A Surface Deformation Framework for 3D Shape Recovery*

Yusuf Sahillioğlu and Yücel Yemez

Multimedia, Vision and Graphics Laboratory
Koç University, Sarıyer, Istanbul, 34450, Turkey
{ysahillioglu, yyemez}@ku.edu.tr

Abstract. We present a surface deformation framework for the problem of 3D shape recovery. A spatially smooth and topologically plausible surface mesh representation is constructed via a surface evolution based technique, starting from an initial model. The initial mesh, representing the bounding surface, is refined or simplified where necessary during surface evolution using a set of local mesh transform operations so as to adapt local properties of the object surface. The final mesh obtained at convergence can adequately represent the complex surface details such as bifurcations, protrusions and large visible concavities. The performance of the proposed framework which is in fact very general and applicable to any kind of raw surface data, is demonstrated on the problem of shape reconstruction from silhouettes. Moreover, since the approach we take for surface deformation is Lagrangian, that can track changes in connectivity and geometry of the deformable mesh during surface evolution, the proposed framework can be used to build efficient time-varying representations of dynamic scenes.

1 Introduction

Deformation models have widely been used in various modeling problems of 3D computer graphics and vision such as shape recovery, animation, surface editing, tracking and segmentation [1]. The main motivation behind employing deformable models is that they in general yield smooth, robust representations that can successfully capture and preserve semantics of the data with well established mathematical foundations. They can easily adapt changes occurring in the geometry of the objects under investigation and can therefore be applied to modeling time-varying characteristics of dynamic scenes.

In this work, we rather focus on the problem of shape recovery with continuous deformable representations. PDE-driven deformation models existing in the computer vision literature can be grouped under two different categories: 1) Level sets (the Eulerian approach) and 2) active contours (the Lagrangian approach). The active contour models, or so called "snakes", were first developed

* This work has been supported by the European FP6 Network of Excellence 3DTV (www.3dtv-research.org).

B. Gunsel et al. (Eds.): MRCS 2006, LNCS 4105, pp. 570–577, 2006.

by Kass et al. [2] for detection of salient features in 2D image analysis and then extended by Terzopoulos et al. [3] to 3D for the surface recovery problem. In this Lagrangian approach, an initial parametric contour or surface is made to evolve towards the boundary of the object to be detected under the guidance of some application-specific external and internal forces that try to minimize the overall energy. The original snake model was not designed to handle possible topological changes that might occur during surface/contour evolution, nor was it capable to represent protrusions and bifurcations of complex shapes. It was nevertheless improved by many successors and has found applications in various domains of computer vision [4].

The level set technique on the other hand was first proposed by Malladi et al. [5] as an alternative to the classical snake approach in order to overcome its drawbacks mentioned above. This technique favors the Eulerian formulation with which the object shape is implicitly embedded into a higher dimensional space as the level set solution of a time varying shape function. The level-set technique, though it can implicitly handle topological changes in geometry, is computationally very expensive and inevitably necessitates a parallel implementation especially in 3D surface recovery problem [6]. More importantly, with the level set approach, the explicit connectivity information of the initial shape model is lost through the iterations between the initial state and convergence. Thus the level set technique becomes inapplicable to building dynamic meshes with fixed or slowly changing connectivity.

In general, 3D reconstruction methods for static scenes can be collected under two groups: active and passive. Active methods make use of calibrated light sources such as lasers and coded light. Most of the active scene capture technologies become inapplicable in the dynamic case. The most accurate active capture method, the shape from optical triangulation, can not for example be used when the object is in motion [7]. A plausible alternative to active methods is the use of passive techniques that usually comprise a set of multiple CCD cameras [6]. Such multicamera systems infer the 3D shape from its silhouettes (in the case of object reconstruction) and/or from multistereo texture, i.e., using color consistency. Silhouette-based techniques however are not capable of capturing hidden concavities of the object surface whereas stereo-based techniques suffer from accuracy problems. Yet, when the object to be captured is not very complicated in shape, passive techniques may yield robust, hole-free and complete reconstructions of an object in motion.

Another challenge for dynamic scene modeling is in representation. A time-varying scene sampled at a standard rate of 30 frames per second would yield enormous 3D model data if no particular care is shown to exploit redundancies between consecutive time frames. The current solution to these problems is the use of object-specific models and to animate the dynamic scene through animation parameters. However this approach is not applicable to general dynamic scenes. The real challenge here is to generate once an initial model for the object under consideration with arbitrary geometry and then to track its motion (or deformation) through time. In this respect, time-varying mesh representations with a connectivity as fixed as possible, but with changing vertex positions, would certainly provide enormous efficiency both for storage, processing and visualization. There have been very few attempts to achieve such time-consistent representations such as in [9], but these works are quite premature and can obtain time-consistent meshes only for very short time intervals. Hence the need for tools such as deformable models to be able to track connectivity changes and for improved reconstructions with passive methods such as shape from silhouette.

This paper proposes a deformation framework that captures the 3D shape of an object from a sequence of multi-view silhouette images. We take the Lagrangian approach for deformation and construct a mesh representation via surface evolution, starting from an initial model that represents the bounding surface. The deformable model is refined or simplified where necessary during surface evolution using a set of local mesh transform operations so as to adapt local properties of the object surface. The final mesh obtained at convergence can adequately represent the complex surface details such as bifurcations, protrusions and large visible concavities, unlike most of the existing snake based deformation techniques proposed in the literature.

2 Deformation Model

We take the Lagrangian approach and assume that the shape to be recovered is of sphere topology with genus 0. We should however note that this limitation can indeed be overcome by employing special procedures to detect possible splitting and merging [10].

The deformation model that we use seeks for an optimal surface S^* that minimizes a global energy term E:

$$E(S, B) = E_{\text{int}}(S) + E_{\text{ext}}(S, B) \tag{1}$$

where the internal energy component E_{int} controls the smoothness of the surface and the external energy component E_{ext} measures the match between the surface S and the object boundary B. This energy term can be minimized by solving the following partial differential equation:

$$\frac{\partial S}{\partial t} = \boldsymbol{F}_{\text{int}}(S) + \boldsymbol{F}_{\text{ext}}(S, B) \tag{2}$$

where the internal and external forces, $\boldsymbol{F}_{\text{int}}$ and $\boldsymbol{F}_{\text{ext}}$, guide the initial surface in a smooth manner towards the object boundary. The discrete form of this differential equation can be solved by surface evolution via the following iteration:

$$S_k = S_{k-1} + \triangle t(\boldsymbol{F}_{\text{int}}(S) + \boldsymbol{F}_{\text{ext}}(S, B)) \tag{3}$$

By iterating the above equation, the surface S_k converges to its optimum S^* at the equilibrium condition when the forces cancel out to 0. The external force component, $\boldsymbol{F}_{\text{ext}}$, is application-specific; its magnitude and direction depend on how far and in which direction the current surface is with respect to the targeted boundary. The external force is commonly set to be in the direction of the surface normal.

3 Shape Recovery

The mesh representation is reconstructed from the multi-view silhouettes of the object by deforming the 3D bounding sphere that encloses the shape. The bounding sphere, which is represented as a mesh, is estimated automatically by using the camera calibration parameters and retro-projecting the 2D bounding boxes obtained from silhouettes into the 3D world coordinate system.

3.1 External and Internal Forces

In our case, the external force component, $\boldsymbol{F}_{\text{ext}}$, is solely based on the silhouette information, though it is also possible to incorporate the texture information, i.e., the color consistency [11]. Following the common practice, we set the direction of the external force so as to be perpendicular to the deformable surface and write the external force at a vertex $\boldsymbol{p}$ of the surface mesh as

$$\boldsymbol{F}_{\text{ext}}(\boldsymbol{p}) = v(\boldsymbol{p}) \cdot \boldsymbol{n}(\boldsymbol{p}) \tag{4}$$

where $\boldsymbol{n}(\boldsymbol{p})$ is the normal vector and $v(\boldsymbol{p})$ is the strength of the external force at vertex $\boldsymbol{p}$. The force strength at each vertex $\boldsymbol{p}$ of the mesh and at each iteration of the surface evolution is based on how far and in which direction (inside or outside) the vertex $\boldsymbol{p}$ is with respect to the silhouettes. Thus the strength v, which may take negative values as well, is computed by projecting $\boldsymbol{p}$ onto the image planes and thereby estimating an isolevel value via bilinear interpolation:

$$v(\boldsymbol{p}) = 2\varepsilon \min_n \{G[\text{Proj}_{I_n}(\boldsymbol{p})] - 0.5\} \tag{5}$$

where $\text{Proj}_{I_n}(\boldsymbol{p})$ is the projection of the point $\boldsymbol{p}(x, y, z)$ to I_n, the n'th binary image in the sequence (0 or 1), and

$$\begin{aligned} G(x', y') = (1-\alpha)((1-\beta)I(\lfloor x' \rfloor, \lfloor y' \rfloor) + \beta I(\lfloor x' \rfloor + 1, \lfloor y' \rfloor)) \\ +\alpha((1-\beta)I(\lfloor x' \rfloor, \lfloor y' \rfloor + 1) + \beta I(\lfloor x' \rfloor + 1, \lfloor y' \rfloor + 1)) \end{aligned} \tag{6}$$

where $(\lfloor x' \rfloor, \lfloor y' \rfloor)$ denotes the integer part and (α, β) is the fractional part of the coordinate (x', y') in the binary discrete image I. The function G, taking values between 0 and 1, is the bilinear interpolation of the sub-pixelic projection (x', y') of the vertex $\boldsymbol{p}$. Thus, the external force strength $v(\boldsymbol{p})$ takes on values between $-\varepsilon$ and ε, and the zero crossing of this function reveals the isosurface. As a result, the isovalue of the vertex $\boldsymbol{p}$ is provided by the image of the silhouette that is farthest away from the point, or in other words, where the interpolation function G assumes its minimum value. The internal force component, $\boldsymbol{F}_{\text{int}}$, controls the smoothness of the mesh as the surfaces evolves towards the object boundary under the guidance of the external force. At each vertex of the mesh, first the external force is applied as specified and then the internal force tries to regularize its effect by moving the vertex to the centroid of its neighbors:

$$\boldsymbol{F}_{\text{int}}(\boldsymbol{p}) = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{p}_i - \boldsymbol{p} \tag{7}$$

where $\boldsymbol{p}_i$, $i = 0, 1, ..., N$, are the vertices adjacent to $\boldsymbol{p}$.

3.2 Surface Evolution

If the deformation model is applied as described above without any further considerations, some problems may arise during surface evolution. These are: 1) Topological problems, i.e., non-manifold triangles may appear, 2) degenerate edges may show up, 3) irregular vertices with high valence values may occur. Non-manifold triangles on the mesh structure may appear as the positions of the vertices are updated by the external forces. To prevent the occurrence of such topological problems, the maximum magnitude of the external force strength is constrained with the finest detail on the mesh, i.e., $\varepsilon < \varepsilon_{\min}/2$, where $\varepsilon_{\min}$ is the minimum edge length appearing on the mesh. To handle the two other problems, we incorporate three special procedures [12], namely edge collapse, edge split and edge flip, to the shape recovery process (see Fig. 1). These operations should carefully be applied so as to avoid illegal moves that would cause further topological problems such as fold-overs and non-manifold triangulations.

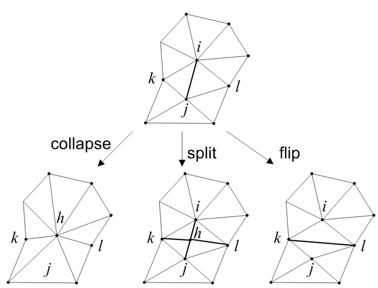

Fig. 1. Edge collapse, split and flip operations

The surface evolution process that incorporates the above operations in an adequate order continues to iterate until convergence. The evolution process includes also a position refinement procedure: Whenever a vertex, which was outside (inside) the object volume at iteration k, becomes an inside (outside) vertex at iteration $k+1$, its exact position on the boundary is computed with binary subdivision and the vertex is frozen in the sense that it is no longer subjected to further deformation. The overall algorithm is thus briefly as follows:

Iterate

- Move each unfrozen vertex $\boldsymbol{p}$ with $v(\boldsymbol{p})$ in the direction of normal $\boldsymbol{n}(\boldsymbol{p})$
- Regularize the mesh using Equation 7.
- Collapse edges with length smaller than $\varepsilon_{\min}$.
- Split edges with length exceeding $\varepsilon_{\max} = 2\varepsilon_{\min}$.
- Flip edges where necessary, favoring the vertices with valence close to 6.

Till convergence

The above algorithm, when converges, may not (and often do not) capture fine details such as protrusions and sharp surface concavities, due to the induced

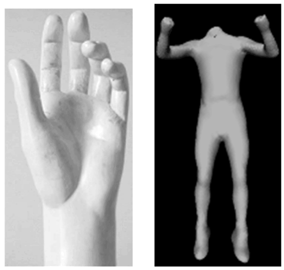

Fig. 2. The original Hand object and the synthetic Human model

regularization and insufficiency of the initial mesh resolution. Therefore after the initial convergence, the edges on the parts of the mesh, where the resolution is not sufficient, are to be split. The criterion to decide whether an edge is to be split or not is as follows: At the equilibrium state, when all the forces cancel out, if there still remain edges with their midpoints detected to be far outside the object boundary, that is, if

$$|\min_{n}\{G[\mathrm{Proj}_{I_n}(\boldsymbol{p})] - 0.5\}| > 0.5 \tag{8}$$

then these edges are split and the surface evolution process is restarted and iterated until convergence. This process is repeated until all the vertices of the deformable mesh strictly get attached onto the object boundary.

4 Experimental Results

We have tested the proposed shape recovery technique on two objects, the Hand object and the synthetic Human model [14], which are displayed in Fig. 2. The Hand object has been pictured horizontally with a calibrated camera from 36 equally spaced view angles and then the corresponding silhouettes have been extracted, whereas the silhouettes of the human model have been created by projecting the synthetic model into the image planes of 16 synthetic cameras. The size of each silhouette image is 2000×1312 for the Hand, and 1024×768 for the Human. The results of the shape recovery process using these silhouettes are displayed in Fig. 3, where we observe each of the models at various iterations as it deforms from the sphere to the recovered object shape. The surface regions marked as white, blue and magenta indicate those parts of the model that are outside, inside and on the boundary of the object volume, respectively. Although the objects, especially the Hand object, contain severe occlusions and concavities, their shapes are accurately recovered, after 143 iterations for the Hand and 177 iterations for the Human model. The final surface representations obtained are smooth, regular and topologically plausible meshes with most vertices on the visual hull and capable to capture fine details.

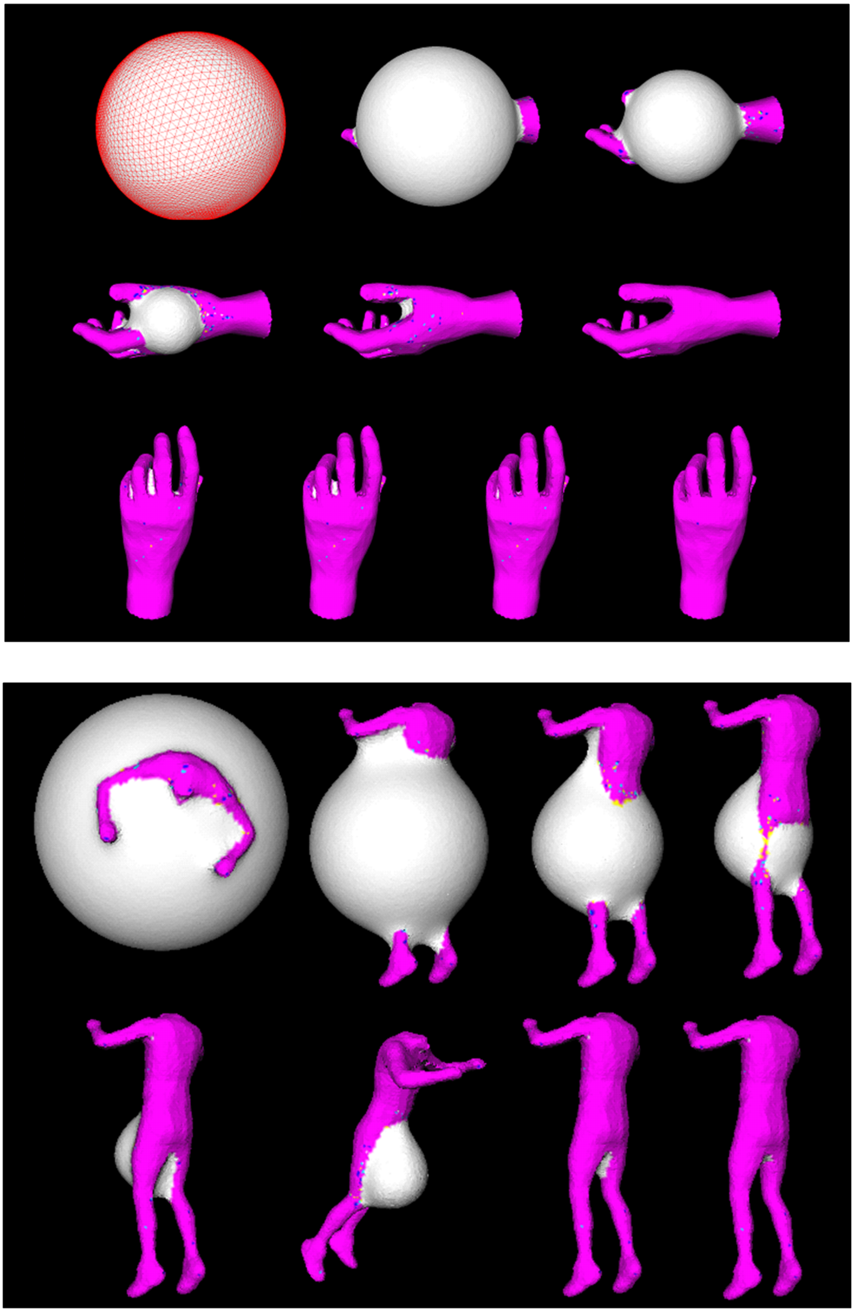

Fig. 3. The Hand model (above) and the Human model (below), from various views and at various iterations, as they deform from the sphere to the corresponding object shape with the proposed technique

5 Conclusion

We have presented a method for surface reconstruction from multi-view images of an object, that is based on surface deformation. The foremost prominent property of the method is that it produces topologically correct and smooth representations. Such representations are eligible for further deformation to serve various purposes, such as to capture hidden concavities of the surface by incorporating stereo texture information. Moreover since the deformation framework is based on Lagrangian approach, the connectivity information is not lost through iterations and thus the presented method can also be employed for building efficient time varying surface representations, that we plan to address as future work.

References

1. Montagnat, J., Delingette H., Ayache N.: A review of deformable surfaces: topology, geometry and deformation. Image Vision and Computing **19** (2001) 1023-1040
2. Kass M., Witkin A., Terzopoulos D.: Snakes: Active contour models. Int. J. Computer Vision **1** (1988) 321-331
3. Terzopoulos D., Witkin A., Kass M.: Constraints on deformable models: Recovering 3D shape and nonrigid motions. Artificial Intelligence **36** (1988) 91-123
4. Cohen, L. D.: On active contour models and balloons. CVGIP: Image Understanding **53** (1991) 211-218
5. Malladi, R., Sethian, J. A., Vemuri, B. C.: Shape modeling with front propagation: A level set approach. IEEE Trans. Pattern Analysis and Mach. Intelligence **17** (1995) 158-175
6. Magnor, M. A., Goldlcke, B.: Spacetime-coherent geometry reconstruction from multiple video streams. Int. Symp. 3DPVT (2004) 365-372
7. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. ACM SIGGRAPH (1996) 303-312
8. Rocchini, C., Cignoni, P., Montani, C., Pingi, P., Scopigno, R.: A low cost 3D scanner based on structured light. Proc. EUROGRAPHICS **20** (2001) 299-308
9. Mueller, K., Smolic, A., Merkle, P., Kautzner, M., Wiegand, T.: Coding of 3D meshes and video textures for 3D video objects. Proc. Picture Coding Symp. (2004)
10. Duan, Y., Yang, L., Qin, H., Samaras, D.: Shape reconstruction from 3D and 2D data using PDE-based deformable surfaces. Proc. ECCV (2004) 238-251
11. Esteban, C. H., Schmitt, F.: Silhouette and stereo fusion for 3D object modeling. Computer Vision and Image Understanding **96** (2004) 367-392
12. Kobbelt, L. P., Bareuther, T., Seidel, H.: Multiresolution shape deformations for meshes with dynamic vertex connectivity. Proc. EUROGRAPHICS **19** (2000)
13. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Mesh optimization. ACM SIGGRAPH (1993) 19-26
14. Anuar, N. and Guskov, I.: Extracting animated meshes with adaptive motion estimation. Proc. of the 9th Int. Fall Workshop on Vision, Modeling, and Visualization. (2004).