

# Line-Stepping for Shell Meshes

Kenny Erleben and Jon Sporning

Department of Computer Science, University of Copenhagen, Denmark  
{kenny,sporning}@diku.dk

**Abstract.** This paper presents a new method for creating a thick shell tetrahedral mesh from a triangular surface mesh. Our main goal is to create the thickest possible shell mesh with the lowest possible number of tetrahedrons.

Low count tetrahedral meshes is desirable for animating deformable objects where accuracy is less important and to produce shell maps and signed distance fields. In this work we propose to improve convergence rate of past work.

## 1 Introduction

Many graphical models of solid objects are given as surface meshes [1,2], since this is an economical representation for visualization, and easily obtainable by laser scanning of real objects or by hand-modeling using a 3 dimensional drawing tool. However, animating deformations of solid objects requires a notion of inner structure which is surprisingly difficult to obtain. Existing algorithms such as [3] are difficult to implement and do not use the natural, intrinsic representation of shape by symmetry sets [4,5].

Shell meshes are attractive since they give a volume representation of a surface mesh with a very low tetrahedral count, which is desirable for animation or similar purposes, where speed is preferred over accuracy of deformation. The shell mesh finds applications in animating solid objects, for shell maps [6], and for the calculation of signed distance field [7,8].

In this paper we will present an extension of [9,10], and our main goal is to create the thickest possible shell mesh with the lowest possible tetrahedral count. I.e. given a polygonal surface mesh, we create a tetrahedral volume mesh representing a thick version of the surface mesh, a shell mesh. The thickness shells are the most challenging to produce, and all thinner shells are a subset of these, hence we will in this article focus on producing the thickest possible shells. In past work vertices of the polygonal surface mesh are displaced inward, thereby creating a new version of the surface mesh. This operation is in the literature termed inward extrusion or just simply extrusion, although intrusion seems a better term. Following an inward extrusion the original surface mesh is used to generate the outside of the shell and the extruded surface mesh is used to generate the inside of the shell mesh. The two meshes is then used to create a triangle prism shell mesh. Finally, the triangle prism mesh are converted into a consistent tetrahedral mesh, also known as tetrahedral tessellation.

A linear randomized tessellation algorithm, the ripple tessellation, was developed in [9]. The ripple method suffers from several problems. It is not deterministic, but relies on picking random ripple directions to fix inconsistencies. Further, no proof has been given on existence of a consistent tetrahedral tessellation of the triangle prism shell. A safe conservative, upper extrusion length limit is used in [9] and later improved in [10]. Nevertheless, both algorithms are very slow due to bad and unpredictable convergence of the bisection search method. In present article we present a new extrusion approach that seek to solve the bad and unpredictable convergence of the bisection search method.

## 2 Adaptive Signed Distance Field Extrusion

The signed distance field for a closed surface mesh is a scalar field,  $\phi$ , whose magnitude is the distance to the closest point on the surface, and whose values are positive outside and negative inside the surface mesh. In the neighborhood of the surface mesh, the gradient of the signed distance field,  $\nabla\phi$ , has length 1 and a direction parallel to the surface normal,  $\mathbf{n}$ , but further away from the mesh, the distance field may experience singular points, exactly where two or more points on the surface are closest. At these points, the signed distance field is first order discontinuous. As an example consider the signed distance map of a circle, which looks like a cone, passing through the circle and whose apex is the center of the circle, since the center is equally close to every point on the circle. The points that are closest to two or more points on a shape are known as the symmetry set [4] and a subset is the skeleton or the Medial surface representation [5].

We perform an inward extrusion of a triangle on the surface, thus producing a prism. This is unproblematic as long as the extrusion is still in the neighborhood of the surface, however, when the extrusion extends beyond the singular points of the distance field, then prisms will overlap with the singular points, and therefore also overlap with an extrusion from another part of the object. Since we are only concerned with inward extrusions, we only need to consider the medial surface part of the symmetry sets, since the medial surface is the locus of the singular points of the signed distance field inside the object. Thus, given the medial surface, the maximum inward extrusion lengths avoiding overlapping prisms would be easily obtained by intersection of the triangle normals with the medial surface. Algorithms do exist to compute the medial surfaces of polygonal models [11], but these are often not applicable to general polygonal models made by artist. In practice one seeks approximations [12,13,14]. We propose to use the signed distance field directly, since it contains the medial surface implicitly. Thus, we avoid the computational burden of the approximation, and gain precision, since the signed distance field is numerically more accurate than an approximation based hereof. In the following we will describe our algorithm.

We attack the problem of detecting the singularities in the signed distance map. Hence, given a signed distance map,  $\phi$ , for a surface, its singularities may be found by a binary search from the surface inwardly along the surface normal, which is the same as in the signed direction of the gradient of the in to find the

medial surface point in the opposite direction of a distance field gradient. Hence, in a neighborhood of the a surface point  $p$  with corresponding normal  $\mathbf{n}$ , we seek the largest value of  $\varepsilon$  fulfilling the criteria

$$\varepsilon = -\phi(\mathbf{p} - \varepsilon\mathbf{n}), \quad \text{and} \quad \varepsilon \leq \varepsilon_{\text{user}}, \quad (1)$$

where we for practical purposes allow for a user-specified, maximum size,  $\varepsilon_{\text{user}}$ . Thus prior to overshooting the medial surface, we must have

$$\varepsilon = -\phi, \quad (2)$$

and immediately after overshooting we must have

$$\varepsilon \neq -\phi. \quad (3)$$

Hence, a root searching algorithm is easily devised, unfortunately the bisection approach works rather badly in practice, since discretization errors and interpolation can be quite large. This implies that

- 1) The gradient direction is not very accurate.
- 2) The distance value at any given position might be a little off.

Thus, given a surface point,  $\mathbf{p}$ , we cannot expect (1) to be very accurate, and an effective stopping criteria for the bisection method is not easily designed. In our tests we used a simple iteration limit of 1000, and a quite large threshold value in the comparison of (1). Figure 1(a) shows a 2D result. Notice that some surface points are extruded quite poorly, and this approach results in too poor results at too high computational cost. In some cases, where we step along symmetry lines, we can not even rely on the assumption in (1). Any root-search method will therefore fail.

An alternative approach is line-stepping, which is based on the idea of stepping along the extrusion line with a fixed increment of  $\Delta\varepsilon$ . In each step the extrusion length is updated by

$$\varepsilon_{i+1} = \varepsilon_i + \Delta\varepsilon, \quad (4)$$

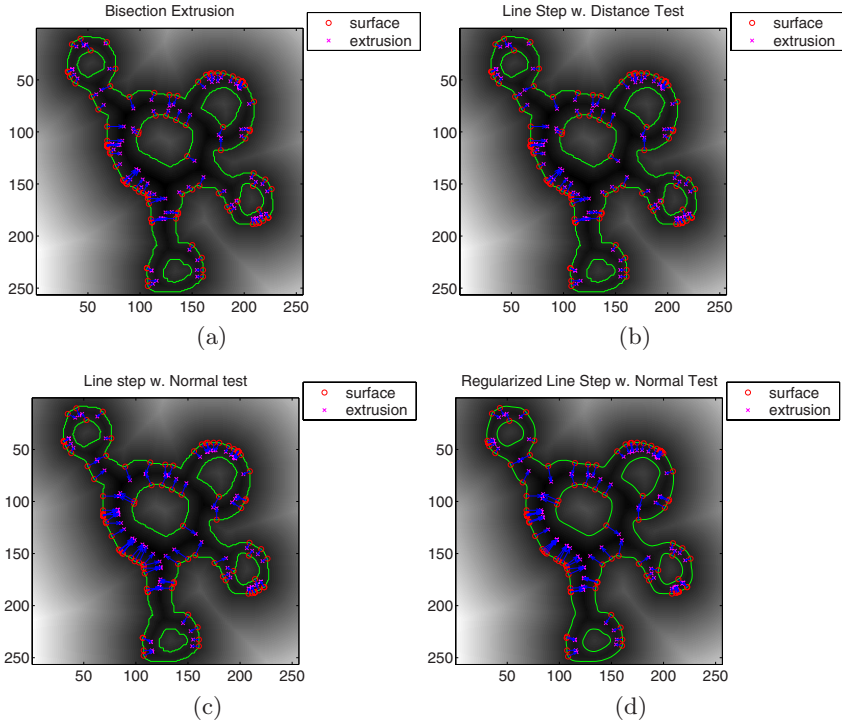
and the current extrusion point,  $\mathbf{q}(\varepsilon)$ , is found by

$$\mathbf{q}(\varepsilon) = \mathbf{p} - \varepsilon\nabla\phi(\mathbf{p}). \quad (5)$$

The stepping is performed as long as  $\nabla\phi(\mathbf{q})$  points in the same direction as  $\nabla\phi(\mathbf{p})$ . If the cell size of the regular sampled signed distance field be given by  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$  then the increment is chosen as

$$\Delta\varepsilon = \frac{\min(\Delta x, \Delta y, \Delta z)}{2}. \quad (6)$$

This ensures that each step along the extrusion line is not faster than the information changes in the signed distance field. This works due to spatial coherence of the values in the signed distance field, since the value at a neighboring grid



**Fig. 1.** 2D examples of inward extrusions on random points on the surface. (a) The bisection method gives poor extrusion lengths. (b) The line stepping using distance testing as stopping criteria results in some points extruded badly. (c) The line stepping using normal testing as stopping criteria, improves the extrusion lengths over (b). (d) The regularized line stepping using normal testing as stopping criteria has the diminished the crossing of extrusion lines.

node in the signed distance field differs by at most  $\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$ . The stopping criteria we use is to keep on increasing  $\varepsilon$ , while

$$\nabla\phi(\mathbf{p}) \cdot \nabla\phi(\mathbf{q}) > \rho, \tag{7}$$

where  $\rho > 0$  is a user specified threshold to control accuracy. We call this the normal-test. The intuition behind the normal-test is: if we pass the medial surface from one side of an object to the opposite side, then the gradients in the signed distance field will flip directions. That is, the sign of the normal-test will flip from positive to negative. Later we will give a more rigorous explanation of this stopping criteria. One should think that a better stopping criteria would be to only step along an extrusion line while

$$|\varepsilon - \phi(\mathbf{q})| < \rho. \tag{8}$$

Unfortunately in practice this distance test is far to sensitive to discretization and interpolation errors in the signed distance field and requires large values of  $\rho$ .

Leading to a questionable approximation of the medial surface. For now we will overlook the problem of walking along symmetry lines, where (8) does not hold. Figure 1(b) shows a test result using the distance test, while Figure 1(c) shows the same test using the normal test. Comparing Figure 1(c) with Figure 1(a) and Figure 1(b) we see that extrusions are much larger and more evenly distributed for the line-stepping approach using normal testing, and therefore we favor this method.

The line-stepping approach with the normal test works quite robustly, although some problems remain: Firstly, the gradient,  $\nabla\phi(\mathbf{p})$ , is not well defined at surface vertices. This is normally not noticeable, since signed distance fields are typically sampled on a regular grid, implying a regularization of the signed distance field gradient. Secondly, typical finite difference approximation schemes for the gradient operator are not accurate enough, which results in surface normals, where the direction of the extrusion lines might cause a swallow tail problem. To circumvent these problems, two solutions may be adopted:

- 1) Instead of using  $\nabla\phi(\mathbf{p})$ , we use the angle weighted pseudo-normals [15]. These can be computed directly from the input surface mesh at high accuracy.
- 2) We can regularize the signed distance field by a curvature flow (or Gaussian convolution etc.), this seems to straighten out the normal directions at small scale features, where the normal direction is poor due to sampling artifacts.

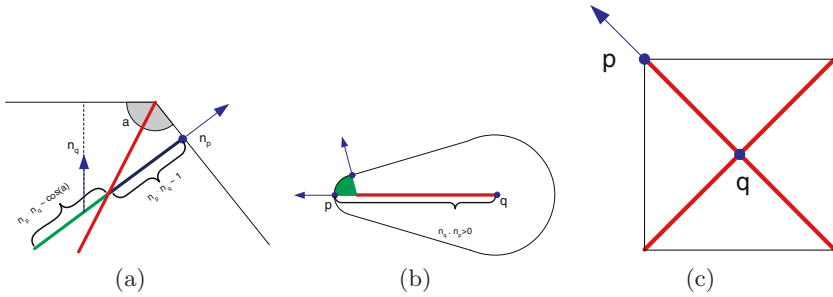
The second approach smooths errors, but it destroys the signed distance field property, this could be recovered by reinitialization [8]. However, in our experience it does not cause major changes in the overall direction of the gradient. Figure 1(d) shows the result using regularization. Comparing Figure 1(d) with Figure 1(c) it is seen that the extrusion lines tends to cross much less, when using regularization.

Smoothing the signed distance field by curvature flow works well in two dimensions, but is costly in three dimensions, where we prefer angle weighted surface normal solution, to be described below.

It is worth noting that left-hand-side of the normal-test in (7) is in fact the directional derivative at position  $\mathbf{q}$ . The sign of the directional derivative therefore tell us, how  $\phi$  changes, as we move in the opposite direction of the surface normal vector  $\mathbf{n}$ . In our specific case the following rules applies:

$$\nabla\phi(\mathbf{q}) \cdot \mathbf{n} \begin{cases} < 0 & \phi \text{ is increasing} \\ > 0 & \phi \text{ is decreasing} \\ = 0 & \phi \text{ is constant} \end{cases} \quad (9)$$

The actual value of the directional derivative tells us how fast  $\phi$  changes in the normal direction. For our inward extrusions, we want to extrude as long as  $\phi$  is decreasing. However, this is not quite enough: as illustrated in Figure 2(a), an extrusion line can cross over the symmetry set, if the surface angle  $a$  is less than the accepted angle difference between  $\nabla\phi(\mathbf{q})$  and  $\mathbf{n}_p$ , which is undesirable. In this case, the directional derivative,  $\nabla\phi(\mathbf{q}) \cdot \mathbf{n}$ , is 1 until the symmetry line is hit, but beyond the symmetry line, the distance value continues to decrease. To stop



**Fig. 2.** (a) An inward extrusion line may cross over the symmetry line. (b) A tentacle is created along the symmetry axis. (c) Not always do we wish to walk along a symmetry line.

at the symmetry line, we must detect the change in the value of the directional derivative. Another approach would be to require that the gradient of the signed distance field does not differ from the surface normal by more than some small user specified angle, i.e.

$$\nabla\phi(\mathbf{q}) \cdot \mathbf{n}_p > \rho \tag{10}$$

where  $\rho$  is the cosine of the accepted angle difference,  $\alpha$ .

$$\rho = \cos(\alpha) \tag{11}$$

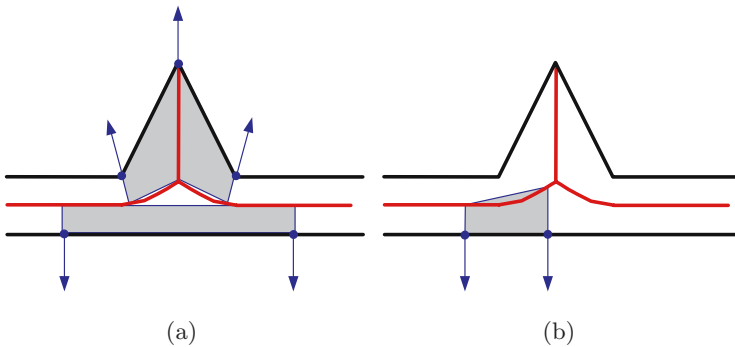
In our test examples we used  $\alpha = 0.4363$  radians, which means  $\rho \approx 0.9$ . We experience that this value greatly reduces the number of cross-overs. Depending on the resolution and accuracy of the signed distance field (10) can be extremely sensitive to numerical errors. In such cases setting  $\rho$  too tight would result in almost no extrusion.

In some rare cases a surface normal can be aligned with the symmetry set in such a way that the signed distance value along the extrusion line keeps on decreasing, while stepping along the symmetry set. That is, the directional derivative is still positive while stepping along inside the symmetry set. This is illustrated in Figure 2(b). The directional derivative,  $\nabla\phi(\mathbf{q}) \cdot \mathbf{n}$ , is 1 until the symmetry axis is hit. However, the distance value is decreasing while stepping along the symmetry axis. The distance value will not increase before we reach the point  $\mathbf{q}$  at the end of the symmetry line. This creates a tentacle for the green prism. The problem is very unlikely in practice, and we have not encountered it in our test-runs, most likely due to noise caused by approximation errors in  $\nabla\phi(\mathbf{q})$  and interpolation errors in  $\phi(\mathbf{q})$ . This observation yields another hint at how to minimize the chance of this problem to occur. An initial slight tangential perturbation of the surface mesh vertices, will destroy any special alignment with the symmetry set.

It should be noted that in some cases we actually do want to walk along a symmetry line. This is illustrated in Figure 2(c). In this case we want to

step along the symmetry line, until we hit the internal junction point. At the junction point we have a singularity of  $\nabla\phi(\mathbf{q})$ . However, along the symmetry line we have a constant value of the directional derivative. If we stepped beyond the junction point, we would see a sign flip of the directional derivative. One possible resolution to both the cross-over and the tentacle problems may be to extend the normal test with an upper extrusion length limit such as the adaptive thin-shell limit [10]. There are some disadvantages of doing this, which we will discuss later on. Another possibility may be to detect the change in the slope of  $\phi(\mathbf{q}(\varepsilon))$ . However, it is numerically very sensitive to make a good estimate of the slope, and we must ask our self the question of how big a numerical error can be accepted?

Other artifacts can occur as well, we attribute these to sampling artifacts, since they are completely dependent on the original placement of the input surface mesh vertices. The first sampling artifact, illustrated in Figure 3(a), is the creation of void regions due to too coarse a sampling. Thus, after extrusion the extruded prisms provide a poor fit to the medial axis leading to an empty void region inside the object. If we add more surface mesh vertices the extruded prisms

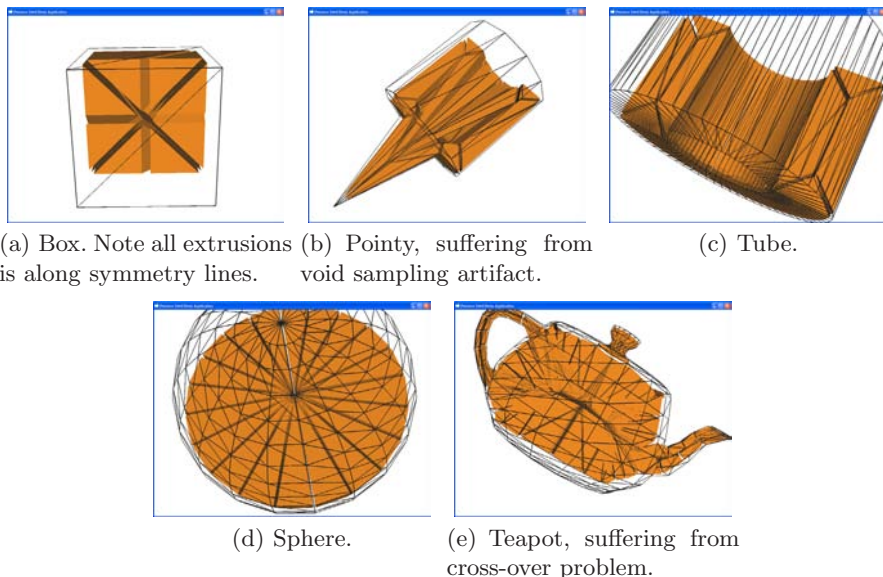


**Fig. 3.** (a) The coarse sampling of the blue surface mesh vertices is not enough to capture the higher order curve of the medial axis, shown in red. (b) The coarse sampling of the blue surface mesh vertices combined with unfortunate placement causes the extruded prism to pass over the medial axis creating a potential overlap with prisms extruded from the opposite side.

will come closer and closer to the medial axis. The coarse sampling combined with unfortunate placement of sampling points may even cause overlapping regions as illustrated in Figure 3(b). Both the creation of void and overlapping regions are due to sampling artifacts, thus one way to improve upon these problems is to re-sample the input surface mesh. Either by detecting good places to insert sample points or simple brute subdivision of mesh faces. Neither of these two solutions have been used for our presented test results using signed distance fields.

### 3 Conclusion on Signed Distance Field Extrusions

In [7] the tetrahedral GPU scan conversion method was presented, and we used this method for generating the signed distance fields in our computations. The narrow-band size was determined by taking half of the diagonal length of a tightly enclosing axis aligned bounding box around each surface mesh.



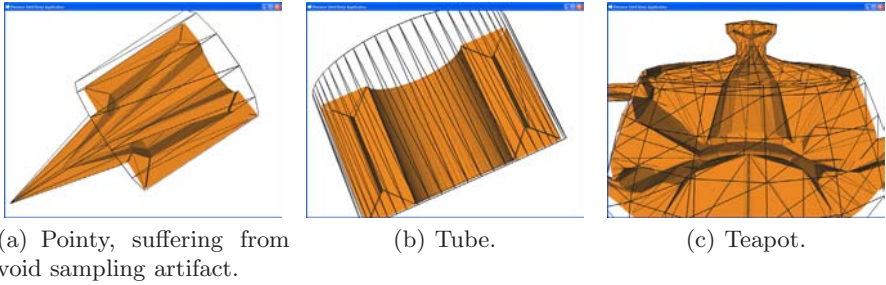
**Fig. 4.** Results of the adaptive distance shell creation method. Using signed distance field of resolution  $256^3$ , and a normal-test with  $\rho = 0.9$ . Observe the overlapping shell mesh of the teapot.

In Figure 4 we have shown planar cross-intersections of a few tetrahedral shell meshes generated using our adaptive distance shell creation method. The more complex teapot shape does suffer from the cross-over problem explained above. However, aside from this it is clear that the presented extrusion method is capable of filling the internal void inside the surface meshes (shown as black wire-frame).

The signed distance field resolution is very important for the quality of the extrusion in thin regions of objects. We suggest the following rule of thumb: the higher the resolution the better the quality. This phenomena is rather trivial because the fixed extrusion length increment  $\Delta\epsilon$  is determined by the resolution of the signed distance field. The higher the resolution the smaller the increment.

To illustrate how the cross-over problem could be handled by combining with an upper extrusion length bound, we have combined the adaptive distance shell method with the extrusion limit from the previous adaptive shell method [10].





**Fig. 5.** Results of combining the adaptive distance shell method with upper bounds from the adaptive thin shell. Observe that overlaps have disappeared, but the artifacts of the extrusion length computation method is present.

Our results is shown in Figure 5. It is clear that the cross-over problem have been resolved. On the other hand the quality is somewhat inferior. This is because the upper limit used is not the maximum upper limit. It is though a safe limit, and it is very dependent on the local shape of the surface mesh faces.

The time-complexity of the line-stepping method is govern by two parameters, the signed distance field resolution,  $N$  and the number of vertices,  $V$ . Each extrusion line is treated independently. Thus, the algorithm scales linear in the number of vertices. The number of steps that can be taken a long an extrusion line, is bounded above by the maximum number of grid nodes encountered on the diagonal of the regular grid. Thus,

$$O(V\sqrt{3\frac{1}{2}N^2}) \approx O(VN). \quad (12)$$

This is extremely fast, since the operations done during each step have very low constants dominated by the computation of the gradient of a scalar field sampled on a regular grid. The major performance cost is the computation of the signed distance fields. We refer to the paper [7] for performance details hereof.

The proposed method for computing extrusion points have resulted in many discoveries: line-stepping for finding the “center-position” is the best solution over any root search method. In terms of numerical robustness normal testing is a far better stopping criteria than distance testing. Finally, thresholding on normal testing is a necessary evil due to fix-precision floating point arithmetic. This leads to possible overlapping interior regions. In practice this calls for parameter tuning. The implication is that it is not always possible to find an acceptable bound on the normal test which restrict internal overlaps and allow for the most aggressive extrusion lengths.

In conclusion, line-stepping has been identified as an algorithm that works exceptionally well on many surface meshes, but there are some cases, in which improvements are needed. The main issue is how to deal with line stepping along symmetry lines, which will be our future point of research.

## References

1. Mantyla, M.: Introduction to Solid Modeling. W. H. Freeman & Co, New York, USA (1988)
2. Botsch, M., Steinberg, S., Bischoff, S., Kobbelt, L.: Openmesh—a generic and efficient polygon mesh data structure. In: Proc. Open SG Symposium (2002)
3. Molino, N., Bridson, R., Teran, J., Fedkiw, R.: Adaptive physics based tetrahedral mesh generation using level sets. (in review) (2004)
4. Diatta, A., Giblin, P.: Pre-symmetry sets of 3D shapes. In: Olsen, O.F., Florack, L.M.J., Kuijper, A. (eds.) DSSCV 2005. LNCS, vol. 3753, pp. 36–49. Springer, Heidelberg (2005)
5. Pizer, S.M., Fletcher, P.T., Joshi, S., Thall, A., Chen, J.Z., Fridman, Y., Fritsch, D.S., Gash, A.G., Glotzer, J.M., Jiroutek, M.R., Lu, C., Muller, K.E., Tracton, G., Yushkevich, P., Chaney, E.L.: Deformable m-reps for 3d medical image segmentation. *International Journal of Computer Vision* 55(2/3), 85–106 (2003)
6. Porumbescu, S.D., Budge, B., Feng, L., Joy, K.I.: Shell maps. *ACM Trans. Graph.* 24(3), 626–633 (2005)
7. Erleben, K., Dohlmann, H.: Scan conversion of signed distance fields. In: Olsen, S.I., (ed.) Proceedings of DSAGM. pp. 81–91 (2006)
8. Sethian, J.A.: Level Set Methods and Fast Marching Methods. In: *Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge Monograph on Applied and Computational Mathematics, Cambridge University Press, Cambridge (1999)
9. Erleben, K., Dohlmann, H.: The thin shell tetrahedral mesh. In: Olsen, S.I., (ed.) Proceedings of DSAGM, pp. 94–102 (2004)
10. Erleben, K., Dohlmann, H., Sporring, J.: The adaptive thin shell tetrahedral mesh. *Journal of WSCG*, pp. 17–24 (2005)
11. Culver, T., Keyser, J., Manocha, D.: Accurate computation of the medial axis of a polyhedron. Technical Report TR98-034, University of North Carolina, Chapel Hill (1998)
12. Bradshaw, G., O’Sullivan, C.: Adaptive medial-axis approximation for sphere-tree construction. *ACM Transactions on Graphics* vol. 23(1) (2004)
13. Bouix, S., Siddiqi, K.: Divergence-based medial surfaces. In: *ECCV ’00: Proceedings of the 6th European Conference on Computer Vision-Part I*, London, UK, pp. 603–618. Springer, Heidelberg (2000)
14. Dimitrov, P., Damon, J.N., Siddiqi, K.: Flux invariants for shape. In: *CVPR (1)*, pp. 835–841 (2003)
15. Baerentzen, J.A.: Signed distance computation using the angle weighted pseudonormal. *IEEE Transactions on Visualization and Computer Graphics*, Member-Henrik Aanaes 11(3), 243–253 (2005)