A Point-Based Approach to PDE-Based Surface Reconstruction

Christian Linz[1], Bastian Goldlücke[2], and Marcus Magnor[1]

[1] Institut für Computergraphik
TU Braunschweig
Mühlenpfordtstr. 23, 38106 Braunschweig, Germany
{linz, magnor}@cg.cs.tu-bs.de
[2] *Graphics - Optics - Vision*
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
bg@mpii.de

Abstract. Variational techniques are a popular approach for reconstructing the surface of an object. In previous work, the surface is represented either implicitly by the use of level sets or explicitly as a triangle mesh. In this paper we describe new formulations and develop fast algorithms for surface reconstruction based on partial differential equations (PDEs) derived from variational calculus using an explicit, purely point-based surface representation. The method is based on a Moving Least-Squares surface approximation of the sample points. Our new approach automatically copes with complicated topology and deformations, without the need for explicit treatment. In contrast to level sets, it requires no postprocessing, easily adapts to varying spatial resolutions and is invariant under rigid body motion. We demonstrate the versatility of our method using several synthetic data sets and show how our technique can be used to reconstruct object surfaces from real-world multi-view footage.

1 Introduction

Many interesting problems in computer vision can be formulated as minimisation problems of an energy functional given as a surface or curve integral over a scalar-valued weight function. The variational formulation of these kinds of problems lead to a curve or surface evolution PDE. Among the well-known variational methods successfully applied in computer vision are *Geodesic Active Contours* [3]. While originally designed for segmentation of objects in 2D it can be easily generalised to 3D [4]. Caselles et al., Zhao et al. and Savadijev et al. use this approach to model surfaces from unstructured point clouds [4,22,19]. Geodesic active contours were also employed for the detection and tracking of moving objects in 2D [16]. Furthermore, minimal surfaces may be employed for 3D reconstruction of static objects from multiple views, as proposed by Faugeras and Keriven [6].

K. Franke et al. (Eds.): DAGM 2006, LNCS 4174, pp. 729–738, 2006.

All of these problems fit into one unifying framework [9]. There, a mathematical analysis of weighted minimal hypersurfaces is given in arbitrary dimension and for a general class of weight functions. An Euler-Lagrange equation is derived that yields the necessary minimality condition. As an application example, the static 3D reconstruction of a surface is generalised towards a global space-time reconstruction of the evolving surface [8]. A common feature of the aforementioned approaches is that object geometry is implicitly defined as the zero level-set of a function extending over the entire space. For an explicit representation, the object shape has to be extracted using marching-cube-like techniques [15] in a post-processing stage.

In contrast to the Eulerian approach, Duan et al. propose a PDE-based deformable model that takes the Lagrangian approach [5], i.e., shape and topology of the deformable object is always explicitly represented throughout the computation. The surface is typically represented as a triangle mesh. This model is used for surface reconstruction from volumetric images, point clouds and reconstruction from 2D multiple views. Goldlücke and Magnor recently also incorporated an explicit surface representation into their framework for space-time coherent reconstruction [7]. While this technique to solve PDEs directly yields an explicit representation of the solution, the topology information encoded in the mesh connectivity requires explicit handling when the surface topology changes. Complex local mesh operations such as the deletion and creation of edges, faces, or vertices render this approach hard to implement robustly.

Recently, purely point-based models have gained increasing popularity in traditional computer graphics as well as in the field of geometric modelling. Those models offer great flexibility since they neither store, nor have to maintain, any connectivity information. In this paper, we make use of this new modelling paradigm in the context of computer vision. Specially, we apply point-based geometry representation to the problem of PDE-based surface reconstruction. We unite several algorithms for point-based geometry processing under a common framework for PDE-based surface evolution. Our approach combines the implicit recovery of the surface topology inherent to level sets with the flexibility of a point based geometry representation stemming from the lack of connectivity information. In particular, our point-based PDE solver does not require any post-processing nor explicit handling of topology changes and easily adapts to varying spatial resolutions. Moreover, it is also invariant under rigid body motion while level sets are vulnerable to numerical diffusion under such circumstances.

The rest of this paper is organised as follows: Section 2 reviews some prerequisites concerning point-based geometry representation that are at the very heart of our work. In Sect. 3, we briefly review the mathematical framework of weighted minimal hypersurfaces and introduce our PDE-based surface reconstruction algorithm. We apply it to the problem of reconstructing surfaces from unstructured point clouds and deal with the problem of surface reconstruction from multiple views in Sect. 4. Section 5 concludes our work and presents some ideas for future work.

2 Review of Point-Based Models

This section briefly summarises some well-known algorithms from point-based modelling. Each of these algorithms is designed for its very special purpose, for example normal estimation or outlier detection. In the following sections, we unite them under a common framework for PDE-based surface evolution on point-based models.

Point Sample Neighbourhood and Normal Estimation. Our surface representation consists of an unstructured point cloud $\mathcal{P}$ in 3D space, made up of n oriented disks that describe an underlying manifold surface S. Contrary to polygonal representations, in a point-based setting all local computations are based on spatial proximity between samples, instead of geodesic proximity and the known connections between mesh vertices. For dense samples and small Euclidean neighbourhoods, both notions are similar [2]. Geodesic neighbourhoods are proposed by Klein and Zachmann [12]. While this work more reliably estimates topologically correct neighbours, it is only applicable to static point sets since the computational cost for building the underlying datastructures is too high for the applications we have in mind. Therefore, our neighbourhood structure relies on the notion of the k-nearest neighbours with respect to the Euclidean distance, denoted $\mathcal{N}_k$, which was already successfully used in [18].

This neighbourhood structure can be computed efficiently using a hierarchical space partitioning technique, for example kD-trees.

Since normal vectors are not necessarily given a-priori or may change if the shape of the model changes, they have to be estimated by analysing the local neighbourhood of a sample point. As has been demonstrated in [11], a surface normal can then be estimated by performing an eigenanalysis of the covariance matrix of the local neighbourhood $\mathcal{N}_k$. The eigenvector with the smallest eigenvalue defines the least-squares plane through the centroid of the neighbourhood $\mathcal{N}_k$ and can therefore serve as an approximation to the local surface normal.

MLS Projection. The set of oriented disks defining our model does not provide a mathematically smooth surface definition. To compute a smooth surface that approximates the sample points $\mathcal{P}$, Levin [13,14] introduces a projection operator based on a Moving Least-Squares (MLS) optimisation. This approach has first been applied to point-based geometry in $\mathbb{R}^3$ by Alexa et al. [1]. The MLS projection takes a point $\mathbf{r}$ in space and projects it onto a polynomial that locally approximates the underlying surface in the vicinity of $\mathbf{r}$. The computation of the polynomial can be split in two steps. First of all, a reference plane H is fitted to the surface samples around $\mathbf{r}$ using a weighted least-squares optimisation. This reference plane provides a local parameterisation of the sample points and is used in a second least-squares fit to compute a bivariate polynomial. A global approximation is built by blending the local polynomials.

Both the computation of the reference domain and the polynomial approximation employ a radially symmetric Gaussian weighting function $\theta(d) = e^{-d^2/h^2}$. The parameter h corresponds to the anticipated spacing between neighbouring samples. In what follows, we always adapt the bandwidth to the local sampling density of the surface as proposed in [17].

Surface Refinement. The quality of the surface reconstruction from an unstructured point set heavily depends on the sampling density of the point set. If the object is undersampled, the reconstructed surface will not be able to recover details present in the original object. Based on the surface definition reviewed in the preceding paragraph, several methods for up- and downsampling of point sets have been proposed [1,17,18]. However, these methods are expensive due to the nature of the projection operator and typically generate oversampling. An algorithm that overcomes these problems was proposed by Guennebaud et al. [10]. We employ this method in our framework to ensure a sufficient sampling density.

To achieve a uniform distribution of all samples, we let neighbouring point samples repel each other. We use an algorithm for point relaxation introduced by Turk [20] for resampling a surface defined by polygons. This approach has been adapted to point-based geometries by Pauly et al. [17].

Outlier Detection. Noise and outliers are almost always present in a point-sampled geometry. Weyrich et al. proposed a set of fast heuristics to detect outliers in point sets [21]. The underlying criteria all deliver an estimator $\chi(\mathbf{p}) \in [0, 1]$ which specifies the likelihood for a point sample $\mathbf{p}$ to be an outlier. All criteria are solely based on the analysis of the k-nearest neighbours $\mathcal{N}_k$ of $\mathbf{p}$. The final classification is then computed as a weighted average of the heuristics. The weighting of the criteria depends on the type of the underlying surface. We refer the interested reader to [21] for details on this.

Overview. In the next section, we integrate the point-based graphics tools reviewed in this section in a framework for PDE-based surface evolution algorithms. Using the point-based models, it is easier and more elegant to obtain a solution to these evolutions. In particular, our approach overcomes the need to keep the surface in a consistent manifold state as it is the case with evolution algorithms based on triangle meshes. Moreover, compared to implicit level set representations, the point-based surface easily adapts to varying spatial resolutions and may readily be rendered without the need for prior surface extraction.

3 PDE-Based Surface Reconstruction and Point-Based Models

We now turn to the mathematical framework we build our work upon. In [9], a mathematical analysis of weighted minimal hypersurfaces is given in arbitrary dimension and for a general class of weight functions. The aim is to find a k-dimensional regular hypersurface $\Sigma \subset \mathbb{R}^n$ which minimises the energy functional

$$\mathcal{A}(\Sigma) := \int_{\Sigma} \Phi(\mathbf{s}) dA(\mathbf{s}). \tag{1}$$

We restrict the weight function Φ to depend solely on the surface point $\mathbf{s}$. The necessary condition for a surface to be a minimum of this functional is to satisfy the Euler-Lagrange equation

$$\Psi := \langle \Phi_{\mathbf{s}}, \mathbf{n} \rangle - \mathrm{Tr}(\mathbf{S}), \tag{2}$$

where $\mathbf{S}$ is the shape operator of the surface. The result presented in [9] is more general in that the weight function may also depend on the surface normal $\mathbf{n}$. We do not consider this general case in this paper.

One of the fundamental questions in practise is how to solve the Euler-Lagrange equation (2). Only in a very limited number of simple cases can an analytic solution be derived directly. In all other cases, one has to numerically solve the surface evolution equation

$$\frac{\partial}{\partial \tau} \Sigma_\tau = \Psi \mathbf{n}, \tag{3}$$

where Σ_τ represents the surface $\Sigma \subset \mathbb{R}^n$ and τ is the evolution parameter. If we start with an initial surface Σ_0 and let it evolve using (3), it will eventually converge to a steady state, yielding a solution to the Euler-Lagrange equation.

We will now present our framework to solve (3) using a point-based approach. For validation, we first test our solver with a surface reconstruction from unorganised 3D sample points, which are distributed on synthetic objects whose geometry is precisely known. Our reconstruction technique and error function is similar to the work of Zhao et al. [22] and Caselles et al. [4], yet we are using the more general framework of Goldlücke and Magnor [9]. The target point cloud defines a point-based model in the sense of Sect. 2, and our surface evolution is implemented using a purely point-based model as well.

The error functional is modelled as the signed distance function $\mathcal{D}(\mathbf{s})$ for each surface sample $\mathbf{s}$ of the evolving surface to the closest point $\mathbf{t}$ on the target surface:

$$\mathcal{A}(\Sigma) := \int_\Sigma \Phi(\mathbf{s}) dA(\mathbf{s}), \tag{4}$$

$$\text{where} \quad \Phi(\mathbf{s}) := \mathcal{D}(\mathbf{s}). \tag{5}$$

The signed distance $\mathcal{D}(\mathbf{s})$ from an arbitrary point $\mathbf{s} \in \mathbb{R}^3$ to a known surface Σ is the distance between $\mathbf{s}$ and the closest point $\mathbf{t} \in \Sigma$, multiplied by ± 1, depending on which side of the surface $\mathbf{s}$ lies. In a point-based setting, the distance is hence computed as

$$\mathcal{D}(\mathbf{s}) = (\mathbf{s} - \mathbf{t}) \cdot \mathbf{n}_\mathbf{t}, \tag{6}$$

that is the normal component of the distance to the closest point on the target surface. In fact, we compute the signed distance function of $\mathbf{s}$ to a local tangent plane in $\mathbf{t}$. Since our surface Σ is closed, this simple rule works well.

According to [5], we add an extra term to (3), yielding

$$\frac{\partial}{\partial \tau} \Sigma_\tau = [v\Phi + \langle \Phi_\mathbf{s}, \mathbf{n} \rangle - \mathrm{Tr}(\mathbf{S})\Phi]\mathbf{n}. \tag{7}$$

The term $v\Phi$ allows the model to capture arbitrary non-convex shapes and avoids that the model gets stuck into local minima during deformation. v is a constant velocity. Using Euler integration, we yield the following iterative formulation of the evolution process:

$$\mathbf{p}^{\tau+\Delta} = \mathbf{p}^\tau + \Delta \Psi \mathbf{n}_\mathbf{p}, \tag{8}$$

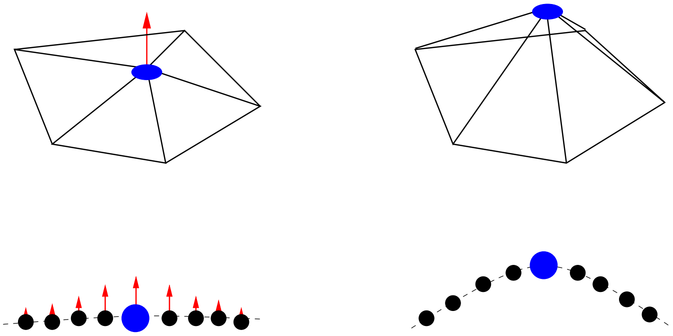


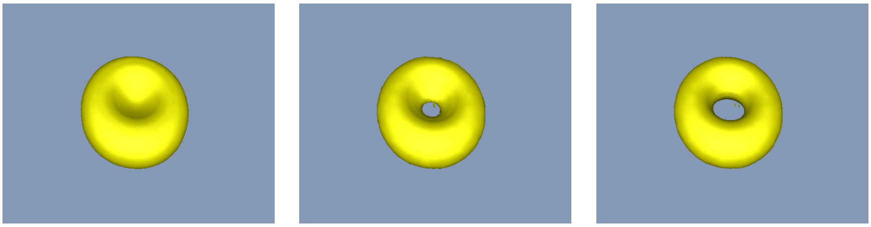
Fig. 1. Evolution on triangle meshes compared to evolution in a point-based setting. The evolution force $\varPsi$ (red vector) has to be distributed to the neighbouring sample points to achieve a behaviour comparable to triangle meshes. The dashed line illustrates the idealised MLS approximation of the sample points.

where $\mathbf{p}^{\tau}$ denotes the position of the surface point of the deformable model $\varSigma_{\tau}$ at time instant τ and $\mathbf{n}_{\mathbf{p}}$ its normal. $\varDelta$ denotes the time step and may be used to control the evolution speed.

We approximate the value $\mathrm{Tr}(\mathbf{S})$ in (7) by the mean curvature values obtained from the MLS approximation presented in Sect. 2. Likewise, $\langle \varPhi_{\mathbf{s}}, \mathbf{n} \rangle$ is approximated using fourth-order accurate central differences. Therefore, we first compute the one-ring neighbourhood of a sample $\mathbf{s}$ and displace the entire neighbourhood structure by a fixed amount in positive and negative normal direction. Since the evolution along the signed distance function does not yield a uniform point distribution and, moreover, often produces undersampled regions, we apply the upsampling scheme in combination with the point relaxation introduced in Sect. 2 to the evolving surface to ensure a good surface approximation in the next iteration. Moreover, we detect outliers that originate from an overshooting evolution force using the heuristics introduced in Sect. 2 to avoid incorrect normals and curvature values. Since the evolution is based on these values, errors would else amplify during the iterative process. Compared to evolutions on triangle meshes, we have to take care that the per-point evolution force $\varPsi$ also affects the sample points in a small neighbourhood, Fig. 1. By weighting the forces using a Gaussian kernel, we are able to mimic an evolution behaviour similar to triangle meshes.

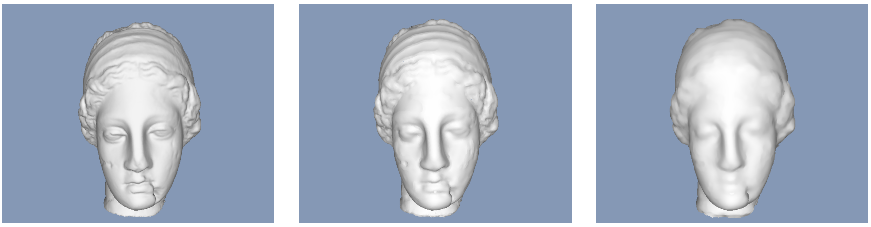
4 Results and Applications

We validate our point-based approach using several models. As a first step, we choose a model of a torus that requires an explicit topology change, Fig. 2. The surface topology is recovered implicitly by the MLS surface approximation without the need for additional operations. Using an explicit surface representation with connectivity information such as triangle meshes would have required complex local mesh operations which render this approach hard to implement



(a) Σ_{10} (b) Σ_{15} (c) Σ_{20}

Fig. 2. Shape recovery with implicit topology change. The initial point surface Σ_0 was a sphere surrounding the target surface.



(a) Original Igea (134345 samples) (b) Reconstructed Igea (160700 samples) using 25 neighbours (c) Reconstructed Igea (160700 samples) using 80 neighbours

Fig. 3. Reconstruction with varying spatial resolutions: The original model is shown on the left, a reconstruction based on a small neighbourhood is depicted in the middle. The reconstruction on the right uses a larger neighbourhood. The initial points are distributed on an sphere enclosing the geometry.

robustly. Figure 3 shows the results of the evolution on a more complex model. It also shows the adaptivity of the point-based model to different spatial resolutions by a varying interpolation radius, determined by the size of the neighbourhood structure. Higher spatial resolutions are easily obtained by placing more sample points in the desired regions. Grid-based level sets on the contrary require a more complex restructuring of the underlying grid. In both cases, the initial point surface Σ_0 was an appropriately scaled sphere surrounding the object.

In a second step, we use our point-based PDE solver to reconstruct real-world object geometry from multiple 2D images. First, however, we need some additional notation for colour and visibility of surface samples. Let I_k denote the image associated with camera k. Each camera projects the scene onto the image plane via a fixed projection of the form $\pi_k : \mathbb{R}^3 \to \mathbb{R}^2$. Then, $I_k \circ \pi_k(\mathbf{s})$ denotes the colour of the projection of $\mathbf{s}$ into the image taken by camera k. For each surface point $\mathbf{s} \in \mathbb{R}^3$, let $\nu_k(\mathbf{s})$ denote whether $\mathbf{s}$ is visible in camera k in the presence of a surface Σ or not. An error measure, taking care of photo-consistency of the evolving surface with the input images, can now be defined as

$$\Phi^C(\mathbf{s}) := \frac{1}{|\mathcal{V}_{\mathbf{s}}|(|\mathcal{V}_{\mathbf{s}}| - 1)} \sum_{i,j=1}^{l} \nu_i(\mathbf{s})\nu_j(\mathbf{s}) \cdot \chi_{i,j}(\mathbf{s}, \mathcal{N}_k) \tag{9}$$

$$\chi_{i,j}(\mathbf{s}, \mathcal{N}_k) := \frac{1}{|\mathcal{N}_k|} \sum_{\mathbf{q} \in \mathcal{N}_k} ((I_i \circ \pi_i)(\mathbf{q}) - \bar{I}_i^{\mathcal{N}_k}) \cdot ((I_i \circ \pi_i)(\mathbf{q}) - \bar{I}_i^{\mathcal{N}_k}). \tag{10}$$

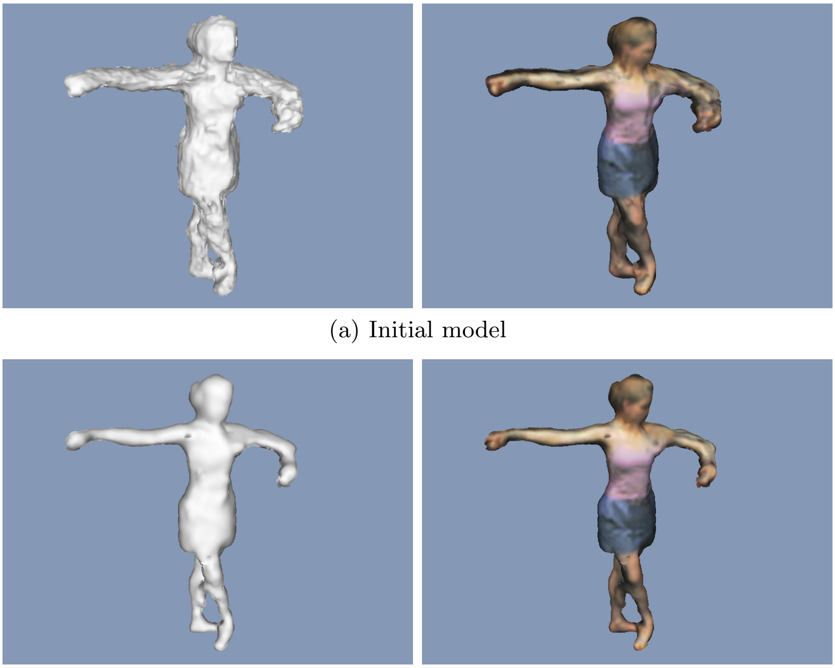
$\bar{I}_i^{\mathcal{N}_k}$ denotes the mean colour value in the k-neighbourhood $\mathcal{N}_k$ of a surface sample. This functional is a reasonable discretization of the error functional introduced in [9] for point-based models.

Using this definition of the error functional, we are able to reconstruct the surface of an object given multiple views. We test our method on multi-view footage of a dancer, recorded from 8 cameras distributed around the scene. All input images are segmented into foreground and background using a thresholding technique. The results obtained with our point-based approach on a fixed frame of the dancer sequence are shown in Fig. 4. Our point-based PDE solver clearly smoothes the initial surface Σ_0 obtained from a space-carving approach and improves photo-consistency. Compared to the approach taken in [8], the use of a point-based model gives comparable results at lower implementation complexity since explicit handling of topology changes is completely avoided.

5 Summary and Conclusions

In this paper, we have introduced a purely point-based technique to reconstruct explicit surfaces from implicit PDE definition. We demonstrated that this representation in combination with the powerful Moving Least-Squares surface approximation unites the advantages of a level set-based representation, i.e., implicit recovery of surface topology, with direct accessibility of an explicit model based on triangle meshes. Our representation does not depend on an underlying grid-structure and hence easily adapts to varying spatial resolutions and is invariant under rigid body motions. We showed the general applicability of point-based geometry representation to surface reconstruction using synthetic data sets as well as real-world data. Compared to a direct representation based on triangle meshes, the point-based model used in our work is more flexible, especially when topology changes are involved. It is thus the more natural choice for iterative surface evolution. Compared to level set-based surface representations, the point-based models are far less memory-consuming. Our, yet unoptimised, point-based implementation already outperforms a similar implementation using level sets.

We believe that with growing interest in point-based models in the research community, the flexibility of this surface representation will be exploited for various tasks in computer vision. One could, for example, extend the implementation of reconstruction from multiple views described in Sect. 4 to reconstructions in space-time as has been outlined in [7,8]. Furthermore, a detailed analysis of the convergence properties and a quantification of the approximation quality as compared to grid-based level sets would be helpful. Also, the computational complexity of our approach needs more investigation.



(a) Initial model

(b) Final result after 180 iterations

Fig. 4. Initial and final point set for a fixed frame, coloured with per-point colour information derived from the best two cameras. Photo-consistency has clearly improved in the final result as can be judged by the decrease of black and grey areas on the arms of the dancer.

References

1. Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15, 2003.
2. Nina Amenta, Marshall Bern, and Manolis Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proc. SIGGRAPH '98*, pages 415–421, New York, NY, USA, 1998. ACM Press.
3. Vincent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. In *Proc. ICCV*, pages 694–699, 1995.
4. Vincent Caselles, Ron Kimmel, Guillermo Sapiro, and Catalina Sbert. Minimal surfaces based object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):394–398, 1997.
5. Ye Duan, Liu Yang, Hong Qin, and Dimitris Samaras. Shape Reconstruction from 3D and 2D Data Using PDE-Based Deformable Surfaces. In *Proc. ECCV (3)*, volume 3023 of *LNCS*, pages 238–251. Springer, 2004.
6. O. Faugeras and R. Keriven. Variational principles, surface evolution, PDE's, level set methods and the stereo problem. In *IEEE Transactions on Image Processing*, volume 3, pages 336–344, 1998.

7. Bastian Goldlücke and Marcus Magnor. Spacetime-continuous Geometry Meshes from Multiple-View Video Sequences. In *Proc. IEEE International Conference on Image Processing (ICIP'05)*, Genoa, Italy, 2005. accepted.
8. Bastian Goldluecke and Marcus Magnor. Space-time isosurface evolution for temporally coherent 3d reconstruction. In *Proc. CVPR*, volume I, pages 350–355, Washington, D.C., USA, July 2004.
9. Bastian Goldluecke and Marcus Magnor. Weighted minimal hypersurfaces and their applications in computer vision. In *Proc. ECCV (2)*, volume 3022 of *Lecture Notes in Computer Science*, pages 366–378. Springer, 2004.
10. Gael Guennebaud, Loc Barthe, and Mathias Paulin. Real-Time Point Cloud Refinement. In *Symposium on Point-Based Graphics*, pages 41–49, 2004.
11. Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proc. SIGGRAPH '92*, pages 71–78. ACM Press, 1992.
12. Jan Klein and Gabriel Zachmann. Point Cloud Surfaces using Geometric Proximity Graphs. *Computers & Graphics*, 28(6):839–850, 2004.
13. David Levin. The approximation power of moving least-squares. *Math. Comput.*, 67(224):1517–1531, 1998.
14. David Levin. Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*, pages 37–49. Springer Verlag, 2003.
15. William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proc. SIGGRAPH '87*, pages 163–169, New York, NY, USA, 1987. ACM Press.
16. Nikos Paragios and Rachid Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, 2000.
17. Mark Pauly, Markus Gross, and Leif Kobbelt. Efficient Simplification of Point-Sampled Surfaces. In *VIS '02: Proceedings of the conference on Visualization '02*, pages 163–170. IEEE Computer Society, 2002.
18. Mark Pauly, Leif Kobbelt, and Markus Gross. Multiresolution Modeling of Point-Sampled Geometry. Technical Report 378, Computer Science Department, ETH Zurich, Switzerland, Computer Science Department RWTH Aachen, Germany, September 2002.
19. Peter Savadijev, Frank P. Ferrie, and Kaleem Siddiqi. Surface Recovery from 3D Point Data Using a Combined Parametric and Geometric Flow Approach. In *Proc. EMMCVPR*, volume 2683 of *LNCS*, pages 325–340. Springer, 2003.
20. Greg Turk. Re-tiling polygonal surfaces. In *Proc. SIGGRAPH '92*, pages 55–64, New York, NY, USA, 1992. ACM Press.
21. Tim Weyrich, Mark Pauly, Simon Heinzle, Richard Keiser, Sascha Scandella, and Markus Gross. Post-processing of Scanned 3D Surface Data. In *Symposium on Point-Based Graphics*, pages 85–94, 2004.
22. Hong-Kai Zhao, Stanley Osher, and Ronald Fedkiw. Fast surface reconstruction using the level set method. In *VLSM '01: Proceedings of the IEEE Workshop on Variational and Level Set Methods*, page 194, Washington, DC, USA, 2001. IEEE Computer Society.