Physically Based Tracking of Cloth

Nils Hasler*, Mark Asbach+, Bodo Rosenhahn*, Jens-Rainer Ohm+, Hans-Peter Seidel*

*Max-Planck-Institut für Informatik
Department 4: Computer Graphics
Stuhlsatzenhausenweg 85, 66123 Saarbrücken, Germany
Email: {hasler,rosenhahn,hpseidel}@mpi-inf.mpg.de
+Chair and Institute of Communications Engineering of RWTH Aachen
52056 Aachen, Germany
Email: {asbach,ohm}@ient.rwth-aachen.de

Abstract

In this work a method for tracking fabrics in videos is proposed which, unlike most other cloth tracking algorithms, employs an analysis-by-synthesis approach. That is tracking consists of optimising a set of parameters of a mass-spring model that is used to simulate the textile, defining on the one hand the fabric properties and on the other the positions of a limited number of constrained points of the simulated cloth. To improve the tracking accuracy and to overcome the inherently chaotic behaviour of the real fabric several methods to track features on the cloth's surface and the best way to influence the simulation are evaluated.

1 Introduction

Due to its applicability in the film industry, and more recently in computer games, cloth simulation has become an important subject of research. Most of the proposed systems, however, focus on producing believable animations, rather than realistic simulations. For tracking of textiles, however, the cloth simulation should be as realistic as possible. Yet, in this work the possibility of tracking fabrics by synthesising an observed scene with a mass-spring simulation is demonstrated. Unfortunately, it was found that, as a result of the inherently instable behaviour of cloth, it was necessary to introduce non-physical forces to bias the simulation towards following the observed behaviour.

The general approach to synthesising a scene can be split into three steps. First silhouettes are extracted and features on the cloth's surface are tracked. An optimisation algorithm then adjusts cloth properties and the positions of constrained points of the fabric to fit the simulation to the observed silhouettes and tracked features. A configuration is evaluated by simulating the configuration and using the difference between captured and simulated silhouettes as the error function.

Section 2 continues by summarising the state of the art in cloth tracking and the developments of the cloth simulation literature. Section 3 outlines the contributions presented in this work. Section 4 contrasts several tracking procedures. The employed optimisation strategy is detailed in section 5 and a summary is given in section 6.

*We gratefully acknowledge funding by the Max-Planck Center for Visual Computing and Communication.

2 Previous Work

Despite strong interest has been shown in cloth simulation on the one hand and cloth tracking and in reconstruction on the other only little effort has been put into combining the efforts by employing analysis-by-synthesis approaches to cloth tracking.

One algorithm was presented by Jojic and Huang [8] which estimates the hidden points a cloth is resting on. They do require 3D-range data of the real cloth to achieve this goal and the two-phased nature of their algorithm restricts its application to static situations only.

An analysis-by-synthesis approach was also employed by Bhat et al. [2] but their goal was fundamentally different. Instead of detecting the drape of the fabric they attempted to extract the static and dynamic fabric parameters from video images. A simulated annealing optimiser was used to demonstrate that a wide variety of cloth types could be reconstructed. Their research, however, lacks verification with a mechanical cloth property extraction method such as the Kawabata Evaluation System (KES) [10]. The method has also been criticised because it is apparently unable to accurately estimate the bend resistance parameter [17].

Other cloth tracking and reconstruction approaches do not make use of a cloth simulation to guide the reconstruction. The algorithm by Pritchard and Heidrich [17] can be divided into three stages. First, stereo correspondence is used to reconstruct most of the textured cloth. Secondly, holes are interpolated and Lowe's SIFT descriptor [12] is employed to map points on the world-space cloth to points on the two dimensional reference cloth. Identified points are then connected by a seed-and-grow algorithm, rejecting spurious points in the process.

Several other algorithms to recover the three dimensional layout of a cloth were published. Scholz and Magnor [18] presented one approach that used optical flow to calculate the three dimensional scene flow. Holes in the model are not interpolated as in Pritchard's approach. Instead a deformable cloth model is matched to the surface, minimising the deformation energy of the patch. Drift is countered by constraining the edge of the simulation to the silhouette of the real cloth. Unfortunately, their algorithm was only demonstrated on synthetic data. Their work was continued with a publication on tracking cloth marked with a pseudo random coloured dots pattern [19]. The proposed algorithm detects coloured ellipses using colour and brightness information and identifies the exact position on the cloth by examining their local neighbourhood. The identified locations are connected in a way similar to Pritchard's approach. Three dimensional coordinates are reconstructed by using a multi-camera setup. As a last step holes are filled by means of a thin-plate spline interpolation technique. Recently, an extension to the approach was presented by White et al. [22] who proposed a stereo-setup to reconstruct a random pattern of coloured triangles printed on a cloth. Their principal contributions are an extension to the seed-and-grow algorithm introduced by Pritchard and a strain minimisation technique that allows them to reconstruct points that are visible in one camera only.

The other important area of research employed in this work, particle system based cloth simulation, was pioneered by Terzopoulos et al. [20]. In their work a number of techniques that are common now such as semi-implicit integration, hierarchical bounding boxes, and adaptive time-step control were proposed. Until Baraff and Witkin reintroduced semi-implicit integration [1], decreasing the computational cost of cloth simulation significantly, explicit integration techniques were common.

In the last few years two major strands of development can be made out in the cloth simulation community. One, aiming for real-time simulation, focusses on computation speed alone, sacrificing expressiveness and accuracy if necessary. Desbrun et al. simplified the equation system that needs to be solved every step by precomputing parts of it [6]. Kang and Choi used a coarse mass-spring discretisation and added wrinkles in a post-processing step by interpolating with a cubic spline [9]. Oh et al. introduced a new semi-implicit integration technique that, besides side-stepping the unnatural damping of Baraff and Witkin's integrator, is reportedly able to run in real-time [15].

The other strand attempts to simulate cloth as realistically as possible. The use of nonlinear cloth properties has been introduced by Eberhardt et al. [7]. Simplified nonlinearities have since been integrated into a number of systems such as [5, 3]. Impressive results have been presented by Volino and Magnenat-Thalmann [21]. The fabric properties employed in their system are not only nonlinear but exhibit hysteretic behaviour.

3 Contributions

The method proposed here makes the novel assumption that the best way to capture scenes containing dynamic cloth motion is to fit a simulated model of the cloth to the observed data. This approach has several advantages. Firstly, parts of the real cloth that are temporarily hidden are still modelled by the simulation. Secondly, no interpolation of holes has to be performed. Thirdly, since a full cloth simulation is implemented strain release as introduced by White et al. [22] becomes superfluous. A full cloth simulation has the additional advantage that the dynamic behaviour of the textile is integrated automatically into the reconstruction.

The resulting procedure can be divided into three logically independent blocks. Their interrelationship is summarised in figure 1. An optimisation module attempts to fit a simulated piece of fabric to video data. The quality of the fit is measured by comparing the silhouettes of the real and the simulated cloth. Additionally, points on the surface of the cloth can be tracked and incorporated into the evaluation function. Thus, the 'Tracking & 3D-Reconstruction' module feeds these informations into the 'Optimiser'. The third block, entitled 'Cloth Simulation', communicates closely with the optimiser. The optimiser instructs the cloth simulation to evaluate a set of parameters. The resulting error values are passed back to the optimiser which uses them to choose the next parameter set.

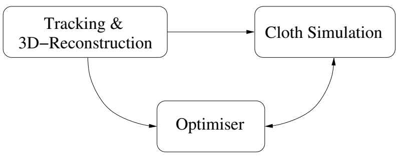


Figure 1: Overview

Unfortunately, even if the cloth simulation was ideal and the initial configuration was known exactly, the simulation and the real world would eventually diverge. This is a result of the inherently chaotic behaviour of cloth. This problem, which was first pointed out by Choi and Ko [5], is demonstrated in figure 2. Consider, for example, the cross-section of a cloth. If the cloth is compressed along its major axis it is undefined in which direction it will buckle to evade the stress. A minimal perturbation of the configuration can in this case cause macroscopically different outcomes. The differences between real world and simulation can be a result of a number of influences such as numerical accuracy, inaccurate initial condition, locally varying fabric properties, coarse quantisation of the cloth, or physical effects that are not simulated.

Several ways to handle this dilemma can be devised but only one was implemented.

The problem can be solved by attaching virtual springs to points detected on the simulated surface, attracting them to tracked world space coordinates. This technique is reasonably fast and fairly easy to implement. Nonetheless, it does has its drawbacks too. First of all it cannot be dismissed that non-physical forces are introduced into the simulation. Yet, if the magnitude of the force is kept small enough it is expected that no negative impact can be

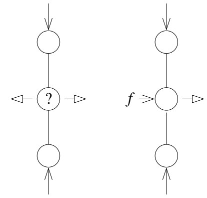


Figure 2: Instable configurations can arise in cloth when it is compressed. A small force $\vec{f}$ acting on the displayed mass-spring arrangement can cause macroscopically significantly different outcomes.

observed. On the other hand if the attraction forces are too small then the desired effect of influencing unstable conditions may not be present.

The next dilemma encountered after committing to non-physical forces is to decide on the type of spring inserted. Regular linear or damped linear springs as used in the cloth simulation increase the attraction force $\vec{f}$ proportional to the distance d between the point on the simulated cloth and the marker ($\vec{f} \sim d$). This approach does not provide for outliers. It would thus require additional handling and more importantly detection of these cases if the simulation is to be prevented from being dominated by outliers. Reciprocal ($\vec{f} \sim 1/d$) or quadratically reciprocal forces ($\vec{f} \sim 1/d^2$) do not have this problem and automatically provide for outlier rejection. Only, another sensitive parameter is introduced when entwining these forces into the cloth model and unfortunately, it cannot be optimised automatically because greater attraction forces produce better fitting results. The spring stiffness, however, should be chosen as small as possible while influencing the simulation sufficiently.

4 Tracking

In order to provide a set of points the simulation can be attracted to, these points have to be extracted from the captured videos. Several variants of two fundamentally different tracking approaches, one working in world space and the other one in two dimensional image coordinates are compared in this section.

First of all, a feature detector and descriptor have to be chosen that allow the stable redetection and recognition of features even in the presence of severe affine transformations. The Scale Invariant Feature Transform (SIFT) as introduced by Lowe [13] was chosen because of its invariance to scale and rotation and its excellent robustness to arbitrary affine transformations while keeping the dimension of the feature descriptor comparatively low. So as a first step to tracking, features are detected in the Laplace pyramid and SIFT feature descriptions are computed independently on all frames of all cameras.

For some of the tracking algorithms, detailed in the following, it is essential that the initial configuration of the cloth is known. For all experiments it is assumed that the cloth is originally resting flatly on the ground. Thus, the initial configuration can be characterised by the position of the corners of the fabric. Using this assumption an unambiguous mapping from camera coordinates into world coordinates with $y = 0$ can be defined. That way 2D-coordinates in cloth space (u, v) can be assigned to features detected in the initial frame. These features in world space, derived from different cameras, can then be combined into markers by combining those with minimal SIFT descriptor differences, whose world coordinates fall within some small fixed radius of each other.

These markers can then be reidentified in subsequent frames either using the first frame as the reference or by tracking them from one frame to the next. Either way the problem is that only a small number of markers is left by the end of the sequence. In the first case the deformation of the cloth can be too strong to reliably reidentify the exact same feature and in the second alternative features are lost gradually. That is, if a marker could not be redetected in the next frame it is dropped and lost to all subsequent frames.

For these approaches several additional constraints can be defined to improve tracking. The primary goal of these filters is to identify outliers, most of which are a result of confusing instances of the repetitive pattern printed on the cloth.

Restricting the maximum distance a marker may move between consecutive frames eliminates a few spurious markers. This filter can be applied either in world coordinates to markers or in camera coordinates to features tracked from one frame to the next or to both.

When identifying features to be combined into markers the differences of the SIFT descriptors are the main criterion to be minimised. However, several constraints can be developed which, when enforced, significantly improve the results. Rays cast from camera origins through the corresponding image planes identifying features that are to be merged, ideally intersect at exactly one point in world space. In practice however, due to inaccuracies capturing the images, calculating camera parameters, and detecting the features, they do not necessarily do so. For two involved rays the closest points on these lines can be computed instead. The 3D-marker is then placed in the middle between their respective closest points. If more than two features are involved the marker is, as visualised in figure 3, placed at the average of the middle points of the shortest connecting lines of all possible ray combinations.

Figure 3: 3D-markers are generated by intersecting rays shot from camera origins through the corresponding image planes into the scene.

An additional useful constraint can be applied when at least three features are considered. Even if the mutual minimum distances between all possible line pairs are small the average of the middle points may be far apart. If this happens the generated marker point is not localised very accurately in space. Thus, the variances of these middle points along coordinate system axes are used to reject spread out markers.

4.1 2D-Tracking

Due to the excessive loss of markers which is primarily a result of the difficulty of combining features from different cameras a different approach is described here which completely avoids this step. The difficulty of identifying features between cameras is predominantly a result of the large angles between the utilised cameras which causes substantial changes to the local feature descriptors when they are transformed from one camera into an other.

Thus, a different class of approaches is proposed here which entirely omits the explicit formation of 3D-markers. The result is that considerably more features can be tracked. Although each of them only constrains a point on the cloth's surface to lie on a ray, the increased number of features and thus constraints improves the result nonetheless.

The procedure works as follows: For the first frame the same method for identifying the cloth-space coordinates of the identified features is used as introduced above for the three dimensional tracker. Namely, under the assumption that the cloth is lying flat on the ground an unambiguous transformation from camera coordinates into cloth-space is computed and cloth coordinates are assigned to all features identified in the frame.

Due to the weaker constraint of this algorithm a possibly less intuitive approach has to be taken when calculating attraction forces. Simulated points are not attracted to fixed world coordinates any more but to arbitrary rays in world space. However, when using Plücker Coordinates to represent the feature rays the calculation of forces perpendicular to these rays becomes achievable. Assume for example that the ray, a feature at world coordinates $\vec{p}$ is attracted to, in Plücker Coordinates is represented by a normalised direction $\vec{d_0}$ and a moment $\vec{m_0}$. The shortest vector $\vec{v}$ pointing from the ray to the point can then simply be calculated by

$$\vec{v} = (\vec{m_0} - \vec{p} \times \vec{d_0}) \times \vec{d_0}$$

As interest points that are tracked are not necessarily located at the positions of mass points of the cloth simulation, forces acting on them have to be distributed to the surrounding mass points. Let ω_1, ω_2, and ω_3 represent the barycentric coordinates of a point contained in a triangle and $\vec{x_1}$, $\vec{x_2}$, and $\vec{x_3}$ the corners of the triangle then the position of the point $\vec{x}$ can be expressed as

$$\vec{x} = \omega_1 \cdot \vec{x_1} + \omega_2 \cdot \vec{x_2} + \omega_3 \cdot \vec{x_3}$$

The force $\vec{f}$ can be distributed continuously to triangle corners as visualised in figure 4 and suggested by Bridson et al. in [4] in the context of collision handling

$$\vec{f_i} = \omega_i \frac{2\vec{f}}{1 + \omega_1^2 + \omega_2^2 + \omega_3^2} \qquad i \in 1, 2, 3$$

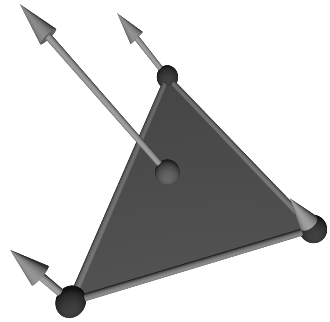


Figure 4: Forces acting on points on the simulated cloth's surface have to be distributed to the corners of the surrounding triangle.

4.2 uv-less Tracking

The 2D-tracking technique described above can be further generalised by skipping the initialisation step. That means the 2D-coordinates (u, v) of a feature on the surface of the cloth are not known during tracking. This constraint on the one hand significantly decreases the complexity of the tracker because it has to work on consecutive frames only. So any sparse optical flow algorithm is suitable for calculating the attractions.

On the other hand the complexity rises on the cloth simulation side as it is no longer possible to attract fixed cloth coordinates to feature rays. Instead, the ray identifying one feature in the video has to be cast at runtime to intersect the simulated cloth. This point is then attracted to the ray identifying the same feature from the same camera in the next frame. Utilising the bounding box hierarchy that is already used for collision detection, ray-triangle intersections can efficiently be calculated as described by Mahovsky and Wyvill in [14].

4.3 Results

Figures 5 and 6 display the number of features or markers tracked with different algorithms of a typical sequence. While the procedures in figure 5 track points in three-dimensional world coordinates those displayed in figure 6 track them independently in every view.

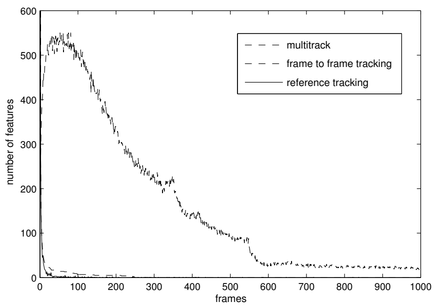


Figure 5: Number of features tracked in 3D

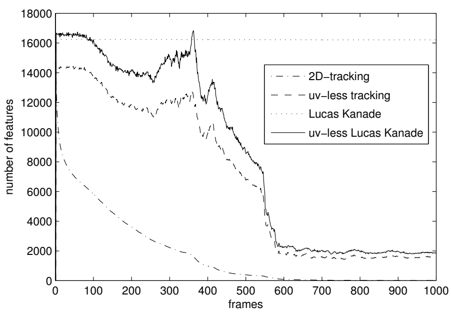


Figure 6: Number of features tracked in 2D

The algorithm here called 'reference tracking' detects features in the first frame, combines them into markers and then attempts to detect these markers by identifying their SIFT-descriptors in subsequent frames similar to the approach Pritchard and Heidrich take in [17]. Unfortunately, due to the strong deformation of the cloth and the repetitive pattern of the cloth only few markers can be recognised. By tracking from one frame to the next slightly better results can be achieved. However, the attainable results are still not satisfactory. This changes when more than one reference frame is used. The method 'multitrack' uses the last 20 frames as a reference for the current frame which explains the warm-up period observed in the first frames.

In figure 6 four variants of a 2D-tracking algorithm are compared. While '2D-tracking' and 'Lucas Kanade' detect features in the first frame and track them throughout the whole sequence the other two trackers track features from one frame to the next independent of whether these features were previously detected or not. The well-known Lucas Kanade tracker seems to perform best but, unfortunately, features that get covered by folds are not dropped but accumulate along the crests of the folds as shown in figure 7.

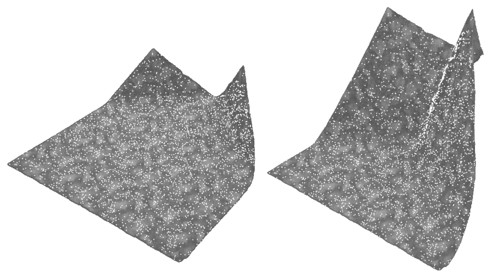


Figure 7: Features tracked by the Lucas Kanade tracker accumulate along folds instead of being dropped.

5 Optimisation

In this section the employed optimisation strategy is introduced and overall results of the proposed technique are presented.

While fabric properties obviously do not change during a sequence the positions of the constrained points of the cloth are valid for a single frame only. Consequently, a two layered optimisation strategy as displayed in figure 8 is chosen. In the outer loop scene parameters, that is parameters that are scene invariant, are optimised. The inner loop optimises parameters that change every frame. Both optimisers employ asynchronous parallel pattern search as described by Kolda [11].

The error function employed in the inner optimisation loop is calculated by summing the differences between the silhouettes of the simulated cloth and the measured silhouettes of all used cameras. The outer optimisation uses an error function that simply sums the error of the inner function for all frames.

As the cloth is initially lying flat on the ground it is easy to apply a simulated annealing opti-

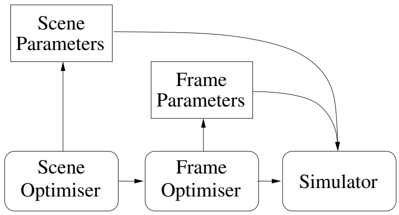


Figure 8: Two layered optimisation

miser [16] to finding the initial position of the simulated cloth.

It is assumed that the initial condition of the actual optimisation run is known a priory. It is also assumed that the basic experiment such as 'lift at one corner' is known. This is unfortunately necessary because the positions of the points that are lifted or dragged have to be added to the local set of parameters and must be marked to be immovable by the cloth simulation.

5.1 Results

In figures 9 and 10 the residual errors are shown. The results were obtained employing different attraction forces normalised to the error attained when optimising without using attraction forces.

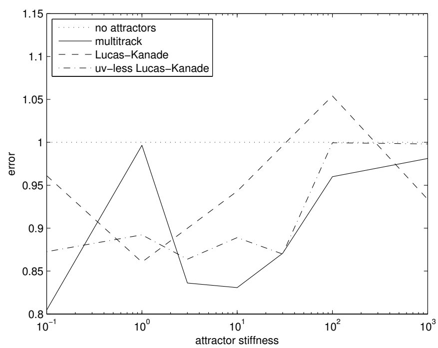


Figure 9: Residual error of optimisation using quadratic reciprocal attraction forces normalised by the optimisation error using no attraction forces.

It is apparent that using attraction forces results in all cases in better fits than without them. The

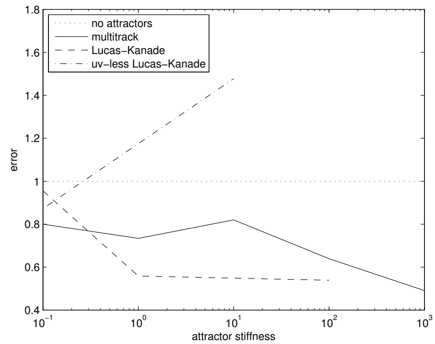


Figure 10: Residual error of optimisation using linear attraction forces normalised by the optimisation error using no attraction forces.

one exception in figure 9 is probably a result of the optimiser's inability to find a better solution. In figure 10 some configurations are missing because the simulation sometimes becomes instable if strong linear attraction forces are employed. When comparing figures 9 and 10 it becomes apparent that the linear attraction forces produce significantly better fitting results but strong attraction forces, in particular the linear variant, negatively influences the ability of the system to estimate underlying geometry. This is not a problem for the current project but for future research this is a critical feature.

6 Conclusion

In this paper an analysis-by-synthesis framework was introduced that demonstrated its applicability to tracking simple textiles. During development special care was taken that the system can easily be extended to more general tracking problems than were investigated here. The performance of a number of different tracking algorithms was evaluated and a few were applied to supporting the optimisation process by introducing different non-physical forces into the simulation.

Overall, the proposed procedure is deemed successful as the tracking results are visually appealing (see figure 11). Only the optimisation is computationally too expensive and in some cases the simulation becomes unstable if strong attraction forces are employed. A single optimisation takes from 10h to

30h if it is run in parallel on 7 AMD Opteron processors at 2.2GHz. So further effort will have to go into improving the speed of the procedure.

References

[1] D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proceedings of ACM SIGGRAPH 98*, pages 43–54. ACM Press, 1998.

[2] K. Bhat, C. Twigg, J. Hodgins, P. Khosla, Z. Popović, and S. Seitz. Estimating cloth simulation parameters from video. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2003)*, pages 37–51. ACM Press, 2003.

[3] R. Bridson. *Computational aspects of dynamic surfaces*. PhD thesis, Stanford University, 2003.

[4] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Transactions on Graphics (ACM SIGGRAPH 2002)*, 21(3):594–603, July 2002.

[5] K.-J. Choi and H.-S. Ko. Stable but responsive cloth. *ACM Transactions on Graphics (ACM SIGGRAPH 2002)*, 21(3):604–611, July 2002.

[6] M. Desbrun, P. Schröder, and A. Barr. Interactive animation of structured deformable objects. In *Proceedings of Graphics Interface (GI 1999)*, pages 1–8. Canadian Computer-Human Communications Society, 1999.

[7] B. Eberhardt, A. Weber, and W. Straßer. A fast, flexible, particle-system model for cloth draping. *IEEE Computer Graphics and Applications*, 16(5):52–59, September 1996.

[8] N. Jojic and T. Huang. On analysis of cloth drape range data. In *ACCV '98: Proceedings of the Third Asian Conference on Computer Vision-Volume II*, pages 463–470, London, UK, 1997. Springer-Verlag.

[9] Y.-M. Kang and H.-G. Cho. Bilayered approximate integration for rapid and plausible animation of virtual cloth with realistic wrinkles. In *Proceedings of Computer Animation*, pages 203–214. IEEE Computer Society, 2002.

[10] S. Kawabata. *The standardization and analysis of hand evaluation*. The Textile Machinery Society of Japan, 1980.

[11] T. Kolda. Revisiting asynchronous parallel pattern search. Technical Report SAND2004-8055, Sandia National Laboratories, Livermore, CA 94551, February 2004.

[12] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, Sep 1999. SIFT first introduced.

[13] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[14] J. Mahovsky and B. Wyvill. Fast ray-axis aligned bounding box overlap tests with plücker coordinates. *The Journal of Graphics Tools*, 9(1):37–48, 2004.

[15] S. Oh, J. Ahn, and K. Wohn. Low damped cloth simulation. *The Visual Computer (to appear)*, 2005.

[16] W. Press, W. Vetterling, S. Teukolsky, and B. Flannery. *Numerical Recipes in C++: the art of scientific computing*. Cambridge University Press, 2nd edition, 2002.

[17] D. Pritchard and W. Heidrich. Cloth motion capture. *Computer Graphics Forum (Eurographics 2003)*, 22(3):263–271, September 2003.

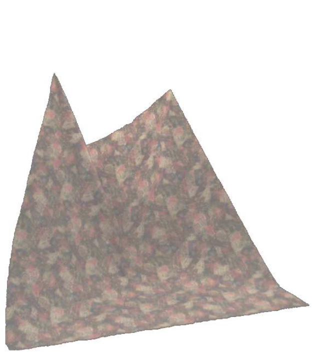
[18] V. Scholz and M. Magnor. Cloth motion from optical flow. In B. Girod, M. Magnor, and H.-P. Seidel, editors, *Proc. Vision, Modeling and Visualization 2004*, pages 117–124, Stanford, USA, November 2004. Akademische Verlagsgesellschaft Aka GmbH.

[19] 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, August 2005.

[20] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *Computer Graphics (Proceedings of ACM SIGGRAPH 87)*, pages 205–214. ACM Press, July 1987.

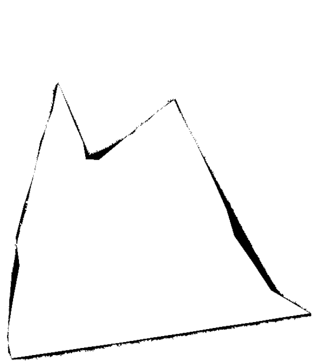
[21] P. Volino and N. Magnenat-Thalmann. Accurate garment prototyping and simulation. *Computer-Aided Design Applications*, 2(5):645–654, 2005.

[22] R. White, D. Forsyth, and J. Vasanth. Capturing real folds in cloth. Technical Report UCB/EECS-2006-10, EECS Department, University of California, Berkeley, February 2006.

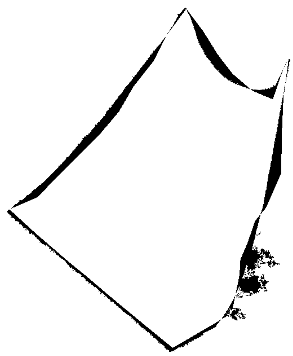


(a) camera 2

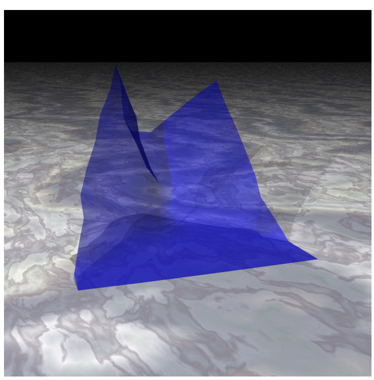
(b) camera 4



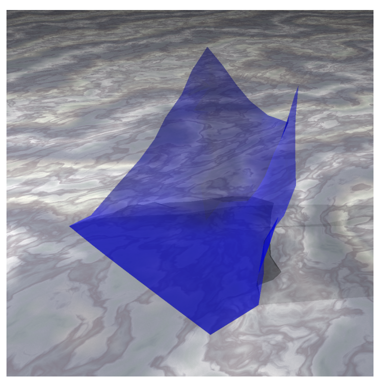
(c) silhouette difference of camera 2



(d) silhouette difference of camera 4



(e) reconstructed cloth (approx. camera 2)



(f) reconstructed cloth (approx. camera 4)

Figure 11: Reconstruction of two views