Dynamic Shape Capture using Multi-View Photometric Stereo

Daniel Vlasic[1] Pieter Peers[2] Ilya Baran[1] Paul Debevec[2]
Jovan Popović[1 3 4] Szymon Rusinkiewicz[3 5] Wojciech Matusik[3]

CSAIL, Massachusetts Institute of Technology[1] USC, Institute for Creative Technologies[2]
Adobe Systems, Inc.[3] University of Washington[4] Princeton University[5]

Figure 1: *Our system rapidly acquires images under varying illumination in order to compute photometric normals from multiple viewpoints. The normals are then used to reconstruct detailed mesh sequences of dynamic shapes such as human performers.*

Abstract

We describe a system for high-resolution capture of moving 3D geometry, beginning with dynamic normal maps from multiple views. The normal maps are captured using active shape-from-shading (photometric stereo), with a large lighting dome providing a series of novel spherical lighting configurations. To compensate for low-frequency deformation, we perform multi-view matching and thin-plate spline deformation on the initial surfaces obtained by integrating the normal maps. Next, the corrected meshes are merged into a single mesh using a volumetric method. The final output is a set of meshes, which were impossible to produce with previous methods. The meshes exhibit details on the order of a few millimeters, and represent the performance over human-size working volumes at a temporal resolution of 60Hz.

1 Introduction

In a variety of industries, ranging from entertainment to manufacturing to medicine, it is necessary to acquire representations of human motion. Even coarse "motion capture," in which only joint angles are obtained, has improved the safety of our vehicles, enhanced our understanding of the human body, and hinted at new forms of human-computer interaction. However, the detailed acquisition of the moving geometry and appearance of people and their clothes will revolutionize the entertainment industry, with digital actors becoming a staple in video games and films.

Advances in moving-surface capture have been based on continuous progress on several fronts, ranging from the development of different algorithms to the design of more sophisticated hardware. However, it remains an open challenge to capture and reproduce detailed geometry (spatial resolution on the order of a few millimeters) at video rates, for full-body performances seen from 360°. Existing techniques each address only a subset of the problems encountered in high-resolution dynamic geometry scanning and do not scale well. On the hardware side, the challenge is to capture enough well-lit data sufficiently fast and without blinding the actor. The captured data streams can easily be on the order of several gigabytes per second, posing a challenge to efficiently process it all at high quality within an acceptable time-frame. For example, binocular and multi-view stereo offer convenience and dynamic capture, although they provide detailed data only in high-texture regions. Active 3D scanning based on structured light provides high-quality static meshes, but does not scale well to moving scenes or large working volumes. Photometric stereo (active shape from shading) offers high-quality geometric detail in the form of normal maps, but it remains difficult to combine the normal maps from multiple views. Template-based regularization may be applied to any of these methods, often with visually-compelling results, but the ultimate quality depends on the similarity of the actual surface to the template — it is difficult to obtain accurate templates for the many degrees of freedom of clothing, for example.

In this work, we describe a *practical* system for capturing highly detailed 3D geometry of an actor's performance at sixty frames per second. We begin by acquiring normal maps from a small number of views (i.e., 8 or 9), using a novel variant of photometric stereo based on a small set of view-independent time-multiplexed light patterns produced by a large lighting dome (Section 3.1). In contrast to many existing systems, we use only four spherical lighting patterns (Section 3.2) to obtain a frame of geometry, which offers numerous practical advantages in acquiring detailed geometrry of full-body performances. The spherical nature of the lighting patterns ensures that each camera, irrespective of its relative location, gains photometric information at each frame. This makes the patterns particularly well-suited for capturing performances from all viewpoints. Furthermore, the large extent of the lighting patterns improves the robustness of the light/camera calibration, as well as allowing the irradiance to be better distributed than with bright point-light sources, and improving actor comfort.

Our reconstruction pipeline is split into several stages to make the large amount of data more manageable. Our algorithm processes the multi-view normal maps together with the corresponding silhouettes to produce high-resolution meshes independently

for each time frame. We first integrate each normal map to obtain an initial surface per view, and estimate the locations of depth discontinuities (Section 4.1). The visual hull, obtained as the intersection of silhouettes, provides constraints on the integration as well as an initial rough estimate for depth discontinuities. Next, we match the partial surfaces and deform them to improve their mutual fit (Section 4.2). We evaluate several matching metrics, including ones based on local shape rather than color. Finally, we use volumetric merging to combine the separate views into a single surface (Section 4.3). We use this surface as an improved proxy replacing the visual hull in a second pass. The final meshes are computed independently between time frames, and recover the majority of a dynamic shape with high quality at sixty frames per second.

Compared to existing systems, our work represents an advance in the combination of high frame rate (60 Hz.), large working volume (human-size), and high spatial resolution (millimeter-scale) necessary for practical, high-quality performance capture. To achieve this, we make contributions in both the capture and reconstruction stages of our pipeline. First, we obtain high-quality high-frame rate normal maps using photometric stereo with spherical lighting. This photometric configuration is more comfortable to the subject, and is easier to construct, calibrate, and time-multiplex than other active illumination schemes. Second, we present a pipeline partitioned into stages that avoids expensive data-intensive optimization strategies on which some recent methods rely. This is more efficient in storage and computation, is trivially parallelizable, and scales well for large data volumes. Third, we demonstrate that matching neighboring views using a surface-based metric yields better and more robust correspondences than image-based approaches, for a sparse view sampling with a wide baseline such as ours.

We anticipate that data from our system will have immediate impact on several problems within geometry processing and physical animation. First, because of our high resolution and our ability to capture surfaces without color detail, we obtain data suitable for analyzing and validating physical simulation models for materials such as cloth. Second, our data is ideally suited as input to algorithms that perform temporal registration and merging of geometry, which have recently received considerable interest in the geometry processing community [Wand et al. 2007; Huang et al. 2008; Li et al. 2008; Sharf et al. 2008; Zhang et al. 2008; Wand et al. 2009]. Looking ahead, we believe that it will be the combination of capture systems such as ours and temporal processing algorithms that will enable detailed full-body performance capture, resulting in even more believable virtual humans in movies and games.

2 Previous Work

Our methods are related to work from the following four research areas: (1) multi-view stereo and volumetric carving, (2) real-time structured light, (3) template-based approaches for performance capture, and (4) multi-view photometric stereo. We will describe the most relevant work in each of these categories.

Multi-view (Wide-baseline) Stereo and Volumetric Carving Multi-view stereo and volumetric space carving approaches use multiple, sparsely spaced cameras to observe a scene from different viewpoints. The color information is employed to carve out the space until the scene is photo-consistent [Seitz and Dyer 1999] or the color information is matched along multiple epipolar lines [Okutomi and Kanade 1993]. Recent approaches additionally employ global optimization to take into account smoothness constraints as well as the silhouettes of the object [Hernandez and Schmitt 2004; Vogiatzis et al. 2005; Furukawa and Ponce 2006; Hornung and Kobbelt 2006]. A comprehensive evaluation of many approaches has been conducted by Seitz and colleagues [2006].

Several characteristics of these methods make it difficult to apply them to high-quality dynamic motion capture. In particular, their performance is highest with dense texture and many camera viewpoints. However, for dynamic capture the number of camera locations is often limited (as opposed to static capture, for which a single camera may be moved to many locations), and the viewpoints must contain separate physical cameras (whose geometric and photometric properties must be calibrated). Although there have been systems that use more than 50 cameras [Rander et al. 1997], typical systems for dynamic scene acquisition [Starck and Hilton 2007] might use only 8 cameras. As a result, reconstruction quality for moving surfaces has typically been lower than for static objects.

Real-time Structured Light Several methods capture dynamic facial performance geometry using structured illumination, usually emitted from a video projector onto the subject, and observed from a single viewpoint [Rusinkiewicz et al. 2002; Davis et al. 2005; Zhang et al. 2004; Zhang and Huang 2006]. While these systems can achieve impressive results for small (e.g., head size) working volumes, extending them to the two-meter working volumes required for full-body performance capture and to multi-view point capture is difficult due to numerous technical limitations of video projectors, such as limited spatial resolution, limited depth of field, and diminishing light levels with increasing working volume.

Template-based Approaches for Performance Capture Geometric templates (3D models of the subjects) can be used to aid in performance capture, for example by deforming them to match (possibly sparse) multiview data, or using them for hole-filling and parameterization. Caranza et al. [2003] use a generic template and deform it using silhouette data from different view points. Theobalt et al. [2007] further estimate surface reflectance and dynamic normal maps. Corazza et al. [2006] also use a generic template, but deform it to the visual hull. Bradley et al. [2008] combine multiview stereo and a template to obtain moving garments. Starck and Hilton [2003] employ silhouettes, stereo, and feature cues to refine a generic humanoid model. Balan and colleagues [2007] use silhouettes from multiple viewpoints to estimate parameters of SCAPE [Anguelov et al. 2005]—a low-dimensional geometric body model derived from a large collection of static 3D scans. Zhang and colleagues [2004] capture facial performances using a multi-camera and projector system. The resulting 3D geometry is regularized with a 3D face template. Most recently, some approaches have used high-quality, person-specific static 3D scans as templates [de Aguiar et al. 2007; Vlasic et al. 2008; de Aguiar et al. 2008]. These are deformed by tracking points on the surface of the object or by using silhouettes and photometric constraints.

The main advantage of template-based methods is that the prior model provides the correct connectivity and topology for the output mesh sequence. This minimizes major artifacts, while also reducing the search space and consequently running time. Furthermore, temporal correspondence is automatically provided since the meshes for all frames share the same parameterization. Therefore, missing data can be interpolated from other frames. However, template-based approaches have significant disadvantages. The quality of the output is low when generic 3D templates are used. The templates do not capture many person-specific details. Using high-quality, person-specific templates requires using an additional 3D range scanner (e.g., Cyberware). Furthermore, deformation details (such as cloth folds) can be baked into the template, and it is difficult to modify or remove them as the corresponding geometric features appear and disappear during the performance. Finally, template-based approaches cannot deal with atypical geometry and props often used during performances.

Multi-view Photometric Stereo The work on multi-view photometric stereo is the most similar to ours. Bernardini et al. [2002] combine 3D range data with multiple normal maps acquired from different viewpoints. These normal maps are used directly during rendering and are not fused with the range data. Nehab and col-

leagues [2005] fuse range data with a single normal map obtained using photometric stereo. The resulting geometry preserves high-frequency details from the normal maps while taking on the overall shape (i.e., low frequencies) of the range data. Nevertheless, this method still requires initial 3D geometry of reasonable quality, and does not address how to combine data from multiple normal maps.

Ahmed et al. [2008] use a template and silhouettes to estimate the large-scale geometry of a performance. Geometric details are added by estimating normals, simultaneously with reflectance properties. In contrast, we do not require geometric templates or reflectance estimation. Campbell et al. [2007] jointly incorporate data captured from multiple views to reconstruct a single shape using a volumetric approach. While elegant in theory, these approaches scale poorly to larger datasets. For example, to obtain millimeter precision in a two meter working volume requires a 2000^3 volumetric grid, which easily occupies several gigabytes of memory. Furthermore, solving a large volumetric optimization rapidly becomes computationally intensive.

Lim et al. [2005] use a sparse set of 3D features to construct a rough depth-map. Using this depth-map, normal directions are computed for each depth-map location. While the quality of the results is high, the method relies on the existence of detectable features, which are not always readily available. Joshi and Kriegman [2007] employ a graph-cut method to find dense correspondences based on a multi-view matching cost function that combines multi-view and photometric stereo. A depth-map from these dense correspondences is fused with photometric normals to yield a high quality shape using a non-iterative method. Unlike the method presented here, this method requires having a large number of views (at least 3) of each surface point, and is limited to 2.5D shapes.

The most similar methods to ours are those by Vogiatzis et al. [2006], and Hernandez et al. [2008]. These methods combine shading and silhouettes from different viewpoints, acquired with a turntable, to derive a 3D geometric model. In contrast, our approach captures dynamic scenes and requires fewer views (8 to 9 as opposed to 36). The method of Vogiatzis et al. performs a local search (essentially gradient descent) to establish correspondences between views. While this is appropriate if adjacent viewpoints are spatially nearby, it can fail to converge to the right correspondence if cameras are 45 degrees apart, as in our system. Furthermore, these methods rely on accurate silhouettes, which can be difficult to obtain in multi-camera setups with active illumination. Finally, our approach explicitly deals with surface discontinuities, which is necessary to obtain correct surfaces for nontrivial performances.

Summary We argue that practical capture of dynamic geometry requires a method that:

- reconstructs full 3D geometry (as compared to 2.5D depth maps) using a small number of views.
- does not use templates, and can handle arbitrary topology.
- remains computationally tractable for high working-volume-to-resolution ratios (several thousand to one) and high frame rates.
- handles fast and complex motion.

The tradeoffs and design decisions made in designing our system are based specifically on satisfying *all* of these requirements.

3 Hardware System and Image Processing

In this section we will discuss our hardware system, and the necessary image processing to compute high-quality normal maps for multi-view dynamic performance capture. We start by describing the hardware setup and calibration in Section 3.1. Next, in Section 3.2 we introduce a novel photometric normal estimation algorithm and its corresponding active illumination conditions. We conclude this section by a detailed discussion of additional optimization strategies and implementation details.

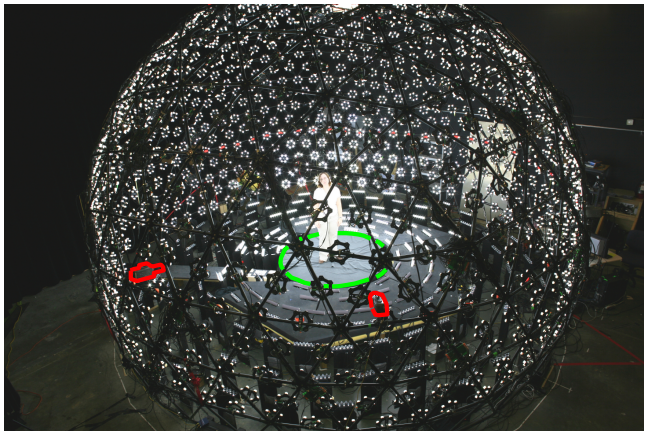


Figure 2: *Our acquisition setup consists of 1200 individually controllable light sources. Eight cameras (of which two are marked in red) are placed around the setup aimed at the performance area (marked in green). An additional ninth camera looks down from the top of the dome onto the performance area.*

3.1 Acquisition Setup and Calibration

Setup We employ a variant of photometric stereo to compute per-camera and per-pixel normal information. This requires an active illumination setup, for which we use the system built by Einarsson and colleagues [2006]. This lighting device consists of the top two-thirds of an 8-meter, 6th-frequency geodesic sphere with 1,200 regularly-spaced individually controllable light sources, of which 901 are on the sphere and the rest are placed on the floor. A central area is reserved for the subject. We capture dynamic performances at a 1024×1024 resolution with eight Vision Research V5.1 cameras. The cameras are placed on the sphere around the subject, at an approximate height of 1.7 meters relative to the central performance area. An optional ninth camera looks down onto the performer from the top of the dome. The performances are captured at a constant rate of 240fps, and the geometry is acquired at an effective rate of 60fps. Figure 2 shows our capture setup, with two selected cameras marked in red and the performance area marked in green.

Calibration Our system requires geometric and photometric calibration of all cameras. We use the LED waving technique of Svoboda et al. [2005] in order to calibrate the intrinsic and extrinsic camera parameters. We photometrically calibrate the cameras by capturing a Macbeth Color Checker under uniform illumination and then solve for the optimal color transfer matrix for each camera.

Silhouettes Our geometry processing algorithms require silhouettes and corresponding visual hulls of the subject in order to provide an initial guess for the surface reconstruction. We use a combination of background subtraction and chroma-keying to automatically extract approximate silhouettes. Though higher quality could be obtained with user assistance, this would be impractical (because so many frames need to be processed) and also unnecessary, since the resulting visual hulls are only used as a rough guide in the initial phase of the geometry reconstruction. However, it is important that the silhouettes not contain spurious holes, so small gaps in the foreground are detected and filled by comparing color statistics (i.e., average and variance) inside and outside the hole.

3.2 Multi-view Photometric Normals

Illumination Design Simultaneously acquiring images for computing photometric normals from multiple viewpoints imposes specific conditions on the design of active illumination patterns. First, capturing data-streams from multiple cameras produces a huge amount of data. Our main objective is to minimize the number

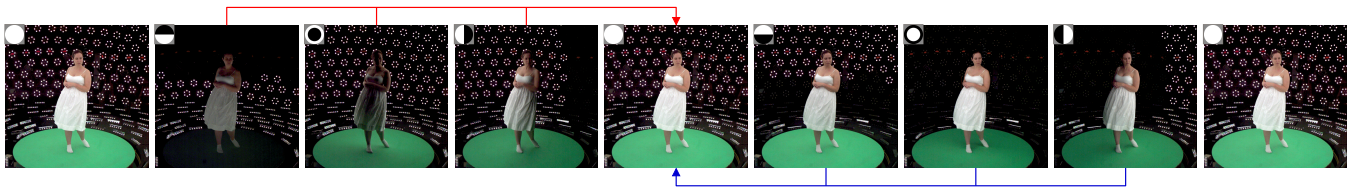


Figure 3: *Captured frames under binary half-on illumination patterns. A complete set plus an additional full-on condition is shown. The insets depict the illumination condition used in each frame. The red and blue arrows indicate the forward and backward motion compensation respectively. High-quality geometry is reconstructed for every full-on tracking frame.*

of required lighting conditions, and thus the number of captured frames. This allows us to maximize an effective frame rate of our capture system and enable using non-specialized (i.e., high-speed) cameras at sufficiently high resolutions. Second, longer exposure times cause motion blur which degrades the quality of the normal estimation. Therefore, to minimize motion blur, we need to minimize camera exposure, and consequently maximize the light levels on the subject. Large area light sources make it easier to maintain a high total light intensity, while remaining comfortable for the subject. Third, to obtain high quality normal estimates, we would like to maximize the signal-to-noise ratio of our measurements.

While conventional photometric stereo [Woodham 1978] is able to estimate normals in a wide range of applications, it has the disadvantage that it only uses a single light source at a time to illuminate the subject. This can result in a low signal-to-noise ratio (i.e., many pixels with low intensity values). Furthermore, a significant number of light sources needs to be placed around the subject in order to obtain a sufficient number of unoccluded directional samples for each pixel in each camera. Recently, Ma et al. [2007] proposed to use spherical gradient illumination to compute per-pixel normal information. The spherical gradient patterns cover the full sphere of incident lighting directions. They are well suited for multi-camera systems and result in a better signal-to-noise ratio. However, these patterns require careful calibration of the emitted illumination conditions such that they exactly conform to the theoretical gradients (both geometrically and radiometrically).

Inspired by [Ma et al. 2007], we propose a novel set of binary half-on illumination patterns that cover half of the sphere of incident directions. They are tailored to efficiently compute photometric normals for multi-view performance captures. Specifically, we employ two sets of 3 illumination patterns. The first set consists of three patterns $\mathbf{X}, \mathbf{Y}$, and $\mathbf{Z}$ defined by

$$\mathbf{X}(x,y,z) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

with $x^2 + y^2 + z^2 = 1$. $\mathbf{Y}$ and $\mathbf{Z}$ are similarly defined. The second set consists of the complements $\bar{\mathbf{X}}, \bar{\mathbf{Y}}$, and $\bar{\mathbf{Z}}$ of the first set, i.e. $\bar{\mathbf{X}}(x,y,z) = 1 - \mathbf{X}(x,y,z)$. We also add a full-on tracking frame $\mathbf{F}$ once every fourth frame in order to improve the temporal alignment and to compensate for motion over the multiple lighting conditions. To summarize, we illuminate the subject repeatedly with the following eight illumination patterns: $[\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{F}, \bar{\mathbf{X}}, \bar{\mathbf{Y}}, \bar{\mathbf{Z}}, \mathbf{F}]$. Figure 3 shows a subject under these illumination conditions.

Normal Estimation Let us initially assume that there is no subject motion during eight consecutive frames; furthermore, let's assume that all surfaces are diffuse. We can reconstruct a normal for each surface point based on the observed radiance under eight lighting conditions. This requires solving a system with three unknowns: the normal direction (2 unknowns), and surface albedo (1 unknown). While it is possible to compute an analytical solution for this system, the solution will be dependent on how accurately the physically emitted illumination conditions match the assumptions (in intensity and geometrical configuration).

Instead, we use a data-driven approach that improves robustness and facilitates calibration. By capturing a known shape with a known BRDF during a calibration step, we can establish a relationship between the observed radiance and normal direction. In our case we use a grey diffuse sphere, and treat the conversion from observed radiance to normal direction as a multi-dimensional lookup problem where the key is defined as $\frac{\mathbf{k}}{||\mathbf{k}||}$, with $\mathbf{k} = [I_{\mathbf{X}} - I_{\bar{\mathbf{X}}}, I_{\mathbf{Y}} - I_{\bar{\mathbf{Y}}}, I_{\mathbf{Z}} - I_{\bar{\mathbf{Z}}}]$, and I_p is the observed radiance under illumination $p \in \{\mathbf{X}, \bar{\mathbf{X}}, \mathbf{Y}, \bar{\mathbf{Y}}, \mathbf{Z}, \bar{\mathbf{Z}}\}$. Normalizing the lookup key removes any dependence of surface albedo from the key.

During calibration we capture a grey diffuse sphere under the binary half-on illumination conditions. For each camera view we extract the sphere's pixels, and create a vector similar to $\mathbf{k}$. We then store these vectors in a kD-tree together with their respective normals. When estimating the normals of a performance frame, we create a similar normalized vector for each camera pixel, search for the best match in the kD-tree, and retrieve the associated normal. In order to further improve the quality and minimize the effects of measurement noise during calibration, we search for the N best matches, and compute the output normal $\mathbf{n}$ as the weighted sum:

$$\mathbf{n} = \sum_{i}^{N} \left(\max_{j \le N} ||\mathbf{k} - \mathbf{k}_j|| - ||\mathbf{k} - \mathbf{k}_i|| \right) \mathbf{n}_i, \tag{2}$$

and renormalize it.

Our normal estimation algorithm has a number of advantages. First, the illumination conditions are binary and therefore they are easier to create in practice. Second, the number and positioning of the cameras is independent of the number and orientation of the lighting conditions. For any possible camera location, all lighting conditions yield sufficient information to compute photometric normals. Finally, this procedure requires very little calibration: a single photograph per camera, per lighting condition of a calibration object with known geometry. The calibration and normal computation is robust to modest variations in light source intensities and light source distribution. Furthermore, the computed photometric normals are in camera space, and thus *independent* of any (multi-view) camera calibration, further improving the robustness of the calibration. In light of the necessary complexity of our setup, this data-driven approach with easily calibratable sub-parts makes the whole acquisition process better manageable.

The presented data-driven method shares some similarities with [Hertzmann and Seitz 2005], where an object of known geometry is used to assist in determining photometric normals. The main difference is that they assume a known (point source) lighting configuration and unknown BRDF, while we assume a Lambertian BRDF and an unknown lighting configuration.

3.3 Implementation

In this section we discuss additional implementation details that are necessary in order to compute high-quality normal maps.

Multiple Calibration Spheres In the previous section, we have assumed that the incident illumination generated by a given illumination pattern is the same in the whole performance area (i.e., the lights sources are at infinity). However, in the current system

this assumption does not hold. For example, the lower third of the light-sphere is placed much closer to the subject. This creates a significantly different illumination depending on the distance to the floor. In order to compensate for this effect, we use multiple normal lookup tables, which depend on the position in the performance volume. We capture images of the grey spheres at 7 different heights. During normal estimation, we compute an output normal by linearly interpolating between normals computed from the two closest in height calibration spheres.

Motion Compensation In the previous section we have also assumed that the subject does not move during the capture of eight illumination conditions required to compute normal maps. In practice, this assumption also does not hold. In order to compensate for subject motion, we compute both forward and backward optical flow between consecutive tracking frames. By assuming a linear motion between full-on tracking frames, we flow the intermediate images under the binary illumination conditions to the central tracking frame. A normal map is then computed for every tracking frame. In our implementation we use the variational approach by Brox et al. [2004] to compute the optical flow. We show the direction of the optical flow to a single key frame in Figure 3. The forward and backward flows are illustrated using the red and blue arrows, respectively. Note that because we have a tracking frame every 4 frames, and the two sets of illumination conditions are complementary to each other, we can compute 2 sets of normal maps per 8 frame cycle.

Optical flow is able to correct for most of the subject's motions. However, flow computation can fail near or at occlusion boundaries. Therefore, we estimate the confidence for both forward and backward flow. The flow confidence is computed for each pixel as the L^2 error of the difference between the tracking frame and the flowed neighboring tracking frame. If this error is below some threshold, we compute the normal as detailed before. Otherwise, the normal is not computed and marked as invalid. When reconstructing the final geometry, we rely on normal estimates from different viewpoints and hole filling to correct for invalid marked normals.

Impact of Low Albedo Another factor that can negatively impact the quality of the estimated photometric normals is the low albedo of the surface points. In particular, camera noise dominates when imaging surface points with low albedo. In this case, the estimated photometric normals become noisy. Similarly, oversaturated pixels lead to incorrectly computed normals. Therefore, we only estimate normals for pixels that have normalized intensities between 0.03 and 0.97. Since albedo is a view-independent quantity, we rely on hole filling to deal with surface points with low albedo.

4 Reconstruction

The multi-view normal maps reproduce the high-resolution geometric detail present in the surface. We combine the information from the normal maps to reconstruct a complete 3D mesh in a three-stage process. First, for each view separately, we integrate a surface from the normal map. During this process, the visual hull acts as a "proxy": it provides rough constraints for the overall position of the integrated surface. Second, we use a similarity metric based on illumination, reconstructed local surface shape, or both to match neighboring views and smoothly deform the integrated surfaces in order to improve their fit. Again, we use the visual hull as a proxy, this time to constrain the possible matches. Finally, we merge the matched surfaces into a single mesh and optionally fill in the holes. The resulting closed mesh is now an excellent approximation to the true shape of the surface, and in particular is a better surface proxy than the visual hull. Therefore, we may repeat all three stages of our reconstruction algorithm, using the new proxy mesh instead of the visual hull wherever needed. The output of the second pass of our algorithm is the final mesh, as we have observed little quality improvement from further passes.

Figure 4: *For a particular view (a), we use the normal map (b) in order to integrate the initial surface (c). Better quality surfaces are obtained when we detect large depth discontinuities (d). The normal maps after surface reconstruction are smoother than the original normal maps, but remove the initial bias, which can be seen in the legs (e).*

4.1 Single-View Surface Reconstruction

We begin by integrating individual normal maps into partial surface meshes. We represent the surface as a depth image, centered at the corresponding camera viewpoint (i.e., the value at each pixel determines how far along the camera ray the surface point lies).

We pose the integration problem as an optimization process, in which the depth values are the unknowns and the observed normals provide constraints. In particular, we enforce that the 3D vector between two neighboring depth samples i and j be perpendicular to the average of the measured normals at those pixels. With only the normals as constraints, the reconstruction problem would be ill-posed: it would be possible to move the surface forward or back while still satisfying the constraints. Therefore, we use the visual hull or the proxy mesh to provide (soft) depth constraints on the reconstruction. The optimization is formulated as a linear system:

$$\arg\min_{\mathbf{z}} \sum_{i,j} \left((\mathbf{n}_i + \mathbf{n}_j)^\top (\mathbf{r}_i z_i - \mathbf{r}_j z_j) \right)^2 + \alpha \sum_i (z_i - \bar{z}_i)^2, \qquad (3)$$

where for a pixel i: z_i is the distance to the surface along the corresponding ray direction $\mathbf{r}_i$, $\mathbf{n}_i$ is the measured normal, and $\bar{z}_i$ is a possible depth constraint at that pixel — the depth of the visual hull or the proxy, if available. The parameter α determines the relative strengths of the normal and depth constraints. We set it to be low (10^{-6}), corresponding approximately to the inverse of the number of our surface points. In the second pass, we expect the proxy mesh to be more accurate, so we increase the weight on the depth constraints by setting α to 10^{-5}.

The optimization process that computes surfaces according to Equation 3 does not intrinsically take into account depth discontinuities. However, integrating normals across depth discontinuities may cause significant distortions, as is evident in Figure 4c, where the legs are connected to the rim of the dress. In order to avoid this issue, we must remove the pixels straddling the depth discontinuities from the linear system (Equation 3). However, detecting depth discontinuities is a difficult problem. We have experimented with a variety of heuristics (e.g., maxima of color and normal gradients, local integrability measures), and have found that the following two simple strategies produce good results. First, large visual hull discontinuities are usually located near the true discontinuities. Removing pixels along them helps keep the surface free of large distortions. In addition to the large visual hull depth discontinuities, our matching algorithm (described in the next section) identifies

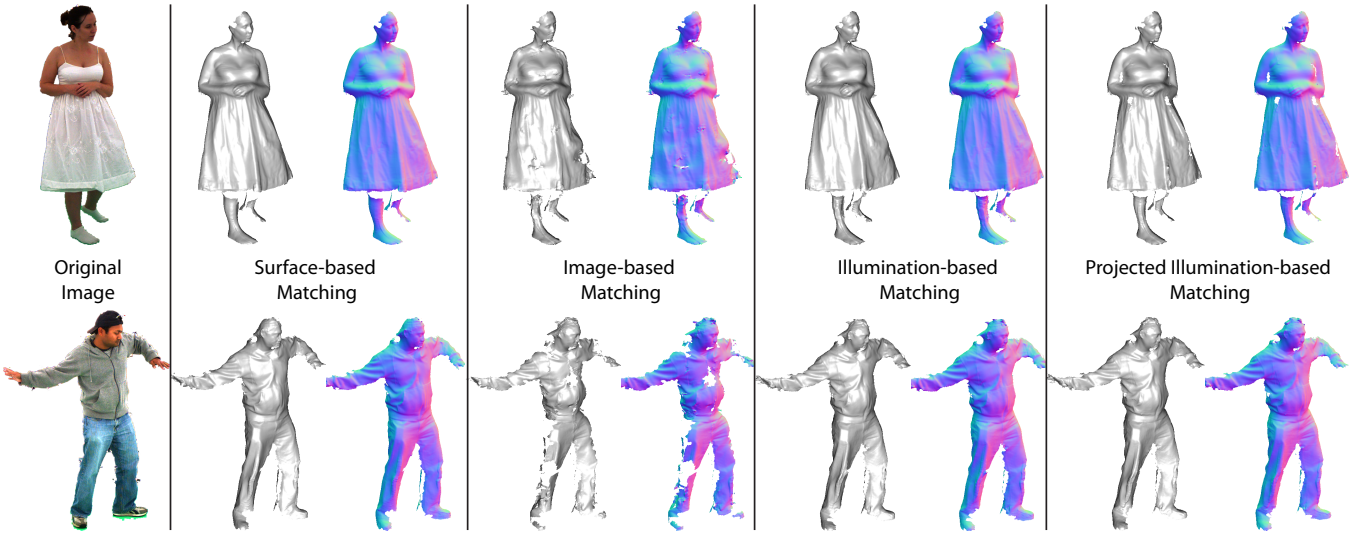


Figure 5: *Results (surfaces and normals) of several matching strategies on two data sets. Our surface-based metric yields the overall best results, especially in regions that lack texture or are shadowed. This is visible in the first row by comparing the reconstructed legs. In the second row, only the surface-based metric does not push the waist inward.*

more precise discontinuities, defined as locations at which the surface jumps by more than 1 cm per pixel. Together, these heuristics alleviate depth discontinuity issues and allow the legs to integrate closer to the center of the skirt in our example (Figure 4d). In the second pass, all the detected discontinuities are closer to their true locations, making the integrated mesh even more precise.

Overall our method is fairly robust with respect to the quality of the visual hull. The accuracy of the visual hull mainly plays a role in determining the depth discontinuities. The closer the visual hull is to the true surface, the more complete our reconstruction will be. If the visual hull's depth discontinuities are far from the true depth discontinuities, then the pixels that are mistakenly connected to a wrong part of the performer (e.g., the leg connected to skirt) will be disregarded (and thus lost from the reconstruction) by the matching phase described below.

4.2 Pairwise Matching

Individual integrated surfaces contain the high-frequency details that are present in the normal maps. However, these surfaces rarely match exactly because of the bias in the normals (due to self-occlusion for example), and so cannot yet be merged into a single model. To correct for this distortion, we warp the surface based on matches computed between pairs of neighboring views.

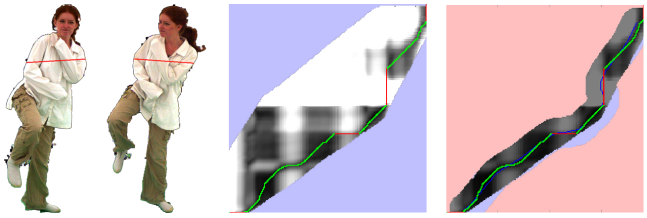
Metrics for Matching We have experimented with a number of matching metrics to perform the correspondences, two based on images alone and two that rely on the integrated surfaces. Our first image-based metric uses only pixel windows under the spherical full-on illumination condition, and therefore reduces to traditional stereo matching. Our second metric is based on comparing an "image stack" of the six illumination conditions. We have found that this increases robustness in many areas that have little color texture, but significant geometric variation.

In practice, the performance of both image-based metrics is limited by different amounts of foreshortening between different views: different views of a region on a surface will not, in general, appear the same from different cameras. Therefore, we have also compared metrics that match the integrated surfaces. Since the distortion is low-frequency, we can assume that a small 3D surface patch in one view and the corresponding patch in the neighboring view will differ only by a rigid transformation. Therefore, we can compare small surface patches (e.g., 5×5 pixels) by computing the mean surface-to-surface distance under the optimal rigid-body alignment. Specifically, given a pair of surfaces ("left" and "right"), we compute the matching error between a point on the left surface and a point on the right surface as follows. First, we find a window of depth samples around the left point. Then, we render the right surface from the left camera's point of view, and find a window of depth samples around the projection of the right point. We assume that the two windows of points correspond, solve for the optimal rigid-body alignment, and compute the mean distance between the pairs of points in the windows. Then, to obtain a symmetric matching score, we repeat the computation with the roles of left and right reversed (i.e., rendering the left mesh into the right camera and finding the windows of samples there). We take the maximum of the two mean distances as the matching error.

Our final matching metric also relies on aligning surface patches, but takes the matching error to be the difference between image stacks (under different illumination) projected onto the surface. In areas of significant color detail, this improves discriminability over surface-distance-based matching, while retaining the advantage of compensating for foreshortening. However, we have found that in areas of little texture the surface-distance-based metric is superior.

We have compared these four metrics on a variety of datasets, as illustrated in Figure 5. As expected, the illumination-stack metric yields better surfaces than simple image-based matching, but, due to the wide baseline, still produces wrong matches. The projected-illumination metric further improves the reconstruction, but exhibits artifacts in shadowed regions. On the whole, we have found that the surface-based metric usually yields the least noisy surfaces.

Global Correspondence To find the best match for a point in the left view, we only need to explore the points in the right view that project to the corresponding epipolar line. Therefore, a simple matching strategy would be to simply take the point with minimum matching error as the correspondence. A more robust approach that takes into account surface continuity considers a whole epipolar plane (e.g., a plane passing through a point in the left view and both of the camera centers) at the same time. We sample this plane by stepping one pixel at a time in each view. We evaluate our matching error on the resulting grid and find the lowest-cost path from one corner to the opposite corner (Figure 6). In particular, we use the 4-step method described by Criminisi et al. [2007] to perform the search. In addition to finding the best matches, this



(a) (b) (c)

Figure 6: *We match the surfaces in neighboring views one epipolar plane at a time (a). Each plane defines a grid of matching errors, where the lowest cost path yields the surface matches (b,c). The green portions of this path denote good matches, while red lines denote depth discontinuities. In the second pass (c), we use the proxy mesh (blue line) to guide this search and only evaluate the nearby matching errors. Light blue denotes points outside the visual hull, while points in pink are not considered in the second pass.*

algorithm detects depth discontinuities, which we then use during surface integration in the next pass. Note that we do not enforce smoothness *across* the sampled planes directly. However, some smoothness is indirectly incorporated because each surface patch spans several epipolar planes.

Good surface matches define absolute depth constraints at certain pixels in each of the views. After computing these matches for all views, we deform the integrated surface to fit these constraints while preserving its high-frequency detail. We achieve this with a thin-plate offset:

$$\arg\min_{\mathbf{d}} \sum_i \sum_j (d_i - d_j)^2 + \beta \sum_i (d_i - \bar{d}_i)^2, \qquad (4)$$

where the depth offset d_i of each pixel i should be equal to the offset $\bar{d}_i$ obtained from the matching. These offsets are smoothly interpolated by pulling d_i toward the centroid of its neighboring offsets d_j. Because the thin-plate offset may introduce significant deformations in parts of the mesh that are far from the constraints, we discard these regions (using a threshold on distance to the nearest constraint point).

Applying the depth offsets obtained from Equation 4 aligns all surfaces much closer (Figure 7). The surfaces are still not perfectly aligned, since the matches are computed on a pixel grid. We address the remaining misalignment errors in the surface merging stage.

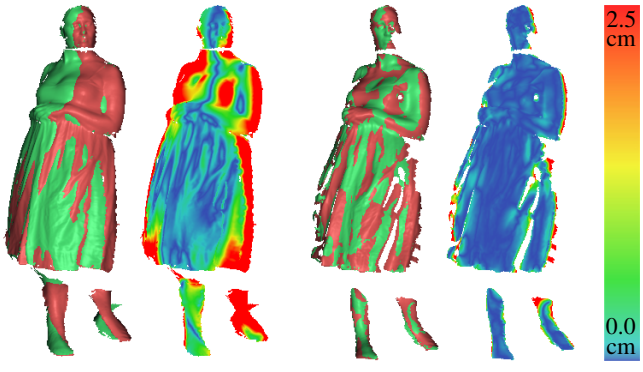
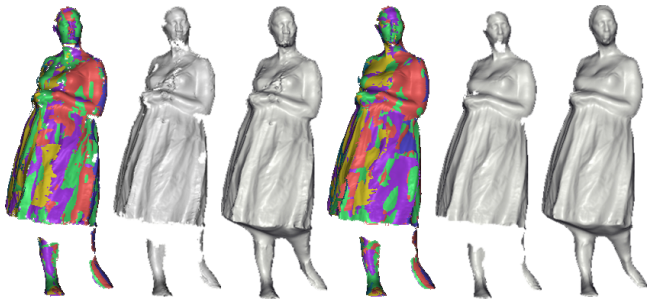


Figure 7: *The integrated surfaces before matching (left) are far from each other, while the deformed surfaces after matching (right) are much closer to each other. Within each pair, the leftmost visualization shows the two meshes (in different colors) overlaid on each other, while the rightmost visualization is a color-coding of mesh-to-mesh distance.*



Pass 1 Pass 2

Figure 8: *From left: false-color visualization of eight meshes after integration, matching, and deformation; result of initial volumetric merging; result of hole-filling; similar visualizations after second-pass matching.*

4.3 Multi-view Surface Reconstruction

After the integration, matching, and deformation stages are performed on all views, we must merge the aligned, yet still logically separate surfaces, into a single mesh. In addition, we must account for the regions in which no data is available by performing hole-filling.

Merging We merge the eight aligned surfaces using the Volumetric Range Image Processing (VRIP) algorithm [Curless and Levoy 1996]. This is a volumetric method that allows for some residual misalignment between scans by averaging signed-distance ramps along the line-of-sight of each mesh. We set the ramp-length to 6 cm. to allow for worst-case misalignment, and reconstruct using a 2 mm. voxel size, which approximately matches the average resolution of our raw data. We also modify the weight computation of VRIP: in addition to weighting each point dependent on its distance to the nearest mesh boundary (to provide for smooth blending) and the cosine of its normal with the view direction (to downweight foreshortened data), we also include a term inversely proportional to the distance between the sample and its camera. The latter down-weights regions of sparsely-sampled data, and is of greater benefit in our setup (in which the distance to the camera can vary significantly) than in typical 3D scanners. A sample result of the merging step is shown in Figure 8. In the first column, we show the meshes that are inputs to the merging, with the output of VRIP in the second column.

Hole Filling As can be seen, the merged mesh contains regions in which there is no surface, typically due to occlusion. Several approaches have been proposed to fill such holes, using techniques such as space carving [Curless and Levoy 1996] and volumetric diffusion [Davis et al. 2002]. In our case, we wish to combine information similar to that used in the two above techniques: we would like to fill small holes smoothly, yet for larger holes we wish to use the information present in the visual hull.

Our hole-filling approach uses the Poisson Surface Reconstruction algorithm of Kazhdan et al. [2006]. As input, we provide oriented point samples taken from both the VRIP reconstruction and the visual hull, giving significantly lower weight (0.01) to the latter. We have found that this produces smooth fills, and draws the final surface towards the visual hull in regions of significant missing data. Moreover, the method guarantees a watertight manifold output, as shown in the result in Figure 8, third column. This mesh may now be used as the proxy mesh, instead of the visual hull, in the second pass of matching. The result of this second pass, together with the merged meshes, are shown in the right half of Figure 8.

We have also experimented with using Poisson Surface Reconstruction for both merging and hole-filling simultaneously (by using samples from the original meshes rather than the VRIP reconstruc-

tion). However, we have found that in regions of significant residual misalignment this results in more smoothing. This is because the Poisson problem inherently treats the influence of each point as isotropic, and hence does not preserve detail as well as VRIP's oriented signed-distance ramps when the merged surface is far from the original samples. We therefore believe that the combination of the two algorithms produces better results than either alone.

Using the visual hull for hole filling is noisy and can yield incorrect topology. This is why we only use it in the first pass, to produce a watertight proxy surface for the second pass. Our final meshes are suitable for filling using more sophisticated approaches, which exploit temporal coherence to aggregate information from multiple frames. This is an active area of research [Wand et al. 2007; Huang et al. 2008; Li et al. 2008; Sharf et al. 2008; Zhang et al. 2008; Wand et al. 2009], and lies outside the scope of this paper.

5 Results

We have acquired and processed five different sequences, including people wearing loose clothing, long skirts, and even a subject covered with a linen sheet. The reconstructions are presented throughout this paper and in Figures 1 and 9, which show a number of individual frames from our sequences. Note that the normals used for rendering are the geometrical (computed) normals, not the (measured) normals from the normals maps. Reprojecting and embossing photometrically measured normals would yield even better results, but would not represent the *true* quality of the reconstructeed geometry. Additional results, including complete animations, are presented in the supplementary video.

These sequences demonstrate that our system can correctly handle characters performing fast motions, as well as non-articulated characters for which template-based approaches fail. Many of our sequences have few or no textured areas, making them challenging for any type of stereo matching algorithm. In contrast, the analogous difficult situation for our algorithm is surfaces that lack geometric detail. However, in these regions, the smoothing performed by our algorithms is the correct action.

Computation Time We typically run the software on a 2.4 GHz PC with at least 2GB of RAM. The total computation time to obtain one final mesh is about an hour, and no user assistance is required. However, most parts of our code have not been optimized or parallelized, which leads us to believe that this time can be significantly reduced. In the current implementation, typical running times of the parts of the pipeline are: normal map computation with motion compensation: 50 min; first pass: 8 min; second pass: 5 min; and merging: 2 min. Normal map computation is dominated by the optical flow (two per camera), while the reconstruction spends most of its time computing the surface matches. While the total processing time is an hour per frame, it should be noted that this includes processing of the raw 240Hz video streams using non-optimized optical flow code. Using a GPU based optical implementation, or a less accurate but more efficient algorithm, could greatly reduce this pre-processing time. The actual processing time starting from the normal maps is only 10 to 15 minutes per frame.

Discussion In this work we have made the deliberate choice of not integrating multiple steps together to obtain a potentially more unified or optimized pipeline. Capturing and processing high quality detailed performance geometry is a complex task that requires significant amounts of hardware, calibration and processing, and rapidly produces huge amounts of data. We therefore try to keep our system as modular as possible. This greatly improves the robustness of the calibration, eases data management, and increases the amount of parallelism in the processing.

Our pipeline currently does not enforce temporal correspondences nor apply temporal filtering. However, we have observed only minor flickering in the matched regions of our meshes. This suggests that our reconstructions are quite close to the true surfaces. While temporal filtering could potentially be used to further smooth the results, it would also remove desirable details of mesh animations. Temporal registration of the acquired mesh sequences could improve the surface coverage by accumulating information through time. However, this is still a very active area of research [Mitra et al. 2007; Sagawa et al. 2007; Pekelny and Gotsman 2008; Chang and Zwicker 2008]. The choice and evaluation of a particular algorithm, or design of a new algorithm, falls outside the scope of this work. Nevertheless, we designed our acquisition and data processing pipeline so that these algorithms can operate as a post-process on our data.

Even though we do not perform any temporal processing, there is little flickering noticable when playing back the processed geometries of a performance. There is some flicker visible near the boundaries of each single-view surface, because data is increasingly bad there, and because the decision of whether to mark a pixel as a discontinuity is binary and independent per frame. Away from the boundary, however, the flicker is very low, showing that we are getting close to the true geometry. Since our resolution is very high (1,000,000 triangles, or 1 triangle per pixel), differences in triangulation do not produce noticeable artifacts either. Finally, the temporal coherence of the computed normals maps is very good to begin with, which reduces the need for temporal regularization.

A second type of coherence not currently exploited by our method is inter-scanline continuity. This could be enforced by using a Markov-Random Field formulation and employing an optimization method such as Belief Propagation or min-cut. However, this would require large amounts of memory and computational power.

Limitations The computed normal maps show significant high frequency detail. However, these normal maps also have a significant bias. This bias is different than the bias observed in traditional photometric stereo (i.e., point light sources versus area light sources), and is especially present in the concavities, since in these regions a smaller-than-expected portion of the sphere of lights illuminates the surface. Furthermore, the normals are typically incorrect in the hair region (due to the complex scattering in the hair volume, which is not consistent with the Lambertian assumption made by photometric stereo), and the generally low albedo of hair. The normal maps are also noisy in areas of low albedo and in areas that were not properly aligned during the motion compensation stage. Figure 4e compares the acquired input normals from one of the viewpoints and the corresponding normal map of the final model. Observe the differences in the left leg, where the captured normals were corrupted by the shadow cast by the dress. This figure also demonstrates that our reconstruction pipeline introduces some smoothing of fine details: the original normal map is sharper than the reconstruction. This smoothing is introduced at several stages during the pipeline, including the surface integration, matching, and scan merging. Most of the original detail, however, is retained.

The current hardware setup is fairly complex, but it should be noted that the setup described is designed as a research prototype, not specialized to the task at hand, and can be refined to reduce complexity and cost. First, not every light source needs to be individually controllable. Only eight distinct groups require individual control, corresponding to the eight quadrants of the sphere of incident directions. Each illumination pattern would then light half of the groups at the same time. Second, using only 10% (i.e., 120) of the number of light sources is sufficient to accurately infer photometric normals from diffuse reflections. Both observations allow us to greatly simplify the hardware setup. Furthermore, we carefully designed the illumination conditions to require only a modest level of control and accuracy. Individual light source intensities do not need to match due to our data-driven scheme to infer normals. The only hard requirement is the ability to quickly toggle lights on and off, which is automatically achieved by using LEDs. Never-

Figure 9: *Each row shows an original image, the corresponding normal map, the reconstructed surface, the hole-filled surface, as well as a novel view of the reconstructed and hole-filled surface for a captured performance. Note that hole-filling was not applied to the versions of these results shown in the accompanying video.*

theless, we expect our processing pipeline to work with alternative setups equally well. For example, six individually triggered flash lights aimed away from the subject at the surrounding walls could achieve similar illumination conditions as used in this paper, and could be directly used to estimate normals, because of the data-driven nature of our normal estimation algorithm. Finally, even though we employ high-speed cameras, our method is specifically designed to work at moderate frame rates that fall in the range of what is possible with readily-available hardware such as Point Grey's Grasshopper, which can capture at 200Hz with a 640x480 resolution.

6 Conclusions and Future Work

Though 3D capture of dynamic performances remains a challenging problem, this paper makes progress towards acquiring high-quality mesh animations of real-life performances. Our system uses novel hardware and image processing algorithms to obtain high-quality normal maps and silhouettes from multiple viewpoints at video rates. The surface reconstruction algorithms process this data to derive high-quality mesh sequences. The resulting mesh sequences can be used in biomechanics to analyze complex motions, in computer games to create next-generation characters, and in movies to create digital doubles.

We take advantage of the high-frequency geometric information present in photometric-stereo normal maps; therefore, our method significantly outperforms multi-view stereo techniques that produce overly smooth surfaces due to lack of texture or photometric calibration errors. Furthermore, our method does not require geometric templates as input and thus it is not restricted by their limitations.

The data produced by the system will stimulate more work in the geometry processing community, whose research endeavors take a sequence of incomplete moving surfaces and produce the best-fitting, congruent, water-tight moving mesh. Further processing of these mesh sequences will prove challenging due to the shear amount of data our system is producing (each mesh contains more than 500,000 vertices).

Acknowledgments This work has benefitted from the assistance and valuable input of many people, and we especially thank Bill Swartout, Randall Hill, Randolph Hall, Max Nikias, the MIT graphics group, and the anonymous SIGGRAPH Asia reviewers. Thanks to Bruce Lamond for hardware assistance and proof reading, Tom Buehler and Boris Ajdin for help with the video, as well as our performance subjects: Saskia Mordijck, Abhijeet Ghosh, and Jay Busch. This work was sponsored in part by the NSF, grants CCF-0347427 and CCF-0541227; the Singapore-MIT Gambit Game Lab; University of Southern California Office of the Provost; U.S. Army Research, Development, and Engineering Command (RDECOM), and software donations from Autodesk and Adobe Systems. The content of the information does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

References

AHMED, N., THEOBALT, C., DOBRE, P., SEIDEL, H.-P., AND THRUN, S. 2008. Robust fusion of dynamic shape and normal capture for high-quality reconstruction of time-varying geometry. In *Computer Vision and Pattern Recognition*, 1–8.

ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. 2005. Scape: Shape Completion and Animation of People. *ACM Transactions on Graphics 24*, 3 (Aug.), 408–416.

BALAN, A. O., SIGAL, L., BLACK, M. J., DAVIS, J. E., AND HAUSSECKER, H. W. 2007. Detailed human shape and pose from images. In *Computer Vision and Pattern Recognition*.

BERNARDINI, F., RUSHMEIER, H., MARTIN, I. M., MITTLEMAN, J., AND TAUBIN, G. 2002. Building a digital model of michelangelo's florentine pietà. *IEEE Computer Graphics & Applications 22*, 1 (Jan./Feb.), 59–67.

BRADLEY, D., POPA, T., SHEFFER, A., HEIDRICH, W., AND BOUBEKEUR, T. 2008. Markerless garment capture. *ACM Transactions on Graphics 27*, 3 (Aug.), 99.

BROX, T., BRUHN, A., PAPENBERG, N., AND WEICKERT, J. 2004. High accuracy optical flow estimation based on a theory for warping. In *Proceedings of the 8th European Conference on Computer Vision*, 25–36.

CAMPBELL, N., VOGIATZIS, G., HERNANDEZ, C., AND CIPOLLA, R. 2007. Automatic 3d object segmentation in multiple views using volumetric graph-cuts. In *British Machine Vision Conference*.

CARRANZA, J., THEOBALT, C., MAGNOR, M. A., AND SEIDEL, H.-P. 2003. Free-viewpoint video of human actors. *ACM Transactions on Graphics 22*, 3 (July), 569–577.

CHANG, W., AND ZWICKER, M. 2008. Automatic registration for articulated shapes. *Computer Graphics Forum (Proceedings of SGP 2008) 27*, 5, 1459–1468.

CORAZZA, S., MÜNDERMANN, L., CHAUDHARI, A., DEMATTIO, T., COBELLI, C., AND ANDRIACCHI, T. P. 2006. A markerless motion capture system to study musculoskeletal biomechanics: Visual hull and simulated annealing approach. *Annals of Biomedical Engineering 34*, 6 (July), 1019–1029.

CRIMINISI, A., BLAKE, A., ROTHER, C., SHOTTON, J., AND TORR, P. H. 2007. Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming. *Int. Journal of Computer Vision 71*, 1, 89–110.

CURLESS, B., AND LEVOY, M. 1996. A volumetric method for building complex models from range images. In *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, 303–312.

DAVIS, J., MARSCHNER, S. R., GARR, M., AND LEVOY, M. 2002. Filling holes in complex surfaces using volumetric diffusion. In *Symposium on 3D Data Processing, Visualization, and Transmission*, 428–438.

DAVIS, J., RAMAMOORTHI, R., AND RUSINKIEWICZ, S. 2005. Spacetime stereo: A unifying framework for depth from triangulation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 196–302.

DE AGUIAR, E., THEOBALT, C., STOLL, C., AND SEIDEL, H.-P. 2007. Marker-less deformable mesh tracking for human shape and motion capture. In *Computer Vision and Pattern Recognition*.

DE AGUIAR, E., STOLL, C., THEOBALT, C., AHMED, N., SEIDEL, H.-P., AND THRUN, S. 2008. Performance capture from sparse multi-view video. *ACM Transactions on Graphics 27*, 3 (Aug.), 98.

EINARSSON, P., CHABERT, C.-F., JONES, A., MA, W.-C., LAMOND, B., HAWKINS, T., BOLAS, M., SYLWAN, S., AND DEBEVEC, P. 2006. Relighting human locomotion with flowed reflectance fields. In *Proc. of Eurographics Symposium on Rendering*, 183–194.

FURUKAWA, Y., AND PONCE, J. 2006. Carved visual hulls for image-based modeling. In *European Conference on Computer Vision*, 564–577.

HERNANDEZ, C., AND SCHMITT, F. 2004. Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding 96*, 3 (Dec.), 367–392.

HERNANDEZ, C., VOGIATZIS, G., AND CIPOLLA, R. 2008. Multiview photometric stereo. *IEEE Trans. Pattern Anal. Mach. Intell. 30*, 3, 548–554.

HERTZMANN, A., AND SEITZ, S. M. 2005. Example-based photometric stereo: Shape reconstruction with general, varying brdfs. *IEEE Trans. Pattern Anal. Mach. Intell. 27*, 8, 1254–1264.

HORNUNG, A., AND KOBBELT, L. 2006. Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In *Computer Vision and Pattern Recognition*, 503–510.

HUANG, Q.-X., ADAMS, B., WICKE, M., AND GUIBAS, L. J. 2008. Non-rigid registration under isometric deformations. *Computer Graphics Forum (Proc. SGP'08) 27*, 5, 1449–1457.

JOSHI, N., AND KRIEGMAN, D. 2007. Shape from varying illumination and viewpoint. In *International Conference on Computer Vision*.

KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. In *Symposium on Geometry Processing*.

LI, H., SUMNER, R. W., AND PAULY, M. 2008. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum 27*, 5, 1421–1430.

LIM, J., HO, J., YANG, M.-H., AND KRIEGMAN, D. 2005. Passive photometric stereo from motion. In *International Conference on Computer Vision*.

MA, W.-C., HAWKINS, T., PEERS, P., CHABERT, C.-F., WEISS, M., AND DEBEVEC, P. 2007. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *Rendering Techniques*, 183–194.

MITRA, N. J., FLORY, S., OVSJANIKOV, M., GELFAND, N., GUIBAS, L., AND POTTMANN, H. 2007. Dynamic geometry registration. In *Proc. Symposium on Geometry Processing*, 173–182.

NEHAB, D., RUSINKIEWICZ, S., DAVIS, J., AND RAMAMOORTHI, R. 2005. Efficiently combining positions and normals for precise 3d geometry. *ACM Transactions on Graphics 24*, 3 (Aug.), 536–543.

OKUTOMI, M., AND KANADE, T. 1993. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15*, 4, 353–363.

PEKELNY, Y., AND GOTSMAN, C. 2008. Articulated object reconstruction and markerless motion capture from depth video. *Computer Graphics Forum 27*, 2 (Apr.), 399–408.

RANDER, P. W., NARAYANAN, P., AND KANADE, T. 1997. Virtualized reality: Constructing time-varying virtual worlds from real world events. In *IEEE Visualization*, 277–284.

RUSINKIEWICZ, S., HALL-HOLT, O., AND LEVOY, M. 2002. Real-time 3D model acquisition. *ACM Transactions on Graphics 21*, 3 (July), 438–446.

SAGAWA, R., OSAWA, N., AND YAGI, Y. 2007. Deformable registration of textured range images by using texture and shape features. In *3DIM '07: Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling*, 65–72.

SEITZ, S. M., AND DYER, C. R. 1999. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision 35*, 2, 151–173.

SEITZ, S. M., CURLESS, B., DIEBEL, J., SCHARSTEIN, D., AND SZELISKI, R. 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer Vision and Pattern Recognition*, 519–528.

SHARF, A., ALCANTARA, D. A., LEWINER, T., GREIF, C., SHEFFER, A., AMENTA, N., AND COHEN-OR, D. 2008. Space-time surface reconstruction using incompressible flow. *ACM Trans. Graph. 27*, 5, 1–10.

STARCK, J., AND HILTON, A. 2003. Model-based multiple view reconstruction of people. In *International Conference on Computer Vision*, 915–922.

STARCK, J., AND HILTON, A. 2007. Surface capture for performance based animation. *IEEE Computer Graphics and Applications 27(3)*, 21–31.

SVOBODA, T., MARTINEC, D., AND PAJDLA, T. 2005. A convenient multi-camera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments 14*, 4 (August), 407–422.

THEOBALT, C., AHMED, N., LENSCH, H., MAGNOR, M., AND SEIDEL, H.-P. 2007. Seeing people in different light-joint shape, motion, and reflectance capture. *IEEE Transactions on Visualization and Computer Graphics 13*, 4 (July/Aug.), 663–674.

VLASIC, D., BARAN, I., MATUSIK, W., AND POPOVIĆ, J. 2008. Articulated mesh animation from multi-view silhouettes. *ACM Transactions on Graphics 27*, 3 (Aug.), 97.

VOGIATZIS, G., TORR, P. H. S., AND CIPOLLA, R. 2005. Multi-view stereo via volumetric graph-cuts. In *Computer Vision and Pattern Recognition*, 391–398.

VOGIATZIS, G., HERNANDEZ, C., AND CIPOLLA, R. 2006. Reconstruction in the round using photometric normals and silhouettes. In *2006 Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, 1847–1854.

WAND, M., JENKE, P., HUANG, Q., BOKELOH, M., GUIBAS, L., AND SCHILLING, A. 2007. Reconstruction of deforming geometry from time-varying point clouds. In *Proc. Symposium on Geometry Processing*, 49–58.

WAND, M., ADAMS, B., OVSJANIKOV, M., BERNER, A., BOKELOH, M., JENKE, P., GUIBAS, L., SEIDEL, H.-P., AND SCHILLING, A. 2009. Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. *ACM Transactions on Graphics 28*, 2 (Apr.), 15.

WOODHAM, R. J. 1978. Photometric stereo: A reflectance map technique for determining surface orientation from image intensity. In *Proc. SPIE's 22nd Annual Technical Symposium*, vol. 155.

ZHANG, S., AND HUANG, P. 2006. High-resolution real-time three-dimensional shape measurement. *Optical Engineering 45*, 12.

ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. M. 2004. Spacetime faces: high resolution capture for modeling and animation. *ACM Transactions on Graphics 23*, 3 (Aug.), 548–558.

ZHANG, H., SHEFFER, A., COHEN-OR, D., ZHOU, Q., VAN KAICK, O., AND TAGLIASACCHI, A. 2008. Deformation-driven shape correspondence. *Proc. Symposium on Geometry Processing 27*, 5 (July), 1431–1439.