Realistic Cloth Augmentation in Single View Video

Anna Hilsmann, Peter Eisert

Fraunhofer Institute for Telecommunications
Heinrich Hertz Institute
Einsteinufer 37, 10587 Berlin, Germany
Email: {anna.hilsmann,peter.eisert}@hhi.fraunhofer.de

Abstract

Augmenting deformable surfaces like cloth in real video is a challenging task because not only geometric parameters describing the surface deformation have to be estimated, but also photometric parameters in order to recover realistic lighting and shading. In this paper we present a method for cloth augmentation in single-view video with realistic geometric deformation and photometric properties without a 3 dimensional reconstruction of the surface. These parameters are jointly estimated from direct image information using an extended version of optical flow with an explicit photometric color model and mesh-based. We demonstrate how the proposed method is used to produce convincing augmented results.

1 Introduction

Merging computer generated content with real video is an important and challenging task in many applications such as movies [5] and augmented reality [11] (see Figure 1). For realistic augmentation, not only geometric parameters describing the motion of the object have to be estimated. In addition to that, also photometric parameters describing lighting and shading have to be recovered to assure that the virtual object merges with the real video content and to increase the perception that the augmented video is truly exhibiting the virtual object. We are particularly interested in *retexturing* non-rigid surfaces, e.g. cloth, in single-view videos for real-time augmented reality applications like the one depicted in Figure 1. As 3D reconstruction of elastically deforming surfaces in single-view video is an ill-posed problem, we approach the problem in the image plane. This is the most intuitive approach as for the *retexturing* purpose we intend to render a virtual texture from the same point of view as the original surface and integrate it into the real video scene such that lighting conditions and shadings have to remain the same.

Current cloth augmentation methods usually do not account for shading and illumination [3] at all or treat geometric and photometric parameter estimation separately [1, 6, 11, 13, 14]. Some approaches require markers to establish a shading map by intensity interpolation [1, 6, 13], others restrict the surface to consist of a limited set of colors which can be easily classified [14]. However, this assumption of a-priori knowledge is problematic in many applications and limits the applicability for arbitrary video sequences.

In [7] we presented a parametric method utilizing direct image information for joint deformation and photometric parameter recovery from single view video and showed in various experiments how taking into account the photometric changes in an image sequence improves the geometric tracking results. We exploit the optical flow constraint equation extended by a specific illumination model which allows the brightness of a scene point to vary with time, as proposed by Gennert and Negahdaripour [4]. Several other researchers have exploited similar ideas to make optical flow based tracking more robust against lighting changes [12, 9, 7]. In this paper, we propose to utilize the additional information about illumination changes to synthesize an augmented *retextured* version of the cloth by incorporating a specific color model, that accounts not only for the light intensity but also for the color of the light. We will show that the addition of lighting substantially improves the realistic impression of the augmented cloth.

The remainder of this paper is structured as fol-

Figure 1: Augmented reality application: in a *Virtual Mirror* a user can see himself wearing virtually textured clothes.

lows. In Section 2 we recall our method for simultaneous estimation of deformation and illumination parameters for deformable surfaces from [7] and expand the method by a specific color model that accounts not only for changes in the light intensity but also in the color of the light, before Section 3 describes our method for cloth augmentation from single view video, i.e. shading map creation and cloth retexturing. Section 4 reports experimental results.

2 Joint Deformable Motion and Photometric Parameter Estimation

The problem of recovering geometric and photometric parameters for realistic retexturing can be seen as an image registration task solving for a warp that not only registers two images spatially but also photometrically. We use an intensity-based parametric approach which leads to a minimization of a cost function over a parameter vector Θ describing the spatial and photometrical warp. In general, the cost function consists of two terms:

$$\hat{\Theta} = \text{argmin}_{\Theta} \left(\mathcal{E}_{\mathcal{D}}(\Theta) + \lambda^2 \mathcal{E}_{\mathcal{S}}(\Theta) \right) \qquad (1)$$

with $\mathcal{E}_{\mathcal{D}}(\Theta)$ being the *data term* and $\mathcal{E}_{\mathcal{S}}(\Theta)$ representing prior knowledge on the shape and illumination model, often called the *smoothness term*. λ is a regularization parameter which weights the influence of this prior knowledge against fitting to the data term.

2.1 Parameterizing the Warp

A 3D reconstruction of elastically deforming cloth from single-view video is an ill-posed task because it faces many ambiguities. Without a 3D reconstruction of the surface we cannot explicitly model the light conditions of the scene. However, we can explain the motion and deformation in the image plane and recover the impact of the illumination on the intensity of a scene point. This is because the shading and lighting properties that should be applied to the virtual texture are already present in the original image. We therefore describe the spatial warp of the surface in a dense 2-dimensional pixel displacement field $\mathbf{D}(\mathbf{x}) = (\mathbf{D}_x(\mathbf{x}), \mathbf{D}_y(\mathbf{x}))^T$ and define the photometric changes in terms of a dense multiplier field $\mathcal{S}(\mathbf{x})$ describing the scale of the pixel intensity, where $\mathbf{x} = (x, y)^T$ represents the pixel position. The reason why we chose to explain the intensity changes in the image by a multiplier field and not by both multiplier and offset fields as in [10] is that the decomposition into multiplier and offset fields yields ambiguities and is not unique. Another advantage is that we do not have to care about gamma correction in the image for the *retexturing* purpose, as explained in Section 3.

We parameterize $\mathbf{D}(\mathbf{x})$ and $\mathcal{S}(\mathbf{x})$ in a parameter vector Θ, using a mesh-based model consisting of K vertices $\mathbf{v}_k, (k=1\ldots K)$. Additionally, a third photometric parameter ρ_i describing the intensity scale at each vertex is incorporated so that each vertex now has three parameters $\theta_k = (\delta v_{kx}, \delta v_{ky}, \rho_k)$ as illustrated in Figure 2. The spatial warp is then parameterized by the vertex displacements $\delta\mathbf{v}_k = (\delta v_{kx}, \delta v_{ky})$ and the photometric change is parameterized by the photometric parameter ρ_k:

$$\Theta = \left(\underbrace{\delta v_{x_1} \ldots \delta v_{x_K}, \delta v_{y_1} \ldots, \delta v_{y_K}}_{\Theta_{\mathbf{v}}}, \underbrace{\rho_1 \ldots \rho_K}_{\Theta_{\rho}} \right)^T \qquad (2)$$

where $\Theta_{\delta\mathbf{v}}$ comprises the geometric deformation parameters of Θ and Θ_{ρ} comprises the photometric parameters. Several parameterizations of the pixel displacement field $\mathbf{D}(\mathbf{x})$ and the brightness scale field $\mathcal{S}(\mathbf{x})$ are possible. One is a parametrization using the Barycentric coordinates of each pixel in the mesh. If a pixel $\mathbf{x}_i$ is surrounded by the three vertices $\mathbf{v}_a, \mathbf{v}_b, \mathbf{v}_c$, and $\beta_a, \beta_b, \beta_c$ are the three corresponding barycentric coordinates ($\beta_a + \beta_b + \beta_c = 1, 0 \leq \beta_{a,b,c} \leq 1$), the pixel displacement field $\mathbf{D}(\mathbf{x})$ and the brightness scale field $\mathcal{S}(\mathbf{x})$ can be

parameterized in the following way:

$$\mathbf{D}(\mathbf{x}) = \sum_{j\in\{a,b,c\}} \beta_j \delta\mathbf{v}_j$$
$$\mathcal{S}(\mathbf{x}) = \sum_{j\in\{a,b,c\}} \beta_j \rho_j \quad (3)$$

Note, that higher order interpolation, like e.g. B-splines or thin-plate splines, are also possible.

When dealing with color images one approach is to estimate a dense muliplier field

$$\mathcal{S}(\mathbf{x}) = (\mathcal{S}_r(\mathbf{x}), \mathcal{S}_g(\mathbf{x}), \mathcal{S}_b(\mathbf{x}))^T \quad (4)$$

for each color separately, where $\mathcal{S}_g(\mathbf{x})$, $\mathcal{S}_r(\mathbf{x})$ and $\mathcal{S}_b(\mathbf{x})$ denote the local intensity changes of the red, green and blue color channels. This is time consuming and would increase the parameter vector Θ by twice the number of mesh vertices. Instead, we assume that the color temperature of the light is spatially constant in the image and only its intensity varies locally. This can be expressed through the following equations:

$$\mathcal{S}(\mathbf{x}) = \mathcal{S}_g(\mathbf{x}) \cdot (1, s_{rg}, s_{bg})^T \quad (5)$$

where s_{rg} and s_{bg} denote global red and blue gains of the light color representing the color temperature. Hence, the parameter vector Θ is extended by two further parameters for color images:

$$\Theta = \left(\underbrace{...\delta v_{x_i}..., ...\delta v_{y_i}...}_{\Theta_{\delta\mathbf{v}}}, \underbrace{...\rho_i...}_{\Theta_\rho}, \underbrace{s_{rg}, s_{bg}}_{\Theta_s} \right)^T \quad (6)$$

such that now Θ_ρ describes the intensity changes in the green channel that can vary spatially in the image and Θ_s comprises the red and blue intensity scale with respect to the green intensity. These parameters are global parameters for one image, i.e they are assumed to be spatially constant.

2.2 Deriving the Data Term

For readability reasons we drop the distinction of the different color channels in the following description of the derivation of the data term from an extended optical flow equation and come back to dealing with the different color channels at the end of this section. In order to account for brightness changes in an image sequence Negahdaripour [10] relaxed the optical flow constraint from its original form, which assumes brightness constancy between two successive image frames [8], allowing for multiplicative and additive deviations from brightness constancy. To account for the uniqueness reasons given above we relax the optical flow constraint equation allowing for multiplicative deviations from brightness constancy only to deduce the data term $\mathcal{E}_\mathcal{D}$:

$$\mathcal{I}(\mathbf{x} + \mathbf{D}(\mathbf{x}, t), t + \delta t) = \mathcal{S}(\mathbf{x}, t) \cdot \mathcal{I}(\mathbf{x}, t) \quad (7)$$

where $\mathcal{I}(\mathbf{x}, t)$ is the image intensity of each of the three color channels at pixel $\mathbf{x}$ and time t, $\mathbf{D}(\mathbf{x}, t)$ is the displacement vector field at pixel $\mathbf{x} = (x, y)^T$ and $\mathcal{S}(\mathbf{x}, t)$ is the intensity scale field. For readability reasons we skip the time dependency of $\mathbf{D}(\mathbf{x}, t)$ and $\mathcal{S}(\mathbf{x}, t)$ in the following equations and denote them with $\mathbf{D}(\mathbf{x})$ and $\mathcal{S}(\mathbf{x})$. The data term $\mathcal{E}_\mathcal{D}$ then is derived from the Sum of Squared Differences

$$\mathcal{E}_\mathcal{D} = \sum_{\mathbf{x}\in\mathcal{R}} \Big(\mathcal{I}(\mathbf{x} + \mathbf{D}(\mathbf{x}), t + \delta t) - \mathcal{S}(\mathbf{x}) \cdot \mathcal{I}(\mathbf{x}, t) \Big)^2 \quad (8)$$

where $\mathcal{R}$ denotes the region of interest in the image.

We proceed in a hierarchical analysis-by synthesis estimation approach where the parameters are not estimated between two successive frames $\mathcal{I}_{n-1}$ and $\mathcal{I}_n$ but between a synthetic version of the previous frame $\hat{\mathcal{I}}_{n-1}$ generated with the previous parameter estimation $\hat{\Theta}_{n-1}$. Between these two images the parameters are estimated in a hierarchical Gauss-Newton method. An initial approximation of the parameters between these two images is computed from low-pass filtered and sub-sampled versions of $\hat{\mathcal{I}}_{n-1}$ and $\mathcal{I}_n$ and the estimation is refined on each level. On each pyramid level we approximate the above equation with a Gauss-Newton approximation by applying a first order Taylor expansion:

$$\mathcal{E}_\mathcal{D} = \sum_{\mathbf{x}\in\mathcal{R}} \Big(\nabla\mathcal{I}(\mathbf{x}) \cdot \mathbf{D}(\mathbf{x}, t) + \frac{\partial\mathcal{I}(\mathbf{x})}{\partial t} + (1 - \mathcal{S}(\mathbf{x}, t)) \cdot \mathcal{I}(\mathbf{x}) \Big)^2. \quad (9)$$

Here, $\nabla\mathcal{I}(\mathbf{x}) = (\mathcal{I}_x(\mathbf{x}), \mathcal{I}_y(\mathbf{x}))$ denotes the spatial derivatives of the image $\mathcal{I}$ and $\frac{\partial\mathcal{I}(\mathbf{x})}{\partial t}$ denotes

the temporal gradient.

To regularize the optical flow field $\mathbf{D}(\mathbf{x})$ and the illumination scale field $\mathcal{S}(\mathbf{x})$ we incorporate the mesh-based motion and illumination model represented by the parameter vector Θ. This leads to a cost function of the following form:

$$\mathcal{E}_{\mathcal{D}}(\Theta) = \left\|\mathbf{J}_{\mathcal{D}} \cdot \Theta - r\right\|^2 \tag{10}$$

where $\mathbf{J}_{\mathcal{D}}$ is the sparse $n \times 3K$ Jacobian matrix of the Gauss-Newton approximation of the data term [7]. The $K \times 1$ parameter vector Θ is given by equation (2) and r is an $n \times 1$ vector given by:

$$r = \left(\tfrac{\partial \mathcal{I}(\mathbf{x}_1)}{\partial t} + \mathcal{I}(\mathbf{x}_1), \ldots, \tfrac{\partial \mathcal{I}(\mathbf{x}_n)}{\partial t} + \mathcal{I}(\mathbf{x}_n)\right)^T \tag{11}$$

n denotes the number of pixels in region $\mathcal{R}$ selected for contribution, i.e. pixels with sufficient spatial gradient.

When dealing with color images $\mathcal{I}(\mathbf{x}) = (\mathcal{R}(\mathbf{x}), \mathcal{G}(\mathbf{x}), \mathcal{B}(\mathbf{x}))^T$ equation (7) is changed to

$$\mathcal{I}\left(\mathbf{x} + \mathbf{D}(\mathbf{x}), t + \delta t\right) = \mathcal{S}\left(\mathbf{x}\right) \circ \mathcal{I}\left(\mathbf{x}, t\right) \tag{12}$$

where $\circ$ denotes the entrywise product or Hadamard product of two vectors and $\mathcal{R}(\mathbf{x}), \mathcal{G}(\mathbf{x}), \mathcal{B}(\mathbf{x})$ denote the red, green and blue color channels of image $\mathcal{I}(\mathbf{x})$. $\mathcal{S}\left(\mathbf{x}\right)$ denotes the local intensity changes of the red, green and blue color channels as given in equation (5). In each iteration step in the hierarchical Gauss-Newton estimation procedure, we approximate $\mathcal{S}\left(\mathbf{x}\right)$ by a first order Taylor expansion

$$\begin{aligned}\mathcal{S}(\mathbf{x}) &= \mathcal{S}_g(\mathbf{x}) \cdot \begin{pmatrix} 1 \\ s_{rg} \\ s_{bg} \end{pmatrix} \\ &\approx \tilde{\mathcal{S}}_g(\mathbf{x}) \cdot \begin{pmatrix} 1 \\ \tilde{s}_{rg} \\ \tilde{s}_{bg} \end{pmatrix} + \mathbf{J}_{\mathcal{S}}|_{\tilde{\mathcal{S}}_g, \tilde{s}_{rg}, \tilde{s}_{bg}} \cdot \begin{pmatrix} \mathcal{S}_g - \tilde{\mathcal{S}}_g \\ s_{rg} - \tilde{s}_{rg} \\ s_{bg} - \tilde{s}_{bg} \end{pmatrix}\end{aligned} \tag{13}$$

where $\tilde{\mathcal{S}}_g$ and $\tilde{s}_{rg}, \tilde{s}_{rg}$ are the current estimates of the local intensity scale field of the green channel and the red and blue gains at which $\mathcal{S}$ is linearized in each iteration step.

2.3 Choosing the Smoothness Term

In order to account for the smooth nature of the deformation and illumination field we penalize the discrete second derivative of the motion and illumination parameters in the mesh in a smoothness term $\mathcal{E}_{\mathcal{S}}$.

$$\mathcal{E}_S(\Theta) = \sum_{k=1}^{K} \left\| \theta_k - w_k \sum_{n \in \mathcal{N}_k} \theta_n \right\|^2 \tag{14}$$

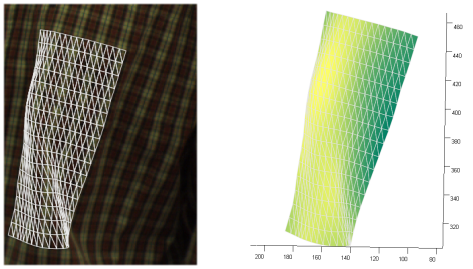

Figure 2: Original frame and estimated model, photometric parameters are represented by the colors.

with $w_k = \frac{1}{|\mathcal{N}_k|}$, where $\mathcal{N}_k$ denotes the neighborhood of vertex $\mathbf{v}_k$, i.e. all vertices $\mathbf{v}_k$ is connected to,K is the number of vertices in the mesh and $\theta = (\delta v_{kx}, \delta v_{ky}, \rho_k)$. In order to give closer neighbors a higher influence on vertex $\mathbf{v}_k$ than neighbors with a larger distance, we can also weight each neighbor according to its distance to vertex $\mathbf{v}_k$. In this case the weight w_k is $w_k = \frac{1/d_{k,l}}{\sum_{n \in \mathcal{N}_k} 1/d_{k,n}}$, where $d_{k,l}$ denotes the distance between the two $\mathbf{v}_k$ and $\mathbf{v}_l$.
In matrix notation, $\mathcal{E}_s(\Theta)$ can be expressed as:

$$\mathcal{E}_s(\Theta) = \left\|\tilde{\mathbf{L}} \cdot \Theta\right\|^2 \tag{15}$$

where $\tilde{\mathbf{L}} = (\lambda_d \mathbf{L}, \lambda_d \mathbf{L}, \lambda_\mu \mathbf{L})$ is composed by concatenating three times the scaled *Laplacian Matrix* $\mathbf{L}$ of the mesh [2]. The scaled *Laplacian Matrix* $\mathbf{L}$ is a $k \times k$ matrix with one row and one column for each vertex and $\mathbf{L}(i,j) = w_i$ if vertex i and j are connected and $\mathbf{L}(i,i) = 1$. All other entries are set to zero. λ_d and λ_μ account for different weights of the smoothness term of the spatial deformation and the change in the brightness scale.

3 Retexturing with Correct Deformation and Shading

In our model the intensity of each color channel at pixel $\mathbf{x}_i$ in the image $\mathcal{I}(\mathbf{x}_i) = (\mathcal{R}(\mathbf{x}_i), \mathcal{G}(\mathbf{x}_i), \mathcal{B}(\mathbf{x}_i))^T$ is described as a product of a factor $\mathcal{S}(\mathbf{x}_i) = (\mathcal{S}_r(\mathbf{x}_i), \mathcal{S}_g(\mathbf{x}_i), \mathcal{S}_b(\mathbf{x}_i))^T$ representing photometric conditions of the scene, comprising e.g. physical lighting conditions, medium properties, spectral response characteristics etc., and the actual color $\mathcal{C}_{p_i} = (\mathcal{C}_r, \mathcal{C}_g, \mathcal{C}_b)^T$

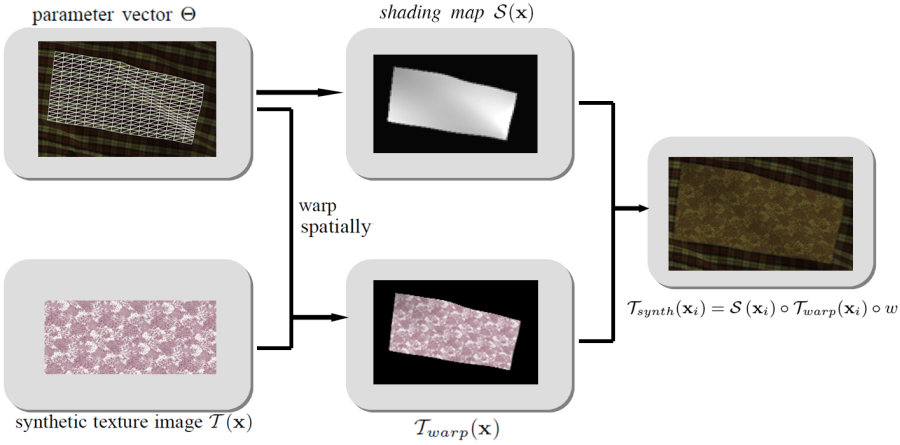

Figure 3: Retexturing with correct shading and deformation

of the underlying surface point p_i:

$$\mathcal{I}(\mathbf{x}_i) = \mathcal{S}(\mathbf{x}_i) \circ \mathcal{C}_{p_i} \tag{16}$$

where $\mathcal{S}_r(\mathbf{x}_i) = \mathcal{S}_g(\mathbf{x}_i) \cdot s_{rg}$ and $\mathcal{S}_b(\mathbf{x}_i) = \mathcal{S}_g(\mathbf{x}_i) \cdot s_{bg}$. The factor $\mathcal{S}(\mathbf{x}_i)$ describes the change of the color intensity at surface point p_i due to changes in the scene light. Here, we assume that the change of the light color, i.e. the fraction of the scale in the red and the blue channel to the scale in the green channel, is spatially constant in the image and only the light intensity varies spatially due to shading. Of course, this is a simplification and the above equation is only true if the first frame of the sequence we analyze is white balanced as we estimate the lighting changes always compared to a reference frame. However, if the first frame is not white balanced but the color value $w = (w_r, w_g, w_b)^T$ of a white surface point in the image is known, the above equations can be rewritten as:

$$\mathcal{I}(\mathbf{x}_i) = \mathcal{S}(\mathbf{x}_i) \circ \mathcal{C}_{p_i} \circ w \tag{17}$$

Retexturing now means to change the underlying surface color $\mathcal{C}_{p_i}$ to a new value, namely to that of the new synthetic texture, and keep the irradiance of the scene the same as well as the white factor w. The change of a pixel intensity in the original image sequence can arise from motion and lighting changes. With our method described in the previous section we can separate the lighting changes from motion and the underlying surface color.

Under the assumption that the reflectance of the new texture is the same as the original texture and the white value w is known we can now use the estimated deformation and photometric parameters to retexture the deforming surface in the video sequence with correct deformation and lighting. The vertex displacements $\delta\mathbf{v}_k$ are used to spatially warp a new synthetic texture image $\mathcal{T}(\mathbf{x})$ such that the original image $\mathcal{I}(\mathbf{x})$ and the warped synthetic texture $\mathcal{T}_{warp}(\mathbf{x})$ are now geometrically registered. Following, a *shading map* $\mathcal{S}(\mathbf{x}) = (\mathcal{S}_g(\mathbf{x}), \mathcal{S}_r(\mathbf{x}), \mathcal{S}_b(\mathbf{x}))^T$ is established as follows. For each pixel $\mathbf{x}_i$ belonging to a triangle with brightness scale parameters ρ_a, ρ_b, ρ_c at the surrounding vertices the shading map is

$$\mathcal{S}(\mathbf{x}_i) = \sum_{j \in a,b,c} \beta_j \rho_j \cdot (1, s_{rg}, s_{bg})^T \tag{18}$$

Again, higher order interpolation is possible. In fact, the parametrization given in equation (18) describes a two-dimensional Gouraud shading. It is furthermore possible to estimate the photometric parameters with the linear interpolation given in equation (3) in the hierarchical Gauss-Newton approach to make the estimation step fast but use a higher order interpolation for the generation of the shading map to produce a smoother *retexturing* result.

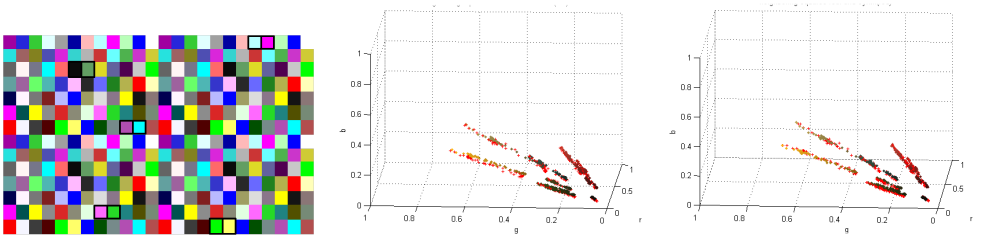

Figure 4: Test pattern (left), real and estimated colors (red crosses) (center) and real and synthesized colors (right) of ten example squares (marked with a black square).

The color channels of the geometrically registered synthetic texture image $\mathcal{T}_{warp}(\mathbf{x}_i)$ are then multiplied with the shading map $\mathcal{S}(\mathbf{x}_i)$ to generate a new synthetic texture image $\mathcal{T}_{synth}(\mathbf{x}_i)$ with correct shading and illumination properties (see also Figure 3):

$$\mathcal{T}_{synth}(\mathbf{x}_i) = \mathcal{S}(\mathbf{x}_i) \circ \mathcal{T}_{warp}(\mathbf{x}_i) \circ w. \quad (19)$$

The synthetic texture is then integrated into the original image via alpha-blending at the mesh borders.

As stated above, equations (16) and (17) are not strictly true, especially in case of saturation or specularities which make the estimation of the photometric parameter unreliable. We treat these cases by simply thresholding the resulting values in $\mathcal{T}_{synth}(\mathbf{x}_i)$ which is mathematically not strictly correct while perceptionally convincing. Gamma correction does not need not to be handled separately as we chose to use a multiplicative photometric model in which a pixel intensity is composed as in equation (16) or (17) which simply means, that the Gamma correction factor is also applied to the multiplicative photometric parameter ρ at each vertex[1].

4 Experimental Results

We applied our method to several video sequences of deforming cloth and generated video sequences of the synthetic texture as explained in the previous section. Figure 5 shows example frames from two video sequences of deforming cloth with different characteristics. The upper row shows thick cloth with very smooth deformations while the lower row shows cloth that produces small crinkles and creases. The figure shows both the tracking mesh on the original frame and the synthetically generated texture image. Figure 6 shows different steps during the *retexturing* approach. Figure 6 shows several frames of tracking and retexturing a fold in a shirt, as well as the associated shading maps. The convincing illusion of the *retexturing* method is best evaluated via visual inspection when viewing the video sequence, such that we refer the reader to the additional video material[2]. In the videos it becomes obvious how crucial illumination recovery is for convincing texture augmentation of deforming surfaces and that the addition of real lighting increases the 3- dimensional perception of the virtual texture.

In order to visually evaluate the accuracy of the estimated color changes in an image sequence we analyzed the color variance due to lighting in a sequence of a checkerboard test pattern that comprises six fully saturated colors (red, green, blue, magenta, cyan, yellow) and black as well as six graduations in saturation and intensity of each color. We plotted the actual variations of each color in the RGB space and overlayed the estimated colors obtained from the parameters Θ_ρ and Θ_s (see Figure 4 center). However, the important question is not, if the color changes were estimated correctly but if a new texture with different colors can be synthesized realistically from these estimations. There-

[1] Let I_1 and I_2 be two geometrically registered image where Gamma correction has been applied. Let $\tilde{I}_1 = I_1^{1/\gamma}$ and $\tilde{I}_2 = I_2^{1/\gamma}$ be the same images without Gamma correction. Applying our approach yields $\tilde{I}_2(\mathbf{x}_i) = \alpha \cdot \tilde{I}_1(\mathbf{x}_i)$ with $\alpha = \sum_{i \in a,b,c} \beta_i \rho_i$. Hence, $I_2(\mathbf{x}_i) = \beta \cdot I_1(\mathbf{x}_i)$ with $\beta = \alpha^\gamma$.

[2] http://iphome.hhi.de/hilsmann/assets/videos/VMV.mov

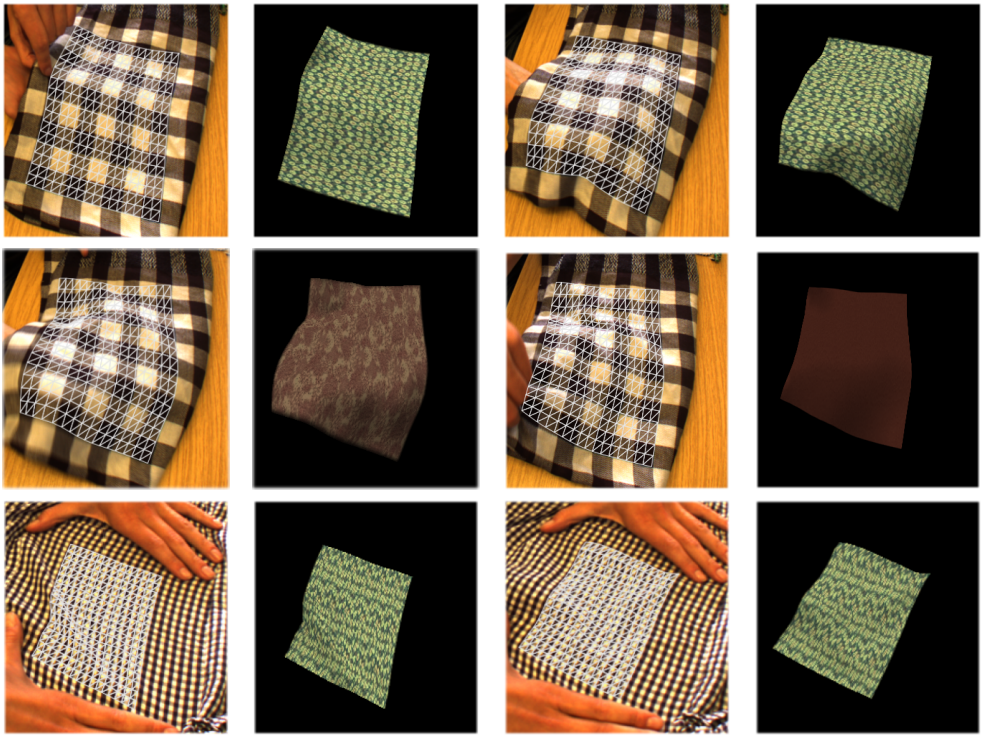

Figure 5: Tracking and retexturing cloth with different characteristics. Note that the mesh is purely 2-dimensional and the 3-dimensional illusion comes from shading.

fore, in the next step we synthesized the colors of the squares with the photometric parameters Θ_ρ and Θ_s estimated at their neighbor squares of different color, saturation and intensity, assuming that they should have experienced similar lighting conditions. Again, we compared the actual colors of the image sequence and the synthesized colors in a RGB plot. This provides us with a visual evaluation of the color variations of the synthesized colors as we treat the neighboring original color as ground truth. The right image of Figure 4 shows an example where we plotted these ground truth colors in an RGB plot and overlayed them with the synthetically generated color values (red crosses). The trajectories of the synthetic colors match with the trajectories of the real colors although they were generated with photometric parameters estimated at colors that were different in hue, saturation and intensity than the selected synthetic color.

5 Conclusion

We presented a method for augmentation of deformable surfaces, like e.g. cloth, from single-view video for augmented reality applications. We exploit the fact that for *retexturing* the retrieval of deformation and motion in the image plane is sufficient as the augmented surface is rendered from the same point of view as the original one. Furthermore, the shadows and lighting conditions that should be cast onto the virtual texture are already present in the original image. To estimate deformation and photometric parameters simultaneously we use an extended optical flow version together with mesh-based models and a specific color model that not only accounts for changes in the light intensity but also in the light color. From the estimated photometric parameters a *shading map* is established which is applied to the spatially warped synthetic texture. We showed several results for cloth retex-

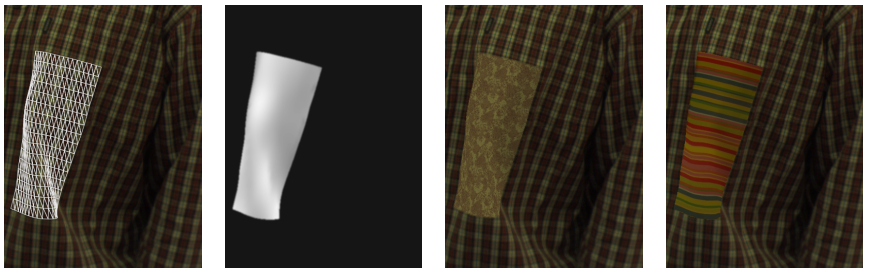

Figure 6: Tracking and retexturing a folding on a shirt. Left to right: Tracking result, shading mask, retexturing result. Note that the mesh is purely 2-dimensional and the 3-dimensional illusion comes from shading.

turing in single-view video. Our method is so far limited to smooth deformations and shading. Future work will therefore concentrate on modeling discontinuities in both the warp, e.g. in case of folding, and in the shading map, e.g. in case of sharp shadows.

References

[1] D. Bradley and G. Roth, G. and P. Bose. Augmented Clothing. In *Proc. Graphics Interface,* May 2005, Victoria, Canada.

[2] D.M. Cvetkovic, M. and Doob and H. Sachs. Spectra of Graphs: Theory and Applications. Wiley,1998.

[3] V. Gay-Bellile, A. Bartoli and P. Sayd. Direct Estimation of Non-Rigid Registrations with Image-Based Self-Occlusion Reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2008.

[4] M. A. Gennert and S. Negahdaripour. Relaxing the Brightness Constancy Assumption in Computing Optical Flow. Cambridge, USA, MIT, 1987

[5] Y. Guo, H.Sun, Q. Peng and Z. Jiang. Mesh-Guided Optimized Retexturing for Image and Video. *IEEE Trans. on Visualization and Computer Graphics,* 14(2): 426–439, 2008

[6] A. Hilsmann and P. Eisert. Optical Flow Based Tracking and Retexturing of Garments. In *Proc. International Conference on Image Processing (ICIP),* October 2008, San Diego, USA

[7] A. Hilsmann and P. Eisert. Joint Estimation of Deformable Motion and Photometric Parameters in Single View Video. *ICCV Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA),* Sept. 2009, Kyoto, Japan,

[8] B.K.P. Horn and B. G. Schunck. Determining Optical Flow. Cambridge, MA, USA, Massachusetts Institute of Technology, 1980.

[9] C. Nastar, B. Moghaddam and A. Pentland. Generalized Image Matching: Statistical Learning of physically based deformations. In *Proc. of European Conf. on Computer Vision (ECCV),* 589–598, April 1996, UK.

[10] S. Negahdaripour and C.H.Yu. A Generalized Brightness Change Model for Computing Optical Flow. In *Proc. Int. Conf. on Computer Vision (ICCV),* 2–11, 1993, Berlin, Germany.

[11] J. Pilet and V. Lepetit and P. Fua. Augmenting Deformable Objects in Real-Time. In *Int. Symposium on Mixed and Augmented Reality,* Oct. 2005, Vienna, Austria.

[12] G. Silveira and E. Malis. Real-time Visual Tracking under Arbitrary Illumination Changes. In *Proc. of Int. Conf. on Computer Vision and Pattern Recognition (CVPR)* 305–312, 2006, Minneapolis, USA.

[13] V. Scholz and M. Magnor. Texture Replacement of Garments in Monocular Video Sequences. In *Rendering Techniques 2006: Eurographics Symposium on Rendering,* 305–312, June 2006, Nicosia, Cyprus.

[14] R. White and D. A. Forsyth. Retexturing Single Views Using Texture and Shading. In *Proc. European Conf. on Computer Vision (ECCV),*70–81, May 2006, Graz, Austria.