

SpringLS: A Deformable Model Representation to provide Interoperability between Meshes and Level Sets

Blake C. Lucas^{1,2}, Michael Kazhdan², Russell H. Taylor²

¹ Johns Hopkins Applied Physics Laboratory, Laurel, MD, USA

² Johns Hopkins University, Baltimore, MD, USA
blake@cs.jhu.edu, misha@cs.jhu.edu, rht@jhu.edu

Abstract. A new type of deformable model is presented that merges meshes and level sets into one representation to provide interoperability between methods designed for either. The key idea is to use a constellation of triangular surface elements (springls) to define a level set. A Spring Level Set (SpringLS) can be interpreted as a mesh or level set and used in place of them in many instances. There is no loss of shape information in the transformation from triangle mesh or level set into SpringLS. As examples, we present results for joint segmentation/spherical mapping of a human brain cortex and atlas/non-atlas segmentation of a pelvis.

Keywords: mesh, level set, deformable model, segmentation, active contour.

1 Introduction

Deformable models are geometric representations of objects that deform (change shape) due to forces applied at their boundary. Deformable models are applicable to a broad range of problems in image analysis and computer vision including: reconstruction, non-rigid registration, image segmentation, atlas, and motion tracking. There are two major model representations: meshes [1] and level sets [2, 3]. Deformable model methods usually favor a particular representation (i.e. meshes for 2D/3D registration and level sets for image segmentation). However, large systems that use a mixture of methods are forced to transform one representation into another in order to use the preferred representation for each method. This strategy leads to loss of information and less flexibility in design of the system. For examples of systems that use a mixture of representations, see Tosun et al. [4] and Wand et al. [5].

The Spring Level Set (SpringLS) representation merges meshes and level sets into a single geometric representation that preserves the strengths of both. SpringLS can be interpreted as a mesh or level set, and no shape information is lost in the transformation from triangle mesh or level set into SpringLS. SpringLS is intended for methods that employ a mixture of mesh and level set techniques, but is applicable to almost all deformable model methods. As examples, we apply SpringLS to joint segmentation/spherical mapping of a human brain cortex and atlas/non-atlas segmentation of a pelvis.

2 Background

Meshes. Meshes were the earliest representation for deformable models [1]. In this framework, the model is deformed by perturbing mesh vertices. The model's boundary is explicitly tracked by remembering the trajectory of each vertex. As a section of the mesh expands or contracts, sharp creases, edges, self-intersections, or triangle flips can develop. Sharp edges and other mesh artifacts violate a common material property that most objects represented by deformable models are smooth and plastic. To reduce artifacts, the mesh must be regularized and re-sampled (remeshed) periodically. Because mesh triangles must be connected to form a watertight model, remeshing is non-trivial [6] and interferes with vertex tracking. Remeshing becomes more of a nuisance if the mesh is not allowed to self-intersect or is allowed to change topology. For these reasons, triangle meshes have become unpopular for applications where the model 1) undergoes large non-rigid deformations that require remeshing; 2) the model is expected to change topology; 3) the model is likely to self-intersect.

Level Sets. The level set method [2, 3] represents a deformable model as a 3D image where the image intensity at each voxel is a distance measurement to the surface of the object. Distance measurements are signed: negative values are inside and positive values are outside the object. A triangle mesh can be extracted by computing the iso-surface corresponding to the zero level set of the image. The level set representation has several advantages over deformable meshes: 1) no need for self-intersection removal; 2) topology change is easy; 3) no need to remesh. These properties have made level sets the popular choice for image segmentation and fluid-like non-rigid deformation.

Level sets are difficult to use for registration and tracking tasks because there's no innate ability to track vertices as in the mesh deformation framework. The surface only exists when an iso-surface is extracted from the level set. Furthermore, the level set is stored as an image that is re-sampled at each time step. Re-sampling an image acts as a low-pass filter that results in feature loss as a function of the number of time steps, even if the motion is rigid (i.e. global registration) or divergence free (i.e. incompressible fluid flow).

Hybrid Representations. Attempts have been made to unify deformable model representations with varying success [7-13]. Of these, the Marker Level Set (MLS) [11] is closest to this work. The MLS method maintains a set of particles located on the level set's zero iso-level. Since particles lie exactly on the zero iso-level, they can be used for tracking the model's boundary. After each level set and particle advection step, the level set is corrected so that particles continue to lie on the level set's zero iso-level. Particles are added to cover the zero iso-level and deleted to prevent over-sampling. The MLS method associates a color with each particle and interpolates the color for new particles based on their neighbors.

The philosophical difference between SpringLS and MLS is that spring surface elements define the model's level set, whereas MLS use particles to correct errors in the level set. MLS requires the deformation method to have an equivalent level set and parametric interpretation in order to deform both representations. Movement of

the auxiliary level set with SpringLS is passive and independent of the deformation method. SpringLS can be applied more broadly to deformation methods for which there is only a parametric interpretation (i.e. Point Distribution Models (PDM) [14]), enabling true interoperability between methods designed for level sets and meshes.

3 Method

A springl is a triangular surface element consisting of a particle and 3 springs connecting the particle to each of the triangle’s vertices (see Fig. 1a). Each springl describes a basis function which defines the unsigned clamped distance $d_n(\mathbf{X})$ to the triangle. The combination of these distance functions form the unsigned level set $\omega(\mathbf{X})$. Support of a springl is represented by a capsule, whose boundary is the iso-surface corresponding to $d_{max} = 0.5$ voxels. Springs, vertices, and particles are coplanar, and the angles between springs are fixed. An auxiliary signed level set $\varphi(\mathbf{X})$ is maintained and evolves with the particles. The level set augments the particle representation in several ways: 1) the signed distance function indicates regions that are inside or outside the model; 2) the signed level set indicates when new springls need to be added or removed; 3) the iso-surface extracted from the level set is a watertight triangle mesh representation of the model. The deformable model is stored as three data structures: a triangle mesh representing surface elements, a point cloud of particles, and a 3D image representing the signed level set.

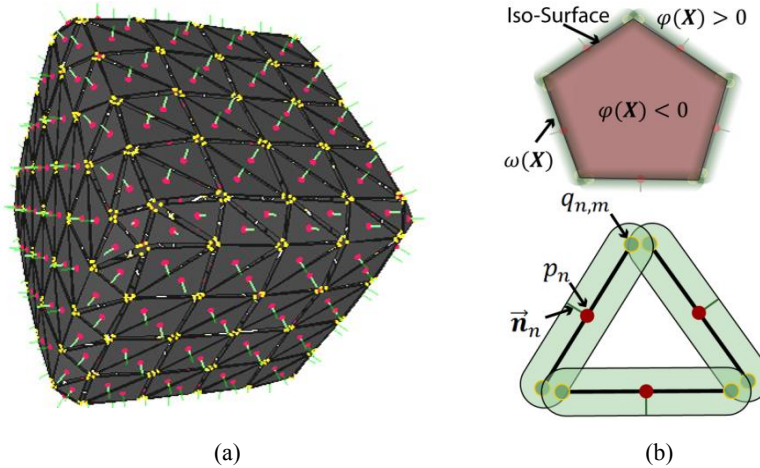


Fig. 1. (a) Model represented by springls showing particles (red), vertices (yellow), normals (green), and surface elements (black). (b) Signed level set, springls, and their capsules (green).

A springl (\mathcal{S}_n) is represented as five points describing the particle location p_n , correspondence point location a_n , and triangle vertices $q_{n,m}$ (see Fig. 1b). The model is deformed by incrementally advecting springls with first-order lagrangian methods (**Advect**). The deformation may be driven by pressure $\rho(\cdot)$ in the normal direction

\vec{n}_n of a springl, external velocity field $\vec{\sigma}(\cdot)$, and/or local affine transformation $\mathbf{A}(\cdot)$ derived from an atlas.

$$\frac{\partial p_n}{\partial t}(t) = \lambda_\rho \vec{n}_n \rho(p_n(t)) + \lambda_\sigma \vec{\sigma}(p_n(t)) + \lambda_{atlas} \mathbf{A}(\mathbf{S}_n) p_n(t), \text{ and} \quad (1)$$

$$\frac{\partial q_{n,m}}{\partial t}(t) = \lambda_\rho \vec{n}_n \rho(p_n(t)) + \lambda_\sigma \vec{\sigma}(p_n(t)) + \lambda_{atlas} \mathbf{A}(\mathbf{S}_n) q_{n,m}(t). \quad (2)$$

After each advection step, particles are fixed and the shape/orientation of each springl is adjusted through a relaxation process. The purpose of relaxation is to orient surface elements so their normals point outward. Motion is dictated by forces that attract neighboring vertices to the edges of nearby triangles. The force $(\vec{v}_{n,m})$ acting on a vertex due to its k closest points on nearby triangle edges $q_{n,m,k} \in \mathbb{R}^3$ is as follows:

$$\vec{v}_{n,m} = \lambda_v \tanh(\lambda_c |\vec{c}_{n,m}|) \frac{\vec{c}_{n,m}}{|\vec{c}_{n,m}|}, \text{ where} \quad (3)$$

$$\vec{c}_{n,m} = \sum_k \text{atanh}(w_{n,m,k}) \frac{q_{n,m,k} - q_{n,m}}{|q_{n,m,k} - q_{n,m}|^r}, \text{ and} \quad (4)$$

$$w_{n,m,k} = R^{-1}(|q_{n,m,k} - q_{n,m}| - 2r). \quad (5)$$

The resultant force due to the spring attached to the vertex is

$$f_\kappa = \kappa(l - |q_{n,m} - p_n|), \quad (6)$$

where $\lambda_v = 0.05$ is the max force, $\lambda_c = 4$ is the kernel smoothness, $l = 0.1$ voxels is the spring rest length, $\kappa = 0.08$ is the spring constant, $r = 0.05$ voxels is the vertex radius, and $R = 0.6$ voxels is the nearest-neighbor range. The rotational moment due to forces applied on the vertex is

$$\vec{m}_{n,m} = \vec{v}_{n,m} \times \vec{s}_{n,m}, \text{ where} \quad (7)$$

$$\vec{s}_{n,m} = (q_{n,m} - p_n) / |q_{n,m} - p_n|. \quad (8)$$

After summing moments, the resultant moment \vec{m}_n indicates the amount and axis of rotation, described by the 3x3 matrix \mathbf{M}_n . The \tanh / atanh weighting functions in eq. (3) and eq. (4) dampen rotational motion that can lead to instability in a surface element's orientation. The final update equation is

$$q_{n,m}^{t+1} = p_n + \mathbf{M}_n \left(q_{n,m}^t - p_n + \vec{s}_{n,m} (\vec{v}_{n,m} \cdot \vec{s}_{n,m} + f_\kappa) \right). \quad (9)$$

The relaxation process (**Relax**) in eq. (9) is repeated for 20 iterations (see *Algorithm 1*). Parameter choices express a tradeoff between minimizing the number of springls needed to cover the zero iso-level while minimizing gap formation. Parameters were selected based on a small number of examples apart from the experiments presented in this paper and have not been changed in these or any subsequent experiments.

Relaxation helps insure that the union of all springl capsules covers the zero iso-level of the signed level set. If springls are unable to cover the zero iso-level through relaxation, gaps must be filled by adding more springls in a subsequent step (**FillGaps**) to prevent the model from tearing. When a zero-crossing of the level set is exposed, a springl is added to cover the zero-crossing. Zero-crossings are computed from the centroids of triangles generated from the signed level set's iso-surface. These triangles are reused to fill the hole with a springl in the same shape and position.

The level set is updated at each time step to track the particles (**Evolve**). This is done by constructing an unsigned level set $\omega(\mathbf{X})$ (eq. (10)) that is the clamped minimum distance to all springls. The signed level set $\varphi(\mathbf{X})$ is evolved to minimize

the energy function in eq. (11). The free parameter λ controls the model's smoothness. A springl is destroyed if after evolving the signed level set, a springl's particle is more than $(1 + \varepsilon)d_{max}$ from the zero iso-level (**Contract**). We choose $\varepsilon = 0.25$ in all cases.

$$\omega(\mathbf{X}) = \min\{d_{max}, d_1(\mathbf{X}) \dots d_N(\mathbf{X})\}, \text{ and} \quad (10)$$

$$E = \int (\omega(\mathbf{X}) + \lambda|\nabla\varphi(\mathbf{X})|)\delta(\varphi(\mathbf{X}))d\mathbf{X}. \quad (11)$$

For large parametric deformations where the CFL number exceeds 1, it is usually faster to convert the unsigned level set to a signed level set [15, 16] than to evolve the signed level set with active contour methods. We perform the conversion by growing the background region and then negating the unsigned level set in the foreground region. This can be done robustly with a coarse-to-fine strategy [16] to prevent the background region from leaking through gaps between springls.

Springls are re-sampled every $M = 5$ iterations to regularize the sampling distribution and triangle quality (**Resample**). Triangles are split along their longest edge if the length of that edge exceeds a threshold (1.5 voxels). If a triangle's angles fall outside a tolerable range $[20^\circ, 160^\circ]$, then the springl is removed. Removing poor quality triangles reduces unstable rotation of springls in the relaxation phase.

Springls maintain a mapping from each particle to the centroid of a triangle on the original model. The initial mapping is an identity mapping ($a_n = p_n$). When a springl is split, the mapping is duplicated and when a springl is added, the mapping is chosen to be the average point mappings for neighboring springls. This method produces mappings that lie slightly off the original surface. To prevent correspondence points from drifting from the original surface, correspondence points are moved along the gradient of the distance (φ_{ref}) to the original surface until convergence (eq. (12)).

$$a_n(t + 1) = a_n(t) - \lambda\varphi_{ref}(a_n(t))\nabla\varphi_{ref}(a_n(t)). \quad (12)$$

The model deformation process (**Deform**) is outlined in *Algorithm 2*.

Algorithm 1. Relax

```

foreach springl n do
  foreach vertex m do
    foreach neighbor k of vertex m do
      Compute  $w_{n,m,k}$ 
      Accumulate  $\vec{c}_{n,m}$ 
      Compute  $\vec{s}_{n,m}$ ,  $\vec{v}_{n,m}$ , and  $\vec{m}_{n,m}$ 
      Accumulate  $\vec{m}_n$ 
    foreach vertex m do
      Compute  $q_{n,m}^{t+1}$ 

```

Algorithm 2. Deform

```

for k = 1:K do
  Advect
  Relax
  if k mod M == 0 then
    Contract
    Resample
    Relax
    Evolve
    FillGaps
  else Evolve

```

4 Results

SpringLS was applied to active contour image segmentation [17] of objects driven under pressure forces from image intensities. It was implemented as a mixture of Java and OpenCL for the CPU on a PC with Dual 2.53GHz Intel Xeon and 12GB of RAM. Parameter settings and grid size ($256 \times 256 \times 256$) were fixed for all

experiments. Reported segmentation errors are measured as the average minimum distance from mesh vertices on the segmented mesh to target iso-surface. Experiments not initialized with an atlas were repeated with 4 different initializations (Fig. 2a).

Fig. 2b,c shows simultaneous segmentation of a T1 MRI image and spherical mapping of a human brain cortex, which are two important tasks in cortical surface analysis [18]. The MRI image was pre-processed with TOADS to produce WM/GM soft-membership images [19]. Spherical mapping was accomplished by initializing the segmentation with a sphere and then explicitly tracking the surface with springls to find the WM/GM surface in the WM membership image. The model was then evolved outward to find the Pial surface in the WM+GM membership image. The resulting reconstruction has a mapping from each point on either surface to a location on the sphere. Table 1 reports runtime statistics and compares SpringLS to an equivalent level set implementation in terms of surface-to-surface distance measured from the SpringLS iso-surface and DICE coefficient between level set segmentations.

Fig. 3a shows segmentation of a pelvis from a CT image when initialized with a cube and highlights SpringLS's ability to change topology. Initializing the segmentation process with a cube or other object shown in Fig. 2a causes over segmentation of the pelvis to include the femurs and spine as well. To mitigate this problem, we incorporate an atlas based approach. A PDM statistical atlas of the pelvis was constructed with the method from Seshamani et al. [20]. Analogous statistical atlas methods have been developed for level sets [21], but these representations are not equivalent.

The segmentation result in Fig. 3a can be improved by combining level set techniques with a parametric atlas. The atlas was registered (rigid + global scale) to the CT image. The first 10 mode weights were optimized in increasing order to reduce the average distance from the atlas to target iso-surface in CT. Because the initial registration was fairly good (1.56 ± 1.36 mm), optimizing the mode weights modestly improved the segmentation result to 1.40 ± 1.32 mm. The registered mesh was then treated as a constellation of springls and advected towards the target iso-level with an external velocity field produced by Gradient Vector Flow (GVF) [22] and pressure forces, reducing the error to 0.95 ± 1.02 mm. This atlas based method produces a better pelvis segmentation (Fig. 3c) than the non-atlas approach and provides a mapping from each springl back to the atlas, enabling transfer of region labels on the atlas to the segmented pelvis. This technique is also significantly faster than the non-atlas based approach (see Table 1).

5 Conclusion

Spring Level Sets (SpringLS) merge meshes and level sets into a single representation to provide interoperability between methods designed for either. The key idea is to use triangular surface elements to define a level set. Insisting the surface elements be triangle shaped insures no shape information is lost in the transformation from triangle mesh into SpringLS. One may choose not to relax or resample a subset of springls to preserve sharp features or tracking information. Because SpringLS uses disconnected surface elements, the object can change topology, track points, and undergo parametric deformations. The auxiliary level set provides a watertight

representation of the model's boundary that cannot self-intersect, and simple rules have been described for adding and destroying surface elements based on the level set representation. We have demonstrated that image segmentation with SpringLS produces results very similar to an equivalent level set implementation, and a registered PDM atlas can be converted into a SpringLS and deformed to produce a better segmentation than without an atlas. SpringLS is open source and distributed as part of the Java Image Science Toolkit (<http://www.nitrc.org/projects/jist>) to encourage the development of new image analysis systems that are true mixtures of mesh and level set methods.

Table 1. Results comparing SpringLS to an equivalent level set method. Algorithm terminated when the DICE coefficient between successive resampling cycles exceeded a threshold.

| Experiment | Surface Distance | DICE | Iterations | Springls | Triangles | Time |
|----------------|------------------|--------|------------|-----------|-----------|-----------|
| Pelvis | 0.14±0.20 mm | 0.9993 | 290-390 | 50K-55K | 200K-217K | 15-19 min |
| Pelvis w atlas | 0.25±0.32 mm | 0.9994 | 110 | 39K | 149K | 4 min |
| WM/GM | 0.18±0.21 mm | 0.9985 | 280-370 | 116K-118K | 366K-372K | 19-30 min |
| Pial | 0.16±0.16 mm | 0.9989 | 60 | 112K-113K | 306K | 5-6 min |

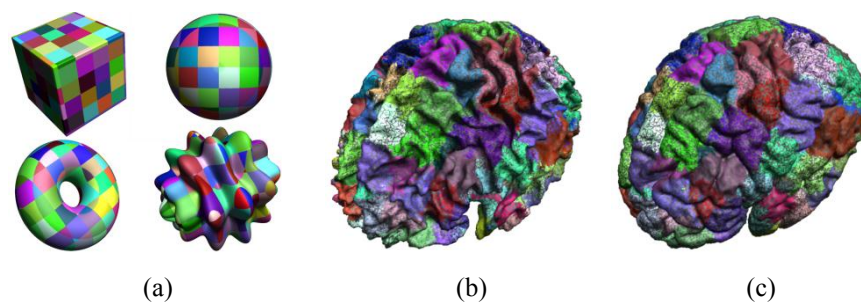


Fig. 2. Ray-cast renderings of SpringLS showing springsls projected onto iso-surface. (a) Initial shapes. (b) WM/GM surface and (c) Pial surface segmentations when initialized with a sphere.

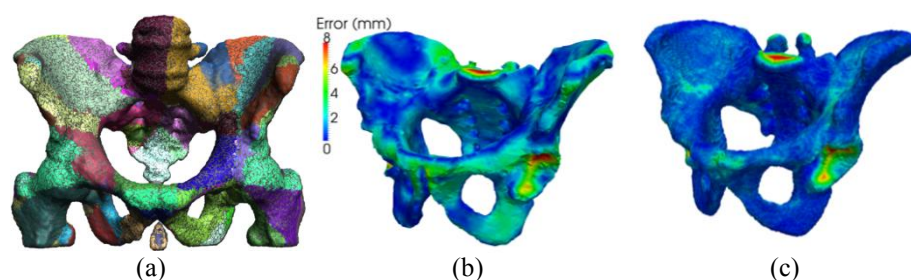


Fig. 3. (a) Pelvis segmentation when initialized with a cube. Atlas based segmentation showing (b) registered PDM and (c) final segmentation.

Acknowledgments. This research was supported by a graduate student fellowship from the Johns Hopkins Applied Physics Laboratory and internal funds from Johns Hopkins University.

References

1. D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," *Computer Graphics*, vol. 21, pp. 205-214, 1987.
2. J. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*: Cambridge Univ Pr, 1999.
3. S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*: Springer Verlag, 2003.
4. D. Tosun and J. L. Prince, "A geometry-driven optical flow warping for spatial normalization of cortical surfaces," *Medical Imaging, IEEE Trans*, vol. 27, pp. 1739-1753, 2008.
5. M. Wand, P. Jenke, Q. Huang, M. Bokeloh, L. Guibas, and A. Schilling, "Reconstruction of deforming geometry from time-varying point clouds," 2007, pp. 49-58.
6. P. Alliez, G. Ucelli, C. Gotsman, and M. Attene, "Recent advances in remeshing of surfaces," *Shape Analysis and Structuring*, pp. 53-82, 2008.
7. D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, "A Hybrid Particle Level Set Method for Improved Interface Capturing," *Computational Physics*, vol. 183, pp. 83-116, 2002.
8. M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa, "Point based animation of elastic, plastic and melting objects," 2004, pp. 141-151.
9. A. Bargteil, T. Goktekin, J. O'Brien, and J. Strain, "A semi-Lagrangian contouring method for fluid simulation," *ACM Trans on Graphics*, vol. 25, p. 38, 2006.
10. M. Müller, "Fast and robust tracking of fluid surfaces," in *SIGGRAPH*, 2009, pp. 237-245.
11. V. Mihalef, M. Sussman, and D. Metaxas, "Textured liquids based on the marker level set," *Computer Graphics Forum*, vol. 26, pp. 457-466, 2007.
12. J. Pons, G. Hermosillo, R. Keriven, and O. Faugeras, "Maintaining the point correspondence in the level set framework," *Computational Physics*, vol. 220, pp. 339-354, 2006.
13. M. Pauly, R. Keiser, L. P. Kobbelt, and M. Gross, "Shape modeling with point-sampled geometry," *ACM Trans on Graphics (TOG)*, vol. 22, pp. 641-650, 2003.
14. T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, pp. 38-59, 1995.
15. P. Mullen, F. De Goes, M. Desbrun, D. Cohen Steiner, and P. Alliez, "Signing the Unsigned: Robust Surface Reconstruction from Raw Pointsets," *Computer Graphics Forum*, vol. 29, pp. 1733-1741, 2010.
16. A. Sharf, T. Lewiner, A. Shamir, L. Kobbelt, and D. Cohen-Or, "Competing fronts for coarse-to-fine surface reconstruction," *Computer Graphics Forum*, vol. 25, pp. 389-398, 2006.
17. V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International journal of computer vision*, vol. 22, pp. 61-79, 1997.
18. M. K. Chung, S. M. Robbins, K. M. Dalton, R. J. Davidson, A. L. Alexander, and A. C. Evans, "Cortical thickness analysis in autism with heat kernel smoothing," *Neuroimage*, vol. 25, pp. 1256-1265, 2005.
19. P. L. Bazin and D. L. Pham, "Topology-preserving tissue classification of magnetic resonance brain images," *Medical Imaging, IEEE Trans*, vol. 26, pp. 487-496, 2007.
20. S. Seshamani, G. Chintalapani, and R. Taylor, "Alternative Statistical Models for Bone Modelling," in *SPIE Medical Imaging*, 2011.
21. A. Tsai, A. Yezzi Jr, W. Wells, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky, "A shape-based approach to the segmentation of medical imagery using level sets," *Medical Imaging, IEEE Trans*, vol. 22, pp. 137-154, 2003.
22. C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *Image Processing, IEEE Trans*, vol. 7, pp. 359-369, 2002.