SPACETIME-CONTINUOUS GEOMETRY MESHES FROM MULTI-VIEW VIDEO SEQUENCES

Bastian Goldluecke and Marcus Magnor

MPI Informatik
Stuhlsatzenhausweg 85, 66121 Saarbruecken, Germany
{bg, magnor}@mpii.de

ABSTRACT

We reconstruct geometry for a time-varying scene given by a number of video sequences. The dynamic geometry is represented by a 3D hypersurface embedded in space-time. The intersection of the hypersurface with planes of constant time then yields the geometry at a single time instant. In this paper, we model the hypersurface with a collection of triangle meshes, one for each time frame. The photo-consistency error is measured by an error functional defined as an integral over the hypersurface. It can be minimized using a PDE driven surface evolution, which simultaneously optimizes space-time continuity as well. Compared to our previous implementation based on level sets, triangle meshes yield more accurate results, while requiring less memory and computation time. Meshes are also directly compatible with triangle-based rendering algorithms, so no additional post-processing is required.

1. INTRODUCTION

Our goal is to reconstruct temporally coherent geometry using multi-video data from only a handful of cameras distributed around a scene. The geometry models obtained that way enable us to render the dynamic scene from arbitrary viewpoints in high quality, using image-based rendering techniques we investigated earlier [1]. In this paper, however, we focus on the aspect of spatio-temporal reconstruction, a computer vision problem that is a primary focus of research interest.

A consistent mathematical model for reconstruction approaches are *weighted minimal surfaces*, which minimize an energy functional given as a surface integral of a scalar valued error function. The variational formulation of these kind of problems leads to a surface evolution PDE. Faugeras and Keriven [2] analyzed how minimal surfaces can be employed for 3D reconstruction of static scenes from multiple views. This technique was recently extended to simultaneously estimate the radiance of surfaces, and demonstrated to give good results in practice [3]. Another well-known technique which utilizes minimal surfaces is *Geodesic Active Contours* [4]. While originally designed for segmentation in 2D, it quickly became clear that it could be generalized to 3D [5], and also applied to other tasks. It is particularly attractive for modeling surfaces from point clouds [6].

In [7], we gave a mathematical analysis of weighted minimal hypersurfaces in arbitrary dimension and for a general class of weight functions. We derived the Euler-Lagrange equation yielding a necessary minimality condition. We employ the mathematical foundations thus established in [8] in order to introduce a space-time reconstruction approach. In this approach, smoothly varying geometry is recovered as a three-dimensional hypersurface embedded in space-time. The intersections of this hypersurface with planes of constant time are two-dimensional surfaces, which yield the geometry of the scene in a single time instant. Our approach defines an energy functional for the hypersurface. The minimum of the functional is the geometry which optimizes photo-consistency as well as temporal smoothness.

In our previous work, we employed level sets in order to solve the resulting PDE. Although the results are good, time and memory requirements are extremely high. In this paper, we will argue that an implementation based on triangle meshes is preferrable. Our main contribution is a way to evaluate the differential operator for a hypersurface given by a collection of triangle meshes, which is much more difficult to achieve than in the level set case. In fact, we have to resort to local level set representations. We arrive at an implementation that is both more efficient as well as more accurate than our previous approach.

In Sect. 2, we will introduce the mathematical foundations of the algorithm and give a rigorous definition of our method in terms of an energy minimization problem. Sect. 3 demonstrates how the minimization can be performed as a surface evolution implemented using triangle meshes. Implementation details are discussed in the second part of Sect. 3, where we describe our parallel scheme which computes the evolution equation. We also propose algorithms necessary to evaluate the more involved terms of the equation. Results obtained with real-world video data are presented in Sect. 4.

2. SPACE-TIME 3D RECONSTRUCTION

In this section, we briefly review the mathematical foundations of our 3D reconstruction algorithm, which were presented in [8]. We assume that we have a set of fully calibrated, fixed cameras. The input to our algorithm are the projection matrices for the set of cameras, as well as a video stream for each camera. We want to obtain a smooth surface Σ_t for each time instant t, representing the geometry of the scene at that point in time. The surfaces shall be as consistent as possible with the given video data. Furthermore, as in reality, all resulting surfaces should change smoothly over time.

2.1. Mathematical Foundations

To achieve these desirable properties, we do not consider each frame of the sequences individually. Instead, we regard all two-dimensional surfaces Σ_t to be subsets of one smooth three-dimensional hypersurface $\mathfrak{H}$ embedded in four-dimensional space-time. From this viewpoint, the reconstructed surfaces

$$\Sigma_t \ = \ \mathfrak{H} \cap \left(\mathbb{R}^3, t\right) \subset \mathbb{R}^3$$

are the intersections of $\mathfrak{H}$ with planes of constant time. Because we reconstruct only one single surface for all frames, the temporal smoothness is intrinsic to our method.

However, we have to take care of photo-consistency of the reconstructed geometry with the given image sequences. We set up an energy functional

$$\mathcal{A}\left(\mathfrak{H}\right) \ := \ \int_{\mathfrak{H}} \Phi \ dA. \tag{1}$$

defined as an integral of the scalar valued weight function Φ over the whole hypersurface. $\Phi = \Phi(s, \mathbf{n})$ measures the photo-consistency error density, and may depend on the surface point s and the normal $\mathbf{n}$ at this point. The larger the values of Φ, the higher the photo-consistency error, so the surface which matches the given input data best is a minimum of this energy functional. In [**?**], we employed a mathematical tool known as the *method of the moving frame* in order to prove the following theorem which is valid in arbitrary dimension.

Theorem. A k-dimensional surface $\mathfrak{H} \subset \mathbb{R}^{k+1}$ which minimizes the functional $\mathcal{A}\left(\mathfrak{H}\right) \ := \ \int_{\Sigma} \Phi\left(s, \mathbf{n}(s)\right) \ dA(s)$ satisfies the Euler-Lagrange equation

$$0 = \Psi \ := \ \langle \Phi_s, \mathbf{n} \rangle \ - \ \mathrm{Tr}\left(\mathbf{S}\right)\Phi \ + \ \mathrm{div}_{\mathfrak{H}}(\Phi_{\mathbf{n}}), \tag{2}$$

where $\mathbf{S}$ is the shape operator of the surface, also known as the Weingarten map or second fundamental tensor. In the remainder of this section, we present a suitable choice for the error measure Φ.

2.2. Continuous Space-time Carving

We need some additional notation for color and visibility of points in space-time first. Let t denote a time instant, then a time-dependent image I_k^t is associated to each camera k. The camera projects the scene onto the image plane via a fixed projection $\pi_k : \mathbb{R}^3 \to \mathbb{R}^2$. We can then compute the color c_k^t of every point (s, t) on the hypersurface:

$$c_k^t(s) = I_k^t \circ \pi_k(s).$$

Here, the image I_k^t is regarded as a mapping assigning color values to points in the image plane.

In the presence of the surface Σ_t, let $\nu_k^t(s)$ denote whether or not s is visible in camera k at time t. $\nu_k^t(s)$ is defined to be one if s is visible, and zero otherwise.

An error measure can now be defined as

$$\Phi^S(s,t) \ := \ \frac{1}{V_{s,t}} \sum_{i,j=1}^{l} \nu_i^t(s)\nu_j^t(s) \cdot \left\| c_i^t(s) - c_j^t(s) \right\|.$$

The number $V_{s,t}$ of pairs of cameras able to see the point s at time t is used to normalize the function.

The same functional for regular surfaces in $\mathbb{R}^3$ was introduced by Faugeras and Keriven [2] for static scene reconstruction. As an additional constraint, we enforce temporal coherence in the form of temporal smoothness of the resulting hypersurface, which makes our method ideal for video sequences. The error functional can be minimized using a surface evolution implemented on a triangle mesh, as derived in the next section.

3. TRIANGLE MESH EVOLUTION

In order to find the minimum of the energy functional, we have to find a solution to the Euler-Lagrange equation (2) according to our theorem. An efficient way to do this is to rewrite it as a surface evolution [9]. Because of the large amount of computations and memory required, a parallel implementation is necessary.

3.1. Evolution Equation

A surface $\mathfrak{H}$ which is a solution to the Euler-Lagrange equation $\Psi = 0$ is likewise a stationary solution to a surface evolution equation, where Ψ describes a force in the normal direction:

$$\frac{\partial}{\partial \tau} \mathfrak{H}_\tau \ = \ \Psi \mathbf{n}. \tag{3}$$

If we start with an initial surface $\mathfrak{H}_0$ and let the surface evolve using this equation, it will eventually converge to a local minimum of $\mathcal{A}$. This surface evolution can be implemented using the well-established level set method [10].

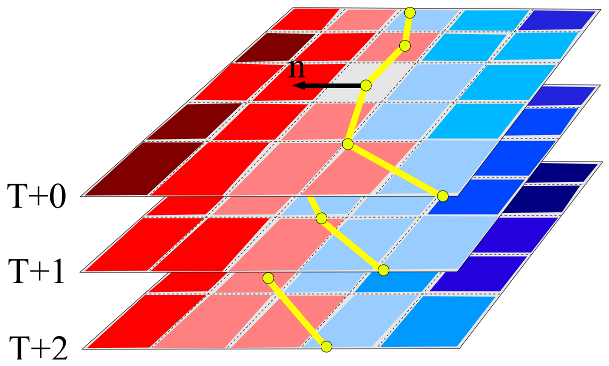


Fig. 1. *On a grid aligned with the normal, a local level set function is initialized as the signed distance transform to the mesh in the adjacent time steps in order to compute the hypersurface curvature.*

In this paper, however, we will discuss how it can be implemented using a triangle mesh representation of the geometry in each time step, and compare it with the original approach.

The main difficulty is to compute the curvature related terms for a hypersurface which is represented by per-frame triangle meshes with no inherent interconnectivity information. For this, we construct local level set representations of the hypersurface. In the mesh vertex where the differential operator is to be evaluated, we place a 5^3-grid aligned with the normal vector. The size of the grid is chosen such that it covers the next two adjacent triangles. Using the grid, we can compute $\operatorname{div}(\Phi \frac{\nabla u}{\|\nabla u\|})$ using finite differences, where u is the signed distance function to the local surface region. This corresponds to evaluating $\langle \Phi_s, \mathbf{n} \rangle - \operatorname{Tr}(\mathbf{S})\,\Phi$, see [8]. Since the error metric does not depend on the normal, the last term of Ψ is zero.

Each mesh vertex is then evolved in the normal direction according to a simple forward-difference scheme, where the time step is chosen so that a maximum velocity is not exceeded.

3.2. Implementation

For each frame the surface geometry is represented by a mesh. The initial mesh to start the evolution is given by the image based visual hull, computed as the intersection of the backprojected silhouette cones of the objects to be reconstructed. One easily calculates that there is a massive amount of data and computation time involved if the sequence is of any reasonable length. In fact, it is currently not yet possible to store the complete data together with all images of all video sequences within the main memory of a standard PC. A parallel implementation distributing the workload and data over several computers is therefore mandatory.

In our implementation, each process is responsible for the evolution of one mesh at constant time t_i. The differential operator must be evaluated for each vertex of the mesh. According to the previous section, we need the surface geometry from up to two frames apart from the current frame in order to evaluate the second order terms. Thus, this geometry has to be communicated over the network. In addition, each process needs to store the image data of its own video frame and the two adjacent frames. After each iteration, the server process may poll the current geometry from any of the other processes in order to give the user feedback about the current state of the iteration. The iteration stops when the flow field is sufficiently close to zero.

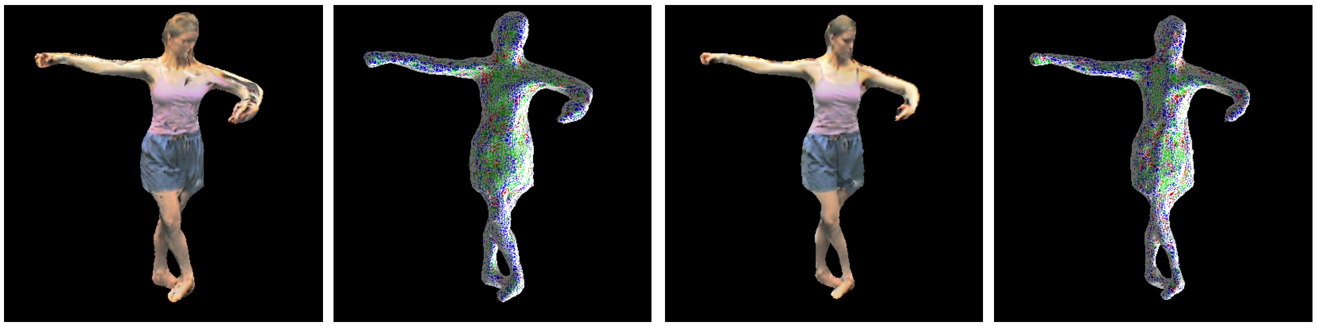
Because the initial mesh topology may be different from the final result, we have to merge or split parts of the mesh as required. This process is quite complicated, and not presented in detail here. Instead, the reader is referred to [11].

4. RESULTS AND DISCUSSION

In order to test our algorithm, we run it on real-world 320×240 RGB video sequences of a ballet dancer, recorded at 15 frames per second. All input images are segmented into foreground and background using the thresholding technique described in [12]. Consequently, we can compute the visual hull to get a starting volume for the PDE evolution. For our test runs, we choose a 50 frame long part of the sequence with the depicted frame in the middle. Our program ran on a Sun Fire 15K with 75 UltraSPARC III+ processors at 900 MHz, featuring 176 GBytes of main memory. In average, we need around one hundred iterations of the surface evolution until the hypersurface has converged to the final result. We achieve very good photo-consistency, as can be observed when the model is rendered with projective texturing from a novel viewpoint, Fig. 2. Each iteration requires 3 minutes with an average number of 8500 mesh vertices.

In comparison to the level set implementation **??**, the mesh based surface evolution requires much less evaluations of the error function, which is comparatively difficult to compute. Memory requirements are also greatly reduced, because the level set method requires storing a lot of temporary data for each cell in order to be efficient. The adaptive mesh geometry requires less data transfer between processes than the fixed level set grid, which also leads to an additional speedup. Another advantage is that the error metric is guaranteed to be evaluated exactly on a point of the surface, so the precise location of the local minimum can be obtained with more accuracy. In fact, with a fixed grid, it can theoretically happen that the minimum is missed, resulting in an incorrect reconstruction.

A problem of meshes is, of course, that topology changes are difficult to implement. However, we found that they do not require much computational overhead.



(a) Initial Mesh (Visual Hull)

(b) Final Mesh after 90 iterations

Fig. 2. *Initial and final mesh for a single frame, projectively textured as well as with color-coded curvature (Red: positive, Blue: negative, Green: close to zero). Photo-consistency has improved a lot in the final result.*

5. SUMMARY AND CONCLUSIONS

Our method takes into account all frames of a multi-video sequence simultaneously. The idea is to optimize photo-consistency with all given data as well as temporal smoothness globally. The algorithm is formulated as a weighted minimal surface problem posed for a 3D hypersurface in space-time. Intersecting this hypersurface with planes of constant time gives the 2D surface geometry in each single time instant. The energy functional defining the minimization problem enforces photo-consistency, while temporal smoothness is intrinsic to our method. The improved implementation that we have presented in this paper computes the surface evolution using a direct mesh representation of the geometry instead of a level set. In comparison to our previous work, it is faster and more memory-efficient. We also obtain more accurate reconstruction results that can be directly visualized using standard graphics hardware.

6. REFERENCES

[1] B. Goldluecke and M. Magnor, "Real-time microfacet billboarding for free-viewpoint video rendering," in *International Conference on Image Processing*, Barcelona, Spain, September 2003, pp. 713–716.

[2] O. Faugeras and R. Keriven, "Variational principles, surface evolution, PDE's, level set methods and the stereo problem," *IEEE Transactions on Image Processing*, vol. 3, no. 7, pp. 336–344, Mar. 1998.

[3] H. Jin, S. Soatto, and A. J. Yezzi, "Multi-view stereo beyond Lambert," in *IEEE Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin, USA, June 2003, vol. I, pp. 171–178.

[4] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," in *Proc. International Conference on Computer Vision*, 1995, pp. 694–699.

[5] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert, "Three dimensional object modeling via minimal surfaces," in *Proc. European Conference on Computer Vision*. Apr. 1996, vol. 1, pp. 97–106, Springer.

[6] H. Zhao, S. Osher, and R. Fedkiw, "Fast surface reconstruction using the level set method," *1st IEEE Workshop on Variational and Level Set Methods, 8th ICCV*, vol. 80, no. 3, pp. 194–202, 2001.

[7] Bastian Goldluecke and Marcus Magnor, "Weighted minimal hypersurfaces and their applications in computer vision," in *Proceedings of ECCV (2)*. May 2004, pp. 366–378, Springer.

[8] Bastian Goldluecke and Marcus Magnor, "Space-time isosurface evolution for temporally coherent 3d reconstruction," in *Proceedings of CVPR 2004*. IEEE Computer Society, July 2004.

[9] S. Osher and J. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on the Hamilton-Jacobi formulation," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.

[10] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Monographs on Applied and Computational Mathematics. Cambridge University Press, 2nd edition, 1999.

[11] Y. Duan, L. Yang, H. Qin, and D. Samaras, "Shape reconstruction from 3D and 2D data using PDE-based deformable surfaces," in *Proceedings of ECCV (3)*. May 2004, pp. 238–246, Springer.

[12] C. Theobalt, J. Carranza, M. Magnor, J. Lang, and H.-P. Seidel, "Enhancing silhouette-based human motion capture with 3d motion fields," *Proc. IEEE Pacific Graphics 2003,* Canmore, Canada, pp. 185–193, Oct. 2003.