Towards A Complete Framework For Deformable Surface Recovery Using RGBD Cameras

Tolga Birdal[1] Diana Mateus[2] and Slobodan Ilic[3]

***Abstract*— In this paper, we study the problem of 3D deformable surface tracking with RGBD cameras, specifically Microsofts Kinect. In order to achieve this we introduce a fully automated framework that includes several components: automatic initialization based on segmentation of the object of interest, then robust range flow that guides deformations of the object of interest and finally representation of the results using mass-spring model. The key contribution is extension of the range flow work of Spies and Jähne [1] that combines Lucas-Kanade [2] and Horn and Shunk [3] approaches for RGB-D data, makes it to converge faster and incorporates color information with multichannel formulation. We also introduced a pipeline for generating synthetic data and performed error analysis and comparison to original range flow approach. The results show that our method is accurate and is precise enough to track significant deformation smoothly at near real-time run times.**

I. Introduction

Deformable surfaces are ubiquitous in real world and, thus, are of great interest to computer vision researchers. While the research in the area is quite new, many advanced methods have already been developed. Most of these methods rely on stereo computations or try to solve the under-constrained problem of recovering surface deformations from monocular scenes. Recently, there has been an increasing number of depth (RGBD) cameras available at commodity prices. These cameras can usually capture both color and depth images in real-time, with limited resolution and accuracy. In this paper, we present a complete framework for automatic capture of surface deformations from RGB-D data. We initialize an area of interest in the RGB-D image and recover the motion of every pixel from frame to frame. This is equivalent to optical flow in RGB images, but here we obtain the flow vectors in 3D and we consider this to be scene or range flow.

Our method is inspired by the approach of Spies and Jähne on range flow [4], which was first proposed as an adaptation of optical flow to RGB-D data. When Spies proposed range flow, commodity RGBD sensors were not available, and thus the technique didn't become popular. By introducing several novelties compared to their approach, we show that recent developments in the field of tracking can well be adapted to the methodology, making it a usable tool for state of the art applications. First, we propose a fully automatic surface segmentation technique to capture the initial pose of a deformable region of interest. This allows complete automatic initialization and does not require any user interaction. In addition the improvements on Range Flow are as follows:

- incorporation of combined Lucas Kanade and Horn and Schunk in order to get more accurate and denser fields
- use of image pyramids to speed up convergence and to converge better
- incorporation of color information with a multichannel formulation

Overall this reduces the global error and together with speed enhancements performs faster and better than the original range flow. Our error analysis and the computation time shows that our technique is reasonable for global deformation capture of moving surfaces and works better the original range flow approach of Spies and Jähne [4]. Our method uses a single RGBD camera in a fully automated manner. This approach is efficient and works in near real-time.

II. Related Work

Because of computational reasons 3D deformable object tracking, at its current state is very immature in comparison to the rigid analogous. Yet, a decent variety of approaches exist to track 3D surfaces.

The work in this field begins with optical flow (OF), which uses photometric information to recover motion (either 2D or volumetric 3D) [3], [2]. Within many years the methods have obtained an impressive level of reliability and accuracy. The relation of optical flow to 2D deformation capture was made clear by [5]. Hilsmann utilizes flow and distance constraints to recover for deformable 2D mesh from 2D images. However, to the best of our knowledge this has not been extended to 3D. Applying optical flow to surfaces still stays in its nut-shell. Recently, researchers made use of multi-camera systems and one of the first surface flow algorithms was so called scene flow [6]. Developed by Vedula and Baker, scene flow was designed to work on multi-view settings and single view experiments were reported to generate very erroneous results. Even though it has then evolved into a variational framework [7], its multi-view nature is retained. For this reason, scene flow is not applicable to our problem.

Other recent studies were carried out on 3D deformation recovery from monocular images [8], [9]. However, these approaches are far from efficiency and they somehow require the point correspondences to be known. Not favoring this requirement, for the deformable case, it is very hard to obtain point correspondences (3D-2D).

Our work is inspired by range flow of Spies and Jähne on range flow[4]. Their approach consisted of taking the well studied Horn & Schunk method and extending to RGBD surface capture. By contrast we introduce robustified depth capture, together with automatic initialization. Our new formulation benefits from recent improvements in optical flow, while maintaining its extendibility. Therefore we improve accuracy and speed. We believe that our framework is a solid demonstration of the possibilities of how range flow could be advanced and of how fully automatic surface flow could be achieved.

Our work is most similar to what B. Petit and A. Letouzey proposed recently [10], [11]. However, their approach relies on flow constraining points, which require the successful extraction of SIFT features in a deforming sequence. Our method is free from such a condition and yet it still maintains a dense flow. Finally, other than computing the flow, we also pose our method as a complete framework of tracking, including the post-processing steps.

III. Surface Segmentation

Our method begins with surface segmentation on Kinect, which acts as an initial step to deformation recovery.

A. Preprocessing

To begin with, we median-filter Kinect's depth over time to gain certain temporal smoothness [12]. The second step of segmentation is the triangulation of the structured mesh. Such triangulation is nothing but connecting the neighboring pixels. Next, we need to filter out the connected mesh elements which we will not use. To accomplish that we treat the entire mesh as a connected sparse adjacency graph and apply Tarjan's algorithm to find the strong connected components [13]. Finally, additional filling is performed on the resulting connected regions to refine the segmentation. This approach, as described achieves to find 3D connected components in realtime. Figure 1 demonstrates an example of segmented Surface of Interest (SOI).

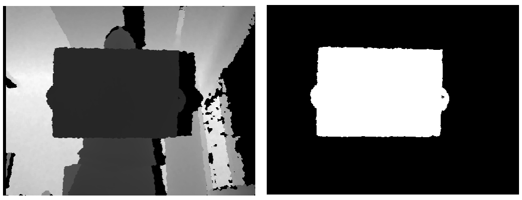


Fig. 1. On the left depth image, on the right segmented initial surface

IV. Surface Motion From Range Flow

After segmenting the surface, a robust, dense 3D tracking is required. For this reason, we will be using an adapted version of Range Flow Algorithm as proposed in [4].

A. Range Flow

Range Flow, proposed by Spies & Barron [4] is actually a modified 3D Horn & Schunk (HS) [3] method to cope with moving surfaces. In this setting, observed surface Z is treated as a depth function $Z = Z(X, Y, t)$ described in terms of space and time. The optical brightness constraint is replaced by range flow motion constraint (RFMC) which reads as

$$Z_X U + Z_Y V + W + Z_t = 0 \tag{1}$$

where $f_R = [U, V, W]^T$ is the range flow. Note that the derivation is not exactly similar to HS. Rather than having a brightness constraint, the assumption that the object is made of locally planar patches is used and the constraint is imposed on the derivative of the depth, rather than the depth values itself (Because depth is not expected to be constant, but its derivative is. Recall that the infinitesimally small motion of the patch is purely translation, causing $\frac{dZ_X}{dt} = \frac{dZ_Y}{dt} = 0$) [4].

1) Global Smoothness: Spies writes a globally regularized energy functional to be minimized as

$$E(U,V,W) = \int_A [(Z_X U + Z_Y V + W + Z_T)^2 + \alpha^2(\nabla U^2 + \nabla V^2 + \nabla W^2)] dX dY dT \tag{2}$$

The integral is computed over the integration area A, where it generally is the entire image and α^2 controls the smoothness. A classical Euler-Lagrange (EL) approach is used to find the solution.

2) Intensity Constraint: For the intensity data the well known brightness change constraint equation $I_X U + I_Y V + I_t = 0$ (BCCE) can be added [1].

Combining the RFMC, the BCCE and a simple membrane model yields the following energy to be minimized:

$$E(U,V,W) = \int_A [(Z_X U + Z_Y V + W + Z_T)^2 + \beta^2(I_X U + I_Y V + I_T)^2 + \alpha^2(\nabla U^2 + \nabla V^2 + \nabla W^2)] dX dY dT \tag{3}$$

As in case of HS, an iterative scheme yields a smooth solution. To combine the two terms (intensity and depth), the weights are chosen as $\beta^2 = \frac{<||\nabla Z||^2>}{<||\nabla I||^2>}$. This prevents the domination of one term to the other.

B. Improvements To Range Flow

1) Range Flow In Multi-Channel Images: Spies [4] states that the color channels can be incorporated into the existing range flow scheme using a simple summation. Furthermore, Lukins [14] provided a more general representation of channels, where the range flow constraint for channel C is

$$C_X U + C_Y V + C_Z W + C_T \tag{4}$$

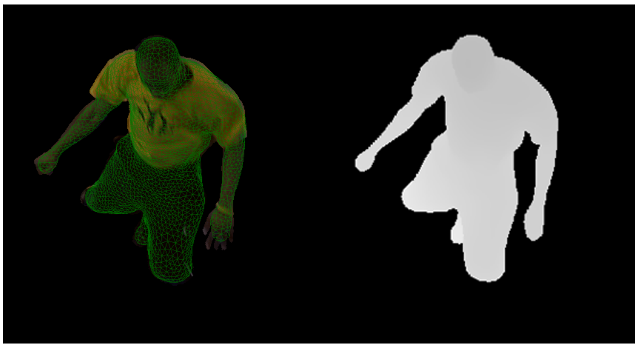
where C_Z is 0 for color and intensity channels and 1 for range channels. Adding color channels to the cost function will help resolve the aperture problem arising in optical and range flow problems. In our method, we choose to merge the color channels into the existing cost function. We formulate the new energy functional incorporating the color channels as

$$E(U,V,W) = \int_A \sum_{i=1}^{M} (\beta^i)^2 (C_X^i U + C_Y^i V + C_Z^i W + C_T^i)^2 + \alpha^2(\nabla U^2 + \nabla V^2 + \nabla W^2) dX dY dT \tag{5}$$

where M is the number of channels (including the depth channel).

The modification in Euler Lagrange equations results in a slight change in the solver. It only requires replacing the depth and intensity channels with the corresponding summations over the multi-channel image. Thus, we could write a more general multichannel solver.

(a) Synthetic RGB-D Data from Salzmann



(b) Synthetic RGB-D Data from Inria (Starck)

Fig. 2. Synthetic Data Visualizations a) Salzmann b) Starck

2) Robustifying Range Flow: Andrs Bruhn and Joachim Weickert propose an interesting method for merging Horn & Schunk algorithm with Lukas Kanade [15]. To come up with a unifying formulation, they alter the notation of the previous approaches. Let $\mathbf{f}$ denote homogeneous pixel-wise flow. So, $\mathbf{f} = (u, v, 1)^T$ and it follows that $\|\nabla \mathbf{f}\|^2 = \|\nabla u\|^2 + \|\nabla v\|^2$. Let $p = I(x, y)$ and hence $\nabla_3 p = (p_x, p_y, p_t)^T$. To clean up the equations, it is comfortable to denote $J_\rho(\nabla_3 p) = K_\rho * (\nabla_3 p \nabla_3 p^T)$ where $*$ is the convolution operation and K_ρ is Gaussian kernel with neighborhood size=ρ. From here, it becomes evident that a combined HS & LK (CLG) formulation can be derived as

$$J_{CLG}(\mathbf{f}) = \int_\Omega \mathbf{f}^T J_\rho(\nabla_3 p)\mathbf{f} + \alpha^{\mathbf{2}} |\nabla \mathbf{f}|^{\mathbf{2}} dxdy \tag{6}$$

It should be noted that these equations are hardly more complicated than the original HS equations and all one has to do is to evaluate the terms containing image data at a nonvanishing integration scale. Because of the simplicity of the results, we find it worthwhile to extend this formulation to range flow.

a) Extension to Range Flow: As described, Spies & Barron succeeded to present Range Flow as a framework not very different from optical flow. Such similarity calls for application of improvements in optical flow to range flow. As mentioned above for CLG method, the only slight modification involves Gaussian convolutions of the data terms. This in fact is very applicable to range flow, because we solve the problem in a very similar way. Before starting, let us note that we will alter the notation to cover multichannel images as well. This requires the terminology used in section IV-B.1. We will take our energy functional to be 5 and rewrite a new set of equations. This brings the entire problem to

$$J_{CLG}(\mathbf{f}_3) = \int_\Omega (\mathbf{f}_3^T J_\rho(\nabla_4 p)\mathbf{f}_3 + \alpha^2 |\nabla \mathbf{f}_3|^2) \tag{7}$$

This formulation adds the robustness of the local methods into the variational framework of dense global methods. Thus it enjoys a dense and smooth flow field.

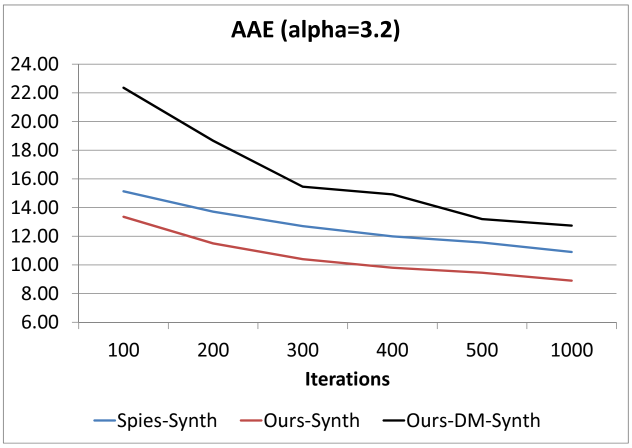
The final contributions are on the implementation causing an enhancement on the speed and robustness. First of all, we use Gaussian pyramids to speed up the convergence. Next, we utilize sparse solvers to achieve high speed and stability. It is known that median filtering robustifies the optical flow estimation [16]. Thus, we choose to apply it after each incremental estimation step to remove outliers. After the range flow is complete, we penalize the unrealistic deformations by imposing the range flow velocities as a proportion of the force field on the mass spring system (MSS). Automatically, MSS pulls together the vertices and constrains the range flow.

V. Data Synthesis and Acquisition for Evaluation

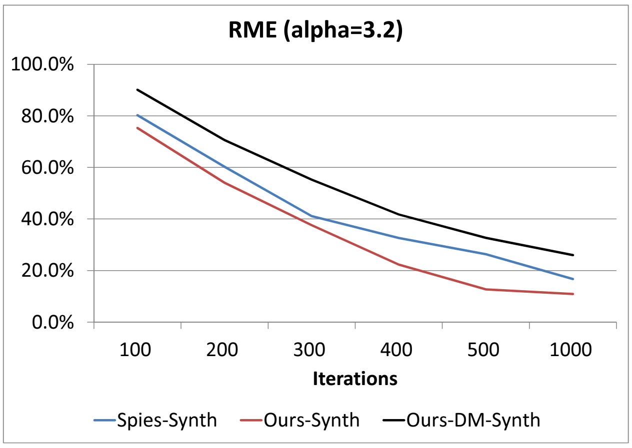
A. Generating Synthetic Data

The ground truth synthesized is composed of : Depth Images Of Deforming Sequence For Each Frame, RGB Images Of Deforming Sequence For Each Frame and Precise Ground Truth Range Flow per Each Pixel in Generated RGB-D Data. For computation of this data we require: 3D Coordinates of Deforming Vertices Over Time, Texture Of The Deforming Surface and 4D Motion of Deformation.

To approximate the real deformation as close as possible we benefit from two different datasets to carry out the synthesis. First one is made public by Mathieue Salzmann [17] and the second by Starck and Cagniart [18], [19]. Salzmann's data is just the mesh of a deforming cardboard. So our algorithm to come up with a synthetic data from this deformation is as follows: First we refine the provided mesh & sort vertices using the motion mapping. Next, the mesh is texture mapped in front of a virtual camera. To obtain a correctly interpolated depth, flow and RGB data, we intersect the rays casted from pixels with the mesh triangles. At each intersection (discarding the non-visible faces), we carry out a barycentric interpolation for depth, color and true flow. Finally we project each data point on a 2D image to generate the "depth image". Such a procedure enjoys from the subpixel precise depth and RGB values. Data from [19] consists of camera calibration matrices, RGB images, and reconstructed 3D meshes. As these meshes are dense and not temporally consistent, we use the consistent meshes from [18]. To generate the required information from such a data, we model the physical cameras in a virtual scene using OpenGL. The deformation of the mesh, range flow and depth are again captured using the successive frames and a similar procedure mentioned above. Note that, because of errors in camera calibration, mesh capture and linear interpolation on the GPU, the ground truth obtained from the second data [19] is less reliable. Figure 2 shows snapshots of synthesized depth from both references.



(a) AAE vs Iterations



(b) RME vs Iterations

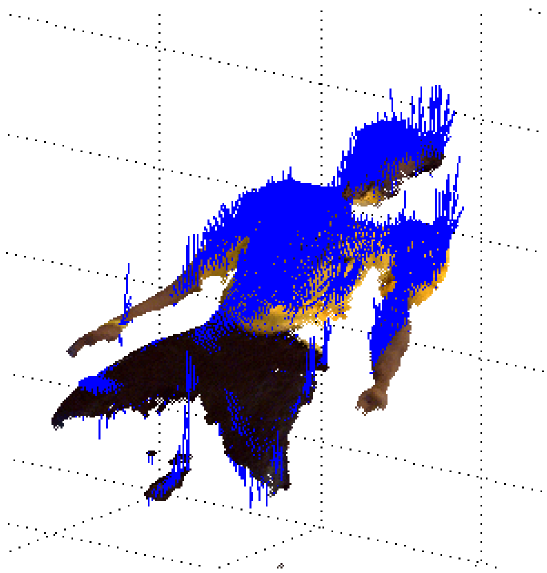
Fig. 3. Error Measurement for Synthetic Datasets

VI. Error Analysis And Evaluation

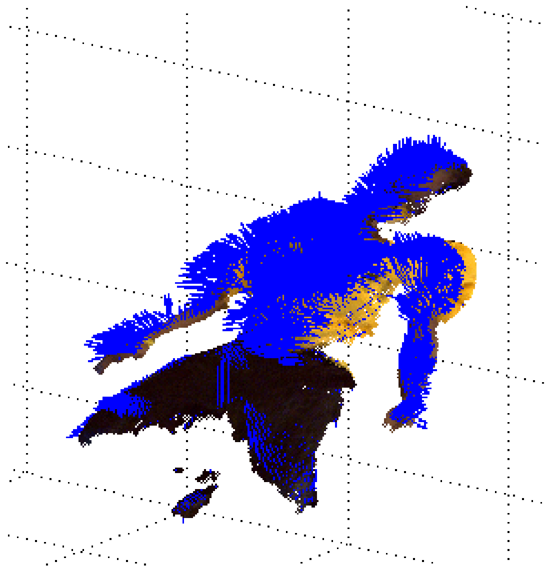
1) Quantitative Error Measurements for Range Flow: We extend the well known techniques optical flow error evaluations to 3D surface flows. We analyze two main error metrics: AAE(Average Angular Error), RME (Root Mean Squared Error).

Figure 3 shows the effect of the change in iterations and regularization parameter (α) to the range flow accuracy. (a) and (b) define AAE and RME for standard range flow versus our implementation. Note that $Ours - DM - Synth$ is the evaluation for dancing man data while $Spies - synth$ is the evaluation of the original algorithm of Spies on Salzmann's synthetic data and $Ours - Synth$ is the evaluation of our novel method on the same dataset. It is clearly noticed that there is an improvement to the standard range flow. On part (c), Figure 3 also demonstrates the quantitative results on Dancing Man Data.

2) Visualization Results & Errors: While quantitative evaluation is very important, it is always necessary to visualize the results. At Figure 4 we present the visual results obtained from sequence of dancing man. Vectors are plotted over the depth image, demonstrating the motion in 3D. Furthermore, Figure 5 demonstrates the performance of range flow on real data. Default parameters as mentioned above are used and median filtering is applied.



(a) Results on Dancing Man with $alpha = 3.0$



(b) Results on Dancing Man with $alpha = 4.5$

Fig. 4. Range Flow On Dancer Head: Different Regularizations

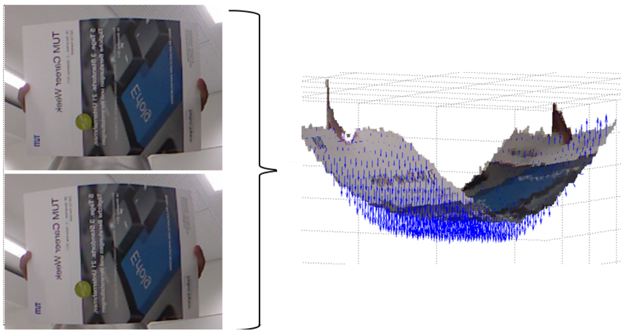


Fig. 5. Range Flow Between 25^{th} and 26^{th} Frames Of Real Data

VII. Conclusion

In this paper we introduce a framework for automatic 3D surface deformation capture. The algorithm starts with automatic initialization based on segmentation, of the area of interest to be tracked. This surface is then tracked using an extended and robustified range flow algorithm. Finally, we used a well studied mass spring model for representing the deforming surface. An experimental evaluation includes a pipeline for synthetic data generation and results demonstrate better performance in comparison to the original range flow approach. The method performs accurate and precise enough to track significant deformations smoothly.

References

[1] H. Spies and B. Jhne, "Dense range flow from depth and intensity data," in *In ICPR*, 2000, pp. 131–134.

[2] B. Lucas and T. Kanade", ""an iterative image registration technique with an application to stereo vision (darpa)"," in *"Proceedings of the 1981 DARPA Image Understanding Workshop"*, "April" "1981", pp. "121–130".

[3] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *ARTIFICAL INTELLIGENCE*, vol. 17, pp. 185–203, 1981.

[4] H. Spies, B. Jähne, and J. L. Barron, "Range flow estimation," *Computer Vision and Image Understanding*, vol. 85, no. 3, pp. 209–231, 2002.

[5] A. Hilsmann and P. Eisert, "Deformable object tracking using optical flow constraints," 2007.

[6] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, pp. 475–480, March 2005.

[7] F. Huguet and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *Proc. Intl. Conf. on Computer Vision*. Rio de Janeiro, Brasil: IEEE, Oct. 2007. [Online]. Available: http://hal.inria.fr/inria-00262139

[8] M. Salzmann, J. Pilet, S. Ilic, and P. Fua, "Surface deformation models for nonrigid 3d shape recovery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 1481–1487, August 2007. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2007.1080

[9] C. Wang, X. Li, and Y. Liu, "Monocular 3d tracking of deformable surfaces using linear programming," in *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ser. ICPR '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 1710–1713. [Online]. Available: http://dx.doi.org/10.1109/ICPR.2010.423

[10] B. Petit, A. Letouzey, E. Boyer, and J.-S. Franco, "Surface flow from visual cues," in *Vision, Modeling and Visualization Workshop*, Berlin, Allemagne, Oct. 2011.

[11] B. P. Antoine Letouzey and E. Boyer, "Scene flow from depth and color images," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2011, pp. 46.1–46.11, http://dx.doi.org/10.5244/C.25.46.

[12] M.-H. Hung, J.-S. Pan, and C.-H. Hsieh, "Speed up temporal median filter for background subtraction," *Pervasive Computing, Signal Porcessing and Applications, International Conference on*, vol. 0, pp. 297–300, 2010.

[13] R. Tarjan, "Depth-First Search and Linear Graph Algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, 1972.

[14] T. Lukins, , T. C. Lukins, and R. B. Fisher, "Colour constrained 4d flow," in *In Proceedings of the British Machine Vision Conference*, 2005, pp. 340–348.

[15] A. Bruhn, J. Weickert, and C. Schnrr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, pp. 211–231, 2005.

[16] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *In Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2010, pp. 2432–2439. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2010.5539939

[17] M. Salzmann, "Datasets for deformable surface reconstruction," Dec. 2008. [Online]. Available: http://http://cvlab.epfl.ch/data/dsr/

[18] C. Cagniart, E. Boyer, and S. Ilic, "Probabilistic deformable surface tracking from multiple videos," in *ECCV*, 2010.

[19] J. Starck, "Surface capture for performance-based animation," vol. 27, 2007.