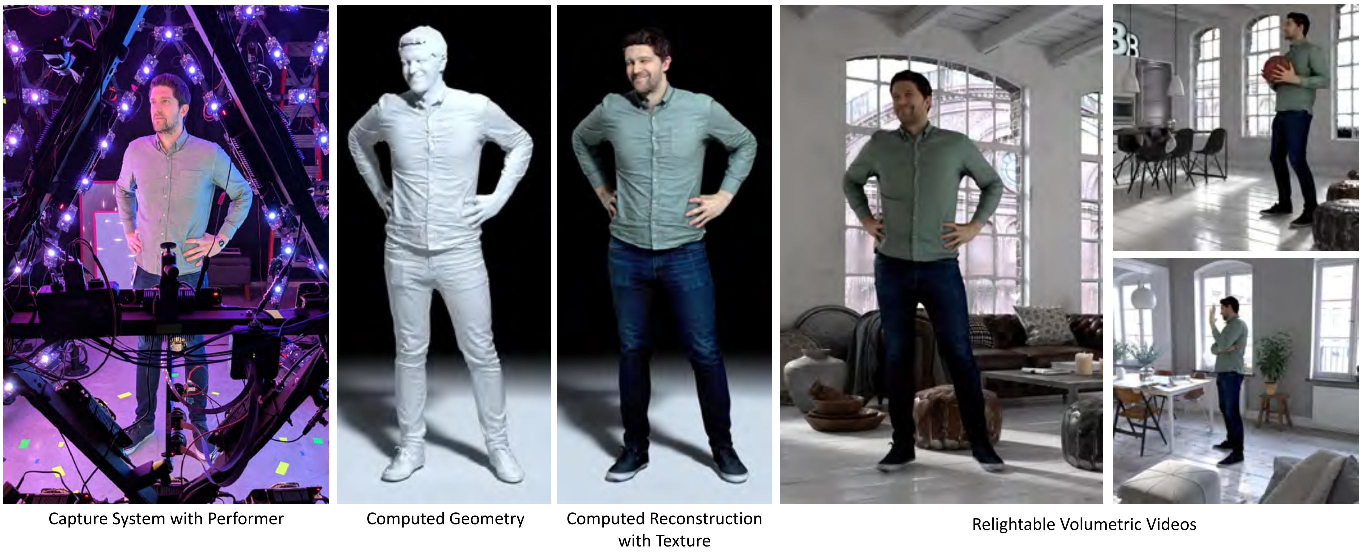
The Relightables: Volumetric Performance Capture of Humans with Realistic Relighting

KAIWEN GUO*, PETER LINCOLN*, PHILIP DAVIDSON*, JAY BUSCH†, XUEMING YU†, MATT WHALEN†, GEOFF HARVEY†, SERGIO ORTS-ESCOLANO, ROHIT PANDEY, JASON DOURGARIAN, DANHANG TANG, ANASTASIA TKACH, ADARSH KOWDLE, EMILY COOPER, MINGSONG DOU, SEAN FANELLO‡, GRAHAM FYFFE‡, CHRISTOPH RHEMANN‡, JONATHAN TAYLOR‡, PAUL DEBEVEC§, and SHAHRAM IZADI§, Google Inc.



Capture System with Performer

Computed Geometry

Computed Reconstruction with Texture

Relightable Volumetric Videos

Fig. 1. The Relightables System. Our volumetric capture setup combines traditional computer vision pipelines with recent advances in deep learning to achieve high quality models that can be relight in arbitrary environments.

We present "The Relightables", a volumetric capture system for photorealistic and high quality relightable full-body performance capture. While significant progress has been made on volumetric capture systems, focusing on 3D geometric reconstruction with high resolution textures, much less work has been done to recover photometric properties needed for relighting. Results from such systems lack high-frequency details and the subject's shading is prebaked into the texture. In contrast, a large body of work has addressed relightable acquisition for image-based approaches, which photograph the subject under a set of basis lighting conditions and recombine the images to show the subject as they would appear in a target lighting environment. However, to date, these approaches have not been adapted for use in the context of a high-resolution volumetric capture system. Our method combines this ability to realistically relight humans for arbitrary environments, with the benefits of free-viewpoint volumetric capture and new levels of geometric accuracy for dynamic performances. Our subjects are recorded inside a custom geodesic sphere outfitted with 331 custom color LED lights, an array of high-resolution cameras, and a set of custom high-resolution depth sensors. Our system innovates in multiple areas: First, we designed a novel active depth sensor to capture 12.4 MP depth maps, which we describe in detail. Second, we show how to design a hybrid geometric and machine learning reconstruction pipeline to process the high resolution input and output a volumetric video. Third, we generate temporally consistent reflectance maps for dynamic performers by leveraging the information

*Authors contributed equally to this work. K. Guo, P. Lincoln, P. Davidson developed major parts of the pipeline.
†Authors contributed equally to this work. X. Yu, J. Busch, M. Whalen, G. Harvey built the hardware and infrastructure components.
‡Authors contributed equally to this work. S. Fanello led the volumetric capture algorithm and pipeline implementation, G. Fyffe led the relightability features and storage infrastructure, C. Rhemann led the capture hardware and software development, J. Taylor led the engineering and algorithmic optimizations.
§Equally last.

Authors' address: Kaiwen Guo, kwguo@google.com; Peter Lincoln, lincolnp@google.com; Philip Davidson, pdavidson@google.com; Jay Busch; Xueming Yu; Matt Whalen; Geoff Harvey; Sergio Orts-Escolano; Rohit Pandey; Jason Dourgarian; Danhang Tang; Anastasia Tkach; Adarsh Kowdle; Emily Cooper; Mingsong Dou; Sean Fanello, seanfa@google.com; Graham Fyffe, fyffe@google.com; Christoph Rhemann, crhemann@google.com; Jonathan Taylor, jontaylor@google.com; Paul Debevec, debevec@google.com; Shahram Izadi, shahrami@google.com, Google Inc.

0730-0301/2019/11-ART217
https://doi.org/10.1145/3355089.3356571

contained in two alternating color gradient illumination images acquired at 60 Hz. Multiple experiments, comparisons, and applications show that The Relightables significantly improves upon the level of realism in placing volumetrically captured human performances into arbitrary CG scenes.

CCS Concepts: • **Computing methodologies → Volumetric models**; *Reflectance modeling*;

Additional Key Words and Phrases: volumetric capture, relightability, photometric stereo, reflectance estimation

ACM Reference Format:
Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, Danhang Tang, Anastasia Tkach, Adarsh Kowdle, Emily Cooper, Mingsong Dou, Sean Fanello, Graham Fyffe, Christoph Rhemann, Jonathan Taylor, Paul Debevec, and Shahram Izadi. 2019. The Relightables: Volumetric Performance Capture of Humans with Realistic Relighting. *ACM Trans. Graph.* 38, 6, Article 217 (November 2019), 19 pages. https://doi.org/10.1145/3355089.3356571

1 INTRODUCTION

Capturing high quality, photorealistic 3D models of humans is an active area of research in the computer vision and machine learning communities [Balakrishnan et al. 2018; Collet et al. 2015; Dou et al. 2017; Neverova et al. 2018; Orts-Escolano et al. 2016; Pandey et al. 2019; Zollhöfer et al. 2014]. Indeed, digital 3D humans can be employed in a variety of applications that range from photography [Meka et al. 2019; Sun et al. 2019] to avatars in augmented and virtual reality [Martin-Brualla et al. 2018; Orts-Escolano et al. 2016]. In the last few years, we have witnessed the rise of state-of-the-art volumetric capture systems such as Collet et al. [2015], as well as companies and start-ups such as 8i, Omnivor, Intel®, and Metastage. Although the technology has made incredible progress, and it has reached a high level of quality, these reconstructions still lack true photorealism. In particular, despite these systems using high end studio setups with green screens (*e.g.*, Collet et al. [2015]), they still struggle to capture high frequency details of performers and they only recover a fixed illumination condition. This makes these volumetric capture systems unsuitable for photorealistic rendering of performers in arbitrary scenes under different lighting conditions, which is a prerequisite for many AR/VR/CG applications.

An orthogonal research trend consists of capturing 2D images of humans under multiple illumination conditions, which enables full relightability in arbitrary environments. These systems usually rely on a Light Stage [Debevec et al. 2000], and provide a high degree of photorealism. Unfortunately, these methods do not estimate the underlying geometry, resulting in a rough proxy rather than an accurate 3D reconstruction, which either limits the viewpoint or generates artifacts when rendering new viewpoints [Peers et al. 2006]. Beeler et al. [2010, 2011] have shown impressive facial performance capture results using passive stereo, but their results lack the reflectance information required for photorealistic relightability. Ultimately, all of these approaches have been very successful and represent the foundation of many industrial applications.

The lack of true photorealism, relightability, and high frequency details has made volumetric capture pipelines fall short of the visual quality of traditional digital cinematography. In this paper we present "The Relightables": a performance capture system that bridges the gap between photorealistic relightable image based systems and volumetric capture. Specifically, we propose a full, end-to-end volumetric pipeline from the ground up. Our main capture setup relies on a Light Stage (a custom geodesic sphere with 331 programmable lights) and novel depth sensors based on active illumination which generate 60 Hz per-viewpoint depth maps of 4112 × 3008 pixels.

We designed the reconstruction pipeline by combining elements from traditional geometry pipelines with recent advances in deep learning. This removes the need for a green screen and allows for more flexible lighting conditions. In particular, we program our custom lights to generate two spherical color gradient illumination patterns as proposed by Fyffe et al. [2009]. These alternating gradient images are captured at 60 Hz and then used to generate the full reflectance maps for each 3D frame at a final output rate of 30 Hz.

Multiple experiments, evaluations, and applications show that our system reaches an unprecedented level of quality for volumetric reconstructions, and we believe it will set the foundation for the next generation of content generation for AR/VR/CG applications.

2 RELATED WORK

Performance capture of humans is one of the most active topics in the field over the last few years. In this section, we present the most representative methods, which we can categorize as *image-based methods*, *model based approaches*, *volumetric capture systems*, and finally *machine learning based algorithms*.

Image-based Rendering (IBR). The seminal system proposed by Debevec et al. [1998] paved the way for many followup research trends on the topic, which are still active and very challenging. These methods find their culmination in very sophisticated systems such as the Light Stage proposed by Debevec et al. [2000]. Although capable of generating very high quality re-rendered images, these systems usually require multiple shots to infer detailed surface normals and reflective properties [Debevec et al. 2000].

Another trend solves the texturing of an object with known geometry using a Conditional Random Field (CRF) model. Lempitsky and Ivanov [2007] rely on a projective texturing approach, where multiple images are blended together according to a certain energy function, whereas Zhou et al. [2005] use a sparse set of correspondences. Despite the impressive results, they do not provide a fully relightable 3D model but instead generate an improved texture map.

Model Based. These methods usually rely on strong priors by adding some constraints in the reflectance and/or lighting models [Barron and Malik 2015; Meka et al. 2017]. Fully parametric models for geometry, reflectance and illumination have been explored for human bodies [Theobalt et al. 2007] and faces [Blanz and Vetter 1999; Garrido et al. 2013, 2016; Gotardo et al. 2018; Ichim et al. 2015; Thies et al. 2016]; however, the results are uncanny and do not cope well with fine grained details that the parametric model cannot capture, such as hair and apparel.

For the special case of faces, relighting has been performed under a diffuse appearance assumption based on radiance environment maps and ratio-images [Wen et al. 2003]. Other approaches jointly estimate parametric Bidirectional Reflectance Distribution Function

(BRDF) models and wavelet-based incident illumination to relight 3D videos of humans [Li et al. 2013b]. Cosine lobe relighting can be performed analytically based on a pair of spherical gradient illumination images [Fyffe et al. 2009], but secondary effects such as shadows are of low quality due to the use of approximations in modeling the face geometry.

Recently, advances in deep learning [Saito et al. 2017; Yamaguchi et al. 2018] show how to estimate the parameters of a predefined reflectance model from single images. Gotardo et al. [2018] extracted a Spatially Varying Bidirectional Reflectance Distribution Function (SVBRDF) and geometry from images captured under uniform lighting, but their approach is restricted to the skin region. Very recent work, also tackle the hard problem of extracting the SVBRDF from a *single* image using a flash [Li et al. 2018a,b; Nam et al. 2018]. However, all of these methods rely on hand-crafted priors and they are usually limited to specific parts of the human body. Often they do not handle specularities and fine grained details and they are restricted to model low-frequency illumination conditions.

Volumetric Capture. These sophisticated systems usually rely on a well orchestrated studio setup such as by Collet et al. [2015], Prada et al. [2017], Starck and Hilton [2007], and Tanco and Hilton [2000], where hundreds of cameras are carefully placed to cover the full capture volume. They often employ a *green screen* with a *fixed illumination condition* to simplify the segmentation and reconstruction problems. In particular, the method by Collet et al. [2015] relies on multi-view depth prediction from multiple sources (active illumination, RGB, and shape from silhouette) to generate rough point clouds. Next, they use Poisson Surface Reconstruction (PSR) [Kazhdan and Hoppe 2013] to retrieve a mesh, followed by multiple post-processing stages. Lastly, a tracking algorithm produces temporally consistent meshes that can be easily compressed and streamed. Prada et al. [2017] extended the previous work to support texture tracking. Despite the high quality reached by these frameworks, they still lack true photo-realism due to the fixed lighting conditions.

An orthogonal trend consists of real-time estimation of a temporally consistent (tracked) model of the performer [Dou et al. 2016; Du et al. 2019; Newcombe et al. 2015; Orts-Escolano et al. 2016; Zollhöfer et al. 2014]. Recent advances in *high speed* 3D capture sensors [Fanello et al. 2017a,b; Kowdle et al. 2018; Tankovich et al. 2018] provide robust high speed-tracking by reducing inter-frame motion [Dou et al. 2017; Guo et al. 2018]. These methods, however, still suffer from both geometric and texture inconsistency, as shown by Dou et al. [2017] and Martin-Brualla et al. [2018]. Despite the incredible efforts, these real-time systems usually lag their offline counterparts in terms of realism.

Machine Learning. Very recent advances in deep learning have enabled realistic synthesis of humans [Balakrishnan et al. 2018; Chan et al. 2018; Ma et al. 2017, 2018; Neverova et al. 2018; Pandey et al. 2019; Si et al. 2018; Zhao et al. 2017]. Zhao et al. [2017] use coarse-to-fine Generative Adversarial Networks (GANs) to synthesize images that are still relatively blurry. Ma et al. [2017] rely on a pose detector in the input, which helps to disentangle appearance from pose, resulting in improved sharpness. More recent extensions of the method [Ma et al. 2018; Si et al. 2018] try to disentangle pose and appearance. Other trends rely on a dense UV map to re-render the target from a novel viewpoint [Neverova et al. 2018].

The very recent work of Pandey et al. [2019] showed how to disentangle appearance, pose, and viewpoint, generating compelling renderings from arbitrary views using just a single sensor. In Lombardi et al. [2019] authors provide an elegant solution to combine mesh based rendering with neural rendering by learning a 3D volume representation from multiple RGB images. However, all these methods usually assume a fixed lighting condition.

Our Approach. The proposed system is a unique combination of traditional geometrical computer vision pipelines enhanced by recent machine learning advances, showing how to obtain high quality relightable volumetric videos of humans. Similar to IBR methods, our system relies on a Light Stage, which we use to generate two spherical gradient illumination conditions that are the key to generating fully relightable 3D models. Different from IBR methods, we do not compute proxy geometry, but we augment the Light Stage with multiple custom depth sensors that can capture high resolution (4112 × 3008) depth maps at 60 Hz. Given the complexity of the studio setup, we do not rely on a green screen to perform segmentation and to guide the reconstruction, but we rather employ a deep learning based segmentation to retrieve precise silhouettes. We then formulate the mesh tracking as a labeling problem and we compute an "optimal" tracked path by solving an a Markov Random Field (MRF) inference problem. As such, the proposed solution largely avoids heuristics used in related work.

We demonstrate the effectiveness of the proposed method in multiple scenarios and applications, showing results that are, for the first time, comparable with the ones obtained with image based renderings, without suffering from their limitations.

3 THE RELIGHTABLES

The Relightables system has three main stages: capture, reconstruction, and rendering. At its core, it relies on a Light Stage combined with multi-view (active) stereo depth sensors: a custom spherical dome with 331 programmable lights and 90 high-resolution 12.4 MP reconstruction cameras.

The capture cameras are a combination of Infrared (IR) cameras (32), which leverage active IR structured light illumination, as well as RGB cameras (58). The IR sensors provide accurate and reliable 3D measurements, while the RGB cameras capture high quality geometry normal maps and textures. The cameras record raw video at 60 Hz, where we interleave two different visible lighting conditions based on spherical gradient illumination [Fyffe et al. 2009].

A capture of 600 frames (*i.e.*, 10 s), generates roughly 650 GB of data. For each session, we also record a small geometric calibration sequence similar to Collet et al. [2015] and a *clean-plate* sequence of 50 frames, *i.e.*, the stage without any performer. The latter is used for segmenting the actor during the actual performance.

Once we upload the data to a common repository, a distributed system processes each frame in parallel. This first phase generates per-camera depth maps, segmentation maps and 3D meshes [Kazhdan and Hoppe 2013].

An alignment algorithm [Newcombe et al. 2015] consumes the sequence of reconstructed meshes so that long subsequences can

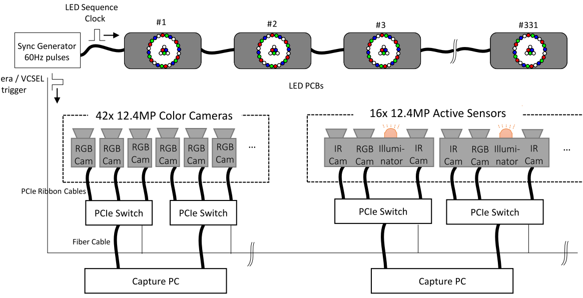
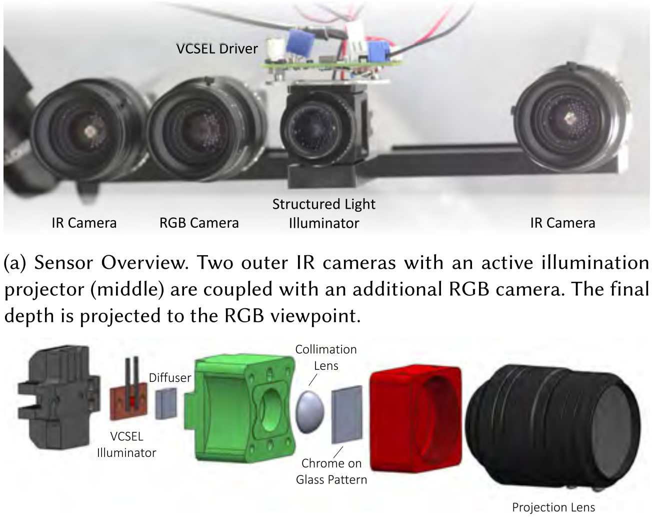


Fig. 2. Hardware System Architecture. 331 lights are synced with 58 RGB and 32 IR cameras running at 60 Hz. Multiple capture stations save the data in real-time on disk.



(a) Sensor Overview. Two outer IR cameras with an active illumination projector (middle) are coupled with an additional RGB camera. The final depth is projected to the RGB viewpoint.

(b) Structured Light Illuminator. The VCSEL emits light at 860 nm, which (as it passes left to right) is diffused, collimated, patterned, and projected. The selection of pattern detail and projection lens determines the effective resolution and FOV of the illuminator.

Fig. 3. Active Depth Sensor Components.

share a common triangulation. We propose a novel formulation to the keyframe (*i.e.*, triangulation) selection problem where it is cast and solved as an MRF inference problem. Each unique triangulation, is parameterized [Sander et al. 2002; Zhou et al. 2004] into a common 2D texture space that can be shared with all frames sharing that triangulation.

Each mesh has two gradient spherical illumination images available, from which, we generate albedo, normals, shininess, and ambient occlusion maps. These maps are compatible with standard rendering engines and can be used to relight the volumetric captures according to any desired illumination condition. The overall pipeline is shown in Figures 8 to 10. In the following sections, we detail each part of the system.

3.1 Hardware and Capture Setup

There are a number of recent works that address the task of full-body volumetric reconstruction using multi-view RGB cameras or multi-view depth sensors. In this work, to capture volumetric relightable performances, we use active stereo depth sensors together with Light Stage technology to efficiently capture photometric normal maps as well as the color, texture, and appearance of the performer.

The overall hardware architecture is illustrated in Figure 2. The system comprises of 58 RGB cameras, 32 IR cameras, 16 IR structured light projectors, and 331 programmable light boards. When recording, all components are synchronized by a pulse generator at 60 Hz. The sections that follow describe this hardware system.

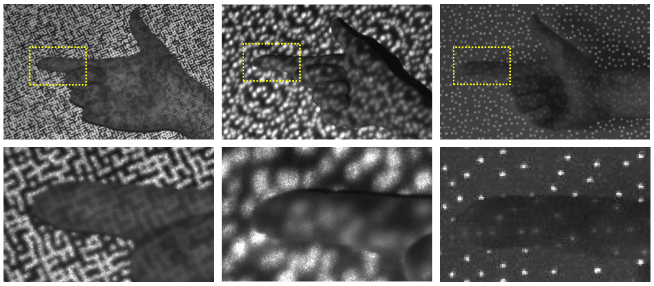
3.1.1 High Resolution Sensors. The 90 12.4 MP cameras in our system work together to capture detailed texture and geometry. Both our RGB and IR cameras are Ximea scientific cameras (Ximea MX124) that use the Sony® IMX253 sensor, which is a CMOS, global-shutter, 4112 × 3008 resolution chip with good quantum efficiency. The high resolution and low noise provides good details and enables robust feature matching between the capture images.

In order to capture at 60 Hz, the cameras are connected via PCIe to the capture PCs. To simplify cabling (data, power, and sync) and to reduce the number of capture machines, sets of 3 to 6 cameras are grouped via PCIe switches, which bridge the copper PCIe Gen2x2 camera interface to a fiber PCIe Gen3x4 or Gen3x8 (depending on camera count). The switch also distributes power and synchronization signals from buses built into the Light Stage. The fiber cables connect back to the capture PCs. We use 16 capture PCs in total, where each PC serves up to two switches.

Viewpoints and Distribution. Cameras on the Light Stage are distributed evenly around the sphere. In order to balance coverage and detail, we selected a C-mount camera lens (Kowa LM16HC) with FOV of about 48° along the long-axis. Given the radius of the sphere, this enables a given camera to capture roughly half the height of a human user. Six cameras are located near the top of the Light Stage, and the remaining cameras are split into three levels: torso and head, mid-section, and legs and feet. To support calibration and provide some redundancy, each level overlaps with its neighbors.

Active Depth Sensing. The 32 IR (and 16 of the RGB) cameras are grouped along with a custom IR structured light illuminator into custom active sensors (*see* Figure 3). The IR cameras are built using monochrome versions of the Ximea MX124 cameras and an IR bandpass filter, centered at 860 nm. As the primary source of 860 nm light in our capture environment comes from our structured light illuminators, the IR cameras are tightly tuned to see only that light. The RGB camera in the sensor ensures that a nearby visible-light reference viewpoint is available for the multi-view depth estimation algorithm.

Our custom structured light illuminators use a chrome-on-glass, direct-projection mask together with a Vertical Cavity Surface Emitting Laser (VCSEL) diode to project a structured pattern into the scene. The design of our projector is detailed in Figure 3b: a VCSEL diode emits IR light at 860 nm (nominal). The light passes through an optical diffuser and an aspheric lens that collimates the light



(a) Our Pattern (b) AMS VCSEL Pattern [Kowdle et al. 2018] (c) Kinect® DOE Pattern [Collet et al. 2015]

Fig. 4. Pattern Comparison. We compare our pattern (a) with those of commercially available solutions (b) and (c) when captured with a 12.4 MP camera at 1.5 m distance using a lens with a 60° FOV. Note that our pattern is much denser which is important to reconstruct small details like the finger shown in this image. In each image, a single pattern projector was active.

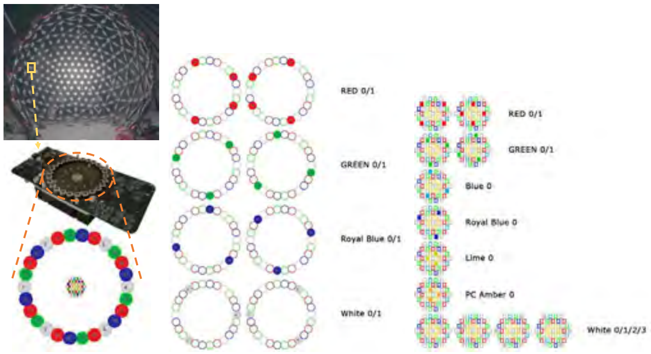
before it hits the chrome-on-glass mask which generates the structured pattern. A projective lens (Evetar N118B0818IRM12) focuses the pattern to give a Field of Illumination (FOI) of about 50°, slightly larger than the FOV of our cameras. Figure 4 shows the pattern generated by our projector and compares it with those of other commercially available solutions used in Kowdle et al. [2018] and Collet et al. [2015]. Note that our custom pattern has much denser features and matches the high resolution of the camera, which is crucial for stereo matching to generate high quality, high resolution depth maps. Additionally, the projective lens component of our illuminator allows us to match the FOI to the cameras' FOVs to support alternative capture conditions. Note that the pattern could be also optimized for a given camera configuration such as in Mirdehghan et al. [2018]. In this work we simply rely on an off-the-shelf grid.

During capture sessions, the distribution of sensors ensures that most parts of the human subject will be covered by a few illuminators. At this quantity, even though the illuminators pulse simultaneously, these patterns do not interfere, but instead combine to provide additional texture.

As our structured light emitter is an invisible laser device, we tested it according to the international standard for eye and skin hazard assessment [International Electrotechnical Commission 2014] and found that the Power-to-Limit Ratio (PLR) to be within the limits of Class 1 laser safety, which means it is safe under all normal use conditions. Furthermore, we ensure that all safety interlocks are in place to ensure a safe end-to-end system.

3.1.2 Light Stage Hardware. The Light Stage is composed of 331 custom programmable light units. All units are linked via a high-speed (min. 600 MHz) daisy-chained network. Data and synchronization signals are transmitted from a PC-controlled master unit, which programs the Light Stage with the desired lighting pattern. Each light unit is populated with a total of 63 high brightness Light-Emitting Diodes (LEDs) that cover a wide spectrum including red, amber, lime, green, blue, and royal blue (*see* Figure 5).

Each LED is controlled by the on-board System on a Chip (SoC) by means of both digital and analog modulation, capable of toggling as fast as 10 MHz. The analog modulation circuit is programmable to set the current limit for each LED when the LED channel is on. We use Pulse Density Modulation (PDM) as the digital method to realize linear grayscale. PDM out-performs traditional Pulse Width Modulation (PWM) [Lincoln 2017] as it achieves a lower grayscale error, eliminating the need for sub-frame synchronization (*see* Figure 6). The 10 MHz toggling speed allows the Light Stage system to generate High-Dynamic Range (HDR) lighting patterns at high frame-rates.



(a) Light Stage, LED Board, and LED Layout (b) Face LED Zones. (c) Body LED Zones.

Fig. 5. Lighting Setup. The Light Stage consists of 331 light boards (a), each with an outer ring of face-optimized (narrow FOI) LEDs (b) and central region of body-optimized (wide FOI) LEDs (c). The board's SoC individually controls each zone.

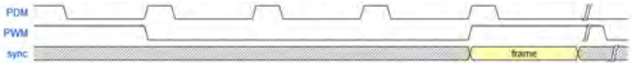


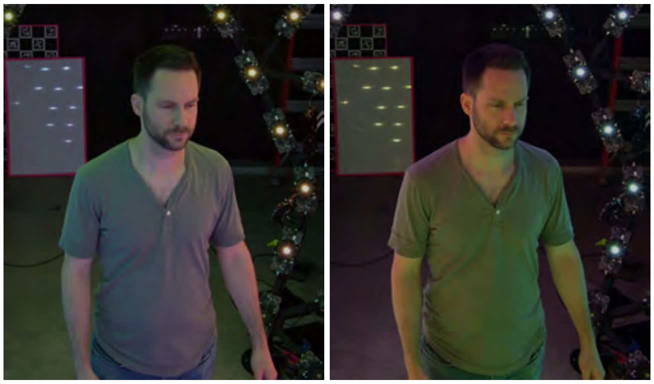
Fig. 6. A misaligned frame causing duty-cycle error with PWM but not with PDM.

The on-board SoC is a custom design of a soft CPU running within a Field-Programmable Gate Array (FPGA) that runs a real-time OS. It delivers a fast response to external trigger signals for the purpose of accurate synchronization between the Light Stage and cameras.

Although our capture system runs at 60 Hz, we alternate the light patterns at 180 Hz in order to be comfortable and imperceptible to the performer [Fyffe et al. 2011].

3.1.3 Data Capture. For each capture session, we program our Light Stage to produce two different lighting patterns using RGB LEDs similar to Fyffe et al. [2009]; however, we use a gradient and an inverse gradient instead of a gradient and white light. The gradient is linear over the sphere, with red from dark to bright along the X axis, green from dark to bright along the Y axis, and blue from dark to light along the Z axis. The inverse gradient uses the same axes, but reverses the bright to dark directions (*see* Figure 7).

The distributed video capture system currently operates the 16 high-performance workstation PCs from a central PC. Collectively, when the system runs in 8 bit mode at 60 Hz (using a 2 ms exposure time), we produce raw data at about 65.3 GB/s. Our custom capture software performs minimal processing during capture to reduce the



(a) Color gradient. (b) Inverse color gradient.

Fig. 7. Two different light patterns (color gradient images) using during capture.

memory and bus bandwidth load on the PC while storing the raw, uncompressed data on a collection of multiple high-performance NVMe SSDs in a distributed manner.

Synchronization and overall control of the system operates over a combination of components. A networked master control program provides centralized control over camera stream activation and recording activation. An external trigger bus ensures that all cameras and light units, and structured IR light projectors trigger in step, simultaneously. While the cameras and light units are directly controlled by the main trigger buses, the structured light illuminators are triggered by the cameras in their bundle, which supports software-defined enabling of their trigger response.

Calibration. One of the critical steps to ensuring high fidelity volumetric reconstructions and high quality textures is high precision geometric calibration of all the cameras in the system. We achieve this using a fairly well established approach [Zhang 2000], where we use a calibration chart with a known pattern of Calibu circle markers that allow us to precisely locate features with sub-pixel precision. We capture many synchronized images from all the cameras. With enough images captured to cover the FOV of each camera and spanning the volume of the stage, we perform a full-bundle Levenberg-Marquardt (LM) solve to obtain, for each camera, the intrinsics and extrinsics (relative to a single camera as the origin). We note that our calibration process achieves a mean reprojection error across all 90 cameras of less than 0.5 pixels which results in high fidelity reconstructions.

While normal captures would use the structured light illuminators, during this calibration, in order to clearly localize the features on the calibration chart, we disable the structured light illuminators in favor of IR flood illuminators. As this is an active multi-view stereo system, we do not need any prior information about the structure of the pattern, nor the projectors' poses.

3.2 Volumetric Reconstruction

In our system, computing accurate 3D geometry information from multiple viewpoints is one of the key building blocks. We adopt a multi-view stereo pipeline, enhanced with deep learning features and point cloud outlier removal before trianglular mesh generation. For efficient mesh processing, we further remove both geometric and topological artifacts from the generated meshes. To obtain a compact representation, we generate a UV atlas to store surface attributes, including normal, reflectance, diffuse textures, etc. To facilitate efficient streaming, we decimate the base mesh to around 25k facets.

3.2.1 Depth Estimation. We now explain how our system reconstructs accurate 3D geometry from raw images. Our system comprises 58 RGB cameras and 32 IR cameras. Although our custom depth sensor is able to provide high quality depth maps, there are still cases due to low SNR or highly reflective surfaces which may return a wrong estimate. To overcome this, we rely on a multi-view stereo algorithm that runs on IR and RGB independently such as in Collet et al. [2015]. Such a multi-view triangulation scheme can be defined by these main components [Scharstein and Szeliski 2002]: view selection, matching cost computation, disparity optimization, and refinement.

View Selection. The view selection defines a set of neighbors for each reference view. In our system, given a view, we use the calibration information to find the closest cameras. In particular, two views belong to the same neighboring set only if their distance is less than 50 cm and their viewing angles are within 30°.

Matching Cost. Traditional matching cost computations rely on Sum of Absolute Differences (SAD), Sum of Squared Distances (SSD), and Normalized Cross-Correlation (NCC); more recent approaches are training Convolutional Neural Networks (CNNs) to perform this task [Žbontar and LeCun 2016].

For IR images, we found NCC to be the preferred choice, which we compute over a small 7×7 window. Thanks to the highly detailed structured light pattern, this window size is sufficient in these high resolution images.

For matching across RGB images that lack texture in many regions, we propose to enhance the cost computation by computing features using a VGG network [Simonyan and Zisserman 2014] pretrained on ImageNet [Deng et al. 2009]. Given two images' patches, a simple SSD is used to aggregate the cost in a very small 3×3 window. Thanks to the learned features, this is sufficient to produce high quality depth even at high resolutions. This method could be thought as a fast approximation to Žbontar and LeCun [2016], which would not have been feasible in our scenario due to the prohibitive computational requirements.

Disparity optimization. To efficiently infer high resolution depth maps for each view point, we resort to the popular PatchMatch algorithm [Barnes et al. 2009; Bleyer et al. 2011; Galliani et al. 2015; Schönberger et al. 2016]. This method parameterizes the 3D scene using a first-order approximation: an array of slanted planes tangential to the ground truth surface. The checkerboard pattern from Galliani et al. [2015] is adopted to infer a slanted plane for each pixel in parallel. Depth maps from different view points are also computed independently in parallel. The depth estimation algorithm generates a pair of depth and normal maps for each input view point.

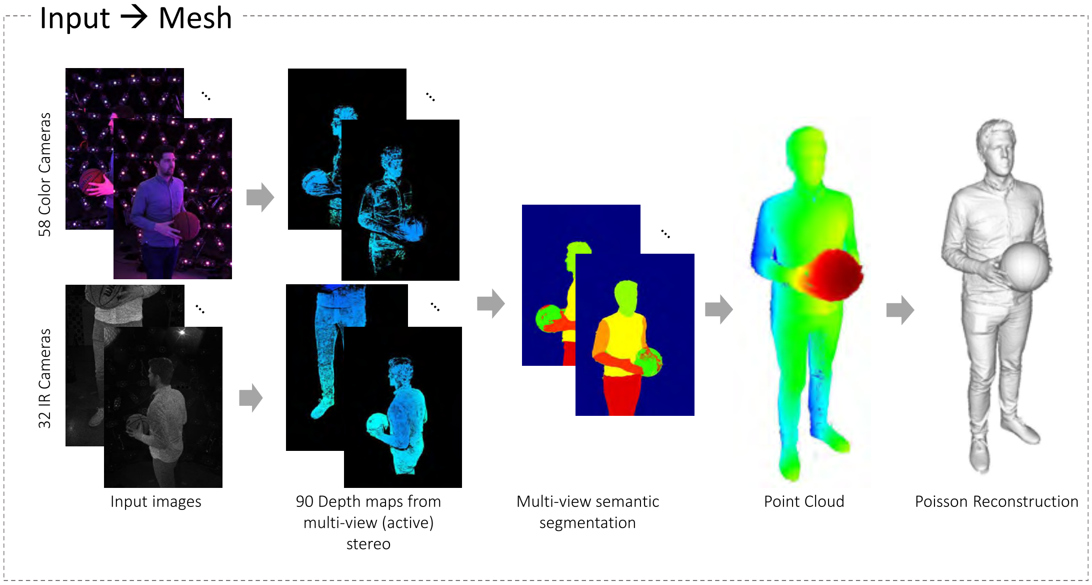


Fig. 8. The Relightables Pipeline (Part 1). First, raw images are used to reconstructed a high quality 3D model.

Depth Refinement. At this stage, some outliers may exist in the raw point cloud. To remove them, we first discard pixels with high matching error located at patches with low spatial color variation. We perform this step by simply computing the variance in a 7 × 7 window and discarding the pixel if the variance is lower than 0.7. Then we verify the view consistency for each pixel by projecting it to each of the selected views where the back-projected point would be visible. We define the mutual point-to-plane error ϵ_{err} as the following:

$$\epsilon_{\text{err}} = D_p(f_q) + D_q(f_p), \tag{1}$$

where $D_p(f_q)$ is a function to the compute distance from point p in the reference view to plane f_q sampled at the projection q of p in another view. If ϵ_{err} is larger than a certain threshold (5 mm), the view is discarded. After this view consistency check, if the number of selected views is less than N (we use $N = 3$), the pixel is invalidated. The view consistency check removes most mismatched points. After that, we attempt to remove any remaining outliers by removing small disconnected regions.

Depth Fusion. Each of the IR depth maps needs to be aligned with one RGB view for the remaining steps of the pipeline such as segmentation. A naive solution might consist of re-projecting the whole 3D pointcloud to all the RGB views, however in practice, this may cause serious issues with occluded pixels with missing depth. Therefore, we re-project each depth map generated from an IR camera to its closest RGB camera. In practice, this is an effective way to minimize issues with occluded areas. Assuming that active illumination provides higher depth accuracy, pixels where IR provides a valid depth value have higher priority and replace any value in the current RGB depth map.

3.2.2 Deep Learning Based Segmentation. Detecting and separating the performer from the background is crucial for any volumetric capture system. As shown by Collet et al. [2015], a green screen can be used to achieve compelling results. However, our setup relies on dynamic illumination conditions, which makes the use of a green screen very challenging. Indeed many cameras and lights have to protrude the screen making estimation of a dense matte hard. Moreover, the screen would cause significant color spill onto the subject which would interfere with the estimation of reflectance maps.

To tackle this problem, we enhance a traditional background subtraction method with a deep learning solution [Chen et al. 2016]. For each performance we record a *clean plate* sequence of 50 frames. For each frame and camera, a depth map is computed and the average over all depth maps is stored as D_{avg}.

At test time, each RGB camera has a depth image D, aligned with an RGB image I, which we use to compute a unary term defined by Equation 2:

$$\psi(D, I) = w_1 \psi_d(D_{avg}, D) + w_2 \psi_{rgb}(I) \ , \tag{2}$$

where $\psi_d(D_{avg}, D)$ is simply defined by evaluating the logistic function on the distance between the current observation D and the average depth D_{avg} as detailed by Orts-Escolano et al. [2016]. The term $\psi_{rgb}(I)$ is the confidence of the semantic segmentation network [Chen et al. 2016]. In all our experiments, we set the contribution of the depth term to $w_1 = 0.6$ and the semantic contribution to $w_2 = 0.4$.

We refine this unary term by solving a CRF which introduces a pairwise potential term to enforce smoothness across neighboring pixels. In practice we rely on Krähenbühl and Koltun [2011] to perform the inference.

This first segmentation pass already achieves very compelling results and does a great job at detecting the performer and most of the foreground objects. However, the machine learned solution was trained to detect people and not objects, therefore some apparel

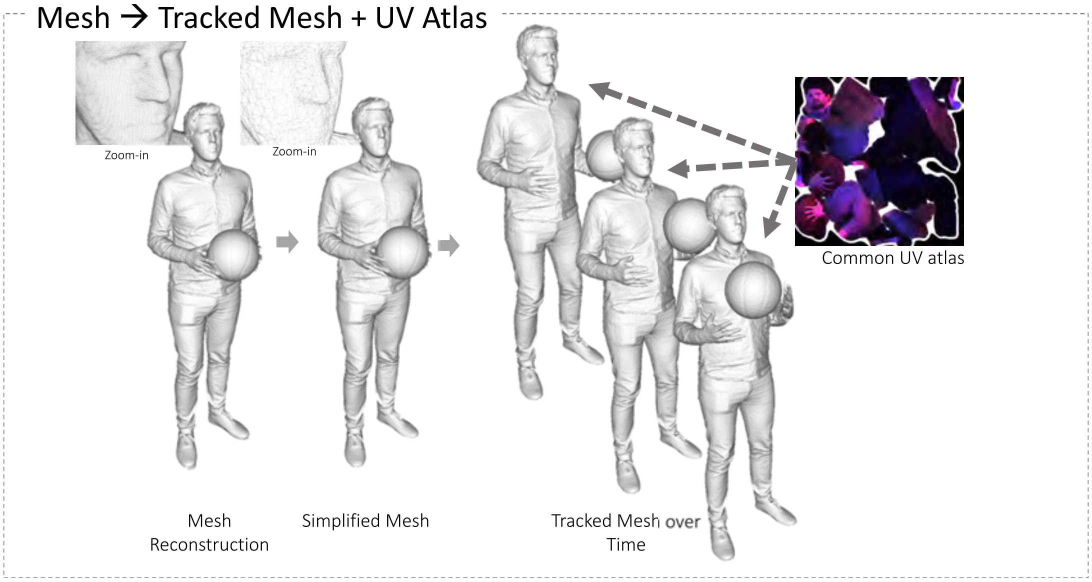


Fig. 9. The Relightables Pipeline (Part 2). This mesh then gets downsampled, tracked over time and parameterized.

will be falsely segmented out (*see* Figure 14b). To solve this, we propose to add a multi-view consistency step. Segmented points are re-projected in 3D space and, using the calibration information, are projected to the original views. As a further refinement, another pass of the CRF produces boundaries that better follow the image edges. The final results can be appreciated in Figure 14c. Notice how all the proposed steps improve the final foreground segmentation: the CRF solver deals with the low resolution output produced by the network and the multi-view refinement step recovers the misdetected object.

3.2.3 Mesh Reconstruction, Simplification, and Post Processing. The segmented depth maps are projected to 3D to generate a point cloud in the Light Stage coordinate system (*see* Figure 8). Due to small miscalibrations between IR and color sensors, we apply an optional ICP-based (Iterative Closest Point) bundle adjustment [Li et al. 2013a] to accurately register the point cloud from multiple views. Then we project each point to a locally fitted plane produced by Moving Least Squares projection [Collet et al. 2015], which compensates for the remaining non-rigid alignment errors. Since we use high resolution sensors, we do not perform the 3D optimization step proposed in Collet et al. [2015] as we noticed doing so introduced additional outliers.

At this stage, the point cloud is well-aligned and clean, and Poisson reconstruction can generate visually pleasing triangular meshes. Similar to Collet et al. [2015], we clamp signed distance values for voxels lying on the background to zero in order to constrain the reconstructed surface within the visual hull. The number of facets on the resulting mesh is around 300k-400k, and it still contains both geometric and topological imperfections. Thus, we run decimation [Garland and Heckbert 1997] on the mesh to bring the number of facets down to 25k, which helps remove most artifacts. Then we remove any remaining mesh islands and surface degeneracies by collapsing edges. During the decimation, we set large penalties on the face and hands to preserve detail, based on semantic information produced by our deep learning-based multi-view segmentation method. Finally, we eliminated topological artifacts through a denoising method [Collet et al. 2015; Guskov and Wood 2001].

3.2.4 Mesh Alignment. At this point, we have N independently reconstructed meshes in a sequence. As a result of noise, the appearance jumps from frame to frame. Furthermore, they do not share a common triangulation which is necessary to efficiently compress both the geometry and texture. Therefore, we seek to represent large contiguous sub-sequences with a single triangulation, for which temporally smooth vertex positions can be used to model the geometry.

Frame to Frame Alignment ($N = 2$). We solve the two-frame mesh alignment problem by deforming one frame to align with the other. Like Li et al. [2009] and Dou et al. [2015], we adopt the embedded deformation graph representation [Sumner et al. 2007] to parameterize the deformation of one mesh so that it can be aligned with another. A deformation graph is a representation of non-rigid motion near the surface. It contains a group of nodes whose positions have been sampled uniformly from the mesh vertices, under the constraint that the distance between two nodes is at least ϵ_{dist}. Each node is also connected to its H ($H = 8$ in our implementation) nearby neighbors, thus forming a graph. Node i is parameterized by an affine transformation $\mathbf{T}_i = (\mathbf{A}_i, \mathbf{t}_i), \mathbf{A}_i \in \mathbb{R}^{3\times3}, \mathbf{t}_i \in \mathbb{R}^3$. Thus, the transformation of a point v close to the surface is represented by linear blend skinning of its K (we use $K = 4$) nearby nodes, $\mathbf{T}(v) = \sum_i \omega_i \mathbf{T}_i$. We define the two-frame mesh alignment as an optimization problem to compute the optimal parameters so that the deformed mesh fits well with the next frame. The objective function of this optimization is defined as the following:

$$E_{\text{align}} = \alpha_{\text{d}} E_{\text{data}} + \alpha_{\text{s}} E_{\text{smooth}} + \alpha_{\text{r}} E_{\text{rigid}} + \alpha_{\text{t}} E_{\text{det}} , \quad (3)$$

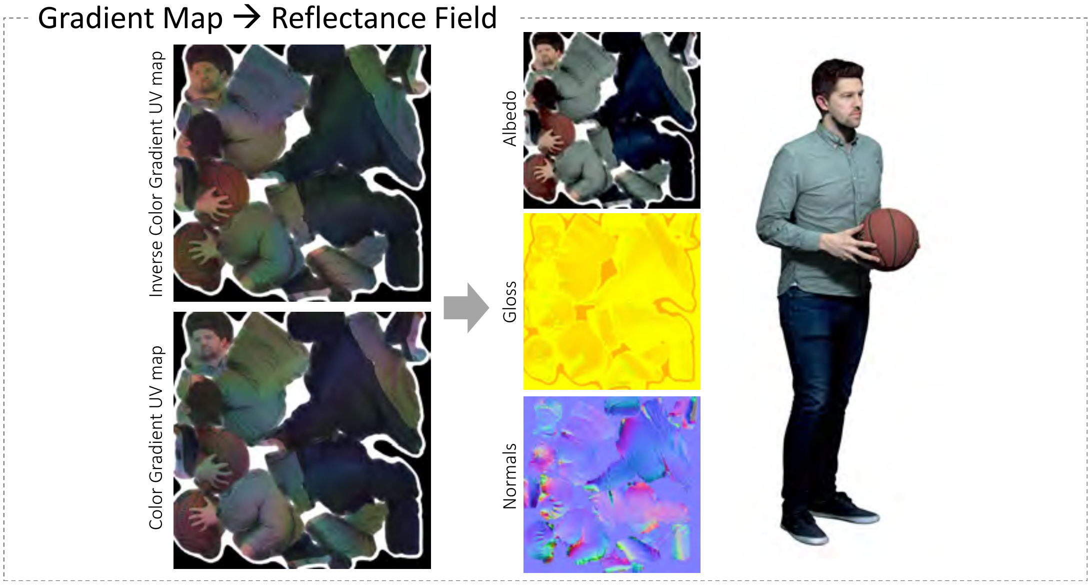

Gradient Map → Reflectance Field
Inverse Color Gradient UV map
Color Gradient UV map
Albedo
Gloss
Normals

Fig. 10. The Relightables Pipeline (Part 3). Finally reflectance maps are inferred from two gradient illumination conditions.

where E_{data} sums point-to-plane distances of all correspondences in both directions, E_{smooth} sums over pairwise smoothness terms between neighboring nodes, and E_{rigid} and E_{det} measure how far affine transformations are from SO3 space to enforce local rigidity. We gradually relax weights of α_s and α_r to generate fine-grained non-rigid deformation. For a detailed description, please refer to Li et al. [2009] and Dou et al. [2015]. The two-frame alignment is agnostic to previous deformations, similar to plastic deformation, and we build a new one for each two-frame tracking problem to prevent accumulating a non-uniform distribution of embedded nodes.

Global Mesh Alignment ($N > 2$). Given $N > 2$ meshes in a sequence, we would like to bring them into alignment leveraging the frame to frame alignment strategy detailed above. As such, for the n-th mesh, we use a sequential alignment algorithm to align to the other meshes. In particular, we first proceed forward in time, sequentially aligning the n-th mesh to all its proceeding meshes. Likewise, we can proceed backward in time, sequentially aligning the n-th mesh to its all its preceeding meshes. As a result, we have aligned the n-th mesh with all other frames.

At the end of this procedure, we obtain a matrix of aligned-meshes $\mathcal{M}$ together with an alignment error matrix $\mathcal{E}$. Entry $\mathcal{M}_{nm}$ of the aligned-meshes matrix contains the aligned mesh of the n-th frame towards the m-th frame (that is, a mesh with the same triangular mesh topology as the n-th input mesh, but with vertex positions that are aligned with m-th input mesh). Entry $\mathcal{E}_{nm}$ of the alignment error matrix contains the misalignment measure between the n-th mesh aligned with the m-th frame.

The goal is to assign to each frame n a mesh $\mathcal{M}_{b_n n}$ with the smallest alignment error $\mathcal{E}_{b_n n}$, where $b_n \in \{0, ..., N\}$. At the same time, we want to minimize the number of times the triangulation changes. We thus formulate this as a discrete MRF with the following energy terms:

$$E(b_1, ..., b_N) = \sum_{n=1}^{N} \mathcal{E}_{b_n n} + \lambda \sum_{n=1}^{N-1} I(b_n \neq b_{n+1}) \,, \tag{4}$$

where $I(\cdot)$ is the indicator function, the first term encourages an assignment with low error and the second term encourages reusing the same triangulation. As this MRF is a single chain and thus is tree-structured, belief propagation can perform inference exactly.

This approach minimizes the number of keyframes used for a given sequence. This is also crucial to reduce "popping" artifacts when a new keyframe is selected. Indeed, changes in the mesh topology could potentially lead to unpleasant flickering during the playback.

3.2.5 *Consistent UV Parameterization.*

The aligned, topologically consistent sequence is not sufficient to render high quality geometrical details. In order to achieve the desired results, we parameterize [Sander et al. 2002; Zhou et al. 2004] these meshes so that we can separate the details from the base geometry using a displacement texture map in UV space. To compute such as parameterization, we use the well established Microsoft UVAtlas software package [Microsoft 2019]. Similar to the system by Collet et al. [2015], we increase vertex weights for facial regions so that the unfolded face takes more texels in the UV space. Projected semantics from segmentation label vertices in facial regions. After mesh alignment, we divide the entire sequence into groups, each of which have the same mesh topology. As such, the meshes within a group can share a common UV parameterization as to enforce temporal and spatial consistency (*see* Figure 9).

3.3 Reflectance Maps Generation

Volumetric capture systems typically operate using fixed lighting and compute a color texture map with the lighting "baked in". Hence it is difficult to produce realistic renderings of the captured subject under novel illumination or in novel environments. These systems often surround subjects by even illumination in an attempt to obtain a color map, but renderings produced using a single color map and a Lambertian reflectance assumption exhibit unrealistic double-shading. They also tend to rely on the geometry to provide surface normals for shading, which is typically far coarser than the details visible in the images. To remedy these issues, we propose to capture more detailed reflectance information, using the two different color gradient lighting conditions in our capture process.

3.3.1 Texture Blending in UV Space. In each frame of a performance, we blend images from the color cameras in the mesh UV space using Poisson blending, with the contribution of each camera weighted by the dot product of the surface normal and view vector. We exclude cameras that are occluded at each point in UV space, using the mesh and ray casting to compute occlusion. Alternating frames contain either the color gradient illumination or the inverse color gradient illumination. Our reflectance map estimation requires both illumination conditions to be aligned in UV space. To achieve this we evaluated two different strategies. The first strategy relies on our mesh alignment step to produce consistent texture parameterizations for adjacent frames, allowing each frame to borrow the complementary *UV texture* from one of its neighbors. The second strategy uses optical flow in image space between consecutive frames [Anderson et al. 2016], therefore the complementary illumination is retrieved in *image space* for each camera prior to UV space blending. Since the mesh alignment step may not accurately track high frequency details in the texture space due to tangential motion, we found this second strategy more effective.

3.3.2 Reflectance Estimation. We use the color gradient and inverse color gradient to compute a reflectance estimate similar to Fyffe et al. [2009]. Equation 5 describes the *rgb* color channels for the color gradient G^+ and inverse color gradient G^- lighting conditions.

$$G_r^+ = \left(\tfrac{1}{2} + \tfrac{1}{2}\Theta_x\right)L;\ G_g^+ = \left(\tfrac{1}{2} + \tfrac{1}{2}\Theta_y\right)L;\ G_b^+ = \left(\tfrac{1}{2} + \tfrac{1}{2}\Theta_z\right)L;$$
$$G_r^- = \left(\tfrac{1}{2} - \tfrac{1}{2}\Theta_x\right)L;\ G_g^- = \left(\tfrac{1}{2} - \tfrac{1}{2}\Theta_y\right)L;\ G_b^- = \left(\tfrac{1}{2} - \tfrac{1}{2}\Theta_z\right)L, \quad (5)$$

with $\Theta \in S^2$ representing the direction from the subject to the (presumed distant) light, and L the overall intensity. Note that $G^+ + G^- = L$, while $G^+ - G^- = \Theta L$. Intuitively, the sum of the color gradient and inverse color gradient photographs contains the albedo at each pixel (as if lit by white light), and the difference between the two photographs encodes the overall reflected direction of the reflectance (times the albedo). Since we rely on the relationship between light direction and light color, we first correct color cross-talk between the light color primaries and camera sensor color primaries using a 3 × 3 color matrix established using photographs of a color chart illuminated by each color of LED. We refer to the color corrected pixel values captured under color gradient illumination G^+ and inverse color gradient illumination G^- as g^+ and g^-, respectively. We also scale the overall magnitude of the color matrix such that a 100 % reflective white material appears with pixel values $\mathrm{g}^+ + \mathrm{g}^- = \{1, 1, 1\}$, established using the same color chart.

Assuming a simple Phong reflectance model having a Lambertian lobe with albedo k_d and surface normal n, and a specular lobe with albedo k_s, lobe axis r, and exponent n, the pixels take on the following values (adapted from [Fyffe et al. 2009]):

$$g^+ + g^- = k_d + \{k_s, k_s, k_s\}; \quad (6)$$

$$g_i^+ - g_i^- = k_{d;i} \int_{\Omega_\mathrm{n}} (\mathrm{n} \cdot \Theta)\Theta_i \; d\omega_\Theta + k_s \int_{\Omega_\mathrm{r}} (\mathrm{r} \cdot \Theta)^n \Theta_i \; d\omega_\Theta;$$
$$\therefore g^+ - g^- = \tfrac{2}{3} k_d \circ \mathrm{n} + \tfrac{n+1}{n+2} k_s \mathrm{r}, \quad (7)$$

with $i \in \{x, y, z\}$, and where Ω_α represents the hemisphere of directions on the positive side of axis α, and $\circ$ represents element-wise multiplication. Note as the color gradients are aligned with the cardinal axes, the cross-talk-corrected color channels $\{r, g, b\}$ are referred to interchangeably as the axes $\{x, y, z\}$.

The magnitude of the difference $g^+ - g^-$ relative to the sum $g^+ + g^-$ encodes information about the narrowness of the scattering, which may be interpreted as a cosine lobe exponent or shininess parameter. Indeed the ratio of these two quantities is $\frac{2}{3}$ for perfect Lambertian materials, and 1 for perfect mirror materials [Fyffe et al. 2009].

In contrast to previous work operating in image space [Fyffe et al. 2009], we operate on blended textures in UV space, which offers several advantages. Examining Equation 7, we see the surface normal n is trivially obtained for Lambertian materials, but is conflated with the reflection vector r for materials having a specular component. Previous single-view work resorted to heuristic conversion from reflected direction to surface normal using assumptions about the BRDF [Fyffe et al. 2009]. However, conveniently, the average value of r over many views surrounding the subject is itself n times a constant factor. Thus a benefit of operating on blended textures from multiple views is that Equation 7 leads directly to a photometric estimate of the surface normal (Equation 12) since view-dependent effects are averaged out. Further, the cosine weighting employed during blending downweights views with large Fresnel gain, yielding a more or less constant specular contribution for dielectric materials.

Despite averaging out view-dependent effects, the blended multi-view gradient illumination images still encode information about the narrowness or broadness of the scattering, as this is a phenomenon derived from the breadth of reflectance lobes rather than their directions with respect to a view vector. We define a narrowness of scattering β, as measured by Equation 8, which may be explained by various phenomena, including shininess, occlusion, and interreflection. We heuristically split the explanation between shininess and occlusion using the following intuition: surfaces with no occlusion will have a photometric surface normal estimate that is largely aligned with the geometric surface normal of the mesh, while surfaces with some occlusion may not. Hence we use the angle between the two surface normals (photometric and mesh) along with the scattering narrowness to estimate a shininess parameter and an ambient occlusion term using Equations 9 and 10. Ambient occlusion can then be removed from the albedo estimate by dividing it out. All told, the shininess s, ambient occlusion term o, albedo $\mathbf{a}$, and surface normal $\mathbf{n}$ are computed as follows:

$$\beta = \tfrac{3}{2}\left(|\mathbf{d}| - \tfrac{1}{3}\right) \quad \text{with} \quad \mathbf{d}_{i\in\{x,y,z\}} = \frac{\mathbf{g}_i^+ - \mathbf{g}_i^-}{\mathbf{g}_i^+ + \mathbf{g}_i^-} \; ; \tag{8}$$

$$s = \beta^{\,(1-\alpha)} \quad \text{with} \quad \alpha = \min(1, \cos^{-1}(\mathbf{n} \cdot \mathbf{n}^m)) \; ; \tag{9}$$

$$o = \beta^{\,\alpha} \; ; \tag{10}$$

$$\mathbf{a} = \frac{\mathbf{g}^+ + \mathbf{g}^- - \langle r_0, r_0, r_0 \rangle}{(1-o)(1-r_0)} \; ; \tag{11}$$

$$\mathbf{n} = \frac{\mathbf{d}}{|\mathbf{d}|} \; ; \tag{12}$$

where $\mathbf{g}^+$ and $\mathbf{g}^-$ are the color gradient illumination pixels g^+ and inverse color gradient illumination pixels g^-, respectively, blended over all non-occluded views, $r_0 = 0.04$ is an approximate dielectric Fresnel term at normal incidence, and $\mathbf{n}^m$ is the mesh normal. The linear mapping in Equation 8 scales β to range from 0 to 1 for rough diffuse to mirror. Depending on the specific shading model employed, this scaling might be omitted. The resulting albedo, surface normal, shininess, and ambient occlusion maps can be used in a real-time rendering engine or offline rendering system without further modification. In practice, shininess (or roughness) and ambient occlusion maps are encoded as a single "gloss" texture map (*see* Figure 10, right column, middle row).

4 RUN TIME

Volumetric capture systems require a considerable amount of computational resources; for instance, the state-of-the-art system by Collet et al. [2015] requires 30 min per-frame using 4 MP cameras. Scaling the system to 12 MP cameras increases the run-time substantially to the point that multiple days are required to process a few seconds of capture.

In order solve this issue, we designed the system to be massively parallel and distributed. Image pre-processing steps such as undistortion, demosaicing, and color correction, as well as multi-view stereo and segmentation computation are parallelized over *camera views.*

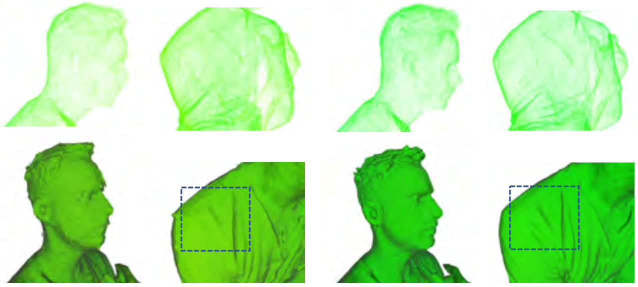
The multi-view segmentation refinement is instead parallelized only over *frames*, as each frame requires all the views to be available. The same approach is used for Poisson Reconstruction, Mesh Simplification, and Denoising.

The mesh alignment step is the most expensive part of the pipeline, since it requires the computation of all the possible tracking solutions across all the frames. Fortunately, each mesh n can be aligned to the others in parallel. Furthermore, for this n-th mesh, the sequential forward alignment and backward alignment's through time can be performed in parallel. Thus the level of parallelism that can be achieved at this granularity is $2n$.

The parameterization stage using UVAtlas runs only on the keyframes, as the tracked meshes all share the same topology. For a sequence of 600 frames we typically find an average of 5 keyframes.

The final stage of texture map computation is instead performed in parallel across all the frames.

A typical sequence of 10 s with 600 frames is processed in about 8 hours. Notice that processing the same amount of data on a single machine with 32 cores would require over a year, proving the importance of a well engineered system for performance capture applications.



(a) 4 MP 3D reconstruction.   (b) 12 MP 3D reconstruction.

Fig. 11. Input image resolution comparison. We compare the 3D reconstructions results generated using 4 MP (a) vs 12 MP (b) IR images. The first row shows the raw point cloud generated by our MVS implementation. The second row shows the reconstructed geometry using PSR. Note that by using 12 MP images we are able to recover small details such as facial expressions and clothing wrinkles.

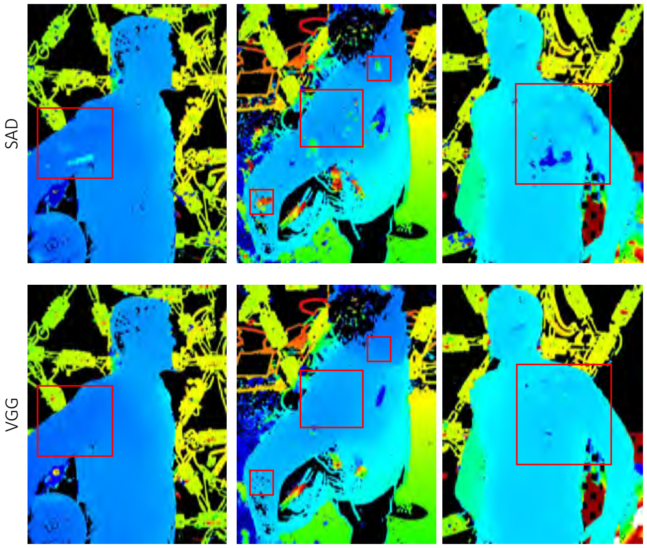


Fig. 12. Comparison of raw stereo matching results with SAD vs. VGG on RGB images. Note that the VGG results contain less gross errors and provide smoother results everywhere. Please see text for details.

These efforts, allow us today to have a production ready system which is orders of magnitude faster of the state of the art. However, we do acknowledge the computational power required is extremely high and that achieving real-time performance requires additional breakthroughs and follow up research.

5 EVALUATION

Our system is a very complex pipeline. In this section, to validate the proposed approach, we analyze the main components of the system and show evidence to justify our design choices.

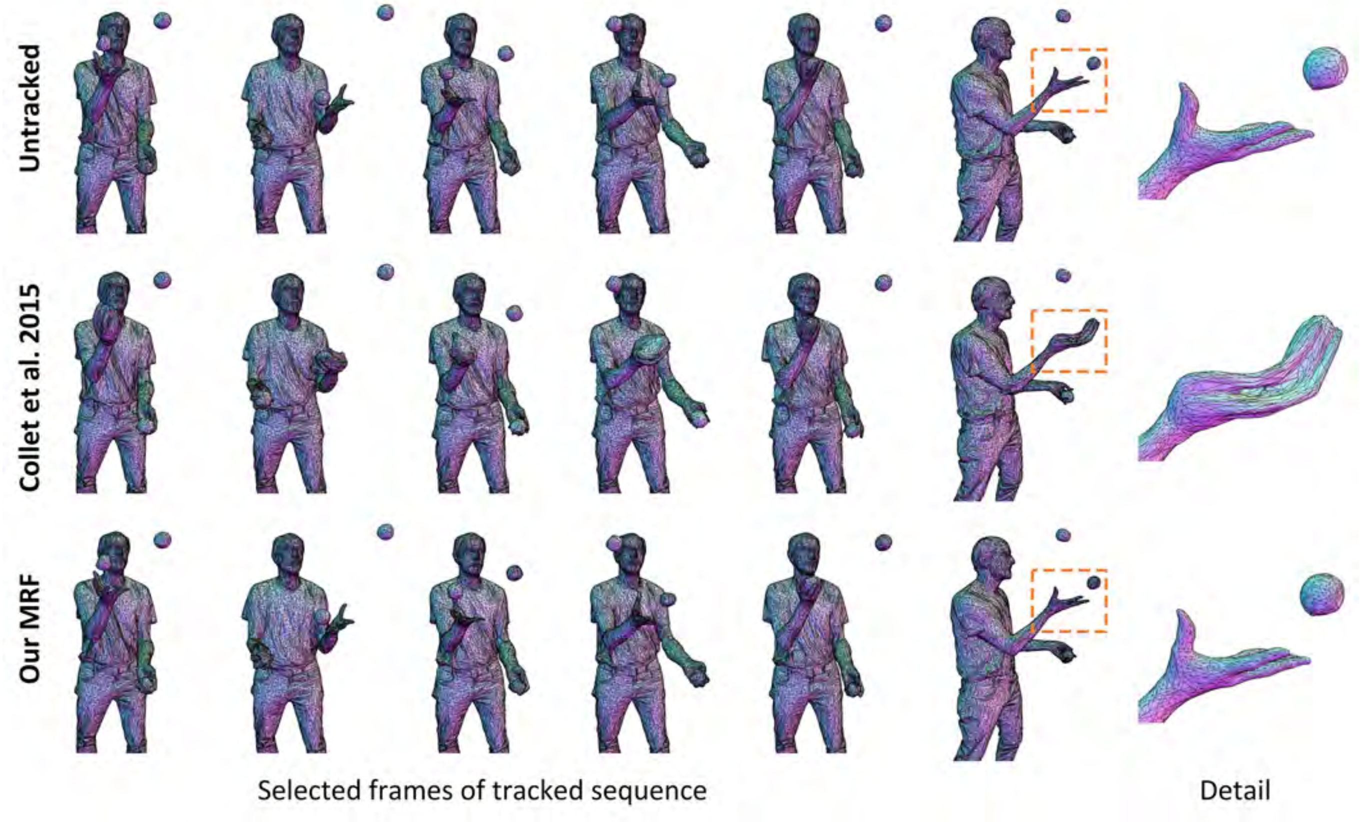


Fig. 13. A comparison of different mesh tracking strategies. Compared to Collet et al. [2015], our approach achieves the lowest error and selects fewer keyframes (25 vs. 41). See text for details.

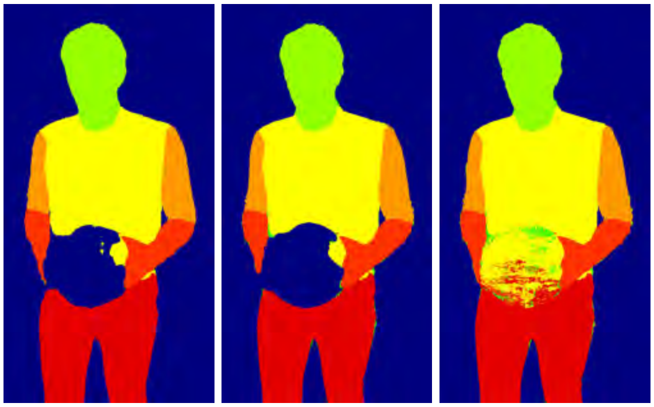
5.1 Depth Quality

Spatial Resolution. Following the Nyquist sampling theorem, it is easy to prove that the spatial surface detail we can reconstruct fundamentally depends on the image resolution. To assess the importance of this, we conducted an experiment where we run our sensors at 4 MP (*e.g.*, Collet et al. [2015]) and compare it with our 12.4 MP solution. In Figure 11, it is quite evident that low resolution sensors have higher level of depth noise and oversmooth the final geometry. In order to achieve the next level of realism, the proposed 12.4 MP cameras are able to capture most of the fine details in geometry including facial features and wrinkles in the shirt.

VGG Features. Our system demonstrates the importance of extracting deep learning features for the multi-view stereo. We conducted an experiment where we run multi-view stereo only on RGB images, without leveraging the active illumination. We invalidate pixels within a 7×7 image patch with a low variance and we do not perform any additional filtering. In one case, we use SAD matching cost with adaptive support weights, and we aggregate the results over a 7×7 patch, in the second case we use SAD on VGG features on a smaller 3×3 patch. VGG features are extracted using the first convolutional layer, resulting in images with 64 channels. Figure 12 shows that the proposed solution consistently reduces gross errors across the full image. Textureless regions, such as the performer's shirt and jeans, are still not perfect; perhaps a more global optimization scheme could be employed to improve this.

5.2 Segmentation

Unlike other volumetric capture systems, we do not rely on a green screen solution to detect the performer. Besides complicated environment setup, the other drawback of green screen segmentation is a baked-in lighting condition. Instead, we embed the prior using a deep learning technique [Chen et al. 2016] into a CRF model with fore-/background color and depth together. Because Chen et al. [2016] is trained to detect people, non-human objects may be misclassified in some view, *e.g.*, the basketball in Figure 14b. On the other hand, our proposed multi-view segmentation method can successfully label this region as foreground, as shown in Figure 14c. Note that the semantic labels assigned to the ball are not crucial: only a foreground mask is required to obtain accurate segmentation of objects. Finally, notice how the proposed CRF solution is able to better follow the edges of the high resolution image. Conversely, the output of the network is usually lower resolution, so it may miss important details such as hair or thin structures.



(a) Single View Segmentation without CRF. (b) Single View Segmentation. (c) Multi-View Segmentation.

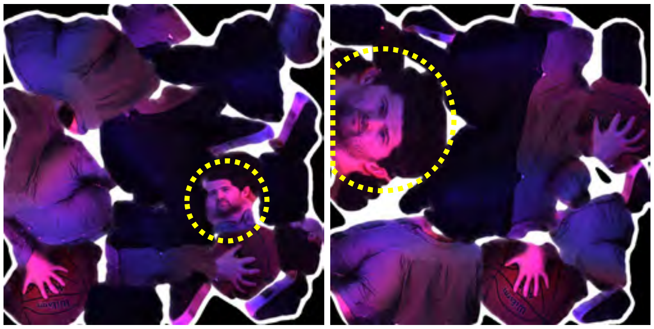
Fig. 14. Our multi-view segmentation is able to label the basketball in the hand of the performer while the single view segmentation cannot label it at all. Note that without CRF the edges are coarse due to the low resolution output of the neural network.

5.3 Optimal Mesh Tracking

The greedy tracking algorithm by Collet et al. [2015] relies on a heuristic score to search for the next keyframe in a priority queue. Even though the aligned surface has a small overall Hausdorff distance to the target frame, misalignment of local structures are inevitable. We present a challenging sequence where a performer is juggling three balls. We expect the tracker to be able to deal with topology changes every time the performer catches a ball or throws the next one in the air. We compared the proposed solution with our re-implementation of the algorithm proposed by Collet et al. [2015] and show the results in Figure 13. Note how our method selects better keyframes, leading to more pleasant reconstructed meshes. Our system also results in quantitative improvements. We result in producing fewer keyframes, only selecting 25 keyframes, as opposed to 41 by Collet et al. [2015]. The average alignment error computed as Hausdorff between tracked meshes in the sequence is 8 cm for Collet et al. [2015], whereas we achieve 3 cm, showing the effectiveness of the proposed MRF formulation.

5.4 UV Parameterization

The semantic weight in the mesh parameterization plays an important role to preserve high frequency details in the face. In Figure 15, we show the results of an experiment where, using UVAtlas, we assign the same importance to all the weights and compare it to one where we increase the weight on the areas belonging to the face of the performer three-fold. Note how the texture map generated when we use semantic information correctly allocates more pixels around the face and downgrading the priority of other components. Indeed, as shown in previous work [Meka et al. 2019; Orts-Escolano et al. 2016], human faces are the areas where artifacts are most noticeable.



(a) Regular Texture Atlas. (b) Texture Atlas with Semantic Allocation.

Fig. 15. Compared to the default atlas (a), using our semantic segmentation (b) for atlasing improves the allocation of texture for important feature like the face.

5.5 Mesh Decimation

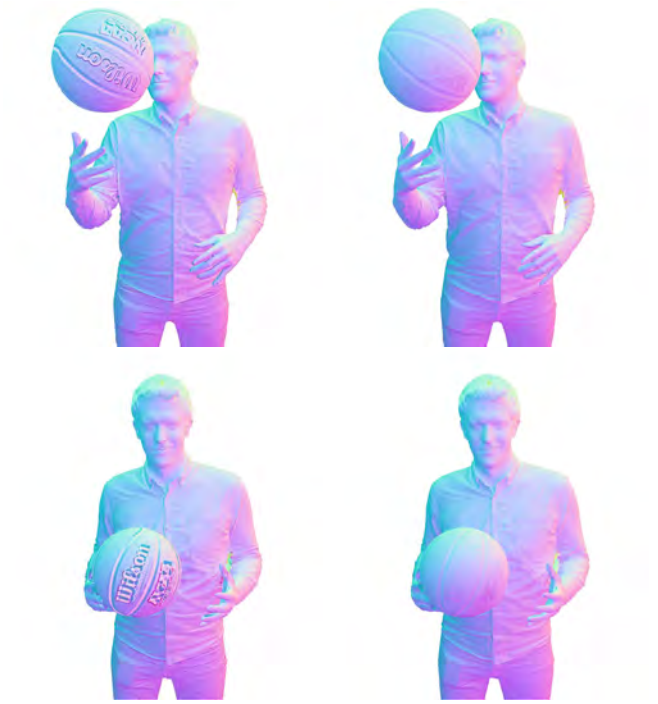
To evaluate the importance of the size of the mesh, we consider a sequence where we set the target decimation to 5k, 25k, and 100k vertices respectively and generate the reflectance maps as described in the previous section. Figure 17 shows a comparison of different decimated meshes using these decimation sizes. By looking at the base mesh (first row) and the photometric normals (second row), as we increase the number of triangles, more and more details start to appear. We argue that between 25k and 100k there is not significant improvement, only highlighting really small details such as facial pores and subtle wrinkles. Nonetheless, when combined with photometric normals, even the 5k decimation results in high quality details.

5.6 Texture Alignment

To evaluate the proposed strategies (*see* Section 3.3.1) for aligning complementary illumination conditions, we demonstrate the importance of an explicit texture alignment step in *image space* when computing the reflectance maps. The mesh tracking algorithm may not accurately track high frequency texture details when tangential motion in the geometry occurs; *e.g.*, a spinning ball. In Figure 16, we show a visual comparison when the alignment step in *image space* is turned on and off. In this case the mesh tracking algorithm fails to align the spinning ball, whereas the explicit texture alignment strategy we propose can compensate for the fast motion resulting in more accurate renderings.

5.7 Comparisons with State-of-the-art

In this section, we compare our system with two state of the art methods. First, we consider the algorithm proposed by Dou et al. [2017], which we ran using a voxel resolution of 2 mm to achieve the highest quality as reported in the original paper. Note that this method generates very accurate reconstructions, however it produces a “tracked” Signed Distance Function (SDF) sequence, which results in meshes that are not topologically consistent. Moreover the meshes obtained with this approach contain millions of vertices,



(a) Without texture alignment.

(b) With texture alignment.

Fig. 16. The texture alignment step substantially improves rendering artifacts in presence of fast tangential motion.

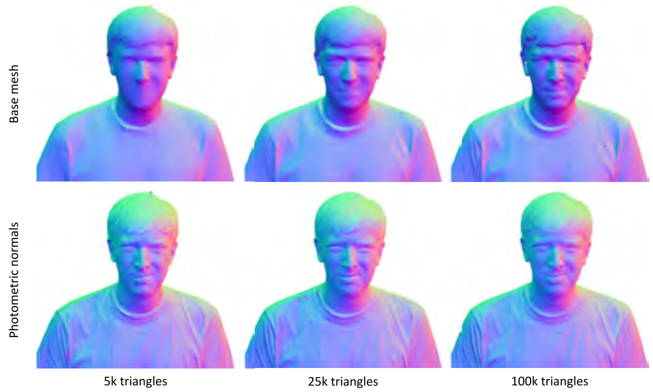


Fig. 17. A comparison of different decimated meshes (base mesh and photometric normals visualization) using 5k, 25k, and 100k triangles respectively.

limiting its application in practice. Nevertheless, we show a side by side comparison in Figure 18.

Note how, despite Dou et al. [2017] relying on a very fine-detailed reconstruction, our results still exhibit better high frequency details in the wrinkles of the shirt, the ball, and the face. Additionally, Dou et al. [2017] cannot interpolate missing geometrical parts, resulting in holes in some areas. This proves the effectiveness of the texture maps as a way to store a displacement from a coarser geometry, making the method more compelling for practical applications.

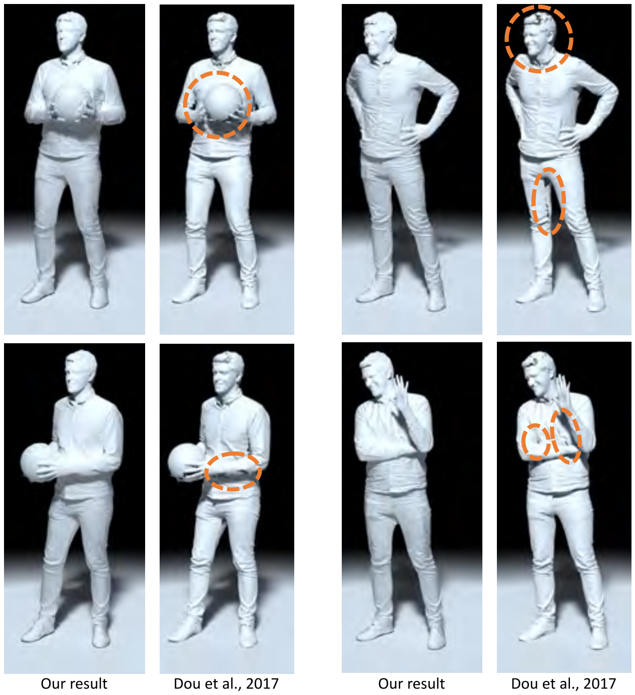


Fig. 18. A comparison of our reconstruction with Dou et al. [2017]. Note that our reconstructions are complete while holes (circled) remain with the approach of Dou et al. [2017].

To compare with the state-of-the-art of volumetric capture proposed by Collet et al. [2015], we reimplemented the majority of its components, except the segmentation algorithm, which relied on a green screen. Nevertheless, we used our proposed segmentation which produces high quality results and compared the final reconstructions in Figure 19. Note how the method by Collet et al. [2015] does an excellent job at generating convincing textured meshes; however, missing high frequency details are noticeable when compared with the proposed method. We push this technology to the next level of photo-realism, recovering fine level details such as facial hair, wrinkles and hair.

Finally, we show the importance of the reflectance maps for relighting purposes. In Figure 20 we compare our results with the ones obtained using geometry and diffuse relighting (e.g. such as in Collet et al. [2015]). Note how the proposed system renders fine details in a more realistic and pleasant way.

6 PHOTO-REALISTIC RENDERINGS

Photorealistic composition of virtual and scanned 3D models into photos or videos is a relevant technique in many areas such as virtual and augmented reality, visual effects and film production. A composition's realism depends on both geometric and lighting related factors. The system that we propose can be employed for various purposes that range from volumetric video playback to highly realistic portrait relighting. For example, given a High-Dynamic

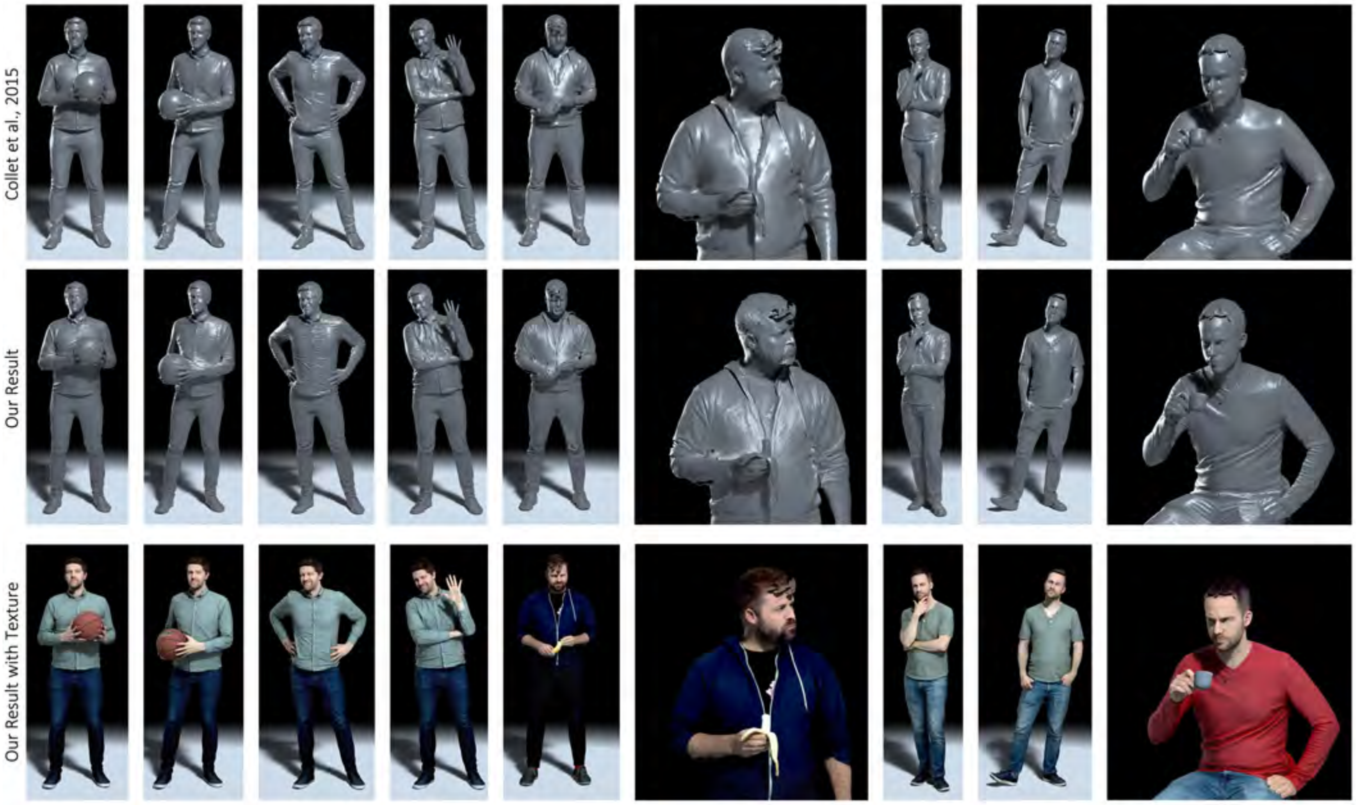


Fig. 19. A comparison of our reconstruction with our software re-implementation of Collet et al. [2015]. Note that our reconstructions exhibits more geometric detail due to higher resolution depth cameras and photometric stereo normal estimation.

Range Image (HDRI) of an environment, we can transport our high-resolution 3D models to photographs of real-world scenes which contain detailed lighting. Figure 21 shows various examples of different 3D models that have been transported to real-world scenes using captured HDRIs.

Using an HDRI provides a good approximation of the scene lighting, but it lacks 3D geometry, cannot render shadows correctly, and misses other light-related effects, such as occlusions. Motivated by this problem, we also used photorealistic, synthetic 3D scenes where the geometry of the environment is known. In this way, we can also properly model shadows and occlusions on scene surfaces. Figure 22 shows a few examples of these renderings using highly realistic synthetic scenes. We used various 3D scenes with different illumination conditions to show how our 3D models blend into the environment making the rendering very realistic, as if the person was recorded at those particular places.

Finally, we took one step further and also created realistic renderings of our models on real world images captured using a regular smartphone camera (*see* Figure 23). We took advantage of a recent learning-based method that estimates plausible HDR, omnidirectional illumination given an unconstrained, Low Dynamic Range (LDR) image from a smartphone camera [LeGendre et al. 2019]. Figure 23 shows multiple renderings of our 3D models on different real world scenes under multiple lighting conditions. Note that the 3D models are blended into the images in a convincing way, rendering consistent lighting and shadows as if the scanned humans were really there.

7 LIMITATIONS

Although our system brings us closer to photo-realistic volumetric videos through their accurate relightability in new scenes, there are limitations. For example, our system still struggles to reconstruct the geometry of thin structures such as hair despite our high-resolution input imagery. Although our reflectance maps somewhat compensate for this by adding high frequency details, we believe that machine learning methods may be the best way to address this. We also struggle with transparent and specular materials, for which we are unlikely to obtain a reconstruction. Figure 24 shows some examples of failures cases for the aforementioned problems: thin structures, such as hair; transparent surfaces, such as glasses; and finally, thin and highly specular surfaces, such as a golf club.

Popping artifacts may be visible when a new key frame is selected, although applying the reflectance maps makes this effect less noticeable at rendering time. Very fast tangential motion could cause wrong reflectance maps estimates even after an explicit texture alignment step: indeed the final quality depends on the accuracy of the

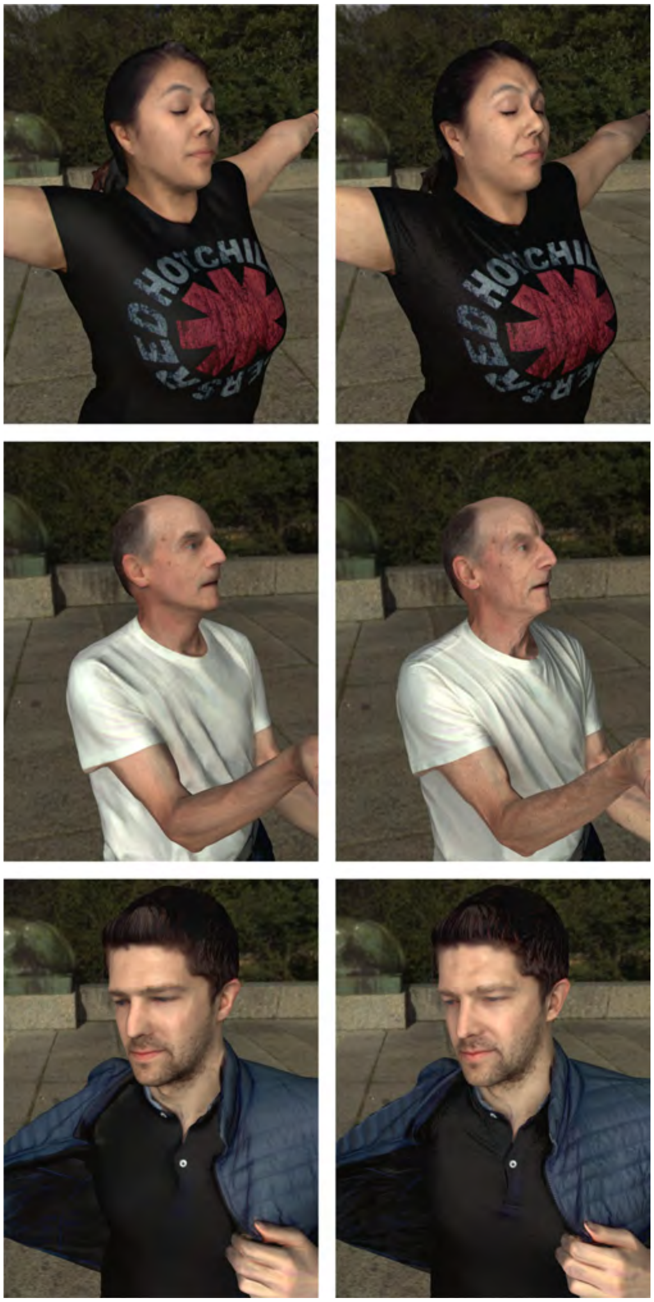


Fig. 20. Left: HDRI relighting of diffuse color and geometry such as in Collet et al. [2015]. Right: our solution using geometry, albedo, photometric normal, and material maps as input. Note the increased sharpness and amount of details with the proposed system.

optical flow. Regions with low SNR (*e.g.*, dark surfaces, hair), could lead to a poor estimate of the normal maps. This can be mitigated by tuning the lights and exposure time ad hoc for a given performer. Other imperfections are due to the spatial bias that increases with the distance from the center of the stage: this could be solved with a precomputed look-up-table.

8 DISCUSSION

In this paper, we presented our system for reconstructing *relightable* volumetric videos of humans. Through the combination of state of the art active illumination, novel high resolution depth sensors, and a high resolution camera array, our system is presented with a plethora of geometric, lighting, and appearance constraints. In order to consume these constraints, we designed a cloud-based reconstruction pipeline. This pipeline adapts state of the art geometric and machine learning methods for use in map reduce style parallelism. As a result of our ability to control the lighting conditions during capture, we are also able to derive reflectance maps. As such, we are able to derive volumetric videos of real humans that can be accurately relit in a new environment without any user intervention. Although this work makes significant progress towards photo-realisism, we leave it as future work to incorporate more complicated lighting models and machine learning methods.

ACKNOWLEDGEMENTS

The authors would like to thank Cynthia Herrera and Peter Denny for organizing and supporting the dataset captures, Damon Wheeler for his work on safety measurements, and Ryan Overbeck for his help during the design of the cloud-based processing system.

REFERENCES

Robert Anderson, David Gallup, Jonathan T Barron, Janne Kontkanen, Noah Snavely, Carlos Hernández, Sameer Agarwal, and Steven M Seitz. 2016. Jump: virtual reality video. *ACM TOG* (2016).

Guha Balakrishnan, Amy Zhao, Adrian V. Dalca, Frédo Durand, and John V. Guttag. 2018. Synthesizing Images of Humans in Unseen Poses. *CVPR* (2018).

Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. 2009. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM TOG* (2009).

Jonathan T. Barron and Jitendra Malik. 2015. Shape, Illumination, and Reflectance from Shading. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 8 (2015).

Thabo Beeler, Bernd Bickel, Paul Beardsley, Bob Sumner, and Markus Gross. 2010. High-quality Single-shot Capture of Facial Geometry. In *ACM SIGGRAPH 2010*.

Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig Gotsman, Robert W. Sumner, and Markus Gross. 2011. High-quality Passive Facial Performance Capture Using Anchor Frames. In *ACM SIGGRAPH 2011*.

Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *SIGGRAPH*, Vol. 99. 187–194.

Michael Bleyer, Christoph Rhemann, and Carsten Rother. 2011. PatchMatch Stereo-Stereo Matching with Slanted Support Windows.. In *Bmvc*, Vol. 11. 1–11.

Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A. Efros. 2018. Everybody Dance Now. *CoRR* (2018).

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2016. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *CoRR* (2016).

Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. 2015. High-quality Streamable Free-viewpoint Video. *ACM TOG* (2015).

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the Reflectance Field of a Human Face. In *SIGGRAPH*.

Paul Debevec, Yizhou Yu, and George Boshokov. 1998. Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping. In *Rendering Techniques*.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Mingsong Dou, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, and Shahram Izadi. 2017. Motion2Fusion: Real-time Volumetric Performance Capture. *SIGGRAPH Asia* (2017).

Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, Pushmeet Kohli, Vladimir Tankovich, and Shahram Izadi. 2016. Fusion4D: Real-time Performance Capture of Challenging Scenes. *SIGGRAPH* (2016).

Mingsong Dou, Jonathan Taylor, Henry Fuchs, Andrew Fitzgibbon, and Shahram Izadi. 2015. 3D scanning deformable objects with a single RGBD sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 493–501.

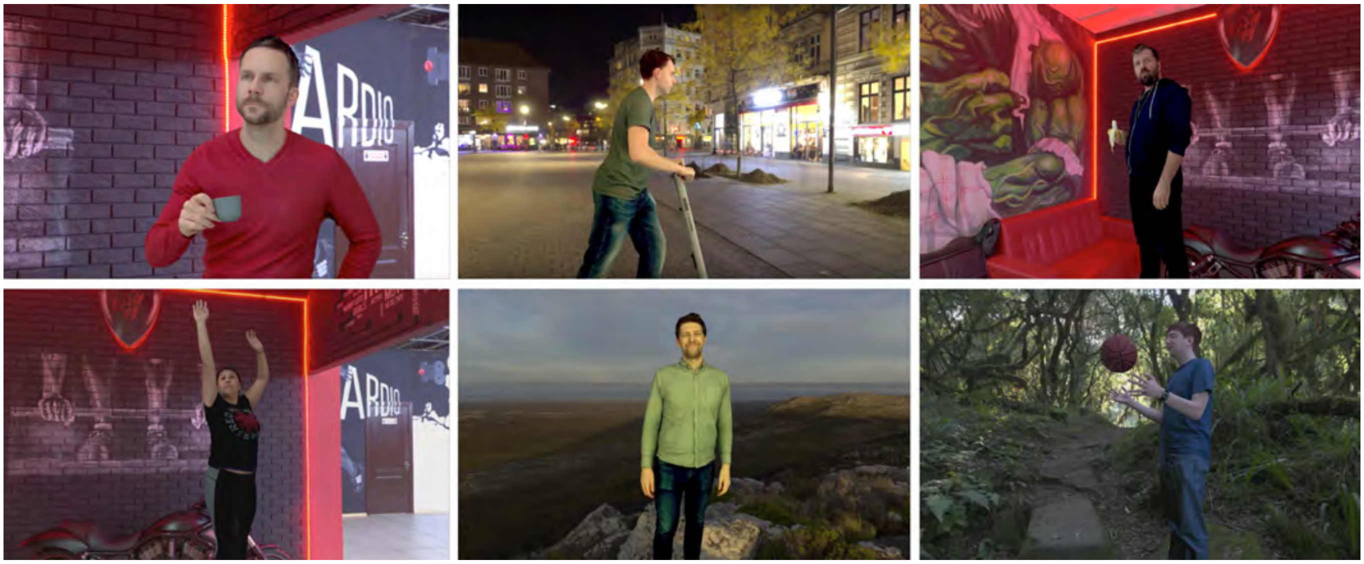


Fig. 21. Our models rendered on real world HDRI maps.

Ruofei Du, Ming Chuang, Wayne Chang, Hugues Hoppe, and Amitabh Varshney. 2019. Montage4D: Real-time Seamless Fusion and Stylization of Multiview Video Textures. *Journal of Computer Graphics Techniques* 8, 1 (17 January 2019).

Sean Ryan Fanello, Julien Valentin, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, Carlo Ciliberto, Philip Davidson, and Shahram Izadi. 2017a. Low Compute and Fully Parallel Computer Vision with HashMatch. In *ICCV*.

Sean Ryan Fanello, Julien Valentin, Christoph Rhemann, Adarsh Kowdle, Vladimir Tankovich, Philip Davidson, and Shahram Izadi. 2017b. UltraStereo: Efficient Learning-based Matching for Active Stereo Systems. In *CVPR*.

Graham Fyffe, Cyrus A. Wilson, and Paul Debevec. 2009. Cosine Lobe Based Relighting from Gradient Illumination Photographs. 100–108. https://doi.org/10.1109/CVMP.2009.18

Graham Fyffe, Tim Hawkins, Chris Watts, Wan-Chun Ma, and Paul Debevec. 2011. Comprehensive Facial Performance Capture. *Eurographics* (2011).

Silvano Galliani, Katrin Lasinger, and Konrad Schindler. 2015. Massively Parallel Multiview Stereopsis by Surface Normal Diffusion. *The IEEE International Conference on Computer Vision (ICCV)*.

Michael Garland and Paul S Heckbert. 1997. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 209–216.

Pablo Garrido, Levi Valgaert, Chenglei Wu, and Christian Theobalt. 2013. Reconstructing Detailed Dynamic Face Geometry from Monocular Video. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 32, 6, Article 158 (Nov. 2013), 10 pages.

Pablo Garrido, Michael Zollhoefer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Perez, and Christian Theobalt. 2016. Reconstruction of Personalized 3D Face Rigs from Monocular Video. (2016).

Paulo Gotardo, Jérémy Riviere, Derek Bradley, Abhijeet Ghosh, and Thabo Beeler. 2018. Practical Dynamic Facial Appearance Modeling and Acquisition. In *SIGGRAPH Asia*.

Kaiwen Guo, Jon Taylor, Sean Fanello, Andrea Tagliasacchi, Mingsong Dou, Philip Davidson, Adarsh Kowdle, and Shahram Izadi. 2018. TwinFusion: High Framerate Non-Rigid Fusion through Fast Correspondence Tracking. In *3DV*.

Igor Guskov and Zoë J Wood. 2001. Topological noise removal. *2001 Graphics Interface Proceedings: Ottawa, Canada* (2001), 19.

Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. 2015. Dynamic 3D Avatar Creation from Hand-held Video Input. *ACM Trans. Graph.* 34, 4, Article 45 (July 2015), 14 pages.

International Electrotechnical Commission 2014. *Safety of laser products – Part 1: Equipment classification and requirements* (3 ed.). International Electrotechnical Commission. IEC 60825-1:2014.

Michael Kazhdan and Hugues Hoppe. 2013. Screened Poisson Surface Reconstruction. *ACM TOG* (2013).

Adarsh Kowdle, Christoph Rhemann, Sean Fanello, Andrea Tagliasacchi, Jon Taylor, Philip Davidson, Mingsong Dou, Kaiwen Guo, Cem Keskin, Sameh Khamis, David Kim, Danhang Tang, Vladimir Tankovich, Julien Valentin, and Shahram Izadi. 2018. The Need 4 Speed in Real-Time Dense Visual Tracking. *SIGGRAPH Asia* (2018).

Philipp Krähenbühl and Vladlen Koltun. 2011. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *NIPS*.

Chloe LeGendre, Wan-Chun Ma, Graham Fyffe, John Flynn, Laurent Charbonnel, Jay Busch, and Paul E. Debevec. 2019. DeepLight: Learning Illumination for Unconstrained Mobile Mixed Reality. *CoRR* abs/1904.01175 (2019). arXiv:1904.01175 http://arxiv.org/abs/1904.01175

V. Lempitsky and D. Ivanov. 2007. Seamless Mosaicing of Image-Based Texture Maps. In *CVPR*.

Guannan Li, Chenglei Wu, Carsten Stoll, Yebin Liu, Kiran Varanasi, Qionghai Dai, and Christian Theobalt. 2013b. Capturing Relightable Human Performances under General Uncontrolled Illumination. *Computer Graphics Forum (Proc. EUROGRAPHICS 2013)* (2013).

Hao Li, Bart Adams, Leonidas J Guibas, and Mark Pauly. 2009. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics (ToG)* 28, 5 (2009), 175.

Hao Li, Etienne Vouga, Anton Gudym, Linjie Luo, Jonathan T Barron, and Gleb Gusev. 2013a. 3D self-portraits. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 187.

Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. 2018a. Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image. In *ECCV (Lecture Notes in Computer Science)*. Springer.

Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. 2018b. Learning to Reconstruct Shape and Spatially-varying Reflectance from a Single Image. In *SSIGGRAPH Asia*.

Peter C Lincoln. 2017. *Low Latency Displays for Augmented Reality*. Ph.D. Dissertation. The University of North Carolina at Chapel Hill.

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural Volumes: Learning Dynamic Renderable Volumes from Images. *SIGGRAPH* (2019).

Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. 2017. Pose guided person image generation. In *NIPS*.

Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. 2018. Disentangled Person Image Generation. *CVPR* (2018).

Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B Goldman, Cem Keskin, Steve Seitz, Shahram Izadi, and Sean Fanello. 2018. LookinGood: Enhancing Performance Capture with Real-time Neural Re-Rendering. In *SIGGRAPH Asia*.

Abhimitra Meka, Gereon Fox, Michael Zollhöfer, Christian Richardt, and Christian Theobalt. 2017. Live User-Guided Intrinsic Video For Static Scene. *IEEE Transactions on Visualization and Computer Graphics* 23, 11 (2017).

Abhimitra Meka, Christian Haene, Rohit Pandey, Michael Zollhoefer, Sean Fanello, Graham Fyffe, Adarsh Kowdle, Xueming Yu, Jay Busch, Jason Dourgarian, Peter Denny, Sofien Bouaziz, Peter Lincoln, Matt Whalen, Geoff Harvey, Jonathan Taylor, Shahram Izadi, Andrea Tagliasacchi, Paul Debevec, Christian Theobalt, Julien

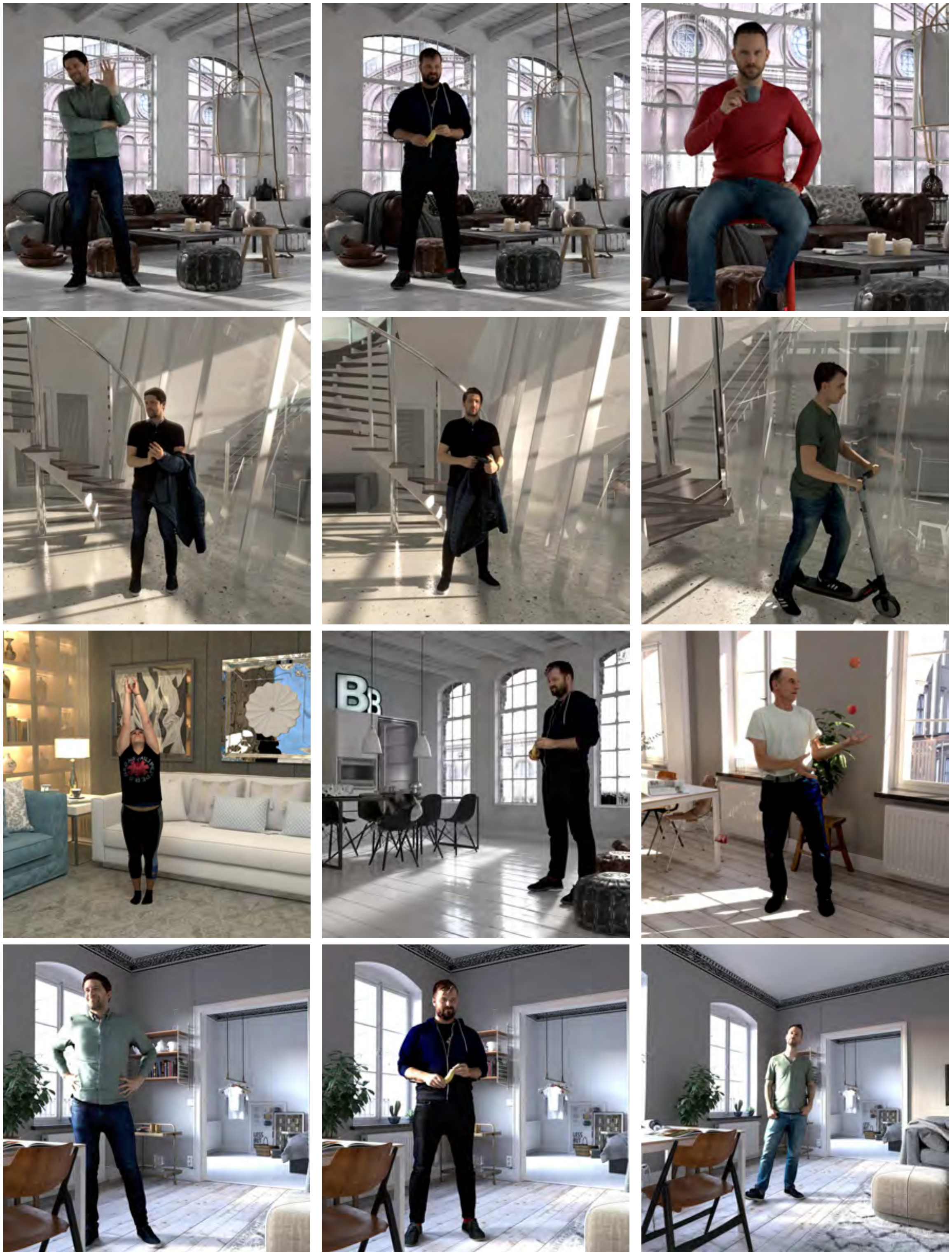


Fig. 22. Our system allows us to realistically integrate and relight our models in virtual 3D scenes.

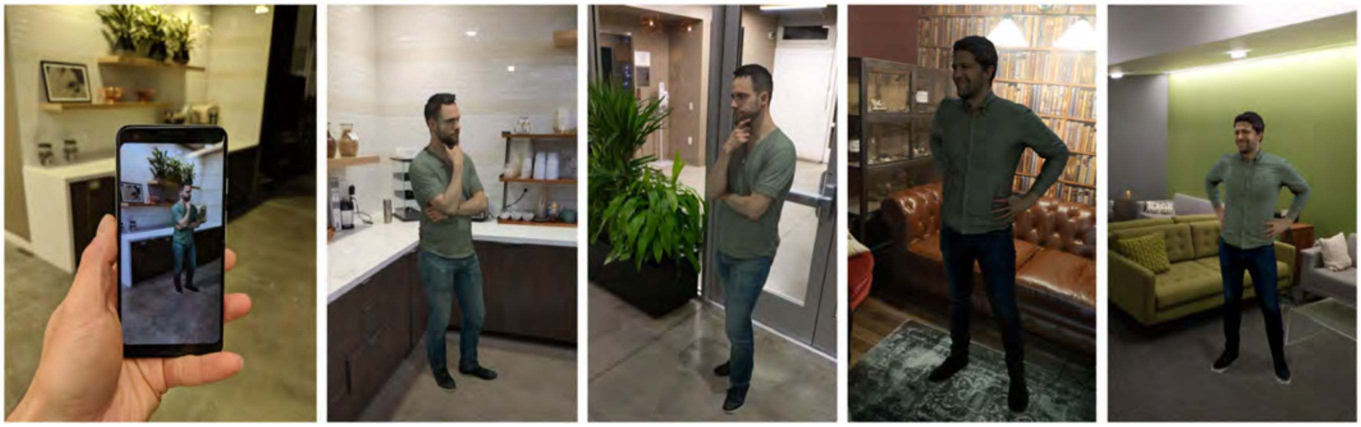


Fig. 23. Our models re-rendered on the smartphone with estimated lighting using [LeGendre et al. 2019].

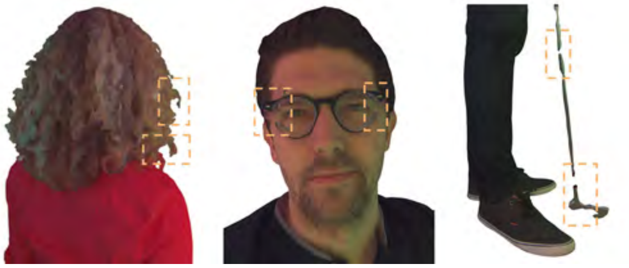


Fig. 24. Our system struggles to reconstruct the geometry of some thin structures (hair), and also transparent and highly specular surfaces.

Valentin, and Christoph Rhemann. 2019. Deep Reflectance Fields - High-Quality Facial Reflectance Field Inference From Color Gradient Illumination. *ACM Transactions on Graphics (Proceedings SIGGRAPH)*.

Microsoft. 2019. UVAtlas - isochart texture atlasing. (2019). http://github.com/Microsoft/UVAtlas

P. Mirdehghan, W. Chen, and K. N. Kutulakos. 2018. Optimal Structured Light a la Carte. In *CVPR*.

Giljoo Nam, Joo Ho Lee, Diego Gutierrez, and Min H. Kim. 2018. Practical SVBRDF Acquisition of 3D Objects with Unstructured Flash Photography. In *SIGGRAPH Asia*.

Natalia Neverova, Riza Alp Güler, and Iasonas Kokkinos. 2018. Dense Pose Transfer. *ECCV* (2018).

R. A. Newcombe, D. Fox, and S. M. Seitz. 2015. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*.

Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L. Davidson, Sameh Khamis, Mingsong Dou, Vladimir Tankovich, Charles Loop, Qin Cai, Philip A. Chou, Sarah Mennicken, Julien Valentin, Vivek Pradeep, Shenlong Wang, Sing Bing Kang, Pushmeet Kohli, Yuliya Lutchyn, Cem Keskin, and Shahram Izadi. 2016. Holoportation: Virtual 3D Teleportation in Real-time. In *UIST*.

Rohit Pandey, Anastasia Tkach, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Ricardo Martin-Brualla, Andrea Tagliasacchi, George Papandreou, Philip Davidson, Cem Keskin, Shahram Izadi, and Sean Fanello. 2019. Volumetric Capture of Humans with a Single RGBD Camera via Semi-Parametric Learning. In *CVPR*.

Pieter Peers, Tim Hawkins, and Paul E. Debevec. 2006. *A Reflective Light Stage*. Technical Report.

Fabián Prada, Misha Kazhdan, Ming Chuang, Alvaro Collet, and Hugues Hoppe. 2017. Spatiotemporal Atlas Parameterization for Evolving Meshes. *ACM TOG* (2017).

Shunsuke Saito, Lingyu Wei, Liwen Hu, Koki Nagano, and Hao Li. 2017. Photorealistic Facial Texture Inference Using Deep Neural Networks. In *CVPR*. IEEE Computer Society, 2326–2335.

Pedro V. Sander, Steven J. Gortler, John Snyder, and Hugues Hoppe. 2002. Signal-specialized Parametrization. In *Eurographics Workshop on Rendering*.

Daniel Scharstein and Richard Szeliski. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision* 47, 1-3 (2002), 7–42.

Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. 2016. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*.

Chenyang Si, Wei Wang, Liang Wang, and Tieniu Tan. 2018. Multistage Adversarial Losses for Pose-Based Human Image Synthesis. In *CVPR*.

K. Simonyan and A. Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

J. Starck and A. Hilton. 2007. Surface Capture for Performance-Based Animation. *IEEE Computer Graphics and Applications* (2007).

Robert W Sumner, Johannes Schmid, and Mark Pauly. 2007. Embedded deformation for shape manipulation. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 80.

Tiancheng Sun, Jonathan T. Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul Debevec, and Ravi Ramamoorthi. 2019. Single Image Portrait Relighting. *ACM Transactions on Graphics (Proceedings SIGGRAPH)*.

L. M. Tanco and A. Hilton. 2000. Realistic synthesis of novel human movements from a database of motion capture examples. In *Proceedings Workshop on Human Motion*.

Vladimir Tankovich, Michael Schoenberg, Sean Ryan Fanello, Adarsh Kowdle, Christoph Rhemann, Max Dzitsiuk, Mirko Schmidt, Julien Valentin, and Shahram Izadi. 2018. SOS: Stereo Matching in O(1) with Slanted Support Windows. *IROS* (2018).

Christian Theobalt, Naveed Ahmed, Hendrik P. A. Lensch, Marcus A. Magnor, and Hans-Peter Seidel. 2007. Seeing People in Different Light-Joint Shape, Motion, and Reflectance Capture. *IEEE TVCG* 13, 4 (2007), 663–674.

Justus Thies, Michael Zollhoefer, Marc Stamminger, Christian Theobalt, and Matthias Niessner. 2016. Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. In *Proc. CVPR*.

Zhen Wen, Zicheng Liu, and T. S. Huang. 2003. Face relighting with radiance environment maps. In *CVPR*.

Shuco Yamaguchi, Shunsuke Saito, Koki Nagano, Yajie Zhao, Weikai Chen, Kyle Olszewski, Shigeo Morishima, and Hao Li. 2018. High-fidelity Facial Reflectance and Geometry Inference from an Unconstrained Image. *ACM Trans. Graph.* 37, 4, Article 162 (July 2018).

Jure Žbontar and Yann LeCun. 2016. Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. *Journal of Machine Learning Research* 17, 65 (2016), 1–32. http://jmlr.org/papers/v17/15-535.html

Zhengyou Zhang. 2000. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 11 (Nov. 2000), 1330–1334. https://doi.org/10.1109/34.888718

Bo Zhao, Xiao Wu, Zhi-Qi Cheng, Hao Liu, and Jiashi Feng. 2017. Multi-View Image Generation from a Single-View. *CoRR* (2017).

Kun Zhou, John Synder, Baining Guo, and Heung-Yeung Shum. 2004. Iso-charts: Stretch-driven Mesh Parameterization Using Spectral Analysis. In *Eurographics*.

Kun Zhou, Xi Wang, Yiying Tong, Mathieu Desbrun, Baining Guo, and Heung-Yeung Shum. 2005. TextureMontage. *ACM TOG* (2005).

Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, and Marc Stamminger. 2014. Real-time Non-rigid Reconstruction using an RGB-D Camera. *ACM TOG* (2014).