# Garment Texture Editing in Monocular Video Sequences based on Color-Coded Printing Patterns

Volker Scholz and Marcus Magnor

MPI–I–2006–5-003 April 2006

## Authors' Addresses

Dipl. Inf. Volker Scholz
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany
vscholz@mpi-inf.mpg.de

Prof. Dr. Ing. Marcus Magnor
TU Braunschweig
Fakultät für Mathematik und Informatik
Institut für Computergraphik
Mühlenpfordtstrasse 23
38106 Braunschweig
Germany
magnor@cg.cs.tu-bs.de

## Publication Notes



## Acknowledgements

We would like to thank our fashion model Julia Luxenburger for making this project possible, especially for her patience during the tailoring and recording process. Thanks also to Michael Neff for voice recordings for the accompanying video.

**Abstract**

In this report we present a video processing algorithm for texture replacement of moving garments. We use a color-coded pattern which encodes texture coordinates within a local neighborhood in order to robustly determine the geometric deformation of the fabric. A time-coherent texture interpolation is obtained by the use of 3D radial basis functions. Shading maps are determined with a technique operating in the image gradient domain. We use the recovered shading maps to realistically apply new textures to the color-coded garment in the video sequence. Our method enables editing and even completely exchanging fabric pattern designs of garments worn by actors as a video post-processing step.

**Keywords**

video editing, texture replacement, RBF reconstruction, Poisson equation

## Contents

# 1 Introduction

Figure 1.1: Input frames (left) and texture replacement result (right).

Movie production has always consisted of elaborate post-recording work. Movie cutting has long been the only means to alter movie content after recording. Even so, the cut has great influence on the perception (and success) of a movie and probably constitutes the artistically most demanding challenge to any movie director. Today's digital image processing tools have greatly advanced movie editing capabilities. However, considerable, time-consuming manual interaction is still necessary if the content of a recorded scene is to be altered long after all actors have gone home.

Our method addresses a specific problem of movie post-production. We propose a tool to realistically alter garment appearance with only a minimum of manual interaction. Our method is also useful for virtual fashion presentation in e-commerce. Current rotoscoping software allows tracking edges or single features in videos for tasks like matting of CGI objects, selective filtering and creating cartoon animation from video [1]. For our purpose, however, an automatic approach is needed which can track several hundred texture features in parallel while handling occlusions automatically. We propose such a system to enable texture replacement with correct texture deformation and lighting. New visual effects by applying video textures are also possible.

This report is organized as follows. In Chapter 2 we describe related work in this area. Chapter 3 gives an overview of our system and in Sections 3.1-3.4 we detail our proposed method. Chapter 4 presents results and we end by drawing conclusions on our work.

# 2 Related Work

Several authors have worked on texture replacement in still images. [32] propose to replace near-regular texture patterns in a plane by learning a statistical texture model and lighting distributions from a sample image. [24] use texture replacement in their image editing system. Depth information is used to generate foreshortening distortions of the texture, while lighting changes are also extracted. Image Analogy [15] and Image Quilting [6] show texture transfer effects which preserve local appearance of the texture. However, texture distortion and lighting effects are not modeled explicitly. [21] present an approach which builds on user-assisted lattice extraction for near-regular texture. A PCA analysis of the obtained geometric and lighting deformation fields allows to control texture regularity. Textureshop [7] introduces the idea of using shape-from-shading to recover a rough set of normals for a non-textured surface in the image and using these normals to introduce distortion in the texture synthesis process. User interaction is required to fix normal recovery errors. [35] present a faster system with improved object selection, texture synthesis and shape-from-shading algorithms. [22] use a shape-from-texture algorithm to recover the shape and irradiance map for textured cloth to replace the texture.

Our work is also related to cloth motion capture where the goal is to capture 3D motion. Approaches for general textures [28, 29] and for color-coded patterns [12, 30] have been proposed. While periodic and general patterns can cause serious correspondence problems, color-coded patterns are able to avoid this problem. These methods, however, all rely on synchronized multi-video footage of the garment.

The major difficulty of replacing texture in monocular video streams consists of achieving temporal coherence. A single-frame method inevitably leads to flickering artifacts. [27] have proposed an algorithm for real-time non-rigid surface detection for arbitrary textures which detects a surface by per-frame feature matching in conjunction with a deformable mesh model. Being a single frame method, however, temporal coherence is not considered. They extend this work in [26] by taking shading effects into account. Texture replacement for video data maintaining temporal coherence has been attempted only recently [18, 19, 20]. The method is based on user-assisted lattice extraction for near-regular texture on cloth. The lattice structure is modeled by a Markov Random Field and tracked with an affine Lucas-Kanade algorithm. Temporal coherence of the texture deformation and shading maps is achieved by spatiotemporal smoothing as a post-processing step (median filtering for the shading maps).

Determining reflectance and shading at each scene point is also referred to as the intrinsic image problem. The goal is to decompose an input image into two images, one containing the shading information and the other the reflectance information. [24] make the simplifying assumption that large-scale luminance variations are due to lighting, while small-scale variations are due to texture. The texture features are blurred with an adaptive bilateral filter. A texture image with uniform lighting is obtained by dividing the initial image by the blurred image. The computer vision literature contains several algorithms to solve the general intrinsic image problem. [31] use machine learning for classifying image pixels while [8] rely on a projection of color onto gray images minimizing image entropy. Our approach is inspired by [10], where the shading field is recovered by removing reflectance changes in the gradient image. Integrating the manipulated gradient field by solving a Poisson equation leads to the shading image. Poisson equations have also been used for seamless editing of images [25] and video [33].

Our approach is most closely related to the work by [19]. Their method is based on near-regular textures while we recover garment shape information by using a special pattern printed on the fabric. Our method is robust to large deformations, lighting changes, and occlusions and is reinitialized at every frame. Tracking is fully automated and does not require any user interaction as in [19]. Knowing the garment's texture, we have more a-priori information available to avoid ambiguous correspondence matches. This enables to robustly cope also with large motion. Seam detection and interpolation are fully automated. By encoding texture coordinates in our color code, our method can also deal with folding topologies where parts of the fabric are hidden by self-occlusion. Due to ambiguous correspondences this is difficult for methods based on periodic textures.

In the following we

- demonstrate a robust automatic method for tracking garment texture deformation and occlusion in uncalibrated monocular video sequences,
- apply 3D spatiotemporal RBF approximation as a general approach to ensure temporal coherence, and we
- determine shading maps by solving a spatiotemporal Poisson equation.

As the result, we are able to manipulate and change garment texture as a post-processing step.

# 3 Processing Pipeline

Figure 3.1 shows an overview of our system. For proper texture replacement, we need to segment the images into garment and background sections. For this

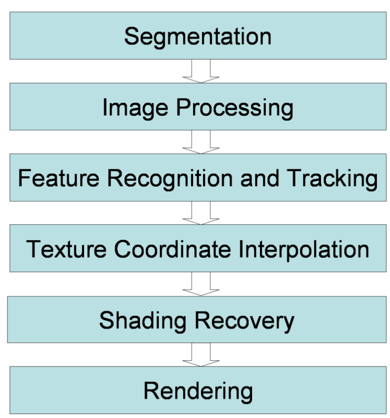


Figure 3.1: Overview of the processing steps of our method.

purpose we use the rotoscoping software by [1]. We are grateful to the authors for having made their source code freely available. In general, any other video segmentation method [17, 34] which delivers alpha mattes could also be used. These methods all require some amount of user interaction. This preprocessing yields boundary curves which are converted into a binary mask for the foreground on which all further processing is done. Next, we perform image processing for feature classification. We use garments with a custom-designed color-coded pattern. The pattern uses five different colors and is constructed in a way that allows to identify each dot in the pattern by its $3x3$ neighborhood. We use the single-frame method from [30] which identifies the dots by their local neighborhood with a region-growing approach. Texture coordinates can then automatically be assigned to each dot. The results of the labeling algorithm are complemented by a feature tracker and fed into a texture interpolation algorithm which determines a time-coherent image texture from the feature positions. For realistic texture shading we determine the shading image. The new texture is rendered into each video image by multiplying texture color with the corresponding shading image.

## 3.1 Image Processing

We convert the input video images into HSV color space in order to increase color recognition robustness against illumination changes. For color classification, we only use the hue channel and learn the five color classes from an example image. The feature pixels are identified with an adaptive thresholding algorithm [11] in the luminance image. From these pixel positions we collect the hue values and fit a Gaussian distribution to each color class (Gaussian mixture model GMM) with a statistical technique [9]. First, we run $k$-means with random initial centers on the input data and apply the EM algorithm for determining GMMs [14]. As the EM algorithm can stagnate in local minima, this procedure is restarted 10 times and the result with the best log-likelihood is kept as the final result. After this step, the pixels segmented by the adaptive thresholding method can be classified into five color classes. For this purpose we compute maximum-likelihood decision boundaries from the Gaussian parameters $\mu_i, \sigma_i$ of each color class. The color-classified image is labeled with a connected component algorithm for every color separately [13]. The obtained features are filtered by an upper and lower bound for their area. Finally, for every feature the center of mass is calculated. We now have 2D image coordinates for a number of color dots on the garment.

## 3.2 Feature Recognition and Tracking

We use two different pieces of apparel for our experiments, a dress and a skirt. A priori known is the fabric's printed pattern matrix $M$ which contains a color label for each dot in the pattern. We identify the outline of the individual cloth panels (three for the dress and two for the skirt, Fig. 3.2) in the pattern matrix manually and identify also the boundary dots adjacent to the seams. Panel boundaries are interactively identified only once per garment. The algorithm proposed in [30] labels the features obtained in the image processing step with their indices $i$, $j$ in the pattern matrix $M$. In a first step, a seed dot with its $3x3$ neighborhood is identified and adjacent dots are found by region growing. The direction of search is directed by the local principal directions $u$, $v$ of the pattern lattice. The obtained indices $i$, $j$ yield the texture coordinates for the feature dots. Details of the algorithm and the construction of the color code can be found in [30]. The original feature labeling algorithm is a robust single-frame method which does

Figure 3.2: Our dress has three panels: one front panel and two back panels (left). The skirt has a front and a back panel (right).

not use tracking history. Its performance deteriorates at oblique surface angles and requires also that a seed with a $3x3$ neighborhood can be identified for each connected texture component in the image. In order to increase the number of recognized features we apply a feature tracker to fill in missing features after labeling [23, 16]. We track the features known from labeling with image patches and set the patch size to the mean distance of neighboring features. In order to handle feature occlusions between pairs of images, we run the tracker forward in time and track the obtained position backward. If the feature is not occluded, the obtained flow vectors should be symmetric. As occlusion test we compute their deviation at the original feature position. If it is below some threshold (1 pixel in our experiments), the feature was tracked successfully and is added to the list of recognized features. Feature tracking is applied to the whole video sequence forward and backward in time in order to take feature occlusions and disocclusions robustly into account.

## 3.3 Texture Coordinate Interpolation

Our garments consist of several panels (Fig. 3.2). Texture interpolation is done separately for each panel. In views where several panels are visible in the image, we have to find the seams between the panels in order to determine the panel segments. We determine the visible seams in the image by identifying dots at the panel boundaries which are a priori known from the pattern matrix. As additional information we know which boundary dots of different panels are adjacent to each other at the seams. The boundary dots lie inside the panel, not on the seam and do not define a smooth boundary due to the discrete nature of the pattern. A smooth boundary polyline is obtained by interpolating a new seam point between each pair of adjacent boundary dots. We use membrane interpolation $\Delta f = 0$ where $\Delta$ is the Laplacian operator and fix the position of the boundary dots. In order to obtain a smooth polyline, we use a weighted Laplacian stencil in the

Figure 3.3: Boundary dots (white) and interpolated seam boundary (red).

corresponding linear system which assigns a higher weight to seam neighbors. The seam points define an estimate of the panel seam which cannot be determined from the images directly (Fig. 3.3). The seam polylines are used to cut out a mask for each visible panel from the foreground mask.

Radial Basis Functions (RBF) are commonly used for scattered data interpolation problems like reconstructing surfaces from point clouds [4]. A trivariate scalar RBF is defined by a set of centers $\mathbf{c}_i \in \mathbb{R}^3$ and weights $w_i \in \mathbb{R}$ as [4]

$$f(\mathbf{x}) = p(\mathbf{x}) + \sum_i w_i \cdot \phi(\mathbf{x} - \mathbf{c_i}) \tag{3.1}$$

where $\phi$ is the basis function and $p(\mathbf{x})$ is a polynomial of low degree. Since basis functions with local support do not provide the same degree of extrapolation and hole filling capabilities as functions of global support, we use the global basis function $\phi(\mathbf{x}) = \|\mathbf{x}\|$ where $\|\,\mathbf{.}\,\|$ is the Euclidean norm, and a linear polynomial $p$. The resulting surface is a biharmonic thin-plate spline. For interpolating texture coordinates $(u\,v)^T \in \mathbb{R}^2$ we use a vector-valued RBF

$$\mathbf{f}(\mathbf{x}) = \mathbf{p}(\mathbf{x}) + \sum_i \mathbf{w_i} \cdot \phi(\mathbf{x} - \mathbf{c_i}) \tag{3.2}$$

with $\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^2$ and $\mathbf{w_i} \in \mathbb{R}^2$, $\mathbf{p} \in \mathbb{R}^2$ are vectors. $\mathbf{f}$ is defined in spatiotemporal 3D space $(x, y, t)$ for temporally smooth texture interpolation. This means we have to add a time coordinate to the obtained feature positions $x$, $y$. The difference $t_{n+1} - t_n$ of adjacent video frames is set to the mean distance of neighboring features in frame $n$ in order to get a homogeneous distribution of centers $\mathbf{c}_i$ in 3D space $(x, y, t)$. We now use RBF approximation (also known as spline smoothing) by

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

Figure 3.4: Input image (a), gradient image (b), thresholded gradient image (c) and shading map (d).

solving [4]

$$\begin{pmatrix} \Phi - 8N\pi\rho I & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{w}_i \\ \mathbf{q}_i \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix} \tag{3.3}$$

where $\Phi_{ij} = \phi(\mathbf{c_i} - \mathbf{c_j})$ and $P_{ij} = p_i(\mathbf{c_j})$ for the polynomial basis $\{p_1, p_2, p_3\} = \{1, x, y\}$. The $\mathbf{q}_i$ are polynomial coefficients, $N$ is the number of centers and $I$ is the identity matrix. $\rho$ is a parameter that determines the trade-off between smoothness of the surface and fidelity to the data. This parameter is found empirically. We use $\rho = 0.005$ for our experiments. The resulting linear system is not sparse due to the global nature of $\phi$ and can be solved directly for our problem size of $N \leq 1000$ centers. RBF approximation is used for overlapping time windows of three video frames to ensure temporal smoothness. The texture coordinates are interpolated for every pixel in the foreground mask of the middle frame.

## 3.4 Shading Recovery

The goal of shading recovery is to remove the reflectance contribution of the color dots from the luminance images $I$ while preserving shading effects. We compute the directional derivatives $I_x$, $I_y$ and gradient magnitude $I_g = \sqrt{I_x^2 + I_y^2}$ of $I$ with forward differences (see Fig. 3.4) after convolution with a narrow Gaussian filter for regularization. For detecting the dot borders in the gradient image $I_g$, we apply a Laplacian of Gaussian filter on the original image $I$ [11]. We assume here that the dot edges have a higher contrast than shadow edges on the garment which is the case in practice. Other methods [10] detect hue changes of reflectance but we found that luminance edges are more stable. The detected edges are dilated with a circle-shaped morphological structure element [11] and softened by applying a Gaussian filter of size $3x3$ pixels. The inverse image yields a thresholding mask which is multiplied with the gradient images. The resulting thresholded gradient images have the property that the gradient is zero at the dot borders (Fig. 3.4c), i.e., the reflectance change of the color dots was removed from the gradient image. In order to reconstruct the image from the new gradient, we have to minimize the integral [25]

$$\iint_{\Omega_2} (\Delta I - \mathbf{v})^2 \, dx\, dy \tag{3.4}$$

with Dirichlet boundary conditions where $\mathbf{v} = (I_x I_y)^T$ is the gradient vector field. We extend this integral with a temporal smoothness term

$$\iiint_{\Omega_3} (\Delta I - \mathbf{v})^2 + \lambda \cdot I_t^2 \, dx\, dy\, dt \tag{3.5}$$

The corresponding Euler-Lagrange equation to be solved is the Poisson equation with an additional temporal term:

$$\Delta I + \lambda \cdot I_{tt} = div\, \mathbf{v} \tag{3.6}$$

and Dirichlet boundary conditions where $I_{tt}$ is the second temporal derivative in an image stack $I^1, \ldots, I^n$ and $\lambda$ a weighting factor for temporal smoothness. We set $\lambda = 1$ in our experiments.

In order to establish correspondences between pixels of different time frames, we compute dense optical flow [3] using code available on their web site. For three adjacent video frames $f_{n-1}$, $f_n$ and $f_{n+1}$, the forward $(f_n \rightarrow f_{n+1})$ and backward $(f_n \rightarrow f_{n-1})$ flow vectors for frame $f_n$ are computed. $I_{tt}$ is then approximated from corresponding pixels in three frames by second order accurate finite differences. The discretization of (3.6) uses a standard $3x3$ stencil for Laplacian and divergence and is applied on a bounding box of the foreground mask. As Dirichlet boundary condition we copy the gray values on the bounding box boundary from the input image $I$. This leads to a sparse linear system where the number of variables equals the number of reconstructed pixels. We limit the number of variables by applying this method to overlapping time windows of three video frames. We use a linear solver for asymmetric sparse matrices [5] for systems up to $n = 400.000$ variables. At the garment boundaries dot removal does not work satisfactory. We omit a border of two pixels and fill it by dilating the shading map.

The obtained shading maps are applied to the new texture during rendering by multiplication per-texel. For the texture lookup we use bilinear interpolation. We adapt the maps to a higher albedo by rescaling. The maps are rescaled by dividing with a reference white value which is obtained by recording a reference image with maximum brightness of the fabric. Note that this is correct for fabric with Lambertian reflectance only. Fortunately, our fabric is close to Lambertian and the gray value gradients still remain correct in the presence of specularities which is more important to the human eye than absolute values.

# 4 Results and Conclusion

The “bridge” video sequence is recorded with a commercial DV camcorder (720x572 pixels) that automatically employs Motion-JPEG video compression. The “catwalk” sequence is acquired using an Imperx 1004C vision camera (1004x1004 pixels) at 25 frames per second. We put the camera on a tripod. However, our method is not limited to static camera position and works just as well for hand-held camera sequences. The Imperx camera is color-calibrated with a reference color checker and a linear regression model (4$x$4 matrix for homogeneous RGB coordinates) in order to obtain a good separation of the garments’ dot colors for feature classification. For the garments we use a cotton fabric with a custom-printed color pattern using a medium gray tone as background. A high-brightness contrast between the color dots and background is needed for robust feature recognition (adaptive thresholding), whereas the shading algorithm needs a reasonably bright fabric, so we meet both requirements with a medium gray tone. The dot spacing is 3.2 cm and the diameter is 2.1 cm which is a compromise between high sampling rate of the surface and sufficient dot size in the image.

All experiments are performed on a Pentium IV 3.2 GHz with 2 GB RAM. The average computation time for the automatic processing steps (Fig. 3.1) in our unoptimized MATLAB implementation for a 1004$x$1004 video frame is 45 seconds (30 seconds for solving the Poisson equation). Dense optical flow is precomputed which takes 90 seconds per frame. Selected algorithms are implemented in C: labeling, RBF evaluation, bilinear texture lookup and both optical flow methods. Fig. 4.1 shows the accuracy of our texture interpolation algorithm. The corners of the overlayed checkerboard texture lie on the geometric centers of the color dots, except for areas where only few features were detected (extrapolation at the garment boundaries). The temporal smoothness of the interpolation can be assessed in the accompanying video. We obtain realistic shading maps which preserve shadows and the shading of cloth folds (Fig. 3.4, 4.2, 4.3, 4.4). Applying them to the new texture enhances the realism of the replacement result.

One limitation of our method is that video segmentation still requires user interaction. The radial basis functions do not model discontinuities explicitly which

Figure 4.1: Accuracy of texture distortion visualized as overlayed checkerboard texture.

Figure 4.2: Shadows are preserved in our renderings.

leads to oversmoothing at these locations. In the future we plan to extend the polynomial basis of the RBFs with step functions at discontinuities. We found that shading recovery works satisfactory only for uncompressed video data, because the algorithm requires accurate gradients. Our spatiotemporal approach reduces flickering artifacts in the shading maps. Using larger time windows, however, would improve the results. Still, our proposed method is able to replace the fabric texture with realistic deformation and lighting for a wide range of real-world scenes. It can robustly deal with large deformations, lighting changes and occlusions. At the end of the video we show the application of video textures as a new visual effect for video post-processing.

We have presented a system for automatic texture replacement of color-coded garments. Visually convincing replacement results are obtained by using 3D spatiotemporal RBF approximation. We also show that the Poisson technique can capture small details at cloth folds.

Currently, the bottleneck in our implementation is the computation of the shading maps. Using faster solvers (e.g. multigrid) would improve our system's performance and memory footprint. Changing the reflectance properties of the fabric and relighting are also interesting topics for future research.

Figure 4.3: Bridge and catwalk sequence. We are able to replace the original, recorded color-dot texture on the apparel with arbitrary pattern designs.

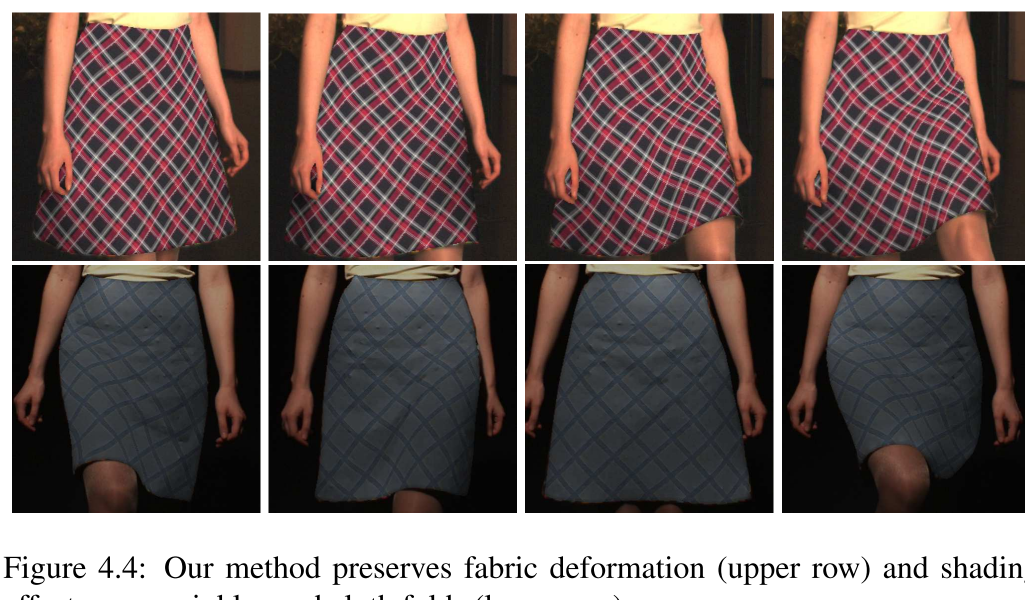

Figure 4.4: Our method preserves fabric deformation (upper row) and shading effects near wrinkles and cloth folds (lower row).

# Bibliography


[1] A. Agarwala, A. Hertzmann, D. Salesin, and S. M. Seitz. Keyframe-based tracking for rotoscoping and animation. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2004)*, 23(3):584–591, 2004.

[2] M. Bell and W. T. Freeman. Learning local evidence for shading and reflectance. In *Proc. ICCV*, pages 670–677, 2001.

[3] M. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Proc. ICCV*, pages 231–236, 1993.

[4] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of ACM SIGGRAPH 2001*, pages 433–442, 2001.

[5] T. A. Davis. Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.*, 30(2):196–199, 2004.

[6] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of ACM SIGGRAPH 2001*, pages 341–346, 2001.

[7] H. Fang and J. C. Hart. Textureshop: texture synthesis as a photograph editing tool. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2004)*, 23(3):354–359, 2004.

[8] G. D. Finlayson, M. S. Drew, and C. Lu. Intrinsic images by entropy minimization. In *Proc. ECCV*, pages 582–595, 2004.

[9] V. Franc and V. Hlaváč. Statistical pattern recognition toolbox for matlab. Technical Report CTU–CMP–2004–08, Center for Machine Perception, Czech Technical University, 2004.

[10] B. V. Funt, M. S. Drew, and M. Brockington. Recovering shading from color images. In *Proc. ECCV*, pages 124–132, 1992.

[11] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 2002.

[12] I. Guskov, S. Klibanov, and B. Bryant. Trackable surfaces. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, pages 251–257, 2003.

[13] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision Volume I*. Addison Wesley, 1992.

[14] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[15] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. Salesin. Image analogies. In *Proceedings of ACM SIGGRAPH 2001*, pages 327–340, 2001.

[16] Intel Corporation. *Open Source Computer Vision Library*, 2001.

[17] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2005)*, 24(3):595–600, 2005.

[18] W. Lin and Y. Liu. NRT-based texture replacement in real videos. In *ACM SIGGRAPH Sketches and Applications*, August 2005.

[19] W.-C. Lin. *A Lattice-based MRF Model for Dynamic Near-regular Texture Tracking and Manipulation*. PhD thesis, Carnegie Mellon University, 2005.

[20] W.-C. Lin and Y. Liu. Tracking dynamic near-regular textures under occlusions and rapid movements. In *Proc. ECCV*, 2006. to appear.

[21] Y. Liu, W.-C. Lin, and J. Hays. Near-regular texture analysis and manipulation. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2004)*, 23(3):368–376, 2004.

[22] A. Lobay and D. Forsyth. Recovering shape and irradiance maps from rich dense texton fields. In *Proc. CVPR (1)*, pages 400–406, 2004.

[23] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Seventh International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[24] B. M. Oh, M. Chen, J. Dorsey, and F. Durand. Image-based modeling and photo editing. In *Proceedings of ACM SIGGRAPH 2001*, pages 433–442, 2001.

[25] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2003)*, 22(3):313–318, 2003.

[26] J. Pilet, V. Lepetit, and P. Fua. Augmenting deformable objects in real-time. In *International Symposium on Mixed and Augmented Reality*, 2005.

[27] J. Pilet, V. Lepetit, and P. Fua. Real-time non-rigid surface detection. In *Proc. CVPR*, pages 822–828, 2005.

[28] D. Pritchard and W. Heidrich. Cloth motion capture. *Computer Graphics Forum (Proc. of Eurographics EG '03)*, 22(3):263–272, 2003.

[29] V. Scholz and M. Magnor. Cloth motion from optical flow. In *Proc. Vision, Modeling and Visualization (VMV)*, pages 117–124, 2004.

[30] V. Scholz, T. Stich, M. Keckeisen, M. Wacker, and M. Magnor. Garment motion capture using color-coded patterns. *Computer Graphics Forum (Proc. Eurographics EG '05)*, 24(3):439–448, 2005.

[31] M. F. Tappen, W. T. Freeman, and E. H. Adelson. Recovering intrinsic images from a single image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(9):1459–1472, 2005.

[32] Y. Tsin, Y. Liu, and V. Ramesh. Texture replacement in real images. In *Proc. CVPR (2)*, pages 539–544, 2001.

[33] H. Wang, R. Raskar, and N. Ahuja. Seamless video editing. In *Proc. ICPR (3)*, pages 858–861, 2004.

[34] J. Wang, P. Bhat, A. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2005)*, 24(3):585–594, 2005.

[35] S. Zelinka, H. Fang, M. Garland, and J. C. Hart. Interactive material replacement in photographs. In *Proc. Graphics Interface*, pages 227–232, 2005.

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.


Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`


---

| | | |
|---|---|---|
| MPI-I-2006-5-003 | V. Scholz | Garment Texture Editing in Monocular Video Sequences based on Color-Coded Printing Patterns |
| MPI-I-2006-5-002 | H. Bast, D. Majumdar, R. Schenkel, M. Theobald, G. Weikum | IO-Top-k: Index-access Optimized Top-k Query Processing |
| MPI-I-2006-5-001 | M. Bender, S. Michel, G. Weikum, P. Triantafilou | Overlap-Aware Global df Estimation in Distributed Information Retrieval Systems |
| MPI-I-2005-5-002 | S. Siersdorfer, G. Weikum | Automated Retraining Methods for Document Classification and their Parameter Tuning |
| MPI-I-2005-4-006 | C. Fuchs, M. Goesele, T. Chen, H. Seidel | An Emperical Model for Heterogeneous Translucent Objects |
| MPI-I-2005-4-005 | G. Krawczyk, M. Goesele, H. Seidel | Photometric Calibration of High Dynamic Range Cameras |
| MPI-I-2005-4-004 | C. Theobalt, N. Ahmed, E. De Aguiar, G. Ziegler, H. Lensch, M.A.,. Magnor, H. Seidel | Joint Motion and Reflectance Capture for Creating Relightable 3D Videos |
| MPI-I-2005-4-003 | T. Langer, A.G. Belyaev, H. Seidel | Analysis and Design of Discrete Normals and Curvatures |
| MPI-I-2005-4-002 | O. Schall, A. Belyaev, H. Seidel | Sparse Meshing of Uncertain and Noisy Surface Scattered Data |
| MPI-I-2005-4-001 | M. Fuchs, V. Blanz, H. Lensch, H. Seidel | Reflectance from Images: A Model-Based Approach for Human Faces |
| MPI-I-2005-2-004 | Y. Kazakov | A Framework of Refutational Theorem Proving for Saturation-Based Decision Procedures |
| MPI-I-2005-2-003 | H.d. Nivelle | Using Resolution as a Decision Procedure |
| MPI-I-2005-2-002 | P. Maier, W. Charatonik, L. Georgieva | Bounded Model Checking of Pointer Programs |
| MPI-I-2005-2-001 | J. Hoffmann, C. Gomes, B. Selman | Bottleneck Behavior in CNF Formulas |
| MPI-I-2005-1-008 | C. Gotsman, K. Kaligosi, K. Mehlhorn, D. Michail, E. Pyrga | Cycle Bases of Graphs and Sampled Manifolds |
| MPI-I-2005-1-008 | D. Michail | ? |
| MPI-I-2005-1-007 | I. Katriel, M. Kutz | A Faster Algorithm for Computing a Longest Common Increasing Subsequence |
| MPI-I-2005-1-003 | S. Baswana, K. Telikepalli | Improved Algorithms for All-Pairs Approximate Shortest Paths in Weighted Graphs |
| MPI-I-2005-1-002 | I. Katriel, M. Kutz, M. Skutella | Reachability Substitutes for Planar Digraphs |
| MPI-I-2005-1-001 | D. Michail | Rank-Maximal through Maximum Weight Matchings |
| MPI-I-2004-NWG3-001 | M. Magnor | Axisymmetric Reconstruction and 3D Visualization of Bipolar Planetary Nebulae |
| MPI-I-2004-NWG1-001 | B. Blanchet | Automatic Proof of Strong Secrecy for Security Protocols |
| MPI-I-2004-5-001 | S. Siersdorfer, S. Sizov, G. Weikum | Goal-oriented Methods and Meta Methods for Document Classification and their Parameter Tuning |
| MPI-I-2004-4-006 | K. Dmitriev, V. Havran, H. Seidel | Faster Ray Tracing with SIMD Shaft Culling |
| MPI-I-2004-4-005 | I.P. Ivrissimtzis, W.-. Jeong, S. Lee, Y.a. Lee, H.-. Seidel | Neural Meshes: Surface Reconstruction with a Learning Algorithm |
| MPI-I-2004-4-004 | R. Zayer, C. Rssl, H. Seidel | r-Adaptive Parameterization of Surfaces |

| | | |
|---|---|---|
| MPI-I-2004-4-003 | Y. Ohtake, A. Belyaev, H. Seidel | 3D Scattered Data Interpolation and Approximation with Multilevel Compactly Supported RBFs |
| MPI-I-2004-4-002 | Y. Ohtake, A. Belyaev, H. Seidel | Quadric-Based Mesh Reconstruction from Scattered Data |
| MPI-I-2004-4-001 | J. Haber, C. Schmitt, M. Koster, H. Seidel | Modeling Hair using a Wisp Hair Model |
| MPI-I-2004-2-007 | S. Wagner | Summaries for While Programs with Recursion |
| MPI-I-2004-2-002 | P. Maier | Intuitionistic LTL and a New Characterization of Safety and Liveness |
| MPI-I-2004-2-001 | H. de Nivelle, Y. Kazakov | Resolution Decision Procedures for the Guarded Fragment with Transitive Guards |
| MPI-I-2004-1-006 | L.S. Chandran, N. Sivadasan | On the Hadwiger's Conjecture for Graph Products |
| MPI-I-2004-1-005 | S. Schmitt, L. Fousse | A comparison of polynomial evaluation schemes |
| MPI-I-2004-1-004 | N. Sivadasan, P. Sanders, M. Skutella | Online Scheduling with Bounded Migration |
| MPI-I-2004-1-003 | I. Katriel | On Algorithms for Online Topological Ordering and Sorting |
| MPI-I-2004-1-002 | P. Sanders, S. Pettie | A Simpler Linear Time 2/3 - epsilon Approximation for Maximum Weight Matching |
| MPI-I-2004-1-001 | N. Beldiceanu, I. Katriel, S. Thiel | Filtering algorithms for the Same and UsedBy constraints |
| MPI-I-2003-NWG2-002 | F. Eisenbrand | Fast integer programming in fixed dimension |
| MPI-I-2003-NWG2-001 | L.S. Chandran, C.R. Subramanian | Girth and Treewidth |
| MPI-I-2003-4-009 | N. Zakaria | FaceSketch: An Interface for Sketching and Coloring Cartoon Faces |
| MPI-I-2003-4-008 | C. Roessl, I. Ivrissimtzis, H. Seidel | Tree-based triangle mesh connectivity encoding |
| MPI-I-2003-4-007 | I. Ivrissimtzis, W. Jeong, H. Seidel | Neural Meshes: Statistical Learning Methods in Surface Reconstruction |
| MPI-I-2003-4-006 | C. Roessl, F. Zeilfelder, G. Nrnberger, H. Seidel | Visualization of Volume Data with Quadratic Super Splines |
| MPI-I-2003-4-005 | T. Hangelbroek, G. Nrnberger, C. Roessl, H.S. Seidel, F. Zeilfelder | The Dimension of $C^1$ Splines of Arbitrary Degree on a Tetrahedral Partition |
| MPI-I-2003-4-004 | P. Bekaert, P. Slusallek, R. Cools, V. Havran, H. Seidel | A custom designed density estimation method for light transport |
| MPI-I-2003-4-003 | R. Zayer, C. Roessl, H. Seidel | Convex Boundary Angle Based Flattening |
| MPI-I-2003-4-002 | C. Theobalt, M. Li, M. Magnor, H. Seidel | A Flexible and Versatile Studio for Synchronized Multi-view Video Recording |
| MPI-I-2003-4-001 | M. Tarini, H.P.A. Lensch, M. Goesele, H. Seidel | 3D Acquisition of Mirroring Objects |
| MPI-I-2003-2-004 | A. Podelski, A. Rybalchenko | Software Model Checking of Liveness Properties via Transition Invariants |
| MPI-I-2003-2-003 | Y. Kazakov, H. de Nivelle | Subsumption of concepts in $DL\ \mathscr{F}\mathscr{L}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete |
| MPI-I-2003-2-002 | M. Jaeger | A Representation Theorem and Applications to Measure Selection and Noninformative Priors |
| MPI-I-2003-2-001 | P. Maier | Compositional Circular Assume-Guarantee Rules Cannot Be Sound And Complete |
| MPI-I-2003-1-018 | G. Schaefer | A Note on the Smoothed Complexity of the Single-Source Shortest Path Problem |
| MPI-I-2003-1-017 | G. Schfer, S. Leonardi | Cross-Monotonic Cost Sharing Methods for Connected Facility Location Games |