

# Tracking and Retexturing Cloth for Real-Time Virtual Clothing Applications

Anna Hilsmann and Peter Eisert

Fraunhofer Heinrich Hertz Institute,  
Einsteinufer 37, 10587 Berlin, Germany  
{anna.hilsmann,peter.eisert}@hhi.fraunhofer.de  
<http://iphome.hhi.de/>

**Abstract.** In this paper, we describe a dynamic texture overlay method from monocular images for real-time visualization of garments in a virtual mirror environment. Similar to looking into a mirror when trying on clothes, we create the same impression but for virtually textured garments. The mirror is replaced by a large display that shows the mirrored image of a camera capturing e.g. the upper body part of a person. By estimating the elastic deformations of the cloth from a single camera in the 2D image plane and recovering the illumination of the textured surface of a shirt in real time, an arbitrary virtual texture can be realistically augmented onto the moving garment such that the person seems to wear the virtual clothing. The result is a combination of the real video and the new augmented model yielding a realistic impression of the virtual piece of cloth.

**Key words:** Cloth Tracking and Augmentation, Augmented Reality, Virtual Clothing

## 1 Introduction

Merging computer generated objects with real video sequences is an important technique for many applications such as special effects in movies and augmented reality applications. In many cases this is time-consuming and processing is done off-line. Real-time applications would make processing easier and open new areas of application. We developed a method for real-time dynamic texture overlay for garments. A single camera captures a moving person in front of a display that shows the mirrored input image. Camera and display are mounted such that the user has the impression of looking into a mirror showing his upper part of the body. Color and texture of the moving garment are exchanged by a virtual one, such that the user seems to wear the virtual cloth in the mirror (see Figure 1). For that purpose we developed a dynamic retexturing method for garments that estimates elastic movements of a surface from monocular images and renders a virtual texture onto the moving garment with correct deformation and realistic illumination.

Deformable tracking and retexturing of surfaces from monocular sequences without 3D reconstruction of the surface encounters the problem of self-occlusions during tracking and realistic lighting during retexturing. We propose a real-time method based on optical-flow that can handle those problems. We address the important problem of



**Fig. 1.** The Virtual Mirror system replaces color and texture of a t-shirt in real time.

self-occlusions during tracking of deformations in 2D using a two-dimensional motion model to regularize the optical flow field and define smoothing constraints locally according to the self-occlusion of a region. Occlusion estimates are established from shrinking regions in the mesh. In order to achieve realistic lighting, we exploit the fact that the original image contains all shadows and illumination to be rendered onto the virtual texture. Hence, shadows and wrinkles are realistically recovered from the original video frame using a simple inpainting algorithm. This enhances the realistic appearance of the augmented garment. Additionally, we segment the shirt and can change the color of the non-textured part of the apparel.

One important characteristic of our system is that we achieve very realistic results in real-time with simple hardware, like a regular PC, a single camera and a display. The deformation estimation is very robust and correct illumination and shading enhance the realistic experience such that the user has the impression of actually wearing the augmented shirt.

The remainder of this paper is structured as follows. Section 2 briefly sums up the related work in garment tracking and retexturing and states the contribution of this paper. Section 3 describes our system as a whole. Section 4 explains the automatic detection and segmentation of the shirt. Section 5 describes the tracking approach for deformable surfaces in monocular sequences used for garment tracking. Section 6 describes the recovery of shading and illumination and the rendering approach. Section 7 presents experimental results.

## 2 Related Work and Contribution

Garment tracking and retexturing has been addressed lately by a number of researchers. Most researchers focused on multi-view systems [11], [4], [15], [19] for cloth tracking but little research has been done in real-time garment tracking from monocular sequences.

Pritchard and Heidrich [11] introduced a feature-based approach to cloth motion capturing with a calibrated stereo camera pair to acquire 3D. They recovered geometry from stereo correspondences using SIFT features. Scholz and Magnor [15] presented a method that uses optical flow to calculate three dimensional scene flow in a multi-view system. They use a cloth model that is matched to the surface, minimizing the deformation energy of the patch. Drift is countered by constraining the edge of the simulation

to the silhouette of the real cloth. Guskov *et al.* [4] introduced a multi-view real-time system based on color-coded markers with a limited size of codewords. Their system needs small motions and the markers have to be quite large. Scholz *et al.* [16] improved upon Guskov *et al.* by using a color-code with more codewords using a pseudo random colored dots pattern. The algorithm identifies the exact position of colored ellipses on the cloth by examining their local neighborhood and makes use of a-priori knowledge on the surface connectivity. The three-dimensional geometry is reconstructed using a multi-view setup. Their approach achieves remarkable results and can cope with fast motion. White *et al.* [21] presented an extension to the approach based on a stereo-setup to reconstruct a random pattern of colored triangles printed on a cloth. Hasler *et al.* [5] used an analysis-by-synthesis approach and they, too, rely on markers in the form of the patterned cloth. Recently, Bradley *et al.* [1] introduced a marker-free approach to capturing garments from sixteen views. However, their method needs some user-assisted key-framing.

All these multi-view methods yield impressive results in 3D reconstruction. However, they need sophisticated acquisition setups and have a high computational cost. We strive for a different goal, i.e. to create an augmented reality in real-time with as little hardware as possible. For that purpose we are interested in deformable tracking from single-view video. Little research has been done in monocular garment tracking and augmentation. One approach in tracking deformable surfaces in monocular sequences is to formulate deformations in 3D [13], [18]. However, recovering 3D position and deformation from monocular surfaces is an ill-posed problem. Another approach is to make use of deformation models in 2D and track the elastic deformations in the image plane. In this case special care has to be taken considering self-occlusion e.g. if a surface is bent or deformed in a way that parts of it occlude other parts or due to the image projection. Furthermore, to retexture the surface shadows and illumination have to be recovered from single images.

Scholz and Magnor [14] modified their approach for multi-view setups using the color-code for garment tracking in monocular sequences. To recover shading maps they interpolate the color coded pattern using thin-plate splines. However, their system needs user interaction and the computation time is quite high.

White and Forsyth [20] presented a method for retexturing non-rigid objects from a single viewpoint without reconstructing 3D geometry using color markers. They limited their method to recover irradiance to screen printing techniques with a finite number of colors. Pilet *et al.* [9] proposed a feature-based method for deformable object detection and tracking in monocular image sequences that uses a wide baseline matching algorithm to establish correspondences. They continued their work by taking shadows and illumination into account by assuming Lambertian illumination in the reference frame [10]. Gay-Bellile *et al.* [3] proposed a direct non-rigid registration method and address the problem of self-occlusions by detecting them as shrinking areas in the 2D warp. The warp is forced to shrink by penalizing a variation in the sign of the partial derivatives of the warp along some directions. A binary decision excludes self-occluded pixels from consideration in the error function.

Some researchers have also addressed virtual clothing scenarios [12], [17], [2]. Saito and Hoshino [12] propose a technique to merge computer generated apparels with hu-

man video by 3D human pose estimation in off-line video sequences. Taguchi *et al.* [17] use two-dimensional plane models and the 2D shape of the garment to recover the 3D shape. Ehara and Saito [2] proposed a texture overlay method for virtual clothing systems. The deformation of the surface of the shirt was estimated from its silhouette in the input image based on a pre-collected database of a number of shapes of the same shirt. Their system needs homogeneous background and markers on the shirt. Unfortunately, they do not take illumination into account.

In [6] we presented an optical-flow based approach for single-view video using a mesh-based motion model together with smoothing constraints that are formulated locally on the mesh. Self-occlusions are taken into account by weighting the smoothness constraints locally according to the occlusion of a region. Thereby we force the mesh to shrink instead of fold in presence of self-occlusion.

The main contribution of this paper lies in a real-time tracking and retexturing method for elastically deforming surfaces in monocular image sequences. The method uses direct image information instead of features to track the deformation in the image plane. The markerless tracking approach can be applied to any surface that is rich enough in detail to exploit optical flow constraints if the initial state of the surface is known. Therefore, we combine this method with a simple detection method for initialization that is based only on a-priori knowledge of the color of the shirt and the knowledge that there is a textured region on the shirt. Additionally, we recover illumination and shadows from the original image. The new virtual texture is deformed in real-time in the 2D image plane and relighted with the recovered shading map. The result is a real-time system with little hardware that exchanges the color and texture of the shirt very realistically while the person in front of the system can move freely.

### 3 System Overview

The system consists of a regular PC, a single camera and a display that shows a combination of real video and computer generated content. The XGA-firewire camera is mounted on top of the display and captures the upper part of the user's body. The placement of the display and the viewing direction of the camera are chosen such that an average sized person sees about the same as he/she would expect when looking in a real mirror located at the same position as the display. Lights are mounted on either side of the display to be more independent from surrounding lighting conditions. As user interface we use a touchscreen on which the user can modify the augmentation result and choose between different colors and textures. In the test scenarios the user wears a green shirt with a rectangular drawing from Picasso as textured region (see Figure 2) to be more independent from the background. Our system does not require that specific pattern. The only assumption we make is that the shirt is green and there is a highly textured rectangular region on the shirt. Other suitable assumptions are possible. Once the line pattern is detected all its movements and deformations are tracked and replaced by any virtual texture while the person can move freely in front of the display. Additionally, we recover illumination and shadows from the original image using a simple inpainting method.



**Fig. 2.** Work Flow: Input image, mesh used for tracking, mirrored augmented output images (left to right). The texture on the shirt is exchanged with a new one and the shirt is re-colored.

The algorithm consists of several parts that all run on a single PC in real-time. Basically, we distinguish two modes. In the Detection Mode the shirt is detected and the mesh that is used for tracking is initialized. Once the shirt is detected the system switches into the Tracking Mode. This mode consists of several parts, that will be described in the following sections: Shirt segmentation and image preprocessing (Section 4), deformable surface tracking (Section 5) and illumination recovery and rendering (Section 6).

## 4 Shirt Segmentation and Image Preprocessing

During the Detection Mode the algorithm searches for the shirt in the image using the a-priori knowledge of the color of the shirt and the assumption that it contains a rectangular highly textured region. Additionally, in this mode the mesh that is used as motion model to estimate the deformation of the texture in the 2D image plane (see Section 5) is initialized. During the Tracking Mode we segment the shirt region in order to give it a new color. We assume that if the shirt is in the image, it is the largest area of that color and all parts of the shirt are connected. With these assumptions we can use a very simple but efficient approach to detect and segment the shirt in the image that is robust against illumination changes and changing background.

First, we transform all pixels in the image into the normalized RGB-colorspace which is more invariant to changes of surface orientation relatively to the light source. We find the largest blob by evaluating the normalized pixel values and fill all holes in that region. Using this very general method for segmentation ensures that all parts of the shirt are segmented, including dark shadows or highlighted regions. During the Tracking Mode we use this segmentation mask to recolor the untextured part of the shirt.

In the Detection Mode we proceed further by searching for a highly textured region in the detected green region to initialize the mesh used for tracking. We find and sample the contours of the texture defining the positions of the border mesh vertices and interpolate all inner vertex position points (see Figure 2). By this method we assure that if the texture is already deformed in the model frame, we get an idea about the deformation at the beginning knowing the rectangular shape of the undeformed texture. The mesh and the texture in this frame are used as model in the optical-flow-based tracking method (see Section 5).

## 5 Deformable Surface Tracking

Once the shirt has been detected and the mesh has been initialized we track the textured region of the shirt in an optical-flow based approach. We exploit the optical flow constraint equation [8] and regularize the optical flow field with a predefined mesh-based motion model. The best transformation can then be determined by minimizing a quadratic error:

$$E = \sum_{i=1}^n \left( \nabla I(x_i, y_i) \cdot \mathbf{d}(x_i, y_i) + \frac{\partial I}{\partial t}(x_i, y_i) \right)^2 \quad (1)$$

where  $\nabla I(x_i, y_i)$  denotes the spatial derivatives of the image  $I$  at pixel position  $[x_i, y_i]^T$  and  $\frac{\partial I}{\partial t}(x_i, y_i)$  denotes the temporal gradient between two images.  $\mathbf{d}(x_i, y_i)$  denotes the displacement vector at position  $[x_i, y_i]^T$  and is defined by the motion model described in Section 5.1.  $n$  is the number of pixels selected for contribution to the error function, i.e. pixels where the gradient is non-zero.

The optical flow equation has some requirements that have to be fulfilled. First, it is valid only for small displacements between two successive frames because it is derived assuming the image intensity to be linear. To account for larger displacements we use a hierarchical framework.

Second, the optical flow equation assumes uniform Lambertian illumination. Illumination changes are taken into account by using highpass-filtered images. By that, high frequencies like sharp edges used by the optical flow algorithm are preserved whereas low frequencies like soft intensity changes due to illumination changes are filtered out and the difference between the two frame are mostly due to motion.

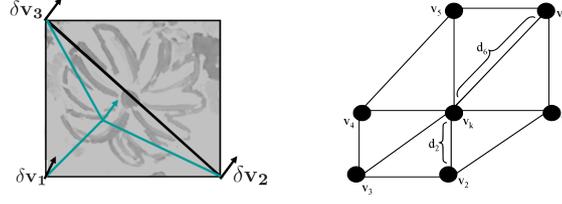
The aperture problem will be addressed by the motion model and the smoothing constraints in the following section.

Drift is countered by using the frame, in which the mesh was initialized, as model represented by the initial mesh. For each frame we estimate the elastic motion from the model frame, that is warped with the previous estimate of the previous frame, to the current frame. Using this model-to-frame tracking approach instead of a frame-to-frame tracking approach avoids error accumulation in the estimation and allows us to recover from small inaccuracies.

### 5.1 Motion Parameterization

The optical flow equation provides one equation for two unknowns, i.e. the displacement in  $x$ - and  $y$ - direction at each pixel position. To overcome this problem we need additional constraints on the pixel displacement. Therefore, we introduce a 2D triangulated regular mesh with  $K$  vertices  $\mathbf{v}_k$ , ( $k = 1 \dots K$ ), as motion model to regularize the optical flow field. The position of each vertex  $\mathbf{v}_k$  is given by its image coordinates  $\mathbf{p}_k = [x_k, y_k]^T$ . Each pixel  $\mathbf{p}_i = [x_i, y_i]^T$  in the image can be represented by the barycentric coordinates of the enclosing three vertices.

$$[x_i, y_i]^T = \sum_{\substack{j=1 \\ \mathbf{v}_j \in \mathbf{v}_k}}^3 B_{jt}(x_i, y_i) \cdot \mathbf{v}_j, \sum_{j=1}^3 B_j(x_i, y_i) = 1, 0 \leq B_j \leq 1 \quad (2)$$



**Fig. 3.** Pixel displacement parameterization used as regularization of the optical flow field (left) and neighborhood of a vertex in the model (right).

where  $B_j(x_i, y_i)$ , ( $j = 1, 2, 3$ ) are the three barycentric coordinates of pixel  $[x_i, y_i]^T$  and  $\mathbf{v}_j$  are the three enclosing vertices. With a mesh deformation,  $[x_i, y_i]^T$  is mapped onto  $[x'_i, y'_i]^T$ . Thus, we are looking for a deformation of the mesh, i.e. a displacement of each vertex  $\mathbf{v}_k$  to  $\mathbf{v}'_k$  such that the barycentric coordinates of  $[x'_i, y'_i]^T$  are those of  $[x_i, y_i]^T$ . The deformation model then yields [7]:

$$\mathbf{d}(x_i, y_i) = \sum_{\substack{j=1 \\ t_j \in t_k}}^3 B_j(x_i, y_i) \cdot \delta \mathbf{v}_j \quad (3)$$

where  $\delta \mathbf{v}_j$  are the three vertex displacements of the enclosing triangle. Inserting the motion models into equation (1) leads to an overdetermined linear equation system that can be efficiently solved in a linear least-squares sense.

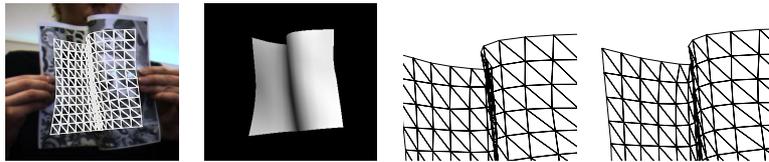
Additionally, we incorporate smoothing constraints for the vertex displacement field. This yields the following error functional to be minimized:

$$E = \sum_{i=1}^n \left( \nabla I(x_i, y_i) \cdot \mathbf{d}(x_i, y_i) + \frac{\partial I}{\partial t}(x_i, y_i) \right)^2 + \lambda \sum_{k=1}^K w_k E_s(\delta \mathbf{v}_k) \quad (4)$$

where  $K$  is the number of vertices and  $\lambda$  is the regularization parameter.  $E_s(\delta \mathbf{v}_k)$  is a local smoothing function for the displacement  $\delta \mathbf{v}_k$  of a vertex  $\mathbf{v}_k$  between two successive frames weighted by  $w_k$ . We choose  $E_s(\delta \mathbf{v}_k)$  to be

$$E_s(\delta \mathbf{v}_k) = \left( \delta \mathbf{v}_k - \frac{1}{\sum_{n \in N_k} \frac{1}{d_n}} \sum_{n \in N_k} \frac{1}{d_n} \cdot \delta \mathbf{v}_n \right)^2 \quad (5)$$

where  $N_k$  denotes the neighborhood of vertex  $\mathbf{v}_k$  and  $\delta \mathbf{v}_n$  denote the neighboring vertex displacements.  $d_n$  denotes the distance between vertex  $\mathbf{v}_k$  and the neighbor vertex  $\mathbf{v}_n$ . Hereby, nearer neighbors have a higher influence on the displacement of vertex  $\mathbf{v}_k$  than neighbors with a larger distance. For example, in Figure 3 vertex  $\mathbf{v}_2$  has a larger impact on the displacement of vertex  $\mathbf{v}_k$  than vertex  $\mathbf{v}_6$  due to its smaller distance to vertex  $\mathbf{v}_k$ .  $E_s(\delta \mathbf{v}_k)$  is a measure of vertex displacement deviation to the displacements of its neighbors. It regularizes the optical flow field. Especially, it is necessary if a vertex displacement is completely unconstrained by the optical flow equation due to



**Fig. 4.** Occlusion handling. Original image, occlusion map, detail of the deformed mesh without and with occlusion handling (left to right). Without occlusion handling the 2D mesh folds at the occlusion boundary which leads to inaccuracies during tracking because the triangles at the occlusion boundary contain wrong texture regions. Dark colors in the occlusion map mark occluded regions.

lack of image gradient in the surrounding triangles. The less a vertex displacement is constrained by the optical flow equation the more it is constrained by the smoothness constraint. Furthermore, defining it locally on the mesh allows us to weight it according to the occlusion of a region to overcome the problem of self-occlusion that usually appears during deformable tracking in 2D (see Section 5.2).

## 5.2 Handling Self-Occlusions

2-dimensional deformation estimation in presence of self-occlusion is a very challenging problem. In the image plane self-occlusion can also appear in parts that are not occluded in 3D due to projection. Naturally, a 2D mesh folds under these conditions which causes inaccuracies during tracking because the mesh triangles contain wrong texture regions (see Figure 4). This is an undefined status in 2D because a texture point cannot uniquely be assigned to one triangle in the mesh. Especially, when the surface is unfolded again the mesh is not able to unfold having reached this undefined status in 2D. We addressed this problem in our previous work [6] but repeat it briefly here because it is essential for the robustness of the presented real-time scenario. One solution to the problem of self-occlusion is to force the mesh to shrink instead of fold in occluded regions [3] because the mesh tracks the visible surface in the 2D image projection plane. We account for the problem of self-occlusions by weighting the smoothness constraints locally according to the occlusion of a region. In Section 5.1 we defined the smoothing constraints locally on the mesh. This allows us to weight them according to the occlusion of a region. This penalizes fold overs and forces the mesh to shrink instead of fold as this results in a smoother deformation field [6].

We estimate an occlusion map from shrinking areas in the deformed mesh (see Figure 4). For each vertex  $\mathbf{v}_k$  in the mesh we calculate the average distance to its vertical and horizontal neighbors and scale it by the initial vertex distance in the reference mesh:

$$D_k = \frac{1}{2|N_{vk}|D_v} \sum_{n \in N_{vk}} \|\mathbf{v}_k - \mathbf{v}_n\|_2 + \frac{1}{2|N_{hk}|D_h} \sum_{n \in N_{hk}} \|\mathbf{v}_k - \mathbf{v}_n\|_2 \quad (6)$$

where  $N_{vk}$  and  $N_{hk}$  are the vertical and horizontal neighborhoods of vertex  $\mathbf{v}_k$  and  $D_v$  and  $D_h$  denote the initial vertical and horizontal distances between two neighboring vertices in the regular reference mesh. By interpolating the average distances that

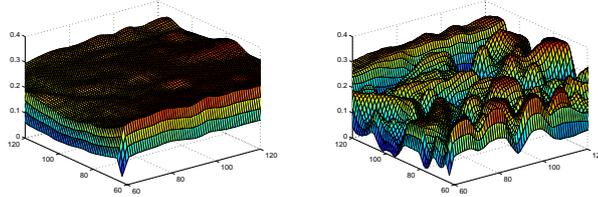
present local mesh shrinking estimates over the entire surface we can establish an estimate of occluded regions in an occlusion map. The occlusion maps are used to adapt the weight  $w_k$  in equation (4) for vertex  $\mathbf{v}_k$  to the degree of its occlusion. Vertices in occluded regions are assigned a higher weight to the smoothness constraint than vertices in non-occluded regions, i.e. the smaller the distance of a vertex to its neighbors the more its displacement is constrained by the surrounding displacements. Hereby, we counter foldings because vertices in mesh shrinking regions are forced to behave like their neighbors. Additionally, when unfolding the surface the increased smoothing weight in occluded regions causes the shrunk (i.e. occluded) region of the mesh to be stretched by the vertices at the occlusion boundary whose displacements are constrained by the optical flow equation. We choose  $w_k$  to be

$$w_k \propto \frac{1}{D'_k} \quad (7)$$

where  $D'_k$  equals  $D_k$  after the vector of all  $D_k$  is normalized so that the maximum of all  $D_k$  is one and zero-values have been set to a value  $\epsilon$  close to zero. Hereby, we do not adapt the weight to the smoothness constraint if the mesh expands or shrinks uniformly, e.g. due to a movement toward or away from the camera.

## 6 Illumination Recovery and Rendering

As we use a monocular video sequence without 3D reconstruction of the surface an estimation of shading and illumination is needed to enhance the augmented reality experience. For the real-time scenario we use a green t-shirt with a texture that consists of black curves. This allows us to establish a shading map in a very simple but efficient way by interpolating the texture pixels from neighboring pixels while preserving main wrinkles and fold overs. Here, we use the fact that the real-world illumination and shadows that should be cast onto the virtual texture are already visible in the original image. That means, the input image exhibits the exact shadows to be rendered onto the virtual texture in the image intensities. The idea is to determine a shading map from the intensity of the original image after the texture lines have been removed. As we know the position of the texture from the tracking process we can easily identify the pixels belonging to the line pattern. We identify the texture pixels as pixels that differ from the mean color of the pixel neighborhood. The intensities of these pixels are then iteratively interpolated from neighboring pixels resulting in a smooth shading map that preserves shadows at main wrinkles and fold overs. Hereby, we preserve smooth intensity changes that result from illumination and shading and filter out the sharp edges of the texture. Figure 5 shows an example where the intensities are represented as height fields. The left height field represents the intensities of the original image. The right height field illustrates the intensities after the texture pixels have been interpolated from neighboring pixels iteratively. It shows a smooth intensity field that still exhibits smooth intensity changes at wrinkles and fold overs. Finally, the texture is rendered into the real scene using an OpenGL shader where the intensity of the filtered image is used to modulate the RGB/A texture.



**Fig. 5.** Recovering shading and illumination. Representation of intensities as height fields before (left) and after (right) interpolation. The result yields a smooth intensity field and preserves smooth intensity changes at main folds and wrinkles.

## 7 Results and Application

Our system has already been tested in our lab and in public at an exhibition. In the environment at the exhibition hall the system was tested under natural conditions like lighting changes, changing background and users unfamiliar with the system. We use a XGA-Firewire camera with a resolution of  $1024 \times 768$  and process 25 frames per second. The system is robust against lighting changes and changing backgrounds. Users can move freely in front of the system and perform elastic deformations with the cloth, i.e. stretching and bending it. Rotation is possible as long as the texture on the shirt is visible. Figure 2 shows the work flow of the deformation estimation. The first image shows the input image of the camera. The second image depicts the position of the mesh and the last two images show tracking and retexturing results. The method is able to recover even strong deformations and incorporating the shading maps enhances the realistic impression of the augmented texture.

Figure 6 shows tracking and retexturing results of different deformations in the image plane. These examples demonstrate that although the elastic deformation is estimated in the 2D image plane the result is a three-dimensional impression. The first two rows of Figure 6 demonstrate realistic augmentation results under self-occlusion. The addition of realistic lighting increases the perception of spatial relations between real and virtual objects. This is even more visible in the second row of Figure 6 that shows a few close ups of the virtually augmented textures demonstrating the correct deformation and illumination of these textures. Note that in Figure 2 and Figure 6 also the segmentation result is visible from the re-colored shirt.

## 8 Conclusion

We presented a robust real-time deformable surface tracking and retexturing approach for monocular image sequences that is incorporated into a Virtual Mirror setup. The system exchanges the color and the texture of a shirt while the person wearing the shirt can move freely in front of the mirror and even perform elastic deformations of the cloth like stretching and bending or move toward or away from the camera. The deformations are estimated in the 2D image plane using an optical-flow based approach and a mesh-based motion model. By taking illumination and shadows into account we achieve very



**Fig. 6.** Real-time cloth tracking: Mesh on the moving garment (left) and virtually augmented textures and colors. The addition of real lighting increases the perception that the cloth is truly exhibiting the virtual texture.

realistic augmentation results. Besides these results, one important characteristic of our system is that it uses very simple hardware.

## References

1. Bradley, D., Popa, T., Sheffer, A., Heidrich, W. and Boubekeur, T.: Markerless Garment Capture, SIGGRAPH, ACM Transactions on Graphics, Los Angeles, USA (2008)
2. Ehara, J., Saito, H.: Texture Overlay for Virtual Clothing Based on PCA of Silhouettes, 5th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2006, pp. 139–142, Santa Barbara, USA (2006)
3. Gay-Bellile, V., Bartoli, A., Sayd, P.: Deformable Surface Augmentation in Spite of Self-Occlusions, 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2007, Nara, Japan (2007)
4. Guskov, I., Klivanov, S., Bryant, B.: Trackable Surfaces, Proc. ACM/EG Symposium on Computer Animation, pp. 251–257, San Diego, USA (2003)

5. Hasler, N., Asbach, M., Rosenhahn, B., Ohm, J.-R., Seidel, H.-P.: Physically Based Tracking of Cloth, Proc. of the International Workshop on Vision, Modeling, and Visualization, VMV 2006, pp.49–56, Aachen, Germany (2006)
6. Hilsmann, A., Eisert, P.: Tracking Deformable Surfaces with Optical Flow in the Presence of Self Occlusions in Monocular Image Sequences, CVPR Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment, Anchorage, USA (2008)
7. Hilsmann, A., Eisert, P.: Deformable Object Tracking Using Optical Flow Constraints, Proc. 4th Int. Conf. on Visual Media Production CVMP 2007, London, UK (2007)
8. Horn, B.K.P., Schunck, B.G.: Determining Optical Flow, Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA (1980)
9. Pilet, J., Lepetit, V., Fua, P.: Real-Time Non-Rigid Surface Detection, Proc. Conference on Computer Vision and Pattern Recognition, CVPR 2005, San Diego, USA (2005)
10. Pilet, P., Lepetit, V., Fua, P.: Fast Non-Rigid Surface Detection, Registration and Realistic Augmentation, Int. Journal of Computer Vision, vol. 96, no. 2, pp. 109–122 (2008)
11. Pritchard, D., Heidrich, W.: Cloth Motion Capture, Proc. Eurographics (Computer Graphics Forum), pp. 263–271, Granada, Spain (2003)
12. Saito, H., Hoshino, J.: A Match Moving Technique for Merging CG and Human Video Sequences, Proc. of the Acoustics, Speech, and Signal Processing, ICASSP 2001, pp.1589–1592, Salt Lake City, USA (2001)
13. Salzmann, M., Pilet, J., Ilic, S., Fua, P.: Surface Deformation Models for Non-Rigid 3–D Shape Recovery, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 29, no. 8, pp. 1481–1487 (2007)
14. Scholz, V., Magnor, M.: Texture Replacement of Garments in Monocular Video Sequences, Proc. of Eurographics Symposium on Rendering, pp. 305–312, Nicosia, Cyprus (2006)
15. Scholz, V., Magnor, M.: Cloth Motion from Optical Flow, Proc. Vision, Modeling and Visualization, VMV 2004, Stanford, USA (2004)
16. Scholz, V., Stich, T., Keckeisen, M., Wacker, M., Magnor, M.: Garment Motion Capture Using Color-Coded Patterns, Proc. Eurographics (Computer Graphics Forum), pp. 439–448, Dublin, Ireland (2005)
17. Taguchi, A., Aoki, T., Yasuda, H.: A Study on Real-time Virtual Clothing System Based on Two-Dimensional Plane Model, Proc. 6th Asia-Pacific Symposium on Information and Telecommunication Technologies, APSITT 2005 (2005)
18. Torresani, L., Yang, D., Alexander, E., Bregler, C.: Tracking and Modeling Non-Rigid Objects with Rank Constraints, Proc. Computer Vision and Pattern Recognition CVPR 2001, Hawaii, USA (2001)
19. White, R., Crane, K., Forsyth, D.: Capturing and Animating Occluded Cloth, SIGGRAPH, ACM Transactions on Graphics, San Diego, USA (2007)
20. White, R., Forsyth, D.: Retexturing Single Views Using Texture and Shading, Proc. European Conference on Computer Vision ECCV 2006, pp. 70–81, Graz, Austria (2006)
21. White, R., Forsyth, D., Vasanth, J.: Capturing Real Folds in Cloth, Technical report, EECS Department, University of California, Berkeley (2006)
22. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)