Computers & Graphics 34(5), pp. 567-574, October 2010.

Realistic Cloth Augmentation in Single View Video under Occlusions

Anna Hilsmann[a,*], David C. Schneider[a], Peter Eisert[a,b]

[a] *Fraunhofer Institute for Telecommunications – Heinrich-Hertz-Institute*
Einsteinufer 37, 10587 Berlin, Germany

[b] *Humboldt University Berlin – Department of Computer Science*
Unter den Linden 6, 10099 Berlin, Germany

Abstract

Augmenting cloth in real video is a challenging task because cloth performs complex motions and deformations and produces complex shading on the surface. Therefore, for a realistic augmentation of cloth, parameters describing both deformation as well as shading properties are needed. Furthermore, objects occluding the real surface have to be taken into account as on the one hand they affect the parameter estimation and on the other hand should also occlude the virtually textured surface. This is especially challenging in monocular image sequences where a 3-dimensional reconstruction of complex surfaces is difficult to achieve. In this paper, we present a method for cloth retexturing in monocular image sequences under external occlusions without a reconstruction of the 3-dimensional geometry. We exploit direct image information and simultaneously estimate deformation and photometric parameters using a robust estimator which detects occluded pixels as outliers. Additionally, we exploit the estimated parameters to establish an occlusion map from local statistical color models of texture surface patches that are established during tracking. With this information we can produce convincing augmented results.

Keywords: augmented reality, cloth retexturing, optical flow, non-rigid tracking

1. Introduction and Related Work

The problem of merging computer generated content with real video is of wide interest in many applications such as movies [1] and augmented reality [2, 3]. Often, a real object in a video sequence is replaced by a virtual, computer generated one. To assure that the virtual object merges with the real video content, not only geometric parameters describing position and shape of the real object have to be recovered, but also photometric parameters describing shading and real lighting conditions on the object. Furthermore, objects occluding the real object must be taken into account as they should also occlude the virtual object in the augmented video.

We are particularly interested in single-view real-time augmented reality applications of non-rigid surfaces, whose deformations are difficult to describe, such as the movement of cloth. One approach would be to model the complete 3-dimensional geometry of the cloth surface from the image and reproject a new 3-dimensional cloth model into the real video. However, an accurate 3-dimensional reconstruction of elastically deforming surfaces like cloth requires a sophisticated multi-view camera setup and is computationally expensive [4, 5]. From single-view video this is an ill-posed problem. Therefore, our approach to augment a piece of cloth in a real video sequence is completely image-based and does not require any 3-dimensional reconstruction of the cloth surface. As we are rather interested in convincing visualization than in accurate reconstruction, we approach the problem in the image plane and *retexture* the moving surface in the image. We blend the virtual texture into the real video such that its deformation in the image projection as well as lighting conditions and shading in the final image remain the same (see e.g. Figure 5 and

*corresponding author, Tel: +49- (0)30 31002 569, Fax: +49 - (0)30 - 392 72 00
Email address: anna.hilsmann@hhi.fraunhofer.de (Anna Hilsmann)
URL: http://iphome.hhi.de/hilsmann (Anna Hilsmann)

Figure 6). For this purpose, we need to recover geometric parameters that describe the deformation of the projected surface in the image plane. Without a 3-dimensional reconstruction of the surface we cannot explicitly model the light conditions of the scene. However, we can recover the impact of the illumination on the intensity of a scene point. Therefore, we model shading or changes in the lighting conditions in additional photometric parameters that describe intensity changes in the image. We refer to this approach as *retexturing* in the following.

The issue of retrieving geometric deformation parameters from single view video has been studied by many researchers in the last few years. Common deformation models are radial basis functions [6] or mesh-based models [2, 7] and the parameters are retrieved by either optimizing over distinct feature points [2] or over the entire image information [6, 7]. In recent years, also the issue of retexturing the tracked surface was addressed. Current retexturing methods of deformable surfaces in single-view video usually either do not account for shading and illumination at all [8] or treat geometric and photometric parameter estimation separately [9, 10]. Some approaches that require markers for tracking use inpainting techniques to establish a shading map [11, 9]. Others restrict the surface to consist of a limited set of colors which can be easily classified [10]. Scholz and Magnor [9] use color-coded patterns and a-priori knowledge about surface connectivity for tracking of garments in single-view sequences. They determine shading maps by removing the color markers used for tracking and interpolating the image intensity in the deleted regions. White and Forsyth [10] presented a similar method for retexturing non-rigid objects from a single viewpoint using color markers. They limited their method to recover irradiance to screen printing techniques with a finite number of colors. However, the assumption of a-priori knowledge like in these papers is problematic in many applications and limits the applicability for arbitrary video sequences.
Recently, also feature-based and direct methods for retexturing of deformable surfaces in videos were proposed that also address the problem of external occlusions. Pilet et al. [2] proposed a feature-based real-time method for deformable object detection and tracking that uses a wide baseline matching algorithm and deformable meshes. They retexure an image by multiplying a shading map, which is the quotient of the input and a warped reference image. This method has problems at texture edges where the registration is not accurate enough such that the old texture is still visible under the synthetic one. They later extended their work by establishing visibility maps taking into account external occlusions in an expectation-maximization algorithm [12]. Generally, image-based methods yield more accurate results in non-rigid deformation estimation than feature-based techniques because they exploit the entire image instead of distinct points. In [8] Gay-Belille et al. proposed a direct method to estimate deformable motion under self-occlusions. External occlusions are only considered in a robust estimator which rejects outliers based on image intensity differences. Retexturing as well as illumination changes are not addressed in this paper such that geometric tracking and the established occlusion map might be sensitive to illumination changes in the scene.

This paper is an extension of our work presented in [7] and [3] where we treat the problem of recovering geometric and photometric parameters for realistic retexturing as an image registration task solving for a warp that not only registers two images spatially but also photometrically. In contrast to [2], we do not handle deformation and illumination parameters separately but estimate them jointly using direct image information instead of distinct features. We exploit the optical flow constraint extended by a specific illumination model and jointly estimate deformation and illumination parameters of a mesh-based model by formulating a non-linear least-squares error functional and minimize it with a Levenberg-Marquardt (LM) approach. This extension was first proposed by Negahdaripour [13] to stabilize optical flow based tracking against illumination changes. In our approach, it not only stabilizes geometric tracking against illumination changes but also allows us to actually retrieve parameters describing these changes. We 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 changes in the light intensity but also in the color of the light. By local weighting of smoothness constraints our approach can cope with self-occlusions. In this paper, this work is extended by the robust handling of external occlusions completing the next task in achieving realistic retexturing. As we are working with monocular image sequences without 3D reconstruction, we cannot determine occlusions from depth, like in stereo vision. Our approach to occlusion handling is two-fold. First, we use a robust estimator in the optimization procedure instead of a least squares estimator which detects occluded pixels and reweights them in the resulting error functional. Second, we establish a dense occlusion map specifying which texture points of the deforming surface are visible and which are occluded. This occlusion map is established from local statistical color models of texture surface patches and a global color model of the occluding object hat are built during tracking.

This paper is structured as follows. Section 2 describes our mesh-based model comprising both deformation and photometric parameters before Section 3 explains our method for image-based parameter estimation. We use a robust estimator to account for outliers due to external occlusion but also need an occlusion map that explicitly indicates the occluded pixels. The establishment of such an occlusion map is described in Section 4. Section 5 explains the final retexturing for augmentation.

2. Combined Mesh-Based Shape and Illumination Model

The spatial motion and deformation of a surface in the image plane is described by a geometric warp $\psi_g(\mathbf{x};\boldsymbol{\theta}) = \mathbf{x} + \mathcal{D}(\mathbf{x};\boldsymbol{\theta})$ of the image coordinates, where $\mathcal{D}(\mathbf{x};\boldsymbol{\theta})$ is a dense 2-dimensional pixel displacement field at each image pixel $\mathbf{x}$, parameterized by a $N \times 1$ parameter vector $\boldsymbol{\theta}$. The photometric differences between the images are explained by a multiplicative function $\psi_p(\mathbf{x};\boldsymbol{\theta})$ to the pixel intensities. We parameterize the warps $\psi_g(\mathbf{x};\boldsymbol{\theta})$ and $\psi_p(\mathbf{x};\boldsymbol{\theta})$ with a deformable model that is presented as a planar regular 2-dimensional mesh with K vertices $\mathbf{v}_k$. The geometric warp is parameterized by the vertex displacements $\delta\mathbf{v}_k$ in x- and y-direction. Additionally, we model the brightness scale between two images in a third photometric parameter ρ_k at each vertex. The resulting parameter vector $\boldsymbol{\theta}$ is then given by concatenating the 3 parameters of each vertex such that the total number of parameters is $N = 3K$

$$\boldsymbol{\theta} = \left(\underbrace{\delta v_{x_1} \ldots \delta v_{x_K} \delta v_{y_1} \ldots \delta v_{y_K}}_{\boldsymbol{\theta}_g \ (2K\times 1)} \underbrace{\rho_1 \ldots \rho_K}_{\boldsymbol{\theta}_p \ (K\times 1)} \right)^T \tag{1}$$

where $\boldsymbol{\theta}_g$ comprises the geometric deformation parameters and $\boldsymbol{\theta}_p$ comprises the photometric parameters. In the following, we present one possible parameterization of the deformation field $\psi_g(\mathbf{x};\boldsymbol{\theta})$ and the brightness scale field $\psi_p(\mathbf{x};\boldsymbol{\theta})$ by the parameter vector $\boldsymbol{\theta}$. We describe a parameterization which uses affine interpolation between the vertex positions but higher order interpolation, like e.g. B-splines or thin-plate splines, are also possible.

If a pixel $\mathbf{x}_i$ is surrounded by a triangle consisting of the three mesh vertices $\mathbf{v}_a, \mathbf{v}_b, \mathbf{v}_c$, and $\beta_a, \beta_b, \beta_c$ are the three corresponding barycentric coordinates, the geometric and photometric warp at that position can be calculated by a weighted sum of the three surrounding vertex parameters:

$$\psi_g(\mathbf{x}_i;\boldsymbol{\theta}) = \mathbf{x}_i + \mathcal{D}(\mathbf{x}_i;\boldsymbol{\theta}) = \mathbf{x}_i + \sum_{j\in\{a,b,c\}} \beta_j \delta\mathbf{v}_j \qquad \psi_p(\mathbf{x}_i;\boldsymbol{\theta}) = \sum_{j\in\{a,b,c\}} \beta_j \rho_j \ .$$

This can also be written as

$$\psi_g(\mathbf{x}_i;\boldsymbol{\theta}) = \mathbf{x}_i + \mathbf{M}_g^{\mathbf{x}_i} \cdot \boldsymbol{\theta} \qquad \psi_p(\mathbf{x}_i;\boldsymbol{\theta}) = \mathbf{m}_p^{\mathbf{x}_i} \cdot \boldsymbol{\theta} \tag{2}$$

where $\mathbf{M}_g^{\mathbf{x}_i}$ and $\mathbf{m}_p^{\mathbf{x}_i}$ are $2 \times N$ and $1 \times N$ matrices of the following form:

$$\mathbf{M}_g^{\mathbf{x}_i} = \underbrace{\begin{pmatrix} \beta_a\ldots & \beta_b\ldots & \beta_c\ldots \\ 0\ldots & 0\ldots & 0\ldots \end{pmatrix}}_{(2\times K)} \underbrace{\begin{pmatrix} 0\ldots & 0\ldots & 0\ldots \\ \beta_a\ldots & \beta_b\ldots & \beta_c\ldots \end{pmatrix}}_{(2\times K)} \underbrace{\begin{pmatrix} 0\ldots & 0\ldots & 0\ldots \\ 0\ldots & 0\ldots & 0\ldots \end{pmatrix}}_{(2\times K)}$$

$$\mathbf{m}_p^{\mathbf{x}_i} = \underbrace{\begin{pmatrix} 0\ldots & 0\ldots & 0\ldots \end{pmatrix}}_{(1\times K)} \underbrace{\begin{pmatrix} 0\ldots & 0\ldots & 0\ldots \end{pmatrix}}_{(1\times K)} \underbrace{\begin{pmatrix} \beta_a\ldots & \beta_b\ldots & \beta_c\ldots \end{pmatrix}}_{(1\times K)} \ .$$

The superscript $^{\mathbf{x}_i}$ denotes that the indices a, b, c, the barycentric coordinates $\beta_a, \beta_b, \beta_c$ and therefore the matrix $\mathbf{M}_g^{\mathbf{x}_i}$ are different for each pixel $\mathbf{x}_i$ in the mesh region.

2.1. Extension of the Model to Color Images

For color images $\boldsymbol{\mathcal{I}} = \begin{pmatrix} \mathcal{I}_R & \mathcal{I}_G & \mathcal{I}_B \end{pmatrix}^T$ we define the photometric warp as

$$\boldsymbol{\psi}_{p,color}(\mathbf{x}_i;\boldsymbol{\theta}_{color}) = \begin{pmatrix} \psi_R(\mathbf{x}_i;\boldsymbol{\theta}_{color}) & \psi_G(\mathbf{x};\boldsymbol{\theta}_{color}) & \psi_B(\mathbf{x};\boldsymbol{\theta}_{color}) \end{pmatrix}^T .$$

It is applied by entrywise multiplication with the three color channels of the image. $\psi_R(\mathbf{x};\boldsymbol{\theta}_{color})$, $\psi_G(\mathbf{x};\boldsymbol{\theta}_{color})$ and $\psi_B(\mathbf{x};\boldsymbol{\theta}_{color})$ denote the photometric warp of the red, green and blue color channels of the image. One approach would be to estimate a dense multiplier field for each color separately. However, this is time consuming and would increase the parameter vector $\boldsymbol{\theta}_{color}$ by twice the number of mesh vertices compared to $\boldsymbol{\theta}$ as three intensity scale parameters ρ_R, ρ_G and ρ_B would be necessary. Another model is to assume that the color of the light is spatially constant in the image and only its intensity varies locally. This can be expressed through the following equation:

$$\boldsymbol{\psi}_{p,color}(\mathbf{x};\boldsymbol{\theta}_{color}) = \psi_p(\mathbf{x};\boldsymbol{\theta})\cdot(c_{rg},1,c_{bg})^T \tag{3}$$

c_{rg} and c_{bg} denote global red and blue gains of the light color. For this color model, the parameter vector $\boldsymbol{\theta}_{color}$ contains two additional parameters ($N = 3K + 2$):

$$\boldsymbol{\theta}_{color} = \Bigg(\underbrace{\ldots\delta v_{x_i}\ldots\ldots\delta v_{y_i}\ldots\ldots}_{\boldsymbol{\theta}_g\ (2K\times1)}\underbrace{\rho_i\ldots}_{\boldsymbol{\theta}_p\ (K\times1)}\underbrace{c_{rg},c_{bg}}_{\boldsymbol{\theta}_c\ (2\times1)}\Bigg)^T \tag{4}$$

such that now $\boldsymbol{\theta}_p$ describes the intensity changes in the green channel that can vary spatially in the image and $\boldsymbol{\theta}_c$ comprises the red and blue intensity scale with respect to the green intensity. These parameters model the changes in the light color over time but are global parameters for one image, i.e. they are assumed to be spatially constant. For the extended parameter vector the geometric warp now is

$$\psi_g(\mathbf{x}_i;\boldsymbol{\theta}_{color}) = \mathbf{x}_i + \tilde{\mathbf{M}}_g^{\mathbf{x}_i}\cdot\boldsymbol{\theta}_{color}$$

where $\tilde{\mathbf{M}}_g^{\mathbf{x}_i}$ is a $2\times N$ matrix and equals $\mathbf{M}_g^{\mathbf{x}_i}$ as explained above with two further zero columns.

3. Image-based Parameter Estimation

We estimate the parameter vector $\boldsymbol{\theta}$ by minimizing a cost function $\mathcal{E}(\boldsymbol{\theta})$ that consists of two terms:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}}\left(\mathcal{E}_D(\boldsymbol{\theta}) + \lambda^2\mathcal{E}_S(\boldsymbol{\theta})\right)$$

$\mathcal{E}_D(\boldsymbol{\theta})$ is the data term and $\mathcal{E}_S(\boldsymbol{\theta})$ represents prior knowledge on the shape and illumination model. It is often called the smoothness term. λ is a regularization parameter which weights the influence of this prior knowledge against fitting to the data term. We minimize the cost function in a Levenberg-Marquardt (LM) approach, iteratively solving for a parameter update $\delta\hat{\boldsymbol{\theta}}$ and updating the parameter vector $\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \delta\hat{\boldsymbol{\theta}}$. The optimization is performed hierarchically on an image pyramid, each level yielding a more accurate parameter estimate.

3.1. Data Term

In the following, we first explain the data term for grayscale images and then address the extension to color images. The data term $\mathcal{E}_D(\theta)$ is often derived by exploiting the optical flow constraint in its original form which assumes brightness constancy between two successive image frames [14]:

$$\mathcal{I}_{n-1}\left(\psi_g(\mathbf{x};\boldsymbol{\theta}_n)\right) = \mathcal{I}_n(\mathbf{x})$$

where $\mathcal{I}_n(\mathbf{x})$ is the image intensity at pixel $\mathbf{x}$ of the n^{th} frame of an image sequence and $\psi_g(\mathbf{x};\boldsymbol{\theta}_n)$ is a geometric image transformation that warps $\mathcal{I}_{n-1}(\mathbf{x})$ onto $\mathcal{I}_n(\mathbf{x})$. For readability reasons we skip the index $_n$ for $\boldsymbol{\theta}$ in the following. The above equation assumes that an image pixel representing an object point does not change its brightness value from frame $n-1$ to frame n and differences between successive frames are due to geometric deformation only. However, this assumption is almost never valid for natural scenes. Furthermore, our aim is not only to retrieve geometrical but also photometric parameters for realistic retexturing. Therefore, we relax the optical flow constraint in the above equation allowing for multiplicative deviations from brightness constancy by introducing a brightness scale field $\psi_p(\mathbf{x};\boldsymbol{\theta})$:

$$\psi_p(\mathbf{x};\boldsymbol{\theta})\cdot\mathcal{I}_{n-1}\left(\psi_g(\mathbf{x};\boldsymbol{\theta})\right) = \mathcal{I}_n(\mathbf{x}) \tag{5}$$

The parameter vector $\boldsymbol{\theta}$ is given by equation (1) and $\psi_g(\mathbf{x};\boldsymbol{\theta})$ and $\psi_p(\mathbf{x};\boldsymbol{\theta})$ are warp functions given by equation (2). 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 [13] is that the decomposition into multiplier and offset fields yields ambiguities and is not unique. The data term $\mathcal{E}_D(\boldsymbol{\theta})$ is now given by the Sum of Squared Differences

$$\mathcal{E}_D(\boldsymbol{\theta}) = \frac{1}{2}\sum_{\mathbf{x}_i\in\mathcal{R}}\left(\underbrace{\psi_p(\mathbf{x}_i;\boldsymbol{\theta})\cdot\mathcal{I}_{n-1}\left(\psi_g(\mathbf{x}_i;\boldsymbol{\theta})\right)-\mathcal{I}_n\left(\mathbf{x}_i\right)}_{r_i(\boldsymbol{\theta})}\right)^2. \tag{6}$$

The Taylor series of the residual $r_i(\boldsymbol{\theta})$ yields:

$$\mathcal{E}_D(\hat{\boldsymbol{\theta}}+\delta\boldsymbol{\theta}) \approx \frac{1}{2}\sum_{\mathbf{x}_i\in\mathcal{R}}\left(r_i(\hat{\boldsymbol{\theta}})+\frac{\partial r_i(\hat{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}}\cdot\delta\boldsymbol{\theta}\right)^2$$

In matrix notation, this can be written as

$$\mathcal{E}_D(\hat{\boldsymbol{\theta}}+\delta\boldsymbol{\theta}) \approx \frac{1}{2}\left\|\mathbf{J}_\mathbf{r}\cdot\delta\boldsymbol{\theta}+\mathbf{r}\right\|^2$$

$\mathbf{r}=\mathbf{r}(\hat{\boldsymbol{\theta}})$ is an $m\times 1$ vector given by concatenating the m pixel values of the residuals and $\mathbf{J}_\mathbf{r}=\mathbf{J}_\mathbf{r}(\hat{\boldsymbol{\theta}})$ is its Jacobian whose rows are given by

$$\frac{\partial r_i(\hat{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}} = \psi_p(\mathbf{x}_i;\hat{\boldsymbol{\theta}})\cdot\nabla\mathcal{I}_{n-1}\Big(\psi_g(\mathbf{x}_i;\hat{\boldsymbol{\theta}})\Big)\cdot\mathbf{M}_g^{\mathbf{x}_i}+\mathcal{I}_{n-1}\Big(\psi_g(\mathbf{x}_i;\hat{\boldsymbol{\theta}})\Big)\cdot\mathbf{m}_p^{\mathbf{x}_i}$$

The parameter update is now calculated via:

$$\delta\boldsymbol{\theta} = -(\mathbf{J}_\mathbf{r}^T\mathbf{J}_\mathbf{r}+\alpha\mathbf{I})^{-1}\mathbf{J}_\mathbf{r}^T\mathbf{r}$$

where α is the damping factor of the LM-method and $\mathbf{I}$ is the identity matrix. When dealing with color images $\mathcal{I}=(\mathcal{I}_R\ \mathcal{I}_G\ \mathcal{I}_B)^T$, equation (5) is changed to

$$\boldsymbol{\psi}_{p,color}(\mathbf{x};\boldsymbol{\theta}_{color})\circ\mathcal{I}_{n-1}\left(\psi_g(\mathbf{x};\boldsymbol{\theta}_{color})\right)=\mathcal{I}_n\left(\mathbf{x}\right) \tag{7}$$

where $\mathcal{I}_R,\mathcal{I}_G,\mathcal{I}_B$ are the red, green and blue color channels of image $\mathcal{I}$ and $\circ$ denotes the entrywise product or Hadamard product of two vectors. The parameter vector $\boldsymbol{\theta}_{color}$ is given by equation (4). $\boldsymbol{\psi}_{p,color}\left(\mathbf{x};\boldsymbol{\theta}_{color}\right)$ denotes the local intensity changes of the red, green and blue color channels as specified in equation (3). Each pixel now contributes three equations to the resulting equation system, one for each color channel.

3.2. Smoothness Term

In order to incorporate prior knowledge of the smoothness of the deformation and illumination fields we penalize the discrete second derivative of the motion and illumination parameters in the mesh in a smoothness term $\mathcal{E}_S(\boldsymbol{\theta})$ by applying a discrete Laplace operator on the vertex parameters of the mesh [15]. For a mesh with K vertices the Laplace matrix $\mathbf{L}$ is a $K\times K$ matrix with one row and one column for each vertex and $\mathbf{L}_{k,l}=-1$ if vertex $\mathbf{v}_k$ and $\mathbf{v}_l$ are connected and $\mathbf{L}_{k,k}=|\mathcal{N}_k|$. Here, the subscripts denote the row and column number of the matrix. All other entries are set to zero. We introduce a scaled Laplace matrix $\tilde{\mathbf{L}}$ with entries $\tilde{\mathbf{L}}_{k,l}=w_{k,l}$ if vertex $\mathbf{v}_k$ and $\mathbf{v}_l$ are connected and $\tilde{\mathbf{L}}_{k,k}=-1$ with $w_{k,l}=\frac{1/d_{k,l}}{\sum_{i\in\mathcal{N}_k}1/d_{k,i}}$ which gives closer neighbors a higher influence. This matrix is applied to the parameter vector such that the smoothness term of the objective function can be rewritten as

$$\mathcal{E}_S(\boldsymbol{\theta}) = \left\|\underbrace{\mathbf{K}\cdot\boldsymbol{\theta}}_{s(\boldsymbol{\theta})}\right\|^2 \tag{8}$$

where $\mathbf{K}$ is a block diagonal matrix composed of three scaled Laplace matrices $\tilde{\mathbf{L}}$ of the mesh, two for the vertex displacements and one for the photometric parameter:

$$\mathbf{K} = \begin{pmatrix} \tilde{\mathbf{L}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{L}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda_p \tilde{\mathbf{L}} \end{pmatrix} .$$

λ_p weights the smoothing terms of the geometric parameters against the smoothing terms of the photometric scale. This is necessary due to the different scaling of the pixel displacement and the photometric parameter as the former is additive while the latter is multiplicative.

The first order Taylor series of the smoothness term is given by

$$\mathcal{E}_S(\hat{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}) = \frac{1}{2} \|\mathbf{J}_s \cdot \delta\boldsymbol{\theta} + \mathbf{s}\|^2$$

were $\mathbf{J_s} = \mathbf{K}$ is the Jacobian of the smoothness term and $\mathbf{s} = \mathbf{s}(\hat{\boldsymbol{\theta}})$. Incorporating the smoothness term into the LM approach now leads to the following parameter update:

$$\delta\boldsymbol{\theta} = -(\mathbf{J}^T\mathbf{J} + \alpha\mathbf{I})^{-1}\mathbf{J}^T\mathbf{b}$$
$$\mathbf{J} = \begin{pmatrix} \mathbf{J_r} \\ \lambda\mathbf{J_s} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \mathbf{r} \\ \mathbf{s} \end{pmatrix}$$

$\mathcal{E}_S(\boldsymbol{\theta})$ penalizes the discrete second derivative of the mesh [15, 16] and regularizes the optical flow field in addition to the mesh-based motion model itself, especially in case of lack of information in the data due to e.g. homogeneous regions with low image gradient in the surrounding triangles. One important advantage of using direct image information is, that less image information, i.e. small image gradients in less textured regions, automatically lead to a higher local weighting of the smoothness constraint in the resulting equation system because less image information in a region leads to less equations in the data term corresponding to the vertices of that region. Also, the smoothing function dominates for vertices detected as outliers when using the Huber function for robust estimation or for occluded vertices that do not contribute to the data term (see Sections 3.3 and 4).

3.3. Robust Estimation

In this section, we address the issue of external occlusions. Occluded pixels can be seen as outliers that should contribute less to the parameter estimation. To make the parameter estimation more robust against these outliers we can embed a robust estimator $\rho(r_i)$ into the data term instead of using the least squares estimator $\rho_{LS}(r_i) = \frac{1}{2}r_i^2$ in equation (6). A very efficient robust estimator is the Huber function [17]

$$\rho_H(r_i) = \begin{cases} \frac{1}{2}r_i^2 & \text{if } |r_i| \le \sigma \\ \sigma\,|r_i| - \frac{1}{2}\sigma & \text{otherwise} \end{cases}$$

which is a parabola in the vicinity of zero, and increases linearly at a given level $|r_i| > \sigma$ of the residuals. Incorporating such an estimator into the data term

$$\mathcal{E}_D(\boldsymbol{\theta}) = \sum_{\mathbf{x}_i \in \mathcal{R}} \rho(r_i(\boldsymbol{\theta}))$$

can be formulated as an equivalent reweighted least-squares problem:

$$\mathcal{E}_D(\boldsymbol{\theta}) = \sum_{\mathbf{x}_i \in \mathcal{R}} w(r_i(\hat{\boldsymbol{\theta}})) \cdot r_i^2(\boldsymbol{\theta})$$

with a weight function

$$w(r_i) = \frac{1}{r_i}\frac{\partial\rho(r_i)}{\partial r_i} = \begin{cases} 1 & \text{if } |r_i| \le \sigma \\ \frac{\sigma}{|r_i|} & \text{otherwise} \end{cases} .$$

This can be easily incorporated into the LM-approach by determining the parameter update by

$$\delta\boldsymbol{\theta} = -(\mathbf{J}_\mathbf{r}^T\mathbf{W}\mathbf{J}_\mathbf{r} + \alpha\mathbf{I})^{-1}\mathbf{J}_\mathbf{r}^T\mathbf{W}\mathbf{r}$$

where $\mathbf{W} = \mathrm{diag}(w(r_i(\hat{\boldsymbol{\theta}})))$ is a weighting matrix which is calculated with the parameter estimate of the previous iteration. The Huber function does not reject outliers, i.e. residuals with $|r_i| > \sigma$, completely. The general idea is to give these outliers less influence on the parameter estimation than in the least-squares approach. However, if σ is chosen too small, we lose a lot of information as too many data points are detected as outliers. We estimate a value for σ using the median absolute deviation (MAD) of the residuals which is an efficient score for outlier rejection [18].

3.4. Analysis-by-Synthesis Approach

Intensity-based differential techniques, which estimate the motion only between two successive frames, often suffer from drift because they accumulate errors indefinitely. This limits their effectiveness when dealing with long video sequences. To avoid error accumulation we make use of an analysis-by synthesis approach where the error minimization is always carried out between a synthesized reference image and the actual camera frame. We use the previous parameter sets $\{\hat{\boldsymbol{\theta}}_1, ..., \hat{\boldsymbol{\theta}}_{n-1}\}$ to generate a synthetic version of the previous frame $\mathcal{I}_{n-1}$ from a model image $\mathcal{I}_0$. The new parameters $\hat{\boldsymbol{\theta}}_n$ are then estimated from this synthetic previous frame $\hat{\mathcal{I}}_{n-1}$ to the current camera frame $\mathcal{I}_n$. This way, the model frame serves as reference frame and we assure that no misalignment of the model and the previous frame occurs. Thereby we allow for recovery from small inaccuracies during parameter estimation. Technically, this means that in equations (5) and (6) the original previous frame $\mathcal{I}_{n-1}$ is replaced by its synthetic version $\hat{\mathcal{I}}_{n-1}$.

4. External Occlusion Handling

The robust estimator explained in Section 3.3 weights detected outliers less than inliers in the resulting equation system to make the parameter estimation more robust against occluded pixels. For larger occluded areas this is not enough to prevent errors in the parameter estimation. These areas should not contribute to the parameter estimation at all. Additionally, for realistic retexturing an occlusion map is needed, i.e. a binary map which indicates the occluded pixels which have to be spared out from retexturing. As we are working with monocular image sequences without 3-dimensional reconstruction we cannot determine occlusions from depth, like in stereo imaging. Therefore, we will establish an occlusion map from local statistical color models of each texture point on the surface. For this purpose, we warp back the current frame $\mathcal{I}_n$ with the previous parameter estimates $\{\hat{\boldsymbol{\theta}}_1, ..., \hat{\boldsymbol{\theta}}_n\}$ such that it is registered onto the model image $\mathcal{I}_0$. We will denote this backwarped image as $\tilde{\mathcal{I}}_n$ in the following. We establish local statistical color models of each texture point, and update these models with every frame in which a texture point is classified as visible. As texture point we denote a fixed point on the deforming texture in contrast to a fixed pixel position. Geometric backwarping is necessary to associate each pixel in the current image with a texture point, whereas photometric backwarping is needed to eliminate changes in the pixel value due to shading or changes in the scene light.

The idea of updating a statistical color model for each texture point is adopted from common background estimation techniques where statistical models are updated for each image pixel over the image sequence. A pixel from a new frame is classified as background if it fits in the model [19]. Background estimation techniques are often used in video surveillance of outdoor scenes and the updating strategy makes the method more robust against changes in the scene lighting. The difference is, that for background estimation the background is assumed to be static while we are interested in a statistical model of texture points on a moving and deforming surface under varying illumination. However, we register the two images spatially as well as photometrically, so that geometric differences due to motion and deformation as well as intensity differences between $\mathcal{I}_n$ and $\mathcal{I}_0$ are removed in $\tilde{\mathcal{I}}_n$ ideally. As there can still be noise in the image due to the sensor or image sampling as well as inaccuracies in the parameter estimation, we also update our color models with each frame. Furthermore, due to the above mentioned reasons we build the local color models from texture patches centered at each texture point instead of a single texture point. In our experiments these patches had a size of 3×3 pixels. We assume that during the first 10 frames of a sequence there is no occlusion so that we can establish an initial color model for each texture patch. Many background estimation techniques not only build a model of the background but also of the foreground, i.e. the occluding object, to make the background estimation more reliable. At the beginning of a sequence we do not have any information on the occluding object. However,

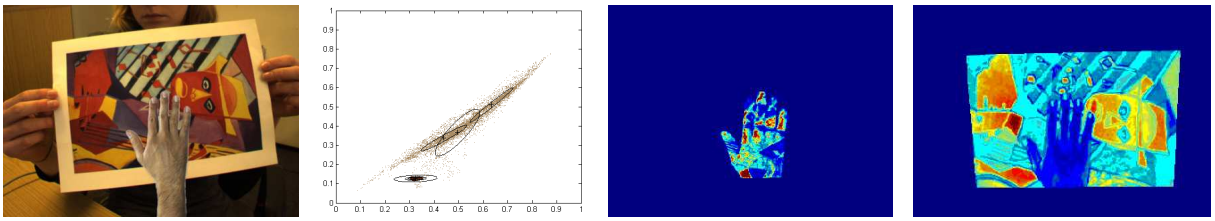

Figure 1: Occlusion handling: Frame with external occlusion, color distribution of a texture patch and occlusion color distribution with $n = 3$ in the RG-plane, color coded Mahalanobis distances to local patch color models and to the global color model of the occluding object.

once an occlusion is detected because pixels do not fit into the corresponding patch color models, we additionally establish a global color model of the occluding object which follows the same updating strategies in the subsequent frames. Having a color model of both the deforming texture and the occluding object makes the occlusion estimation more reliable. In the following we will illustrate the statistical color models, the updating strategy and the occlusion detection in detail.

The local patch color distributions are each associated with a mean color value in RGB-space $\boldsymbol{\mu}_i$ and a Gaussian distribution about that mean described by the covariance matrix $\boldsymbol{\Sigma}_i$. For each new frame we calculate the Mahalanobis distance between the pixel color values of the backwarped image $\tilde{\mathcal{I}}_n$ and the color distribution of the corresponding texture patch in RGB-space:

$$D_i = \sqrt{(\mathbf{c}_i - \boldsymbol{\mu}_i)^T \, \boldsymbol{\Sigma}^{-1} \, (\mathbf{c}_i - \boldsymbol{\mu}_i)}$$

where $\mathbf{c}_i$ denotes the RGB-color values at pixel $\mathbf{x}_i$. If no occlusion has been detected in the previous frames, the external occlusion map $\mathcal{M}_{EO}(\mathbf{x})$ is established by putting a threshold on the Mahalanobis distances to the patch color distributions. This threshold is determined using the median-absolute-deviation (MAD) which is a resistant score for successful outlier rejection [18]:

$$\mathcal{M}_{EO}(\mathbf{x}_i) = \begin{cases} 1 & \left|\frac{D_i - \bar{D}}{MAD}\right| > \upsilon_{EO} \\ 0 & \text{otherwise} \end{cases}$$

Here, $\bar{D}$ denotes the median of all distances and $MAD = \text{median}\left|D_i - \bar{D}\right|$ denotes the median of absolute deviations. $\mathcal{M}_{EO} = 1$ denotes occluded pixels in the occlusion map and $\mathcal{M}_{EO} = 0$ denotes visible pixels. For visible pixels the parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are updated. An occluded pixel that is misclassified as visible can lead to error accumulation in the patch color model. Therefore, to account for this uncertainty we only update those color models with a score that is smaller than $0.7\upsilon_{EO}$. The accuracy of the occlusion map is improved by enforcing spatial constraints, e.g. dense regions without holes. For this purpose, erosion and dilation operators are applied on $\mathcal{M}_{EO}(\mathbf{x})$.

Once an occlusion is detected, we also establish one global color model for the occluding object to make the occlusion estimation more reliable in the subsequent frames. In contrast to the local color models of the texture patches which are each modeled as one Gaussian distribution, the global color model of the occlusion consists of a mixture of n Gaussian distributions, each associated with a mean $\tilde{\boldsymbol{\mu}}_j$ and a covariance matrix $\tilde{\boldsymbol{\Sigma}}_j$, $(j = 1...n)$. To classify whether a pixel is visible or not, we now calculate the Mahalanobis distance of a pixel color value to the n Gaussian distributions of the occlusion and to the color distribution of the corresponding texture patch. Pixels are now classified as occluded if the Mahalanobis distance to one of the occlusion color distributions is smaller than the distance to the distribution of the texture patch.

$$\mathcal{M}_{EO}(\mathbf{x}_i) = \begin{cases} 1 & \min(\tilde{D}_j) < D_i, \, j = 1...n \\ 0 & \text{otherwise} \end{cases}$$

where n is the number of Gaussians in the occlusion color model, D_i denotes the Mahalanobis distance to the corresponding patch color distribution and $\tilde{D}_j$ denotes the Mahalanobis distances to the color distributions of the occlusion. Figure 1 shows an example frame with an external occlusion, example color models of one texture patch and the global

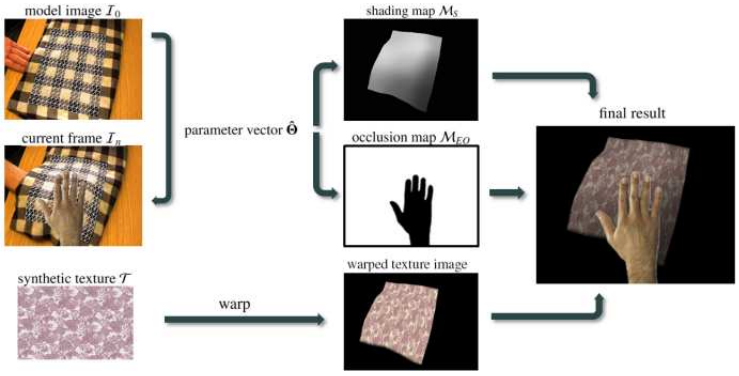

Figure 2: Illustration of the retexturing approach.

occlusion color model as well as color coded Mahalanobis distances to the color models in the image. As the color models of the texture patches are established on local positions in the image and the color distribution of the occluding object is a global distribution, the number of data in the local color models is much smaller than in the global occlusion model (see Figure 1). Therefore, we model the local color distributions with a minimum variance of $\sigma_{pc} = 0.001$ in the principal component direction of the distribution. The occlusion map is used to retexure the surface as explained in the following section. Additionally, occluded pixels do not contribute to the parameter estimation in Section 3 such that equation (6) is changed to

$$\mathcal{E}_D(\boldsymbol{\theta}) = \frac{1}{2} \sum_{\mathbf{x}_i \in \mathcal{R}} \left(1 - \mathcal{M}_{EO}(\mathbf{x}_i)\right) \cdot \left(\psi_p(\mathbf{x}_i; \boldsymbol{\theta}) \cdot \mathcal{I}_{n-1}\left(\psi_g(\mathbf{x}_i; \boldsymbol{\theta})\right) - \mathcal{I}_n(\mathbf{x}_i)\right)^2 .$$

5. Realistic Retexturing

In our model as explained in Section 2 the intensity of each color channel at pixel $\mathbf{x}_i$ in the image $\boldsymbol{\mathcal{I}}(\mathbf{x}_i) = (\mathcal{I}_R\ \mathcal{I}_G\ \mathcal{I}_B)^T$ is described as an entrywise product of a factor $\boldsymbol{\psi}_{p,color}(\mathbf{x}_i; \hat{\boldsymbol{\theta}})$ representing photometric conditions of the scene, comprising e.g. physical lighting conditions, medium properties, spectral response characteristics etc., and the actual color $\mathbf{c}_{\mathbf{p}_i} = (\mathbf{c}_R\ \mathbf{c}_G\ \mathbf{c}_B)^T$ of the underlying surface point $\mathbf{p}_i$:

$$\boldsymbol{\mathcal{I}}(\mathbf{x}_i) = \boldsymbol{\psi}_{p,color}(\mathbf{x}_i; \hat{\boldsymbol{\theta}}) \circ \mathbf{c}_{\mathbf{p}_i} \tag{9}$$

where $\boldsymbol{\psi}_{p,color}(\mathbf{x}_i; \hat{\boldsymbol{\theta}})$ is given in equation (3). It describes the change of the color intensity at surface point $\mathbf{p}_i$ 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. Retexturing now means to change the underlying surface color $\mathbf{c}_{\mathbf{p}_i}$ to a new value, namely to that of the new synthetic texture, and keep the photometric conditions of the scene the same. The change of a pixel intensity in the original image sequence can arise from motion and lighting changes. With our method described in Section 3 we can separate the lighting changes from motion and the underlying surface color.

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 $\boldsymbol{\mathcal{T}}(\mathbf{x})$ such that the original image $\boldsymbol{\mathcal{I}}(\mathbf{x})$ and the warped synthetic texture $\boldsymbol{\mathcal{T}}(\psi_g(\mathbf{x}; \hat{\boldsymbol{\theta}})$ are now geometrically registered. A shading map $\mathcal{M}_S(\mathbf{x})$ is established from the photometric warp:

$$\mathcal{M}_S(\mathbf{x}_i) = \boldsymbol{\psi}_{p,color}(\mathbf{x}_i; \hat{\boldsymbol{\theta}})$$

Again, higher order interpolation is possible. In fact, the above parametrization describes Gouraud shading. The color channels of the geometrically registered synthetic texture image $\mathcal{T}(\psi_g(\mathbf{x}_i;\hat{\boldsymbol{\theta}})$ are then multiplied with the shading map to achieve realistic shading and illumination properties

$$\mathcal{T}_{synth}(\mathbf{x}_i) = \mathcal{M}_S\,(\mathbf{x}_i) \circ \mathcal{T}(\psi_g(\mathbf{x}_i;\hat{\boldsymbol{\theta}}).$$

As stated above, equation (9) is 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 showed to be perceptionally convincing. The synthetic texture is finally integrated into the original image via alpha-blending at the mesh borders and the external occlusion map. Our retexturing method is schematically illustrated in Figure 2.

6. Experimental Results

We applied our method to several real video sequences with a resolution of 1024×768 pixels and a frame rate of 25 fps showing deforming surfaces, in particular pieces of paper and cloth, under varying illumination conditions. To this end we use 4 levels of resolution in the image pyramid and experiments with synthetic image sequences with this hierarchical scheme showed that it is able to estimate displacements of up to 25 pixels between two frames with an average error of 0.2 pixels. In our experiments we evaluated different aspects which are explained in the following.

6.1. Registration Accuracy

We evaluate the registration results based on the Root Mean Squared Error (RMSE) between the synthetic image $\hat{\mathcal{I}}_n$ generated from the parameter estimates $\hat{\boldsymbol{\theta}}_n$ and the original current frame $\mathcal{I}_n$ computed over all image pixels in the mesh region $\mathcal{R}$ for several video sequences and compared our approach with the classical optical flow approach. With *classical optical flow* approach we refer to the original optical flow constraint that does not account for illumination changes, the geometric deformation model and optimization method are equal. Figure 3 shows the progress of the RMSE over frames for two video sequences. The solid line shows the RMSE with our approach and the dashed line shows the RMSE with the classical optical flow approach. Experiments with nine sequences showed that taking illumination parameters into account significantly reduces the mean RMSE over the entire sequence by up to 74%.
Additionally, we manually labeled prominent feature points in every 50th frame of two test sequences which serve as ground truth points. We then warped the ground truth points of the reference frame with the geometric deformation parameters of these frames. The mean difference between the estimated positions and the manually labeled ground truth position describes the geometric registration error. This additional evaluation approach is chosen to evaluate geometric registration accuracy separately from photometric registration. We can reduce the mean distance between the estimated and the ground truth position by approximately 40% when taking illumination into account. The right images in Figure 3 show frame 250 of one test sequence with the real and estimated positions of the ground truth points overlayed for our approach and the classical optical flow approach. Here, classical optical flow leads to geometric misregistration, e.g. in the upper left corner, in contrast to our method.

6.2. Retexturing

We used the estimated deformation and illumination parameters to establish shading maps and produce augmented versions of several video sequences which are best evaluated by visual inspection. Figure 4 shows example frames from two video sequences of deforming cloth with different characteristics. The left images show thick cloth with very smooth deformations while the right images show cloth that produces small wrinkles and creases. The figure shows both the tracking mesh on the original frame and the synthetically generated texture image. Figure 5 shows retexturing results of a video sequence showing a shirt folded in front of the camera. Retexturing results with and without illumination recovery as well as the established shading map are depicted. These examples demonstrate how crucial illumination recovery is for convincing texture augmentation of deforming surfaces. The addition of realistic lighting increases the perception of spatial relations between the real and virtual objects. Note, that spatial deformation is purely 2-dimensional and the 3-dimensional impression comes from shading.

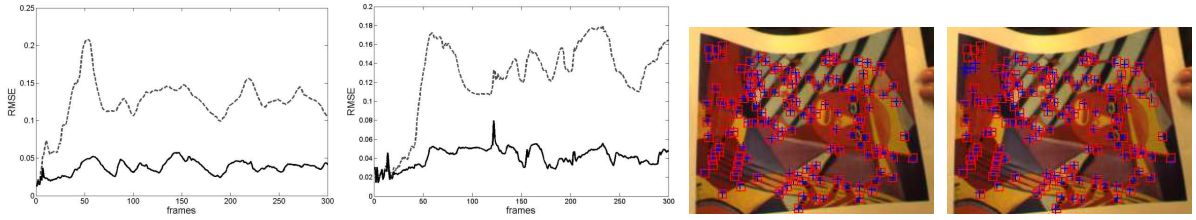

Figure 3: Evaluation of the registration accuracy. Left images: RMSE of two video sequences with our approach (solid line) and the classical optical flow approach (dashed line). Right images: Estimated (crosses) and ground truth (squares) positions of the feature points with (left) and without (right) illumination consideration.

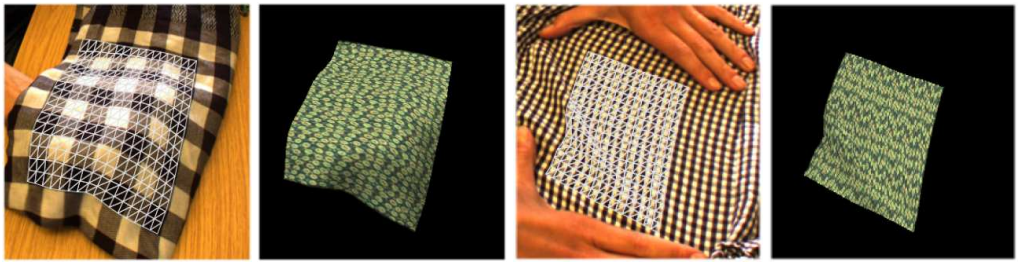

Figure 4: Tracking and retexturing cloth with different characteristics. The left images show the original frame with the deformation mesh and the right images show synthetic results which we achieved with our retexturing approach. Note that the mesh is purely 2-dimensional and the 3-dimensional illusion comes from shading.

6.3. External Occlusions

To evaluate our approach to occlusion detection we synthetically rendered occlusions in video sequences and evaluated the occlusion masks by calculating the pixel differences to the alpha mask we used for rendering. In our experiments we used a threshold of $\upsilon_{EO} = 3.5$ and described the color model of the external occlusion with $n = 3$ Gaussians. 96.84% of the pixels were correctly classified as visible or occluded. Figure 2 shows an example where we artificially rendered a hand into a video sequence. The color distributions of the occluding object and one texture patch are depicted in the RG-plane. The right images show the color-coded Mahalanobis distance to the local texture color distribution and the global distribution of the occluding object. Figure 6 shows two retexturing results under external occlusions. The left images depict the augmented results and the right images show the estimated occlusion maps.

7. Conclusion

In this paper, we presented a method for augmentation of deformable surfaces, like e.g. cloth, in single view sequences. We exploit the fact that for the visualization of the virtually textured surface in the augmented video sequence a 3-dimensional reconstruction of the surface geometry is not needed. We rather retrieve geometric and photometric parameters which describe the appearance of the surface in the image. These parameters are estimated using an extended optical flow constraint and a specific color model that not only accounts for changes in the light intensity but also in the light color. We account for external occlusions in an occlusion map classifying whether a pixel is visible or not based on local texture patch color distributions and a global occlusion color distribution. The final visualization is achieved by warping the new texture with the geometric parameters and multiplying it with a shading map built from the photometric parameters. The synthetic texture is rendered in the original video sequence via alpha-blending at the mesh-borders and the occlusion map. Our method is currently limited to video sequences with smooth deformations and shading due to the smoothness terms. Future work will e.g. 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.

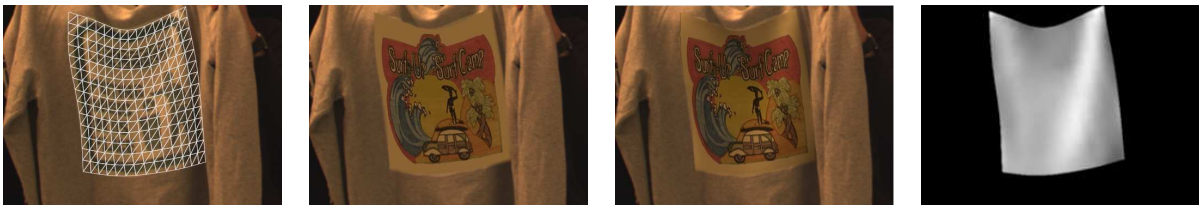

Figure 5: Tracking and retexturing a shirt. Left to right: original image, retexturing without illumination considering, retexturing with our approach, shading map. The addition of real lighting increases the perception that the paper is truly exhibiting the virtual texture.

Figure 6: Retexturing under external occlusions. Left images: Retextured image, right images: occlusion maps.

References

[1] Guo Y, Sun H, Peng Q, Jiang Z. Mesh-guided optimized retexturing for image and video. IEEE Trans on Visualization and Computer Graphics 2008;14(2):426–39.

[2] Pilet J, Lepetit V, Fua P. Fast non-rigid surface detection, registration and realistic augmentation. Int J of Computer Vision 2008;76(2):109–22.

[3] Hilsmann A, Eisert P. Realistic cloth augmentation in single view video. In: Proc. of Vision, Modeling, and Visualization Workshop 2009. 2009, p. 55–62.

[4] Bradley D, Popa T, Sheffer A, Heidrich W, Boubekeur T. Markerless garment capture. In: Proc. of ACM SIGGRAPH 2008. 2008, p. 99–108.

[5] de Aguiar E, Stoll C, Theobalt C, Ahmed N, Seidel HP, Thrun S. Performance capture from sparse multi-view video. In: Proc. of ACM SIGGRAPH 2008. 2008, p. 1–10.

[6] Bartoli A, Zisserman A. Direct estimation of non-rigid registrations. In: Proc. of British Machine Vision Conf. 2004, p. 221–31.

[7] Hilsmann A, Eisert P. Joint estimation of deformable motion and photometric parameters in single view video. In: ICCV 2009 Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment. 2009, p. 1–6.

[8] Gay-Bellile V, Bartoli A, Sayd P. Direct estimation of non-rigid registrations with image-based self-occlusion reasoning. IEEE Trans on Pattern Analysis and Machine Intelligence 2010;32(1):87–104.

[9] Scholz V, Magnor M. Texture replacement of garments in monocular video sequences. In: Rendering Techniques 2006: Eurographics Symposium on Rendering. 2006, p. 305–12.

[10] White R, Forsyth DA. Retexturing single views using texture and shading. In: Proc. Europ. Conf. on Computer Vision 2006. 2006, p. 70–81.

[11] Bradley D, Roth G, Bose P. Augmented reality on cloth with realistic illumination. Machine Vision and Applications 2009;20(2):85–92.

[12] Pilet J, Lepetit V, Fua P. Retexturing in the presence of complex illumination and occlusions. In: International Symposium on Mixed and Augmented Reality. 2007, p. 249–58.

[13] Negahdaripour S. Revised definition of optical flow: Integration of radiometric and geometric cues for dynamic scene analysis. IEEE Trans on Pattern Analysis and Machine Intelligence 1998;20(9):961–79.

[14] Horn B, Schunck B. Determining optical flow. Tech. Rep.; Cambridge, MA, USA; 1980.

[15] Taubin G. A signal processing approach to fair surface design. In: Proc. of ACM SIGGRAPH 1995. 1995, p. 351–8.

[16] Wardetzky M, Mathur S, Kälberer F, Grinspun E. Discrete Laplace operators: No free lunch. In: Proc. of 5th Eurographics Symposium on Geometry Processing. 2007, p. 33–7.

[17] Huber P. Robust Statistics. John Wiley & Sons,; 1981.

[18] Hoaglin DC, Mosteller F, Tukey JW. Understanding Robust and Exploratory Data Analysis. John Wiley & Sons.; 1993.

[19] Radke RJ, Andra S, Al-Kofahi O, Roysam B. Image change detection algorithms: A systematic survey. vol. 14. 2005, p. 294–307.