}} e^{-\left(\tan^2(\theta_\mathbf{h})/\alpha^2\right)}, \end{aligned} \tag{1}$$

where $\mathbf{i}$ and $\mathbf{o}$ are the incident light vector and the view vector, $\mathbf{h} = (\mathbf{i}+\mathbf{o})/||\mathbf{i}+\mathbf{o}||$ is the half-angle vector, $\theta_\mathbf{i}$, $\theta_\mathbf{o}$ and $\theta_\mathbf{h}$ are an angle between the normal $\mathbf{n}$ and each vector $\mathbf{i}$, $\mathbf{o}$ and $\mathbf{h}$, respectively, at point $\mathbf{p}$.

Rendering. With an objective of per-frame inverse rendering, we capture an HDR environment map as scene illumination over solid angle Ω as input. Suppose we have incident light $L(-\mathbf{i}; \mathbf{p})$ over angle Ω. Using the rendering equation [Kaj86], we calculate reflected light $L(\mathbf{o}; \mathbf{p})$ as

$$\begin{aligned} L(\mathbf{o}; \mathbf{n}, \mathbf{p}) &= \int_\Omega L(-\mathbf{i}; \mathbf{p}) f_r(\mathbf{i}, \mathbf{o}; \boldsymbol{\rho}_d, \rho_s, \alpha, \mathbf{n}, \mathbf{p})(\mathbf{n}\cdot\mathbf{i}) d\mathbf{i} \\ &\approx B(\boldsymbol{\rho}_d, \mathbf{n}, \mathbf{p}) + S(\mathbf{o}; \rho_s, \alpha, \mathbf{n}, \mathbf{p}). \end{aligned} \tag{2}$$

First, for computational efficiency, we approximate diffuse reflection as spherical harmonics of radiosity [WZN*14, RH01] from given normals, assuming fixed environment illumination: $B(\boldsymbol{\rho}_d, \mathbf{n}, \mathbf{p}) = \boldsymbol{\rho}_d \sum_{k=0}^{8} l_k H_k(\mathbf{n})$, where l_k are the nine spherical harmonics coefficients of incident environment illumination (up to the second order) over Ω, and the spherical harmonics basis functions $H_k(\mathbf{n})$ take normals $\mathbf{n}$ as input to calculate diffuse shading in the global space. Second, we calculate specular reflection $S(\mathbf{o}; \rho_s, \alpha, \mathbf{n}, \mathbf{p}) = \int_\Omega L(-\mathbf{i}, \mathbf{p}) f_s(\mathbf{i}, \mathbf{o}; \rho_s, \alpha, \mathbf{n}, \mathbf{p})(\mathbf{n}\cdot\mathbf{i}) d\mathbf{i}$ by integrating the spherical illumination map using uniform sampling of the upper hemisphere in the normal space.

5.1.2. *Capturing Shape*

Simultaneously estimating SVBRDF, geometry and motion is a chicken-and-egg problem because they are strongly correlated. Once the first-frame observation of the RGBD camera is stored in the canonical space, we begin with estimating the per-frame motion field by formulating the following optimization problems. Before explaining SVBRDF estimation in motion, we briefly explain how to estimate the motion field to accumulate dynamic photometric samples in our hierarchical data structure. This motion part is inherited from the traditional fusion-based framework [NFS15]. Refer to Table 1 for symbols and notations used in this paper.

Global Registration. To improve robustness, we first estimate global transformation that registers the input frame of a depth camera to the voxel grid in each frame, which is formulated as a 6-DOF rigid body transformation (RBT) matrix $\mathbf{T}^t_{\mathcal{D}\to\mathcal{K}} \in \mathbf{SE}(3)$ such that point $\mathbf{p}^t_\mathcal{D}$ in the depth camera space $\mathcal{D}$ at frame t is transferred into the canonical space of voxel grid $\mathcal{K}$ via $\mathbf{p}^t_\mathcal{K} = \mathbf{T}^t_{\mathcal{D}\to\mathcal{K}}\mathbf{p}^t_\mathcal{D}$. The matrix can be optimized by solving the iterative closest point (ICP) method [RL01].

Capturing Shape via Motion. Following the previous work of DynamicFusion [NFS15], we first estimate the local non-rigid motion fields per frame and update the shape of the deformable objects. We define a motion field $\mathcal{W}$ from the canonical space $\mathcal{K}$ to the current warped frame t as $\mathcal{W}^t=\{[\mathbf{q}_i, \sigma_i, \mathbf{T}_i]\}^t$, where $\mathbf{q}_i$ is a position of ith node from the total N number of nodes ($i \in \{1, \ldots, n\}$) in the deformation graph $\mathcal{G}$, $\sigma_i \in \mathbb{R}^+$ is a radius parameter for the distance weight w_i between node $\mathbf{q}_i$ and point $\mathbf{p}_\mathcal{K}$ in the canonical space: $w_i(\mathbf{p}_\mathcal{K}, \sigma_i) = \exp(-||\mathbf{p}_\mathcal{K} - \mathbf{q}_i||^2/(2\sigma_i^2))$ and $\mathbf{T}_i \in \mathbf{SE}(3)$ is a 6-DOF RBT of the ith node. The motion field $\mathcal{W}^t$ at a point $\mathbf{p}_\mathcal{K}$ is defined by dual-quaternion blending [KCvO07] using the k-nearest neighbour nodes with its convex weights. The motion field $\mathcal{W}^t$ warps a point $\mathbf{p}_\mathcal{K}$ and a normal $\mathbf{n}(\mathbf{p}_\mathcal{K})$ in the canonical space by $[\tilde{\mathbf{p}}^\intercal_\mathcal{K}, 1]^\intercal = \mathcal{W}^t(\mathbf{p}_\mathcal{K})[\mathbf{p}^\intercal_\mathcal{K}, 1]^\intercal$ and $[\tilde{\mathbf{n}}(\mathbf{p}_\mathcal{K})^\intercal, 0]^\intercal = \mathcal{W}^t(\mathbf{p}_\mathcal{K})[\mathbf{n}(\mathbf{p}_\mathcal{K})^\intercal, 0]^\intercal$. Given depth image $\mathbf{D}^t$ and the estimated warp motion field, we obtain a weighted average of the projective TSDF values for every voxel $\mathbf{x}$ to reconstruct the shape. Finally, we conduct the marching cube algorithm on the TSDF volume to create a polygonal mesh model per frame and update deformation graph. For more detail, refer to the previous work [NFS15] and the supplemental material.

5.2. Estimating Motion with SVBRDF

State-of-the-art fusion methods [ZNI*14, DNZ*17, NFS15, GXY*17] evaluate only diffuse colour and geometry differences to estimate motion field. In contrast, we can estimate the current motion field $\mathcal{W}^t$ by minimizing the following energy function making use of given geometry and SVBRDF:

$$E_{\text{motion}}\left(\mathcal{W}^t\right) = E_{\text{depth}} + \lambda_{\text{dreg}} E_{\text{dreg}} + \lambda_{\text{pcolour}} E_{\text{pcolour}}, \tag{3}$$

where E_{depth} and E_{dreg} are the data term and its regularizer for geometry, E_{pcolour} is our novel data term for SVBRDF. λ_{dreg} and λ_{pcolour} are the corresponding weights.

Geometric Energy. Our geometric energy terms E_{depth} and E_{dreg} are similar to those terms used in [NFS15]. E_{depth} optimizes the motion parameter by minimizing the plane-normal distance between the warped mesh from the previous frame and its correspondence point in the current depth image. To enforce the local smoothness of motion and prevent overfitting, E_{dreg} minimizes the distance when the node is warped by its own motion parameter and when it is warped by the motion of the neighbouring nodes. Refer to [NFS15] or the supplemental document for more details.

Colour Energy. Assuming that SVBRDF of the captured object does not change over time, our novel motion estimation term E_{pcolour} considers object appearance to enforce the photometric consistency of object surfaces at the ith node in the camera space $\mathcal{C}$ as follows:

$$E_{\text{pcolour}}(\mathcal{W}^t) = \sum_{u \in \mathcal{P}^t_{\mathcal{C}}} \left\| \mathbf{C}^t(\tilde{u}_{\mathcal{C}}) - L^t\left(\tilde{\mathbf{O}}^t_{\mathcal{C}}(u); \tilde{\mathbf{N}}^t_{\mathcal{C}}(u), \tilde{\mathbf{V}}^t_{\mathcal{C}}(u)\right) \right\|_2^2, \quad (4)$$

where $\mathcal{P}^t_{\mathcal{C}}$ is a set of visible pixels u obtained by rendering the warped static model to the current colour camera space $\mathcal{C}^t$, $\tilde{\mathbf{V}}^t_{\mathcal{C}}{:}\mathbb{N}^2 \to \mathbb{R}^3$ is the vertex map of the warped mesh $\tilde{\mathcal{V}}^t_{\mathcal{K}}$ transformed by $\mathbf{T}^t_{\mathcal{K}\to\mathcal{C}}$ from the canonical space to the current colour camera space, $\tilde{\mathbf{O}}^t_{\mathcal{C}}$ is the view direction of $\tilde{\mathbf{V}}^t_{\mathcal{C}}$ to the colour camera, $\tilde{\mathbf{N}}^t_{\mathcal{C}}{:}\mathbb{N}^2 \to \mathbb{R}^3$ is the normal map of $\tilde{\mathcal{V}}^t_{\mathcal{K}}$ transformed by $\mathbf{T}^t_{\mathcal{K}\to\mathcal{C}}$, $\tilde{u}_{\mathcal{C}} = P(\mathbf{K}_{\mathcal{C}}\tilde{\mathbf{V}}^t_{\mathcal{C}}(u))$ is the pixel in the colour image $\mathbf{C}^t$ that corresponds to u, $\mathbf{K}_{\mathcal{C}}$ is the intrinsic matrix of the colour camera, and the reflected light $L^t = B^t + S^t$ is rendered by Equation (2). As unestimated specular components degrade the estimate quality of the estimating motion, this term helps to correctly estimate the photometric difference between a colour image and our reconstructed objects. Refer to Figure 9 to see how geometric accuracy has been improved by accounting for SVBRDF in estimating motion.

Motion Optimization. In order to solve Equation (3), we reformulate it as the sum of squared residuals f so that we can define a new vector field $\mathbf{F}$ to find out the vector of motion parameters $\mathcal{X}$, satisfying: $E(\mathcal{X}) = \sum f(\mathcal{X})^2 = ||\mathbf{F}(\mathcal{X})||^2$. Then, the optimization formulation can be solved by the Gauss–Newton method. The reformulated optimization needs the linearization of three terms of motion, diffuse reflectance and specular reflectance.

For the first two approximation steps of motion and diffuse colour, we follow an existing method of using twist representation [MSZ94] that represents each node's motion parameters $\mathcal{X}$ (3D for rotation and 3D for translation), and converting it to $\mathbf{SE}(3)$ using an exponential map. We also linearize the diffuse colour image using the first-order Taylor approximation [WVT12, NFS15, GXY*17].

However, linearizing our novel SVBRDF term is not trivial. Different from view-invariant diffuse reflection B^t at frame t, specular reflection S^t at vertex $\tilde{\mathbf{V}}^t_{\mathcal{C}}$ depends on the outgoing angle variable $\tilde{\mathbf{O}}^t_{\mathcal{C}}$ with appearance parameters (ρ_s, α, $\mathbf{n}$) and also is formulated by the integration of the incident light (Equations (1) and (2)). Therefore, the computational cost for the direct minimization of Equation (4) with the SVBRDF term is highly expensive. Instead, we first render specular reflection S^t with given environment illumination in the current colour camera space C^t and then substitute S^t from captured colour image $\mathbf{C}^t$ for comparison with pure radiosity B^t, based on Equation (2). This solution increases colour optimization very efficiently and enabling us to consider SVBRDF when estimating motion fields.

(a) Frame #720 (b) Frame #740 (c) Motion fields

Figure 4: *(a) and (b) Input photographs of 720th and 740th frames. (c) Our estimated motion fields showing the deformation of the cloth at the 740th frame.*

Finally, in each Gauss–Newton iteration, we find parameters of $\Delta\mathcal{X}$ by solving a linear least-squares problem [DNZ*17]:

$$\Delta\hat{\mathcal{X}} = \arg\min_{\Delta\mathcal{X}} \left\| \mathbf{F}(\mathcal{X}^{(j-1)}) + \mathbf{J}_{\mathbf{F}}(\mathcal{X}^{(j-1)}) \cdot \Delta\mathcal{X} \right\|. \quad (5)$$

To obtain $\Delta\hat{\mathcal{X}}$, we set the partial derivatives of the above equation with respect to $\Delta\mathcal{X}$ as zero to solve the following equation: $\mathbf{J}^{\mathsf{T}}_{\mathbf{F}}(\mathcal{X}^{(j-1)})\mathbf{J}_{\mathbf{F}}(\mathcal{X}^{(j-1)}) \cdot \Delta\hat{\mathcal{X}} = -\mathbf{J}^{\mathsf{T}}_{\mathbf{F}}(\mathcal{X}^{(j-1)})\mathbf{F}(\mathcal{X}^{(j-1)})$. We solve this problem with pre-conditioned conjugate gradient method (Section 6). Finally, we update motion field as follows: $\mathbf{T}^t_i = e^{\Delta\hat{\mathcal{X}}} \cdot \mathbf{T}^{t-1}_i$. Figure 4 shows an example of the estimated motion field using our SVBRDF-aware motion optimization.

5.3. Capturing SVBRDF in Motion

The state-of-the-art methods for estimating material appearance have focused on SVBRDF of static objects [PNS18, NLGK18] or only diffuse albedo of dynamic objects [GXY*17]. As we estimate per-vertex motion and shape, we then estimate complete SVBRDF parameters per vertex in a progressive way through our novel optimization method.

5.3.1. *Specular Parameters*

There are two main technical challenges for estimating specular parameters: First, specular reflectance depends on both light and view directions, whereas diffuse reflectance is a constant. In particular, specular parameter estimation requires a set of multiple photometric samples with known light and view directions before optimization. Second, per-frame progressive optimization of specular parameters is therefore supposed to suffer from a lack of samples more than the traditional offline methods. The appearance parameters of the same materials need to be shared with spatial and temporal coherence for efficient sampling. We handle these challenges as follows.

Point-Light Illumination for Specular Reflection. As mentioned earlier, in an RGBD camera, there is a TOF camera module that consists of an infrared light and an infrared camera to measure depth

(Figure 3). We utilize the pair of the infrared illumination and the infrared camera module to capture photometric samples to estimate specular parameters.

First, we have geometrically calibrated these two devices beforehand to obtain the light and view vectors ($\mathbf{i}_\mathcal{I}$, $\mathbf{o}_\mathcal{I}$). The relative position and orientation of both $\mathbf{i}_\mathcal{I}$ and $\mathbf{o}_\mathcal{I}$ with respect to the surface geometry are obtained using the estimated motion field. Given the known light and view vectors in the normal space, we can remove the integral over hemisphere Ω in Equation (2) using the point light assumption:

$$S(\mathbf{i}_\mathcal{I}, \mathbf{o}_\mathcal{I}; \rho_s, \alpha, \mathbf{n}, \mathbf{p}) = L_{i_\mathcal{I}}(\mathbf{p}) f_s(\mathbf{i}_\mathcal{I}, \mathbf{o}_\mathcal{I}; \rho_s, \alpha, \mathbf{n}, \mathbf{p})(\mathbf{n} \cdot \mathbf{i}_\mathcal{I}). \quad (6)$$

It allows us to solve the inverse problem efficiently per frame, assuming that the surface roughness of microfacets is consistent in both visible and infrared illumination, following [WZ15, PNS18].

Hierarchical Data Structure. Different from capturing diffuse albedo, estimating specular parameters requires dense observation samples, and thus existing SVBRDF methods [TAL*07, LWS*13, WZ15, PNS18] have used a hierarchical data structure to accumulate sparse samples of specular appearance per each cluster to be used for inferring the specular parameters. In addition, existing dynamic fusion methods [NFS15, GXY*17] make use of a hierarchical data structure to regularize motion vectors of moving objects. In this work, to estimate the SVBRDF of objects in motion, we combine these two data structures into a novel hierarchical data structure that allows us to estimate motion vectors and appearance parameters together. The structure consists of three main components: surface clusters, deformation graphs and a TSDF volume, where diffuse albedo is estimated per voxel, and specular parameters are estimated per cluster (a set of deformation graph nodes that are associated with motion fields) by assuming that surfaces are dichromatic, and that roughness is locally smooth [WZ15, PNS18].

Once these attributes are optimized per frame, they are interpolated to each vertex in the static model. Our hierarchical structure is beneficial in two aspects: First, we can efficiently estimate both appearance and geometry *in motion* per frame, which requires *expensive* optimization, by working on the small number of clusters compared to the number of voxel grids. Second, we can achieve observations with various angles of θ_h to optimize SVBRDF parameters per frame by working on a large range of surfaces with potentially different angles of θ_h. Figure 5 visualizes our hierarchical data structure that accumulate photometric samples.

Fine-to-Coarse Sample Accumulation. We accumulate these photometric samples in the hierarchical data structure of the *half-angle buffer* based on spatiotemporally coherent clustering using the motion fields. We first store the reflection observations of the infrared point light in the fine-grained TSDF voxel grid. To this end, we first warp the positions of the canonical voxels $\mathbf{x}_\mathcal{K}$ into the current depth camera frame via $\tilde{\mathbf{x}}^t_\mathcal{D} = \mathbf{T}^t_{\mathcal{K}\to\mathcal{D}} \mathcal{W}^t(\mathbf{x}_\mathcal{K}) \mathbf{x}_\mathcal{K}$. We then calculate the perspective projection of $\tilde{\mathbf{x}}^t_\mathcal{D}$ to check visibility and correspondence of $\tilde{u}_{\mathbf{x}_\mathcal{D}} = P(\mathbf{K}_\mathcal{D} \tilde{\mathbf{x}}^t_\mathcal{D})$ with respect to camera pixels $\mathbf{I}^t$. Once we find out the corresponding camera intensity $\mathbf{I}^t(\tilde{u}_{\mathbf{x}_\mathcal{D}})$, we calculate the specular intensity v with respect to the half-angle vector angle θ_h (a.k.a. the discrete normal distribution function (NDF)) by normalizing the gamma-corrected intensity with shading $1/(\mathbf{n} \cdot \mathbf{i}_\mathcal{I})$ and distance d^2 at point $\tilde{\mathbf{x}}^t_\mathcal{D}$ as follows:

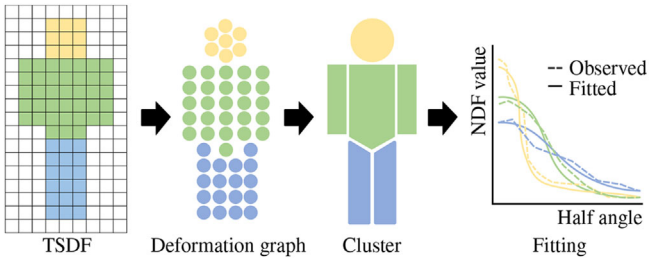


Figure 5: *We accumulate shape and SVBRDF parameters in a hierarchical data structure. First, we store every observation from the RGBD camera into the high-resolution TSDF structure. We then transfer the observation into the deformation graph structure for efficient appearance estimation. Nodes are associated with motion fields to yield the spatiotemporal coherence of appearance estimates. Finally, the deformation nodes are clustered, providing enough samples for fitting BRDF parameters for each cluster.*

$$v = \frac{d^2(\tilde{\mathbf{x}}^t_\mathcal{D})}{\kappa} \cdot \frac{\left(\mathbf{I}^t(\tilde{u}_{\mathbf{x}_\mathcal{D}})\right)^{\gamma_\mathcal{I}}}{\tilde{\mathbf{n}}^t_\mathcal{D} \cdot \tilde{\mathbf{i}}^t_{\mathcal{I},\mathcal{D}}}, \quad (7)$$

where $\tilde{\mathbf{n}}^t_\mathcal{D}$ is a normal at $\tilde{\mathbf{x}}^t_\mathcal{D}$, $\tilde{\mathbf{i}}^t_{\mathcal{I},\mathcal{D}}$ is incident IR illumination vector at $\tilde{\mathbf{x}}^t_\mathcal{D}$, $\gamma_\mathcal{I}$ is the infrared camera gamma and κ is a normalization constant. Both $\gamma_\mathcal{I}$ and κ are calibrated, following [PNS18]. We assume that the infrared emitter and receiver are close enough that both $\mathbf{i}$ and $\mathbf{o}$ are the same as $\mathbf{h}$ to simplify Equation (7) similar to [WZ15]. Per-voxel specular reflectance, $v^t_\mathbf{x}$, of point $\mathbf{x}$ at current frame t is updated in the half-angle buffer through weighted average in the static model:

$$v^t_\mathbf{x}(\theta_\mathbf{h}) = \frac{v \cdot \omega + v^{t-1}_\mathbf{x}(\theta_\mathbf{h}) \cdot \omega^{t-1}_\mathbf{x}(\theta_\mathbf{h})}{\omega + \omega^{t-1}_\mathbf{x}(\theta_\mathbf{h})}, \quad (8)$$

where $\omega = \Phi_{\text{bell}}(u) \cdot \tilde{\mathbf{n}}^t_\mathcal{D} \cdot \tilde{\mathbf{o}}^t_\mathcal{D}$, $\tilde{\mathbf{o}}^t_\mathcal{D}$ is camera view direction at $\tilde{\mathbf{x}}^t_\mathcal{D}$, Φ_{bell} is the bell-shaped filter kernel to suppress extreme noise at the edge of the image. We also update the corresponding weight as follows: $\omega^t_\mathbf{x}(\theta_\mathbf{h}) = \omega + \omega^{t-1}_\mathbf{x}(\theta_\mathbf{h})$. As we estimate specular parameters in the hierarchical data structure, we lift the discrete NDF values stored in the high-resolution TSDF structure to the deformation graph's nodes. Specifically, we assign the target deformation node to a TSDF voxel based on the diffuse albedo values of the node and the voxel. We then cluster deformation graph nodes $\mathbf{q}_i$ with normalized diffuse albedo using the k-mean clustering algorithm (k varies up to eight).

Specular Parameters Optimization. For each cluster $m \in \mathcal{M}^t$, we estimate infrared diffuse albedo $\rho_{d_\mathcal{I}}$ by finding out the minimum value of $v^t_m(\hat{\theta}_\mathbf{h})$ such that $\frac{\Phi_{\text{box}}(v^t_m(\theta_\mathbf{h}+1))}{\Phi_{\text{box}}(v^t_m(\theta_\mathbf{h}))} \geq 1+\epsilon$, where Φ_{box} is the box filter kernel, and ϵ is a user-defined value (0 – 0.01). We then estimate $\hat{\alpha}(m)$ and initial $\hat{\rho}_s(m)$ of each cluster m by minimizing the objective function:

$$\underset{\alpha,\ \rho_s}{\text{minimize}} \sum_{\theta_\mathbf{h}=0}^{\pi/3} \left(\omega' \left| v^t_m(\theta_\mathbf{h}) - \rho_{d_\mathcal{I}} - f_s(\theta_\mathbf{h}, \alpha, \rho_s) \right|^2 \right), \quad (9)$$

where $\omega'=\cos^2(\theta_{\mathbf{h}})\max(\omega_{\max}, \omega_z^t(\theta_{\mathbf{h}}))$ when $\omega_{\max}$ is a pre-defined clamp parameter and v_z^t is the sums of $v_{\mathbf{x}}^t$ that belong to the cluster z. Here, ω indicates the level of confidence of observation. We set $\omega_{\max}$ to a certain level empirically to exclude the diffuse-like observation from the regression of the specular parameter. We solve the optimization problem through the brute-force search with a tabulated function f_s (70 levels: 0 – 70°) in Equation (1), pre-computed with discrete α (66 levels: 0.05 – 0.7) and ρ_s (100 levels: 0.01 – 1.00). Note that the deformation graph's nodes are associated with the motion field, allowing for spatiotemporal coherence. Figure 6 shows the estimated clusters, accumulated observations for each cluster and our fitting results.

Specular Albedo Adjustment. We utilize the active infrared illumination and the infrared camera to estimate the specular parameters: specular albedo $\hat{\rho}_s$ and roughness $\hat{\alpha}$. However, the albedo of the infrared wavelength is independent of that of the visible wavelength. To estimate specular reflection captured by the RGB colour camera, we estimate the albedo scalar $\lambda_{\mathcal{I}\to\mathcal{C}}$ that adjusts the infrared albedo to the visible specular albedo, that is, $\lambda_{\mathcal{I}\to\mathcal{C}}$ is then multiplied to the infrared specular albedo ρ_s, yielding visible albedo $\hat{\rho}_s$. Note that the infrared roughness parameter α is independent of albedo so that the same value is copied to the visible roughness $\hat{\alpha}$. The albedo scalar $\lambda_{\mathcal{I}\to\mathcal{C}}(m)$ of each cluster m is estimated as

$$\lambda_{\mathcal{I}\to\mathcal{C}}(m) = \frac{\sum_{\mathbf{x}\in\mathcal{T}_{\mathcal{C}}^t\cap\mathcal{M}_m} \left(max\left(Y(\mathbf{C}^t(\tilde{u}_{\mathbf{x}_{\mathcal{C}}})) - Y(B_{\mathcal{C}}^t(\tilde{\mathbf{x}}_{\mathcal{C}}^t)), 0\right)\right)}{\sum_{\mathbf{x}\in\mathcal{T}_{\mathcal{C}}^t\cap\mathcal{M}_m} S_{\mathcal{C}}^t\left(\hat{\rho}_{s,\mathcal{I}}(m), \hat{\alpha}_{\mathcal{I}}(m), \tilde{\mathbf{x}}_{\mathcal{C}}^t\right)}, \quad (10)$$

where $\mathcal{T}_{\mathcal{C}}^t$ is a set of visible surface voxels $\mathbf{x}$ warped to the current $\mathcal{C}^t$, $\mathcal{M}_m$ is a set of voxels which cluster to m, $\tilde{\mathbf{x}}_{\mathcal{C}}^t=\mathbf{T}_{\mathcal{D}\to\mathcal{C}}\tilde{\mathbf{x}}_{\mathcal{D}}^t$ is a voxel transformed from the current depth camera space $\mathcal{D}^t$ to the colour camera space $\mathcal{C}^t$, $B_{\mathcal{C}}^t(\tilde{\mathbf{x}}_{\mathcal{C}}^t) = B_{\mathcal{C}}^t(\rho_d(\tilde{\mathbf{x}}_{\mathcal{C}}^t), \mathbf{n}, \tilde{\mathbf{x}}_{\mathcal{C}}^t)$ is diffuse shading rendering using the diffuse albedo of voxel at $\mathcal{C}^t$, $Y(\cdot)$ is a luminance function that converts a colour to the luminance intensity, $Y(\mathbf{C}^t) - Y(B^t)$ is the difference between the captured colour and rough diffuse albedo of voxels subject to $Y(\mathbf{C}^t) > Y(B^t)$, yielding initial specular shading in the colour camera and $S_{\mathcal{C}}^t(\tilde{\mathbf{x}}_{\mathcal{C}}^t) = S_{\mathcal{C}}^t(\mathbf{o}; \hat{\rho}_{s,\mathcal{I}}, \hat{\alpha}_{\mathcal{I}}, \mathbf{n}, \tilde{\mathbf{x}}_{\mathcal{C}}^t)$ is specular shading rendered at $\mathcal{C}^t$ with the IR specular parameters using Equation (2). In order to calculate the diffuse shading image, we use the $(t-1)$ frame estimated diffuse albedo. Our algorithm refines the diffuse albedo and the specular albedo progressively over time.

Coarse-to-Fine Propagation of Parameters. Before we render the specular shading of each voxel $S_{\mathcal{C}}^t(\tilde{\mathbf{x}}_{\mathcal{C}}^t)$, we propagate the visible specular parameters from the deformation graphs to the resolution of TSDF. Each deformation-graph node takes the appearance values from its associated cluster. Each TSDF voxel obtains the parameters from the deformation nodes based on the k-nearest neighbours classified by diffuse albedo. As every voxel $\mathbf{x}$ is associated with four k-nearest neighbour nodes, we propagate per-cluster $\hat{\alpha}(\mathbf{x})$ and $\hat{\rho}_s(\mathbf{x})$ to every voxel $\mathbf{x}$ by the minimum difference of albedos in each voxel and the node within the k-nearest neighbour.

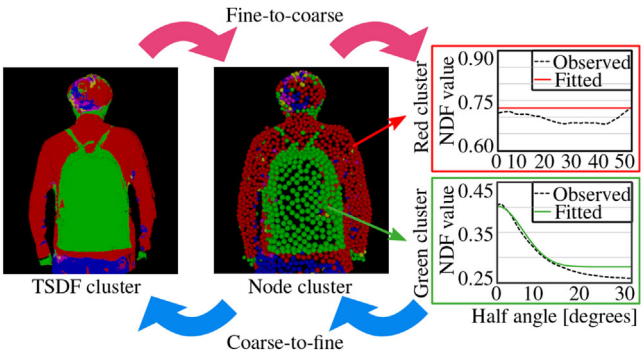


Figure 6: *We accumulate all the photometric samples from the fine to the coarse levels: TSDF, deformation graph and cluster. After we estimate specular appearance per cluster, we propagate the estimated appearance from the coarse to the fine levels.*

5.3.2. *Diffuse Albedo Estimation*

Existing fusion-based methods that estimate appearance account for diffuse reflection, assuming that surfaces have pure diffuse albedo only. The traditional fusion-based methods can integrate averaged photometric observations as diffuse albedos per voxel without separating specular reflection from them [NFS15]. The state-of-the-art method [GXY*17] accounts for shading when calculating diffuse albedos by capturing the environment illumination additionally. However, these methods still cannot account for specular reflection from diffuse albedo computation. In contrast, our method separates specular reflection from the entire reflection, yielding pure diffuse reflection.

5.3.3. *SVBRDF Optimization*

Given the motion field $\mathcal{W}^t$, we estimate the surface properties of SVBRDF $\mathcal{A}^t=\{[\boldsymbol{\rho}_d, \rho_s, \alpha]\}^t$: diffuse albedo, specular albedo and surface roughness per voxel $\mathbf{x}$ in the TSDF volume $\mathcal{T}$ by formulating the following energy function:

$$E_{\text{SVBRDF}}(\mathcal{A}^t) = E_{\text{vcolour}} + \lambda_{\text{treg}}E_{\text{treg}} + \lambda_{\text{sreg}}E_{\text{sreg}}, \quad (11)$$

where E_{vcolour} is the per-voxel colour data term, E_{treg} is the temporal regularizer and E_{sreg} is the spatial regularizer for the diffuse SVBRDF parameters.

The colour data term E_{vcolour} enforces photometric consistency of the SVBRDF parameters (on each voxel warped to the camera $\mathbf{x}_{\mathcal{C}}^t$) to make rendering with them satisfy given camera observation $\mathbf{C}^t$:

$$E_{\text{vcolour}} = \sum\nolimits_{\mathbf{x}\in\mathcal{T}_{\mathcal{C}}^t} \Phi\left(\left\|\tilde{\mathbf{n}}_{\mathcal{C}}^t - \tilde{\mathbf{o}}_{\mathcal{C}}^t\right\|\right)\left\|\mathbf{C}^t(\tilde{u}_{\mathbf{x}_{\mathcal{C}}}) - L^t(\tilde{\mathbf{x}}_{\mathcal{C}}^t)\right\|_2^2, \quad (12)$$

where $\tilde{u}_{\mathbf{x}_{\mathcal{C}}}=P(\mathbf{K}_{\mathcal{C}}\tilde{\mathbf{x}}_{\mathcal{C}}^t)$ is a corresponding pixel of $\tilde{\mathbf{x}}_{\mathcal{C}}^t$ at the current colour image $\mathbf{C}^t$, $\tilde{\mathbf{n}}_{\mathcal{C}}^t$ and $\tilde{\mathbf{o}}_{\mathcal{C}}^t$ are normals and camera vectors at $\tilde{\mathbf{x}}_{\mathcal{C}}^t$, respectively and Φ is a robust kernel where $\Phi(x) = 1/(1+5x)^3$, following [ZDI*15]. Here, $L^t(\tilde{\mathbf{x}}_{\mathcal{C}}^t) = L^t(\tilde{\mathbf{o}}_{\mathcal{C}}^t; \tilde{\mathbf{n}}_{\mathcal{C}}^t, \tilde{\mathbf{x}}_{\mathcal{C}}^t)$ is the outgoing radiance under visible environment illumination, which is the sum of diffuse radiosity B^t and specular reflection S^t of the voxel in the colour camera space (Equation (2)).

Regularizer E_{treg} in Equation (11) suppresses the temporal overfit of diffuse albedo $\boldsymbol{\rho}_d$ towards specular reflection:

$$E_{\text{treg}} = \sum\nolimits_{\mathbf{x}\in\mathcal{T}_{\mathcal{C}}^{t}\cap\mathcal{T}_{\mathcal{C}}^{t-1}} \left\| \boldsymbol{\rho}_d^{t}(\mathbf{x}) - \boldsymbol{\rho}_d^{t-1}(\mathbf{x}) \right\|_2^2, \tag{13}$$

where $\mathcal{T}_{\mathcal{C}}^{t-1}$ is a set of visible surface voxels $\mathbf{x}$ at the previous colour camera frame $\mathcal{C}^{t-1}$.

In addition to the colour data term, we enforce local smoothness of diffuse albedo by formulating E_{sreg}:

$$E_{\text{sreg}} = \sum\nolimits_{\mathbf{x}\in\mathcal{T}_{\mathcal{C}}^{t}} \sum\nolimits_{\mathbf{y}\in N(\mathbf{x})\cap\mathcal{T}_{\mathcal{C}}^{t}} \Phi\left(\left\| \Gamma\left(\tilde{u}_{\mathbf{x}_{\mathcal{C}}}\right) - \Gamma\left(\tilde{u}_{\mathbf{y}_{\mathcal{C}}}\right) \right\|\right) \left\| \boldsymbol{\rho}_d^{t}(\mathbf{x}) - \boldsymbol{\rho}_d^{t}(\mathbf{y}) \right\|_2^2, \tag{14}$$

where $N(\mathbf{x})$ is a set of the neighbouring voxels $\mathbf{x}$, $\tilde{u}_{\mathbf{x}_{\mathcal{C}}}$ and $\tilde{u}_{\mathbf{y}_{\mathcal{C}}}$ are pixels obtained by transforming voxels $\mathbf{x}$ and $\mathbf{y}$ to the current colour camera space $\mathcal{C}^t$, respectively, $\Gamma{=}\mathbf{C}^t/Y(\mathbf{C}^t)$ is the ratio of chromaticity to luminance Y of each pixel.

To implement this optimization progressively, we render visible specular reflection S^t with the specular parameters $\hat{\rho}_s$ and $\hat{\alpha}$ at voxel $\mathbf{x}^t$ that we have estimated in Section 5.3.1, using $\mathbf{i}_{\mathcal{C}}$ and $\mathbf{o}_{\mathcal{C}}$ under visible environment illumination (captured by a 360 camera). We then subtract the estimated specular components from the captured image so that Equation (11) can be optimized only with respect to the pure diffuse albedo. This can be solved with the preconditioned conjugate gradient optimization as it becomes a least-square problem.

6. Implementation Details

Radiometric Calibration. We have conducted radiometric calibration for the RGB camera module and the infrared TOF camera module inside an RGBD device, Kinect 2 (Figure 3) in order to quantify the sensor responses in the red, green, blue and infrared channels. First, we estimate the RGB irradiance of the illumination $(r_{\text{n}}, g_{\text{n}}, b_{\text{n}})$ by capturing the standard reflectance tile, Spectralon (Labsphere SRM99) for white balancing with the gamma value of $\gamma_{\mathcal{C}} = 2.2$. Then, we calibrate the infrared camera parameters by solving the following optimization [PNS18]:

$$\min_{\kappa,\ \gamma} \sum_{u\in\mathcal{P}_s} \left(\mathbf{I}(u) - \left(\kappa \cdot \psi \frac{\mathbf{n}(u)\cdot\mathbf{i}(u)}{\pi \cdot d^2(u)} \right)^{\gamma_{\mathcal{I}}} \right)^2, \tag{15}$$

where $\mathcal{P}_s$ is a set of pixels u in the region where the spectralon is captured, κ is the illumination intensity of the infrared emitter in the Kinect 2 sensor, $\gamma_{\mathcal{I}}$ is the gamma exponent of the infrared camera, $\mathbf{I}(u)$ is the infrared value at the pixel u, $\mathbf{n}(u)$ is the normal of the pixel u, $\mathbf{i}(u)$ is the incident light direction of the pixel u and $d(u)$ is the distance between the IR emitter and the pixel u. We have estimated the values of κ and $\gamma_{\mathcal{I}}$ as 0.46 and 0.92 through nonlinear optimization [BGN00]. Given the radiometric parameters r_{n}, g_{n}, b_{n}, κ, $\gamma_{\mathcal{C}}$ and $\gamma_{\mathcal{I}}$ in the pre-processing of calibration, we linearize each RGB and infrared images and normalize them with irradiance.

Pre-conditioned Conjugate Gradient for GPU. We have implemented a GPU-based data-parallel pre-conditioned conjugate gradient (PCG) solver [WBS*13]. The main computational bottleneck is the part of calculating matrix–vector multiplication. Following [ZNI*14], we have made use of two sparse matrix–vector multiplication kernels.

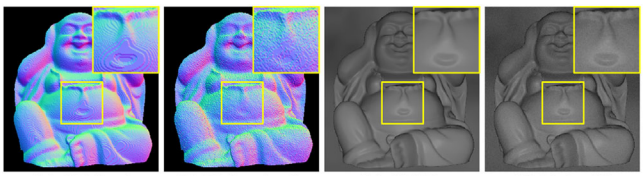
Table 2: *Per-frame processing time of our method. Our method takes 456 ms in total to process each frame inputs.*

Algorithm	Processing time (ms)
Global registration	8
Motion estimation	224
TSDF integration	26
Specular estimation	89
Diffuse estimation	43
Marching cube	63
Etc.	2
Total	456

Environment Map Capture. To estimate incident illumination of scenes, we have captured scene environment maps as high-dynamic-range (HDR) radiance maps using a 360 camera (Ricoh Theta) with multiple exposures. In this paper, we have used monochromatic illumination maps by converting RGB radiance maps to luminance maps for computational efficiency. We then represent the environment maps with spherical harmonics coefficients for efficiently computing shading.

7. Results

We built our capture setup using a Kinect 2 RGBD camera that consists of both an RGB and an infrared camera with an infrared illuminator in the TOF camera module (see Figures 1 and 3). Our method is implemented in C++, where CUDA-based GPU acceleration is extensively used for parallel processing, along with the OpenGL Shading Language for rendering. We set the resolution of the TSDF volume as $512 \times 512 \times 512$, and each TSDF voxel is defined as a cube with a width of 2 mm. Each node in the deformation graph has a radius of 20 mm. For the ground-truth data, we use 1.5 mm voxel size and 15 mm deformation graph radius. The truncation range for TSDF is five times wider than the voxel size. We pre-compute a discrete table of the BRDF function for pre-defined samples of parameters: The half-angle is sampled from 0 to 60° with a step size of 1°. Then, the Ward BRDF model is pre-computed with the values of α and ρ_s from 0.05 to 0.70 and 0.01 to 1 both with 0.01 intervals, respectively. We tested our algorithm on a desktop computer with an Intel Core i7-7700K 4.20 GHz and a graphics card of



(a) GT depth (b) Noisy depth (c) GT IR (d) Noisy IR

Figure 7: *Synthetic input example. (a) Ground-truth depth image with normals. (b) Synthetic depth image with Gaussian noise. (c) GT IR image. (d) Synthetic IR image with Gaussian noise.*

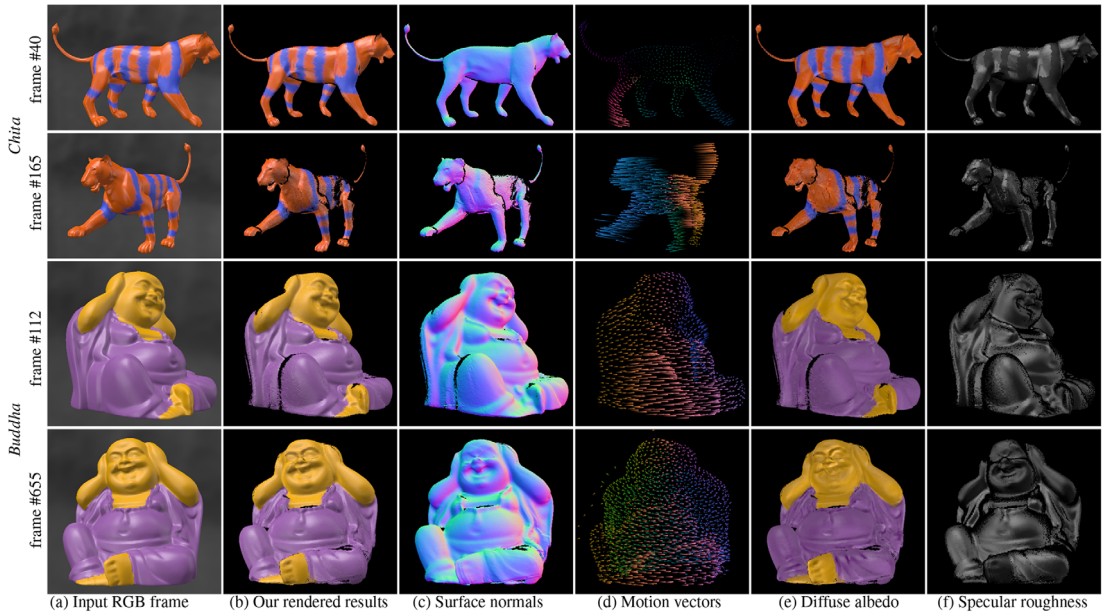


Figure 8: *We have evaluated the reconstruction accuracy of geometry and appearance parameters, compared with the ground truth dataset synthetically created with Gaussian noise. (a) Input RGB images, (b) our results of reconstructed 3D models at each frame, followed by (c) our results of surface normals, (d) motion vectors, (e) diffuse albedo and (f) specular roughness.*

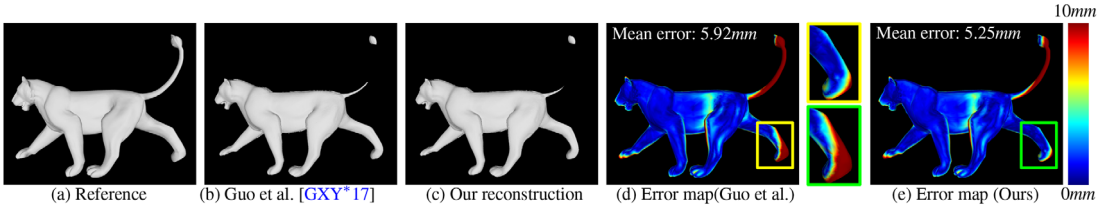


Figure 9: *We compared the geometric accuracy of our method with a state-of-the-art method, Guo* et al. *[GXY*17], implemented by ourselves. We used the synthetic dataset that we created with Gaussian noise. Our method accurately reconstructs motion and geometry resulting in a low geometric error of average 5.25 mm. (a) Reference ground truth geometry. (b) Result by Guo* et al. *(c) Our reconstructed geometry. (d) Error map of Guo* et al. *compared with the GT geometry. (e) Error map of our results compared with the GT. Close-up boxes compares our method with that of Guo* et al.

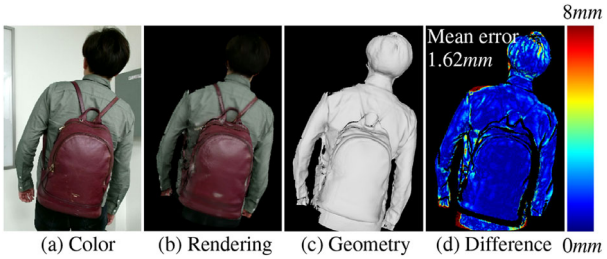


Figure 10: *We evaluate the accuracy of the reconstructed geometry, comparing the warped geometry with the depth map of the current frame. (a) Input RGB image, (b) rendered result, (c) warped geometry, and (d) a difference map between (c) and the current depth map. The mean error of the depth values is just 1.62 mm.*

NVIDIA Titan V (12 GB). Our entire algorithm took 456 ms to process and render each frame. Our method is designed to be online, progressively processing input frames. Table 2 shows the detailed timestamps of our method taking 456 ms to process each frame. We provide our experimental results and comparison as follows.

Quantitative Evaluation. We created a synthetic dataset with known shape, SVBRDF and motion of different objects using OpenGL rendering. To make our synthetic dataset closer to the real sensor input, we also added Gaussian noise to the ground-truth (GT)depth images with $\mathcal{N}(0, 0.002^2)$ and the GT IR images with $\mathcal{N}(0, 1000^2)$, as shown in Figure 7.

Using the synthetic dataset created with Gaussian noise, we have evaluated the accuracy of our reconstruction algorithm compared with the ground truth. Figure 8 presents our reconstruction results compared with the ground-truth SVBRDF, shape and motion. In order to quantitatively evaluate the reconstruction accuracy of shape

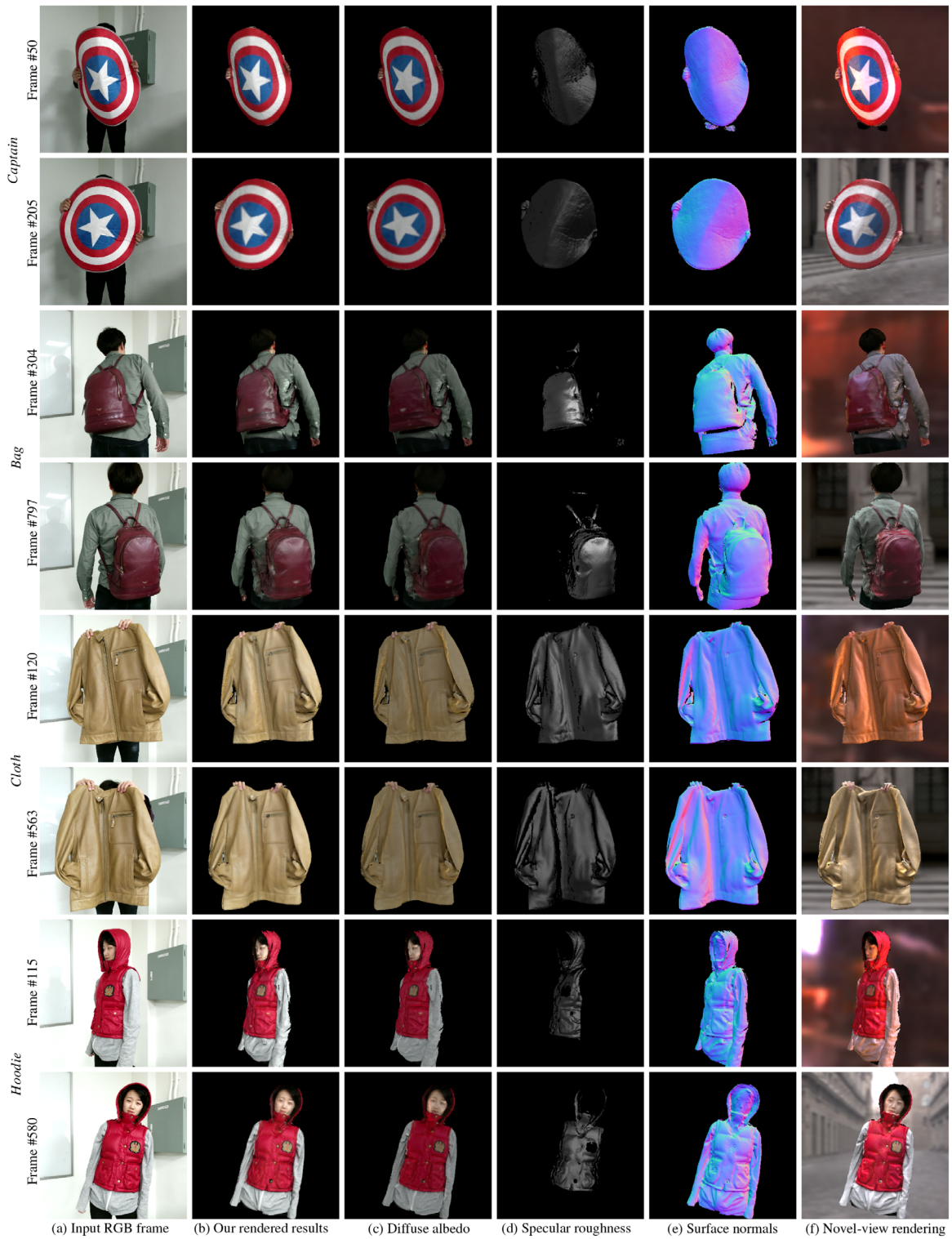


Figure 11: *Our method faithfully reconstructs SVBRDF, geometry and motion of real-world dynamic objects. (a) Input RGB frame. (b) our results rendered with (c) reconstructed diffuse albedo, (d) specular roughness and (e) surface normals. (f) Results rendered under novel environment illumination. Note that we have multiplied the specular intensities by a factor of two for visualization purpose only. Refer to the supplemental video for more results.*

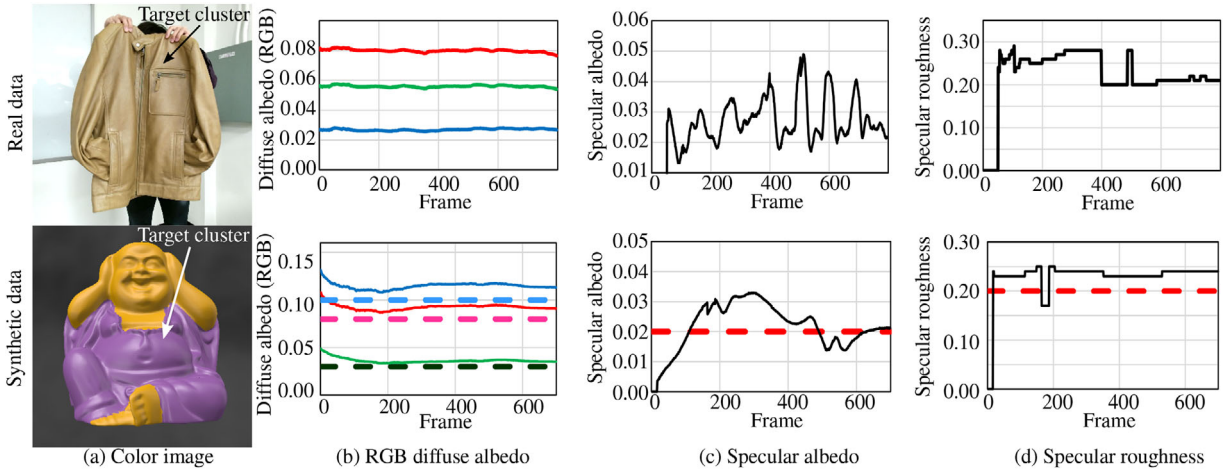


Figure 12: *Our method progressively estimates SVBRDF by accumulating observations. We evaluate the stableness of our SVBRDF reconstruction for the real and synthetic dataset. (a) Real* (top) *and synthetic* (bottom) *colour photographs and target clusters noted by arrows. (b)–(d) Estimated SVBRDF parameter values for the target clusters over frames. We plot our estimates with solid lines and the ground truth values with dotted lines (synthetic only).*

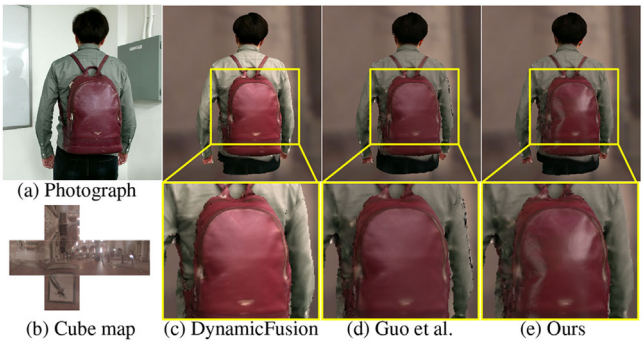


Figure 13: *We compare our method with two online scanning methods. (a) Reference photograph. (b) An environment map. (c) DynamicFusion [NFS15] presents fixed specular reflection while (d) Guo* et al. *[GXY*17] can reconstruct only diffuse shading. (e) Our method can acquire both specular and diffuse appearance. Note that specular reflection changes realistically in our results when the environment illumination spins. Refer to the supplemental video for more results.*

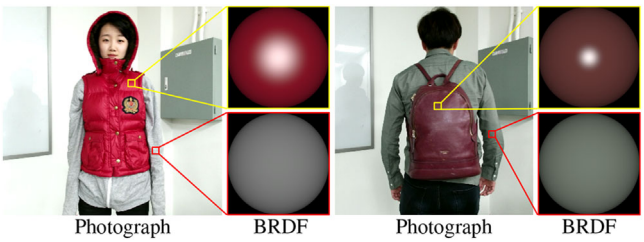


Figure 14: *We present captured BRDFs of dynamic objects, reconstructed by our method. Even though objects are dynamic, we reconstruct diffuse colour and specular lobe successfully.*

and motion together, we warp the estimated geometry to the current camera frame with motion estimates for each frame. The averaged error between the ground-truth shape and the reconstructed shape with motion is very low at just 5.25 mm in the Hausdorff distance, as shown in Figure 9.

Moreover, we evaluate the accuracy of the estimated flow by our method in the real scene. We compare the differences between the actual depth map at #400 frame (of the *Bag* scene) and the warped geometry of our reconstructed model in Figure 10. The average distance error of the entire human body is just 1.62 mm. It is not surprising that there are some large errors around challenging geometry, such as hair and the silhouette of the body. Overall, our method successfully reconstructs SVBRDF, shape and motion for not only synthetic data but also real data. Refer to the supplementary video for every reconstruction results.

Qualitative Evaluation. Figure 11 presents the results of real-world dynamic objects. We present (a) input colour frame, (b) our reconstructed 3D objects rendered with a point light, (c) diffuse albedo, (d) specular roughness, (e) surface normals and (f) novel light-and-view rendering with an environment illumination map to qualitatively evaluate the overfitting problem of inverse rendering. The results of diffuse albedo and specular roughness demonstrate the effectiveness of our decomposition of material properties. Our results rendered under novel environmental lighting and view conditions present no typical blinking artefacts of overfitting. Refer to the supplemental video for every reconstruction result.

Progressive Reconstruction. Our method reconstructs SVBRDF and geometry progressively per frame. Figure 12 shows our estimates of specular albedo, specular roughness and diffuse albedo for each frame for a real scene and a synthetic scene. For the real-world case, the optimization for diffuse albedo converges fast thanks to our robust clustering. In contrast, optimization for specular

parameters requires a long iteration to be stabilized, showing fluctuation at an early stage. Our method provides fast and accurate convergence when optimizing both diffuse and specular appearance parameters: RGB diffuse albedo, specular albedo and specular roughness.

Comparison. Figure 13 compares our method with other two fusion-based methods of capturing objects using a single RGBD camera: DynamicFusion [NFS15] and Guo *et al*. [GXY*17]. As there is no available public source, we implemented both methods. DynamicFusion [NFS15] does not separate diffuse and specular reflection, that is, it stores the sum of diffuse and specular colours as a single colour while geometry and motion can be faithfully recovered. Guo *et al*. [GXY*17] extend the estimation of geometry and motion to capture the diffuse appearance of objects. Their method can cover only diffuse shading rendering, missing specular reflection. Our method is capable of estimating the full SVBRDF appearance of diffuse and specular components, in addition to the motion and geometry of dynamic objects. Figure 14 presents our reconstructed BRDFs of the dynamic objects that present very different characteristics.

8. Discussion

Although our method can handle various scenes, it is not free from limitations.

Infrared Illumination & Camera. In order to estimate appearance parameters, we have made use of the infrared illumination and camera in the TOF module of Kinect 2. As these light and sensor modules are originally designed for measuring depth information, the RGBD camera API does not provide any custom control of exposure level of the IR emitter/sensor. Therefore, we were not able to capture high-dynamic-range (HDR) images as input for estimating specular reflection. When surface normals of specular objects look at the camera directly, not only depth map but also IR images have often been saturated, resulting in suboptimal results. See Figure 15(a).

Frequency of Texture. For estimating specular roughness parameters per pixel, we have used the infrared camera in the TOF module. The resolution of the IR image sensor is 512-by-424, while the resolution of the RGB image sensor is 1920-by-1080. The resolution of the IR sensor is four-time lower than that of the RGB sensor. Accordingly, when a target object has high-frequency patterns, the estimated specular parameter cannot reflect the object's appearance in high frequency. Figure 15(b) shows that the high-frequency structure of diffuse surface lines over the green tree cannot be estimated properly.

Rapid motion. When estimating motion vectors of dynamic objects, our method inherits the traditional linearized approximation of differential motion using the twist representation [MSZ94]. We found that the performance of this approach has become suboptimal when motion occurs very dynamically, or input frames are captured with severe motion blurring. See Figure 15(c) for an example. In addition, when motion causes deformation of objects surface with texture, our method cannot account for the stretch of the texture surfaces. This would be interesting future work to explore.

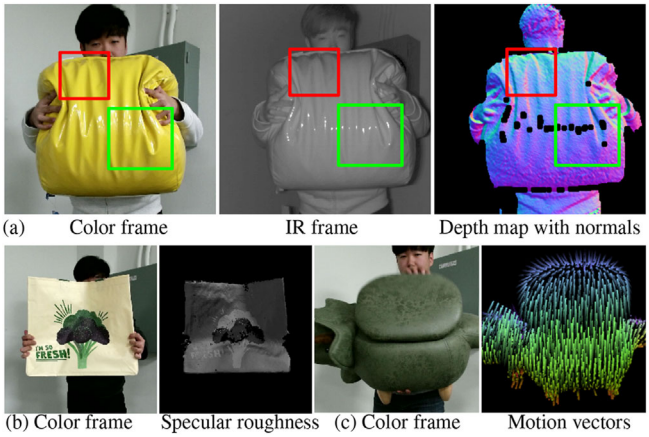


Figure 15: *(a) Owing to the limited dynamic range of the image sensor, input signals from the IR camera are saturated when surface smoothness is high. It often results in no depth values. The performance of our method becomes suboptimal when object surfaces are very smooth. (b) The resolution of the IR camera is four-time lower than that of the RGB camera, and thus our method fails in capturing high-frequency patterns of specular roughness. (c) When motion is large or rapid, our motion estimation often suffer from suboptimal reconstruction results due to motion blur.*

Spatial Resolution. The spatial resolution of SVBRDF and shape is determined by the spatial resolution of the TSDF volume. As we currently store this information for each vertex, the current resolution degrades the spatial resolution of the final results. Applying texture mapping to our framework would be interesting future work.

Frame Rate. The current frame rate is about two frames per second, which is lower than the real-time performance thus far due to challenges of heavy optimization in factorizing SVBRDF, geometry and motion simultaneously. Accelerating computation for real-time applications would be an interesting avenue to explore.

9. Conclusions

We have presented a novel material acquisition method that estimates SVBRDF, geometry and motion simultaneously using a single RGBD camera. We have proposed an inverse rendering framework that can efficiently estimate material appearance using the voxel grid and the deformation graph in the two different scales. We have also provided the appearance-aware motion estimation algorithm so that the specular appearance can be considered to improve the motion estimation accurately. We have experimented with real-world objects. Finally, we have carefully discussed the limitations, evaluations and comparisons with other methods to validate the performance of our method.

Acknowledgements

Min H. Kim acknowledges Samsung Research Funding Center of Samsung Electronics (SRFC-IT2001-04), in addition to a partial support of Korea NRF grants (2019R1A2C3007229, 2013M3A6A6073718), Samsung Research, the CT Research & Development Program of KOCCA in MCST of Korea, MSIT/IITP of Korea, MSRA and Cross-Ministry Giga KOREA (GK17P0200). The authors also thank Joo Ho Lee for helping GPU implementation and helpful comments.

References

[AWL15] Aittala M., Weyrich T., Lehtinen J.: Two-shot SVBRDF capture for stationary materials. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 110:1–13.

[BGN00] Byrd R. H., Gilbert J. C., Nocedal J.: A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming 89*, 1 (2000), 149–185.

[BJTK18] Baek S.-H., Jeon D. S., Tong X., Kim M. H.: Simultaneous acquisition of polarimetric SVBRDF and normals. *ACM Transactions on Graphics 37*, 6 (2018), 268:1–15.

[CL96] Curless B., Levoy M.: A volumetric method for building complex models from range images. In *SIGGRAPH '96: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive* (New York, NY, 1996), ACM Press, pp. 303–312.

[DCP*14] Dong Y., Chen G., Peers P., Zhang J., Tong X.: Appearance-from-motion: Recovering spatially varying surface reflectance under unknown lighting. *ACM Transactions on Graphics (TOG) 33*, 6 (2014), 1–12.

[DDF*17] Dou M., Davidson P., Fanello S. R., Khamis S., Kowdle A., Rhemann C., Tankovich V., Izadi S.: Motion2fusion: Real-time volumetric performance capture. *ACM Transactions on Graphics 36*, 6 (November 2017), 246:1–246:16.

[DKD*16] Dou M., Khamis S., Degtyarev Y., Davidson P., Fanello S. R., Kowdle A., Escolano S. O., Rhemann C., Kim D., Taylor J., Kohli P., Tankovich V., Izadi S.: Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics 35*, 4 (July 2016), 114:1–114:13.

[DNZ*17] Dai A., Niessner M., Zollhofer M., Izadi S., Theobalt C.: Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics 36*, 4 (2017), 76a.

[DRS10] Dorsey J., Rushmeier H., Sillion F.: *Digital Modeling of Material Appearance*. Amsterdam, Netherlands: Elsevier, 2010.

[FHW*11] Fyffe G., Hawkins T., Watts C., Ma W.-C., Debevec P.: Comprehensive facial performance capture. *Computer Graphics Forum 30*, (2011), 425–434.

[GCHS10] Goldman D. B., Curless B., Hertzmann A., Seitz S. M.: Shape and spatially-varying BRDFS from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 6 (2010), 1060–1071.

[GHP*08] Ghosh A., Hawkins T., Peers P., Frederiksen S., Debevec P.: Practical modeling and acquisition of layered facial reflectance. *ACM Transactions on Graphics 27*, 5 (2008), 139.

[GXY*17] Guo K., Xu F., Yu T., Liu X., Dai Q., Liu Y.: Real-time geometry, albedo, and motion reconstruction using a single RGB-D camera. *ACM Transactions on Graphics 36*, 3 (2017), 32.

[HLZ10] Holroyd M., Lawrence J., Zickler T.: A coaxial optical scanner for synchronous acquisition of 3d geometry and surface reflectance. *ACM Transactions on Graphics 29*, 4 (2010), 99.

[HSL*17] Hui Z., Sunkavalli K., Lee J.-Y., Hadap S., Wang J., Sankaranarayanan A. C.: Reflectance capture using univariate sampling of brdfs. In *Proceedings of the IEEE International Conference on Computer Vision* (Piscataway, NJ, 2017), IEEE, pp. 5362–5370.

[IZN*16] Innmann M., Zollhofer M., Niessner M., Theobalt C., Stamminger M.: Volumedeform: Real-time volumetric nonrigid reconstruction. In *European Conference on Computer Vision* (Berlin, Heidelberg, 2016), Springer, pp. 362–379.

[Kaj86] Kajiya J. T.: The rendering equation. In *SIGGRAPH '86: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, 1986), ACM Press, pp. 143–150.

[KCvO07] Kavan L., Collins S., Žára J., O'Sullivan C.: Skinning with dual quaternions. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2007), ACM Press, pp. 39–46.

[LPG19] Lin Y., Peers P., Ghosh A.: On-site example-based material appearance acquisition. *Computer Graphics Forum 38*, (2019) 15–25.

[LWS*13] Li G., Wu C., Stoll C., Liu Y., Varanasi K., Dai Q., Theobalt C. (2013) Capturing relightable human performances under general uncontrolled illumination. *Computer Graphics Forum*, *32* (2pt3), 275–284.

[LZG18] Li C., Zhao Z., Guo X.: Articulatedfusion: Real-time reconstruction of motion, geometry and segmentation using a single depth camera. In *ECCV: European Conference on Computer Vision* (Berlin, Heidelberg, 2018), Springer.

[MSZ94] Murray R. M., Sastry S. S., Zexiang L.: *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, FL, 1994.

[NFS15] Newcombe R. A., Fox D., Seitz S. M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA, 2015), IEEE, pp. 343–352.

[NJR15] Nielsen J. B., Jensen H. W., Ramamoorthi R.: On optimal, minimal brdf sampling for reflectance acquisition. *ACM Transactions on Graphics 34*, 6 (2015), 1–11.

[NLGK18] Nam G., Lee J. H., Gutierrez D., Kim M. H.: Practical SVBRDF acquisition of 3D objects with unstructured flash photography. *ACM Transactions on Graphics 37*, 6 (2018), 267:1–12.

[NLW*16] Nam G., Lee J. H., Wu H., Gutierrez D., Kim M. H.: Simultaneous acquisition of microscale reflectance and normals. *ACM Transactions on Graphics 35*, 6 (2016), 185.

[PNS18] Park J. J., Newcombe R., Seitz S.: Surface light field fusion. In *2018 International Conference on 3D Vision (3DV)* (Verona, Italy, 2018), IEEE, pp. 12–21.

[RH01] Ramamoorthi R., Hanrahan P.: A signal-processing framework for inverse rendering. In *SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, 2001), ACM Press, 117–128.

[RL01] Rusinkiewicz S., Levoy M.: Efficient variants of the ICP algorithm. In *3DIM* (Washington, DC, USA, 2001), IEEE Computer Society, pp. 145–152.

[RPG16] Riviere J., Peers P., Ghosh A.: Mobile surface reflectometry. *Computer Graphics Forum 35*, 1 (February 2016), 191–202.

[RRFG17] Riviere J., Reshetouski I., Filipi L., Ghosh A.: Polarization imaging reflectometry in the wild. *ACM Transactions on Graphics 36*, 6 (2017), 206.

[SBCI17] Slavcheva M., Baust M., Cremers D., Ilic S.: Killingfusion: Non-rigid 3d reconstruction without correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI, 2017), IEEE, p. 7.

[SBI18] Slavcheva M., Baust M., Ilic S.: Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT, 2018), pp. 2646–2655.

[SSWK13] Schwartz C., Sarlette R., Weinmann M., Klein R.: Dome ii: A parallelized BTF acquisition system. In *Proceedings of the Eurographics 2013 Workshop on Material Appearance Modeling: Issues and Acquisition* (Goslar, Germany, 2013), Eurographics Association, pp. 25–31.

[SWK19] Stotko P., Weinmann M., Klein R.: Albedo estimation for real-time 3d reconstruction using RGB-D and IR data. *ISPRS Journal of Photogrammetry and Remote Sensing 150* (2019), 213–225.

[TAL*07] Theobalt C., Ahmed N., Lensch H., Magnor M., Seidel H.-P.: Seeing people in different light-joint shape, motion, and reflectance capture. *IEEE Transactions on Visualization and Computer Graphics 13*, 4 (2007), 663–674.

[TFG*13] Tunwattanapong B., Fyffe G., Graham P., Busch J., Yu X., Ghosh A., Debevec P.: Acquiring reflectance and shape from continuous spherical harmonic illumination. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 109:1–12.

[War92] Ward G. J.: Measuring and modeling anisotropic reflection. *ACM SIGGRAPH Computer Graphics 26*, 2 (1992), 265–272.

[WBS*13] Weber D., Bender J., Schnoes M., Stork A., Fellner D.: Efficient gpu data structures and methods to solve sparse linear systems in dynamics applications. *Computer Graphics Forum 32*, 2 (2013), 16–26.

[WVT12] Wu C., Varanasi K., Theobalt C.: Full body performance capture under uncontrolled and varying illumination: A shading-based approach. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part IV* (Berlin, Heidelberg, 2012), Springer, pp. 757–770.

[WWZ16] Wu H., Wang Z., Zhou K.: Simultaneous localization and appearance estimation with a consumer RGB-D camera. *IEEE Transactions on Visualization and Computer Graphics 22*, 8 (2016), 2012–2023.

[WZ15] Wu H., Zhou K.: AppFusion: Interactive appearance acquisition using a kinect sensor. *Computer Graphics Forum 34*, 6 (2015), 289–298.

[WZN*14] Wu C., Zollhofer M., Niessner M., Stamminger M., Izadi S., Theobalt C.: Real-time shading-based refinement for consumer depth cameras. *ACM Transactions on Graphics 33*, 6 (2014), 200.

[XSH*19] Xu L., Su Z., Han L., Yu T., Liu Y., Lu F. (2019) UnstructuredFusion: Realtime 4D Geometry and Texture Reconstruction using CommercialRGBD Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1. https://doi.org/10.1109/tpami.2019.2915229.

[YGX*17] Yu T., Guo K., Xu F., Dong Y., Su Z., Zhao J., Li J., Dai Q., Liu Y.: Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera. In *The IEEE International Conference on Computer Vision (ICCV)* (Venice, Italy, October 2017).

[YZG*18] Yu T., Zheng Z., Guo K., Zhao J., Dai Q., Li H., Pons-Moll G., Liu Y.: Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* (Salt Lake City, UT, June 2018).

[YZZ*19] Yu T., Zheng Z., Zhong Y., Zhao J., Dai Q., Pons-Moll G., Liu Y.: SimulCap : Single-View Human Performance Capture With Cloth Simulation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 5499–5509, (2019).

[ZDI*15] Zollhofer M., Dai A., Innmann M., Wu C., Stamminger M., Theobalt C., Niessner M.: Shading-based

refinement on volumetric signed distance functions. *ACM Transactions on Graphics 34*, 4 (2015), 96.

[ZNI*14] Zollhofer M., Niessner M., Izadi S., Rehmann C., Zach C., Fisher M., Wu C., Fitzgibbon A., Loop C., Theobalt C., et al.: Real-time non-rigid reconstruction using an RGB-D camera. *ACM Transactions on Graphics 33*, 4 (2014), 156.

[ZYL*18] Zheng Z., Yu T., Li H., Guo K., Dai Q., Fang L., Liu Y.: Hybridfusion: Real-time performance capture using a single depth sensor and sparse imus. In *European Conference on Computer Vision (ECCV)* (Munich, Germany, September 2018).

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Data S1

Data Video S2