TEXTURE OVERLAY ONTO NON-RIGID SURFACE USING COMMODITY DEPTH CAMERA

Tomoki Hayashi, Francois de Sorbier and Hideo Saito
Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Japan

Keywords: Deformable 3-D Registration, Principal Component Analysis, Depth Camera, Augmented Reality.

Abstract: We present a method for overlaying a texture onto a non-rigid surface using a commodity depth camera. The depth cameras are able to capture 3-D data of a surface in real-time, and have several advantages compared with methods using only standard color cameras. However, it is not easy to register a 3-D deformable mesh to a point cloud of the non-rigid surface while keeping its geometrical topology. In order to solve this problem, our method starts by learning many representative meshes to generate surface deformation models. Then, while capturing 3-D data, we register a feasible 3-D mesh to the target surface and overlay a template texture onto the registered mesh. Even if the depth data are noisy or sparse, the learning-based method provides us with a smooth surface mesh. In addition, our method can be applied to real-time applications. In our experiments, we show some augmented reality results of texture overlay onto a non-textured T-shirt.

1 INTRODUCTION

Recent progress in computer vision significantly extended the possibilities of augmented reality, a field that is quickly gaining popularity. Augmented reality is a young field that can be applied to many domains like entertainment and navigation (Azuma, 1997).

For clothes retail industry, examples of virtual clothes fitting system have been presented. In these systems, users can try on clothes virtually. It can be applied to a tele-shopping system over the internet, clothes designing, etc.

In order to realize such a virtual fitting using a monocular 2-D camera, many methods, that register a deformable mesh onto user's clothes and map the clothes texture to the registered mesh, have been presented (Ehara and Saito, 2006) (Pilet et al., 2007) (Hilsmann and Eisert, 2009). For the deformable mesh registration, they need that rich textures or the silhouette of the clothes can be extracted.

In the last few years, a new kind of depth cameras has been recently released with a reasonable price. By utilizing the 3-D data captured by the depth camera, some industrial virtual cloth fitting systems have been presented. However, those systems roughly, or just do not, consider the shape of the clothes that a user wants to wear. On the contrary, there are some methods which register a 3-D deformable mesh onto captured depth data of a target surface. Although those registration methods are very accurate, most of them require high processing time and are then not suitable for real-time applications.

In this paper, we present a real-time method that registers a 3-D mesh and overlays a template texture onto a non-rigid target surface like a T-shirt. Our method consists of an off-line phase and an on-line phase. In the off-line phase, we generate a number of representative sample meshes by exploiting the inextensibility of each edge of the triangles. Then the PCA (Principal Component Analysis) is applied for reducing the dimensionality of the mesh. In the on-line phase, we quickly estimate few parameters for generating the mesh according to input depth data. The target region where the template texture should be overlaid is defined by few color markers. Finally, the generated mesh is registered onto the target surface and the template texture is mapped to the registered mesh.

There are some contributions in our research. First, we overlay a texture which has a feasible shape onto a non-rigid surface captured by a commodity depth camera. Even though input depth data is noisy, our method can generate a natural and smooth shape. Second, we also do not need to use any texture to generate the surface mesh that fits the real shape. Finally, we achieve a real-time process by taking the advantage of the PCA that is a simple method of reducing the dimension of meshes.

Hayashi T., de Sorbier F. and Saito H..
TEXTURE OVERLAY ONTO NON-RIGID SURFACE USING COMMODITY DEPTH CAMERA.
DOI: 10.5220/0003867200660071
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2012), pages 66-71
ISBN: 978-989-8565-04-4

2 RELATED WORKS

Traditionally, methods that aim at overlaying a texture onto a non-rigid surface are applying a two dimensional or three dimensional deformable model reconstructed from a commodity color camera.

2-D Deformable Model. Pilet *et al.* have presented a feature-based fast method which detects and tracks deformable objects in monocular image sequences (Pilet et al., 2007). They applied a wide baseline matching algorithm for finding correspondences. Hilsmann and Eisert proposed a real-time system that tracks clothes and overlays a texture on it by estimating the elastic deformations of the cloth from a single camera in the 2D image plane (Hilsmann and Eisert, 2009). Self-occlusions problem is addressed by using a 2-D motion model regularizing an optical flow field. In both of these methods, the target surface requires a rich texture in order to perform a tracking.

3-D Deformable Model. Several methods are taking advantage of a 3-D mesh model computed from a 2-D input image for augmenting a target surface. Shen *et al.* recovered the 3-D shape of an inextensible deformable surface from a monocular image sequence (Shen et al., 2010). Their iterative L_2-norm approximation process computes the non-convex objective function in the optimization. The noise is reduced by applying a L_2-norm on re-projection errors. Processing time is, however, too long to satisfy a practical system due to their iterative nature.

Salzmann *et al.* generated a deformation mode space from the PCA of sample triangular meshes (Salzmann et al., 2007). The non-rigid shape is then expressed by the combination of each deformation mode. This step does not need an estimation of an initial shape or a tracking. Later, they achieved the linear local model for a monocular reconstruction of a deformable surface (Salzmann and Fua, 2011). This method reconstructs an arbitrary deformed shape as long as the homogeneous surface has been learned previously.

Perriollat *et al.* presented the reconstruction of an inextensible deformable surface without learning the deformable model (Perriollat et al., 2010). It achieves fast computing by exploiting the underlying distance constraints to recover the 3-D shape. That fast computing can realize augmented reality application. Note that most of those approaches require correspondences between a template image and an input image.

Depth Cameras. Recent days, depth cameras have been becoming popular and many researchers have been focusing on the deformable model registration using it (Li et al., 2008) (Kim et al., 2010) (Cai et al., 2010). The depth camera has a big advantage against a standard camera because it captures the 3-D shape of the target surface with no texture.

Amberg *et al.* presented a method which extends the ICP (Iterative Closest Point) framework to non-rigid registration (Amberg et al., 2007). The optimal deformation can be determined accurately and efficiently by applying a locally affine regularization. Drawback of this method is that the processing cost increases due to the iterative process. Papazov and Burschka proposed a method for deformable 3-D shape registration by computing shape transitions based on local similarity transforms (Papazov and Burschka, 2011). They formulated an ordinary differential equation which describes the transition of a source shape towards a target shape. Even if this approach does not require any iterative process, it still requires a lot of computational time.

In addition, we are aware that most methods using a depth camera assume that the input depth data is ground truth. Therefore, they may result in an unnatural surface if the depth data is noisy.

3 TEXTURE OVERLAY ONTO NON-RIGID SURFACE

In this section, we describe our method to overlay a texture onto a non-rigid surface. Fig. 1 illustrates the flow of our method. First, in the off-line phase, we generate deformation models by learning many representative meshes. That deformation models were proposed by Salzmann *et al.* (Salzmann et al., 2007). Because the dimension of the mesh in the model is low, we can quickly generate an arbitrary deformable mesh to fit the target surface in the on-line phase. In addition, thanks to the models, even though the input data is noisy, we can generate a natural mesh that has smooth shape.

In Salzmann's method, the iterative processing is required because it is not easy to generate a 3-D mesh only from a 2-D color image. In our case, we can generate a 3-D mesh directly by taking advantage of 3-D data from a depth camera.

3.1 Surface Deformation Models Generation

In the off-line phase, we generate the deformation models by learning several representative sample meshes. This part is based on Salzmann's method (Salzmann et al., 2007) that can reduce dras-

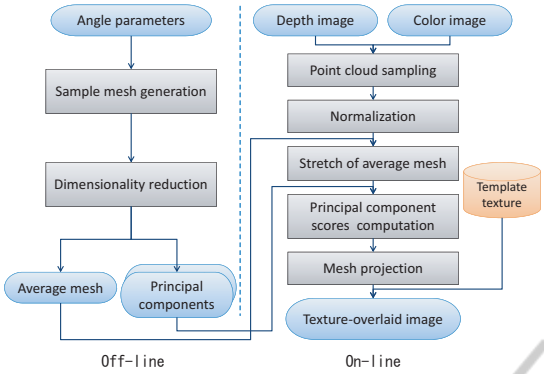


Figure 1: Flow of our method.

tically the number of degrees of freedom (dofs) of a mesh by assuming that the original length of the edge is constant and utilizing PCA.

In our approach, the target surface and the template texture are rectangular, so we introduce a rectangular surface mesh made of $m = M \times N$ vertices $\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_m\} \subset \mathbb{R}^3$.

3.1.1 Sample Mesh Generation

Thanks to Salzmann's work, we can generate sample meshes that are variously deformed by setting small angle parameters. The number of the parameters is considerably smaller than $3 \times m$ that is the original dofs of the mesh $\mathbf{V}$.

We randomly constrained the range of the angle parameters to $[-\pi/8, \pi/8]$ and discarded the generated meshes that may still not preserve the topology.

Finally, all the sample meshes are aligned in order to uniformize the result of the PCA.

3.1.2 Dimensionality Reduction

For more sophisticated expression of the mesh, Salzmann proposed the method to conduct the dimensionality reduction by running the PCA on the sample meshes presented in Sec. 3.1.1. As a result, we can get the average mesh $\bar{\mathbf{V}} = \{\mathbf{y}_1, ..., \mathbf{y}_m\} \subset \mathbb{R}^3$ and N_c principal components $\mathbf{P} = \{\mathbf{p}_1, ..., \mathbf{p}_m\} \subset \mathbb{R}^3$ which represent some deformation modes. Then an arbitrary mesh can be expressed as follows:

$$\mathbf{V} = \bar{\mathbf{V}} + \sum_{k=1}^{N_c} \omega_k \mathbf{P}_k \qquad (1)$$

where $\mathbf{V}$ is the vertices of the target surface mesh that we want to generate, ω_k denote k^{th} principal component score or weights, and $\mathbf{P}_k$ denote the corresponding principal components or deformation modes. N_c is the number of the principal components, which is determined by looking at the contribution rate of the PCA. For example, setting N_c to 40 principal components is enough to reconstruct over 98% of the original shape. Any mesh can then be expressed as a functions of the vector: $\Theta = \{\omega_1, ..., \omega_{N_c}\}$. Once Θ is known, the surface mesh can be easily reconstructed using Eq. 1.

In the following section, we explain our method to generate a 3-D mesh from an input depth image and to overlay a texture onto a non-rigid surface by using the surface deformation models.

3.2 Mesh registration

The goal in the on-line phase is to overlay a template texture $\mathcal{S}$ onto a target surface $\mathcal{T}$ by using a color image and its corresponding depth image. In order to create the mesh onto which the texture is mapped, we need to estimate the optimal principal component score vector Θ. In general, the principal component score ω_k is described as:

$$\omega_k = (\mathbf{V} - \bar{\mathbf{V}}) \cdot \mathbf{P}_k \qquad (2)$$

where $\bar{\mathbf{V}}$, $\mathbf{P}_k$ and $\mathbf{V}$ were defined in Eq. 1.

This equation means that ω_k will be higher if $(\mathbf{V} - \bar{\mathbf{V}})$ is similar to $\mathbf{P}_k$. In that case, $\mathbf{P}_k$ considerably affects the shape of the generated surface mesh, and vice versa. The generated surface mesh will then receive the template texture and will be overlaid onto the target surface.

3.2.1 Point Cloud Sampling

Eq. 2 means that each data needs to have the same dimension. Although $\mathbf{V}$ is unknown, the input point cloud of the target surface $\mathcal{T}$ is useful as a good candidate to replace $\mathbf{V}$. The simplest idea is to set $\mathbf{V}$ by finding the corresponding points between the point cloud of $\mathcal{T}$ and the vertex coordinates of $\bar{\mathbf{V}}$. However, $\mathcal{T}$ is a big data set without any special order, implying that the computational time may become high.

Therefore we sample the point cloud of $\mathcal{T}$ to match its dimension to the dimension of $\mathbf{V}$. The sampling is done on the input depth image by using color markers. We have eight points defined by eight color markers on $\mathcal{T}$ and add an additional point which is a centroid of them. Based on their image coordinates, we sample the region covered by the color markers such that the number of vertex becomes the expected sampling resolution N_D. N_D is set to m that has the same dimension as $\mathbf{V}$. Each sampled coordinate are computed as linear interpolation of the image coordinates of adjacent 4 points as presented in Fig. 2.

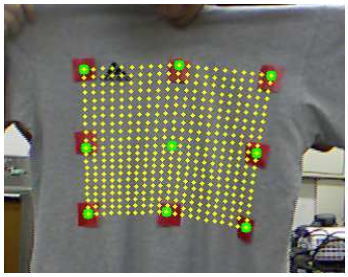


Figure 2: Markers and sampled coordinates. Green points on the surface border denote the markers that are detected based on its color. The center green point is an average coordinate of the green points on the surface border. Yellow points are the sampled coordinates.

3.2.2 Normalization

Even if the dimension of the sampled point cloud has become same as the one of $\bar{\mathbf{V}}$, it can not still be used in Eq. 2 because the scale and the orientation of the point cloud is different from those of $\bar{\mathbf{V}}$ and $\mathbf{P}_k$. In order to match them, we define a normalized coordinate system and a rigid transformation matrix $\mathbf{M}$ which transforms the data from the world coordinate system to the normalized system.

$\mathbf{M}$ is computed using the captured depth data. We first define a rigid transformation which makes all sampled points of $\mathcal{T}$ to fit the coordinates of the undeformed mesh. The undeformed mesh is aligned in the normalized system so that its four corner vertices correspond to the coordinates $\left(-\frac{W-1}{2}, -\frac{H-1}{2}, 0\right)$, $\left(\frac{W-1}{2}, -\frac{H-1}{2}, 0\right)$, $\left(-\frac{W-1}{2}, \frac{H-1}{2}, 0\right)$ and $\left(\frac{W-1}{2}, \frac{H-1}{2}, 0\right)$. Note that W and H respectively represent the width and the height of the undeformed mesh.

The estimation of $\mathbf{M}$ is done by a least-squares fitting method. The sampled point cloud of $\mathcal{T}$ is normalized as $\mathbf{V}'$ of $\mathcal{T}_N$ by $\mathbf{M}$. The alignment of $\bar{\mathbf{V}}$ and $\mathbf{P}_k$ in the normalized system can be pre-processed during the stage presented in the Sec. 3.1.1.

3.2.3 Stretch of Average Mesh

The x and y coordinates of $\mathbf{V}'$ do not match the ones of $\bar{\mathbf{V}}$ in the normalized coordinate system when $\mathcal{T}$ is deformed. If their difference is too big, we can not reconstruct the optimal ω_k. Therefore we stretch the shape of $\bar{\mathbf{V}}$ to roughly match the one of $\mathbf{V}'$.

The stretch of $\bar{\mathbf{V}}$ is applied on the XY plane direction in the normalized coordinate system. The vertex coordinates of $\bar{\mathbf{V}}$ which correspond to the color markers are transformed to the marker's x and y coordinates while keeping each z coordinate. On top of that, those stretching transformation vectors are used for the other coordinates of $\bar{\mathbf{V}}$. The remaining coordinates are transformed by applying a weight on the previously computed vectors. Each weight is pre-processed based on the square distance between the viewing coordinate of $\mathbf{V}$ and each vertex coordinate corresponding to the markers of $\bar{\mathbf{V}}$. The resulting vertex coordinates are expressed by $\bar{\mathbf{V}}'$. As a result of the stretching, x and y coordinates of $\mathbf{V}'$ and $\bar{\mathbf{V}}'$ become similar.

3.2.4 Principal Component Scores Computation

Thus, we can adapt Eq. 2 to:

$$\omega_k = (\mathbf{V}' - \bar{\mathbf{V}}') \cdot \mathbf{P}_k. \tag{3}$$

Because each ω_k that is calculated in the Eq. 3 is applicable to $\bar{\mathbf{V}}'$, we generate the mesh using this new equation:

$$\mathbf{V} = \bar{\mathbf{V}}' + \sum_{k=1}^{N_c} \omega_k \mathbf{P}_k. \tag{4}$$

Then we get the mesh $\mathbf{V}$ corresponding to $\mathcal{T}_N$.

3.2.5 Mesh Projection

Once we generate $\mathbf{V}$, the last stage is to transform it to the world coordinate system. Because we already know transformation $\mathbf{M}$ from the world coordinate system to the normalized coordinate system, we can transform $\mathbf{V}$ by using $\mathbf{M}^{-1}$.

For the rendering, we define the texture coordinates for each vertex of a surface mesh. Therefore, the texture is overlaid on the target surface obtained by $\mathbf{M}^{-1}\mathbf{V}$.

4 EXPERIMENTAL RESULTS

All the experimental results have been done on a computer composed of a 2.50 GHz Intel(R) Xeon(R) CPU and 2.00 GB RAM. We use the depth camera Microsoft Kinect with an image resolution of 640×480 pixels and a frame rate of 30 Hz. The target surface is a region of a T-shirt without any textures, but defined by eight basic color markers. For the resolution of the rectangular mesh, we set both M and N to 21 vertices.

4.1 Depth Comparison

We evaluated our method by the comparison of depth value between the registered mesh and the depth camera. We plot the depth data of the depth camera and the registered mesh in the direction of the horizontal image axis in Fig. 3.

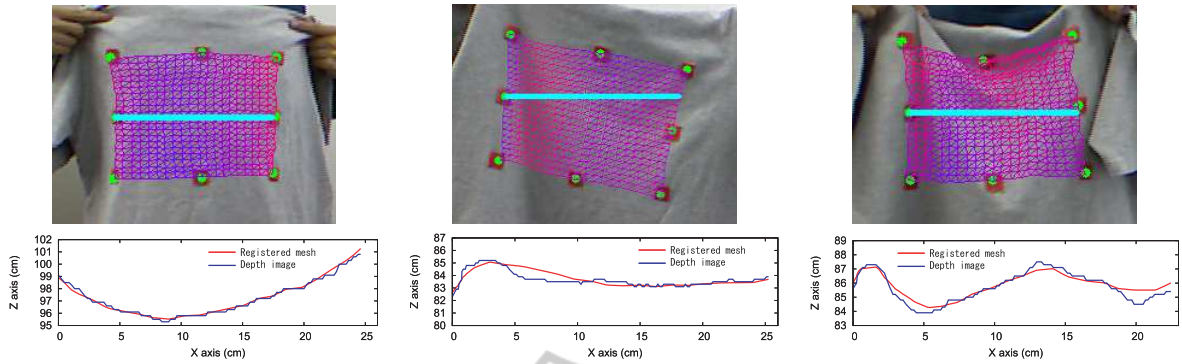


Figure 3: Plot of depth data of depth camera and generated mesh. The blue points in the top row images denote the positions which are used for plotting. The bottom row images are the plot of the depth data.

The registered mesh is deformed following the depth data. Despite of discrete depth data from the depth camera, our method can generate similar and smooth shape.

4.2 Processing Time

We calculated the computational time because our method is supposed to be used for a real-time application. The result of the average computational time in 100 frames is shown in Table 1. Note that the computational time of the mesh registration including the sampling, the normalization and the principal component scores computation is quite small. As a whole, the average processing speed was over 25 frames per second.

Table 1: Processing time.

Task	Time(msec)
Capturing	11
Marker Detection	11
Mesh Registration	7
Image Rendering	4

4.3 Visualization of Mesh Registration and Texture Overlay

Finally, we illustrate the visualization result of the texture overlay onto the non-rigid target surface of the T-shirt as augmented reality in Fig. 4. Even if the target has no texture inside the target region, our mesh is deformed to fit the surface according to the data obtained by the depth image.

5 DISCUSSION

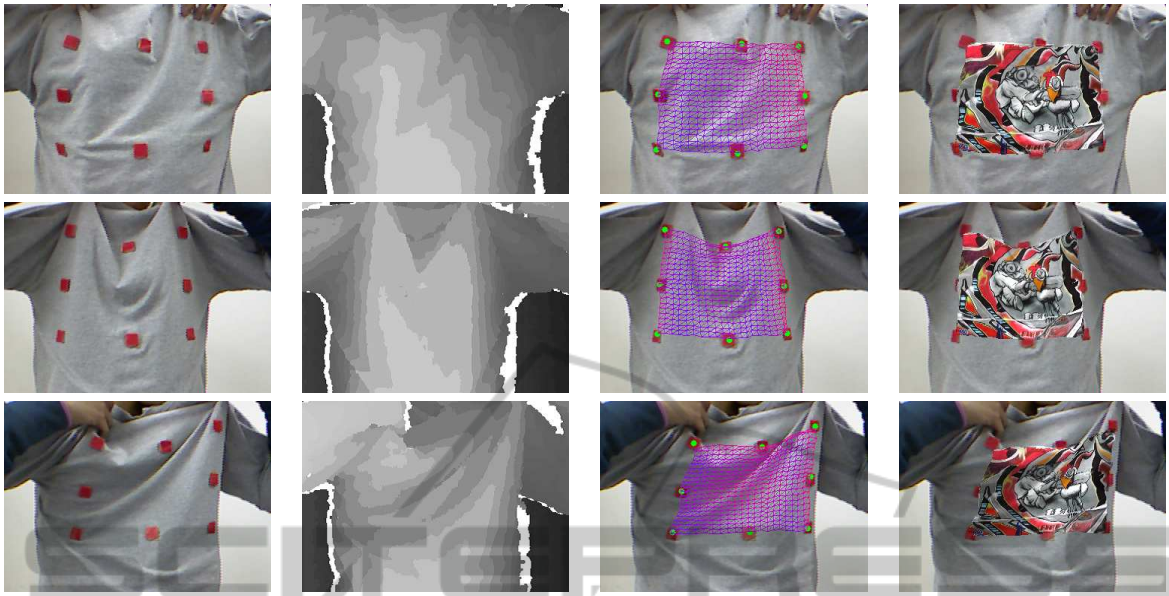
Following the description about our method and experimental results, we summarize the characteristics of our method. In terms of the processing speed as the principal advantage of our method, we achieved quite high processing speed thanks to the reduction of the dofs of the mesh and the non-iterative mesh registration method. Since we also regard the mesh registration accuracy is sufficient, our method can be applicable to a practical virtual fitting system. Moreover, since the mesh registration is based on the deformation models, our method is robust to the noise of the input depth image.

On the other hand, if the cycle of the spatial frequency of the target surface is shorter than the sampling interval of the mesh, we can not generate a mesh having appropriate shape. Although this problem is supposed to be solved by shrinking the sampling interval, additional processing will be required because average mesh needs to be sampled.

Basically, the deformation models produce not an optimal but a broken mesh in case that the target surface is deformed more sharply than the learning angle. To attack this problem, it might be effective to learn much more varieties of the representative meshes.

6 CONCLUSIONS AND FUTURE WORK

We presented a registering method of the 3-D deformable mesh using a commodity depth camera for a texture overlay onto a non-rigid surface. Our method has several advantages. First, it is not required to attach a rich texture onto the target surface by taking the advantage of using a depth camera. Second, the PCA



(a) (b) (c) (d)

Figure 4: Some visualization results. All images are cropped for the visualization and the shadow is reflected from pixel value of a color image. (a): Input color images. (b): Input depth images. (c): Registered mesh. (d): Texture overlay onto the target T-shirt.

enables us to generate a feasible shape mesh even if the depth data are not very accurate or noisy. Furthermore the PCA reduced the dofs of the mesh and helps to obtain a real-time processing.

As a future work, we will implement the sampling of the depth image using GPU shader programming since it will provide a more dense sampling and then more precise registration with faster processing. In addition, we expect to replace our template mesh by a T-shirt model in order to achieve a global virtual T-shirt overlaying and remove the color markers.

ACKNOWLEDGEMENTS

This work was partially supported by Grant-in-Aid for Scientific Research (C) 21500178, JSPS.

REFERENCES

Amberg, B., Romdhani, S., and Vetter, T. (2007). Optimal Step Nonrigid ICP Algorithms for Surface Registration. In *Proc. CVPR*, pages 1–8. Ieee.

Azuma, R. (1997). A survey of augmented reality. *PTVE*, 6(4):355–385.

Cai, Q., Gallup, D., Zhang, C., and Zhang, Z. (2010). 3D Deformable Face Tracking with a Commodity Depth Camera. In *Proc. ECCV*, number 2.

Ehara, J. and Saito, H. (2006). Texture overlay for virtual clothing based on PCA of silhouettes. In *Proc. ISMAR*, pages 139–142. Ieee.

Hilsmann, A. and Eisert, P. (2009). Tracking and Retexturing Cloth for Real-Time Virtual Clothing Applications. In *Proc. MIRAGE*, pages 1–12.

Kim, Y. S., Lim, H., Kang, B., Choi, O., Lee, K., Kim, J. D. K., and Kim, C.-y. (2010). Realistic 3d face modeling using feature-preserving surface registration. In *Proc. ICIP*, pages 1821–1824.

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.

Papazov, C. and Burschka, D. (2011). Deformable 3D Shape Registration Based on Local Similarity Transforms. *Computer Graphics Forum*, 30(5):1493–1502.

Perriollat, M., Hartley, R., and Bartoli, A. (2010). Monocular Template-based Reconstruction of Inextensible Surfaces. *IJCV*, 95(2):124–137.

Pilet, J., Lepetit, V., and Fua, P. (2007). Fast Non-Rigid Surface Detection, Registration and Realistic Augmentation. *IJCV*, 76(2):109–122.

Salzmann, M. and Fua, P. (2011). Linear local models for monocular reconstruction of deformable surfaces. *PAMI*, 33(5):931–44.

Salzmann, M., Pilet, J., Ilic, S., and Fua, P. (2007). Surface deformation models for nonrigid 3D shape recovery. *PAMI*, 29(8):1481–7.

Shen, S., Shi, W., and Liu, Y. (2010). Monocular 3-D tracking of inextensible deformable surfaces under L(2) - norm. *image processing*, 19(2):512–21.