Hedgehog Shape Priors for Multi-object Segmentation

Hossam Isack† Olga Veksler† Milan Sonka‡ Yuri Boykov†

habdelka@csd.uwo.ca olga@csd.uwo.ca milan-sonka@uiowa.edu yuri@csd.uwo.ca

†Computer Science, University of Western Ontario, Canada
‡Electrical and Computer Engineering, University of Iowa, USA

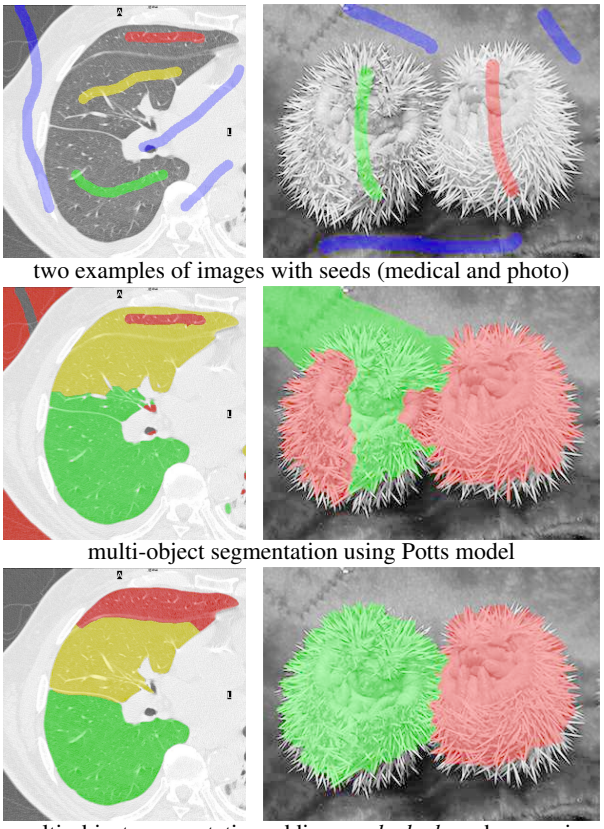
Abstract

Star-convexity prior is popular for interactive single object segmentation due to its simplicity and amenability to binary graph cut optimization. We propose a more general multi-object segmentation approach. Moreover, each object can be constrained by a more descriptive shape prior, "hedgehog". Each hedgehog shape has its surface normals locally constrained by an arbitrary given vector field, e.g. gradient of the user-scribble distance transform. In contrast to star-convexity, the tightness of our normal constraint can be changed giving better control over allowed shapes. For example, looser constraints, i.e. wider cones of allowed normals, give more relaxed hedgehog shapes. On the other hand, the tightest constraint enforces skeleton consistency with the scribbles. In general, hedgehog shapes are more descriptive than a star, which is only a special case corresponding to a radial vector field and weakest tightness. Our approach has significantly more applications than standard single star-convex segmentation, e.g. in medical data we can separate multiple non-star organs with similar appearances and weak edges. Optimization is done by our modified α-expansion moves shown to be submodular for multi-hedgehog shapes.

1. Introduction

Distinct intensity appearances and smooth contrast-aligned boundaries are standard segmentation cues. However, in most real applications of image segmentation there are multiple objects of interest with similar or overlapping color appearances. Intensity edges also could be cluttered or weak. These common practical problems require additional regularization, see the second row in Fig.1.

Our paper proposes a simple and sufficiently general shape regularization constraint that could be easily integrated into standard MRF methods for segmentation. Shape



two examples of images with seeds (medical and photo)

multi-object segmentation using Potts model

multi-object segmentation adding our *hedgehog* shapes prior

Figure 1. *Hedgehog shapes* prior for multi-object segmentation.

priors have been successfully used in binary graph cut segmentation [15, 23, 11]. While our "hedgehog" shape prior extends popular *star-convexity* constraint [23] in ways different from previous related methods [15, 11] and it has its own merits, our main contribution is a *multi-hedgehog* prior in the context of multi-object segmentation problems. We observe that similarity between object appearances and edge clutter are particularly problematic in larger multi-label segmentation problems, *e.g.* in medical imaging. Pre-

vious graph cut methods did not address shape regularization constraints in such problems. Our multi-hedgehog prior is fairly flexible, has efficient optimizers, and shows significant potential in resolving very common ambiguities in multi-label segmentation problems, see Fig.1 (last row).

Our general multi-object segmentation framework allows to enforce "hedgehog" shape prior for any of the objects. The class of all possible hedgehog priors is sufficiently representative yet each specific hedgehog constraint offers sufficiently powerful regularization to address color overlaps and weak edges. One extreme case of our prior is related to the standard *star* shape prior [23]. The other extreme case allows shapes with restricted *skeletons* [18, 21].

The main contribution of our method is that it offers a practical and efficient way to integrate multiple shape priors into popular **multi-label** MRF segmentation framework [6]. Our work also allows to extend previous multi-surface graph cut methods [17, 8]. For example, [17] compute multiple nested segments using one fixed *polar* grid defined by some non-overlapping *rays*. Besides particular image discretization, these rays introduce two constraints: one star-like shape constraint shared by the nested segments and a smoothness constraint penalizing segment boundary jumping between adjacent rays. In contrast, our method defines independent shape constraints for each segment. Similarly to [15], shape normals are constrained by arbitrary vector fields, rather than non-overlapping rays [17] or trees [23, 11]. Our use of Cartesian grid allows to enforce standard boundary length smoothness [4]. While this paper is focused on Potts model with distinct shape constraints, hedgehog shapes can be easily combined with inter-segment *inclusion* or *exclusion* constraints [8]. The use of distinct (not necessarily nested) shape priors extends [17]. See [12] for a more detailed discussion.

There are other earlier methods for segmenting multiple objects with shape priors, e.g. [1] in the context of convex optimization and [24] in MRF-frameworks. These approaches require training data sets, which could be limiting when objects of interest have complex shapes that do not conform to any predefined structure, e.g. lesions. It is hard to train for such shapes. In contrast, our shape prior does not require training data and it is fairly flexible: even complex shapes could be defined via simple user interaction. Furthermore, [24] is not suitable for 3D medical segmentation where there no occlusion, since [24] allows a pixel or voxel to be assigned to multiple objects by construction.

Overview of contributions: We propose a new multi-label segmentation model and the corresponding optimization algorithm. Our contributions are summarized below.

- *hedgehog shape constraint* - a new flexible method for segmentation regularization based on simple and intuitive user interactions.
- new multi-object segmentation energy with *multi-hedgehog* shape priors.
- we provide an extension of α-expansion moves [6] for the proposed energy.
- experimental evaluation showing how our multi-object segmentation method solves problematic cases for the standard Potts model [6].

The rest of the paper is organized as follows. Section 2 defines our hedgehog shape prior for a simpler case of binary segmentation of one object. We discuss its properties and show how it can be globally optimized with s/t graph cuts. Section 3 defines multi-hedgehog shape constraint in the context of multi-label MRF segmentation and proposes an extension of α-expansion optimization algorithm. Our experiments in Section 4 includes multi-object segmentation of real photos and 3D multi-modal medical data. Finally, we discuss possible extensions and variations of the hedgehog prior in Section 5.

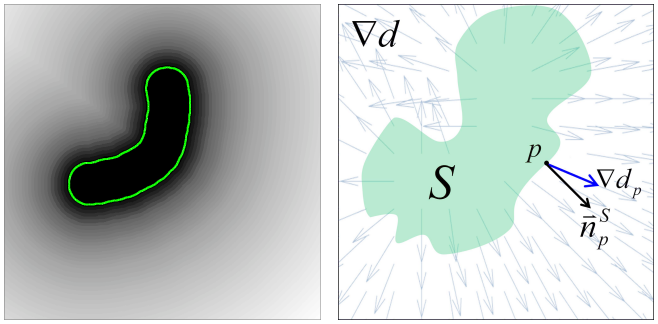
2. Hedgehog shape constraint for one object

This section describes *hedgehog* shape prior for a single object in case of binary segmentation. Section 3 describes a more general *multi-hedgehog* segmentation prior where multiple objects can have separate hedgehog constraints. While *multi-hedgehog* prior helps in a much wider range of problems, *e.g.* in medical imaging, binary segmentation with one "hedgehog" is easier to start from and it has merits on its own. In particular, single-hedgehog prior generalizes popular *star-convexity* [23, 11] in binary segmentation.

Similarly to *star* prior [23], hedgehog prior could be defined interactively. Instead of a single click in the star center, hedgehog shape allows an arbitrary scribble roughly corresponding to its *skeleton*. Hedgehog can also be defined by an approximate user-defined outline of a desired shape or by a shape template. In any case, such scribble, outline, or template define the corresponding (signed) *distance transform* or *distance map* $d : \Omega \to \mathcal{R}$ and a field of its gradients ∇d, as illustrated in Fig.2. Section 5 explores the potential/merits of using other vector fields. Our *hedgehog constraint* for segment S is defined by vector field $\{\nabla d_p \mid p \in \Omega\}$ and angular threshold θ restricting orientations of surface normals $\bar{n}_p^S$ at any point p on the boundary of S to a cone

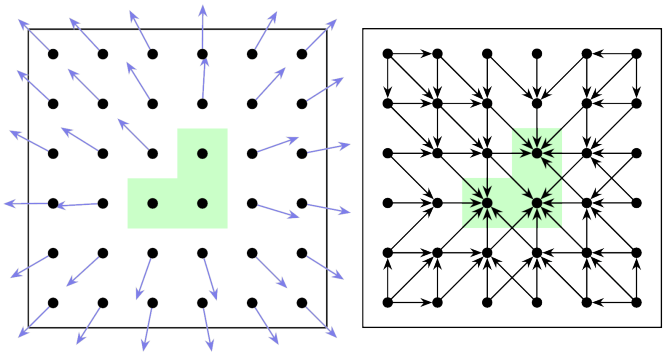
$$C_\theta(p): \quad \angle(\bar{n}_p^S, \nabla d_p) \le \theta \qquad \forall p \in \partial S \qquad (1)$$

assuming gradient ∇d_p is defined at p. More generally, *hedgehog constraint* for segment S could be defined by any given vector field $\{\bar{v}_p \mid p \in \Omega\}$ defining preferred directions for surface normals, Similarly to [15], we can use *dot product* to define allowed normals cones $C_\theta(p): \ \langle \bar{n}_p^S, \bar{v}_p \rangle \ge \tau$ where width varies depending on the magnitude of $\bar{v}_p$. In case $\bar{v}_p = \nabla d_p$ this constraint reduces to (1) for $\tau = cos(\theta)$ since $|\nabla d_p| = 1$ at all points where gradient ∇d_p exists.



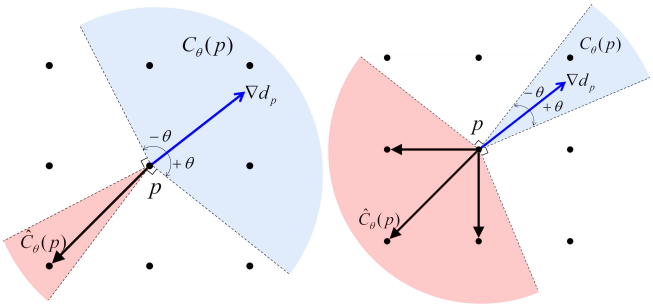
(a) scribble's distance map d (b) constraint $\angle(\bar{n}_p^S, \nabla d_p) \le \theta$

Figure 2. *Hedgehog prior* for segment S. (a) User-scribble or shape template define a (signed) distance map d. (b): Orientations of surface normals $\bar{n}_p^S$ for S are constrained by $\angle(\bar{n}_p^S, \nabla d_p) \le \theta$.



(a) gradients ∇d (b) graph edges $\mathcal{E}(\frac{\pi}{4})$

Figure 3. Hedgehog constraint (1) for user seeds (green) and the corresponding distance map gradients ∇d in (a) is approximated by infinity cost directed edges $\mathcal{E}(\theta)$ in (b) selected as in Fig.4.



(a) wide cone of normals C_θ (b) tight cone of normals C_θ

Figure 4. Approximating *hedgehog constraint* (1) at grid node p. Cone C_θ of allowed surface normals (blue) is enforced by ∞ cost directed edges (pq) in the corresponding *polar cone* $\hat{C}_\theta$ (red).

2.1. Single hedgehog properties

Even a single hedgehog shape prior discussed in this section could be useful in practice. For example, if $\theta = \pi/2$ it closely approximates a popular star convexity [23] in case of a single click. However, our formulation uses locally defined constraints, which can be approximated by a simple rule for selecting local edges, see Section 2.2. Unlike [23, 11], we do not enforce a global tree/forest structure, see Fig.3(b). Similar to [15, 11], hedgehog prior allows a much larger variability of shapes for scribbles different from a point. In our case, a scribble defines a rough *skeleton* of a shape. For smaller values of θ our cone constraints (1) give a tighter alignment of surface normals with vectors ∇d_p forcing the segment boundary to follow the level-sets of the scribble's distance map d. Under certain conditions[1] this implies consistency of segment's skeleton with the skeleton of the given scribble, outline, or template.

2.2. Single hedgehog via graph cuts

We show an approximation for *hedgehog constraint* (1) for object S in the context of binary N-dimensional image segmentation via graph cuts [3]. All cone constraints (1) for any given θ and distance map gradients ∇d, see Fig.3(a), correspond to a certain set of infinity cost directed edges, see Fig.3(b). For example, consider cone of allowed surface normals $C_\theta(p)$ at some point p illustrated in Fig.4 for two different values of parameter θ. It is easy to see that a surface/boundary of segment S passing at p has normal $\bar{n}_p^S \in C_\theta(p)$ iff this surface does not cross the corresponding *polar cone*

$$\hat{C}_\theta(p) \;:=\; \{(py) \,|\, \langle (py), (pz) \rangle \le 0 \;\; \forall z \in C_\theta(p)\}. \quad (2)$$

This reformulation of our hedgehog constraint (1) is easy to approximate via graph cuts by setting infinity cost to all directed edges adjacent to p whose directions agrees with polar cone $\hat{C}_\theta(p)$, see Fig.4. To avoid clutter, the figure only shows such directed edges $(pq) \in \hat{C}_\theta(p)$ starting at p, but one should also include similarly oriented directed edges $(qp) \in -\hat{C}_\theta(p) := \{(yp) \,|\, \langle (yp), (pz) \rangle \le 0 \;\; \forall z \in C_\theta(p)\}$ pointing to p. The set of all directed graph edges consistent with local polar cones orientations, see Fig.3(b), is

$$\mathcal{E}(\theta) = \{(pq) \in \mathcal{N} \mid (pq) \in \hat{C}_\theta(p) \; or \; (pq) \in -\hat{C}_\theta(q)\}. \quad (3)$$

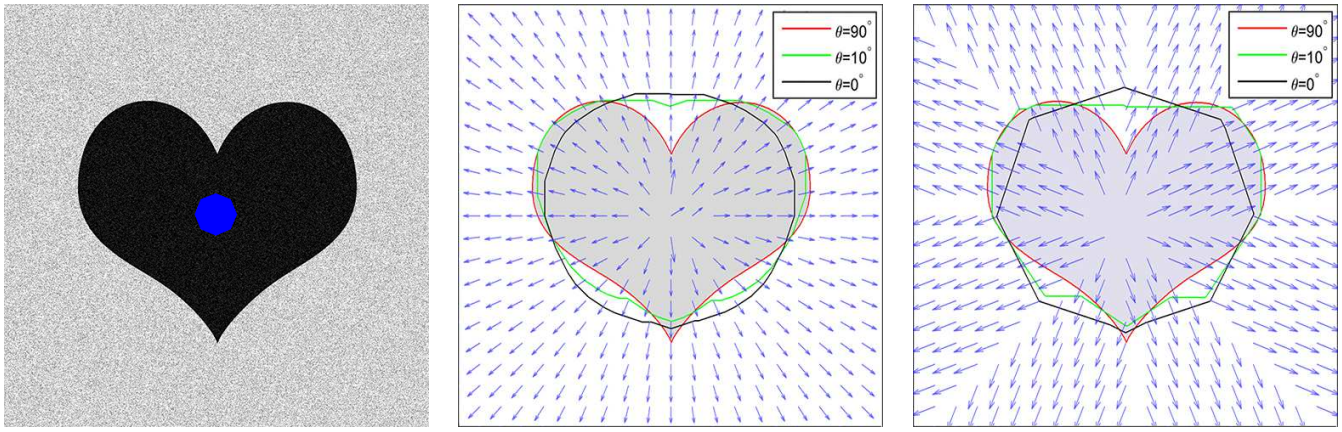
Obviously, hedgehog constraints are better approximated by large neighbour systems $\mathcal{N}$. The reader is referred to [12] for a more detailed discussion and experiments regarding discretization artifacts and how to avoid them.

It is easy to see that set (3) of infinity cost edges corresponds to a *submodular* pairwise energy approximating hedgehog shape constraint (1) for binary labeling $\mathbf{f} = \{f_p\}$ representing segment $S = \{p \mid f_p = 1\}$

$$h_\theta(\mathbf{f}) \;=\; \sum_{(pq) \in \mathcal{E}(\theta)} w_\infty \cdot [f_p = 1, f_q = 0] \quad (4)$$

where w_∞ is an infinitely large scalar.

[1] In general, *skeleton consistency* is subject to smoothness and *radial curvature condition*, see Definition 9 and Theorem 1 in [22] due to [7].



(a) (b) (c)

Figure 5. (a) image and scribble (octagon shown in blue). (b-c) show our results for varying θ, ground truth (gray) and the vector field used to enforce hedgehog constraints (blue). In (b) the vector field used to constraint the surface normals is the gradient of the scribble's distance transform and in (c) it is the gradient of a function where the levelsets are scaled versions of the scribble. Notice how the used vector field affects the segmentation for the same value of θ in (b) and (c). The hedgehog prior reduces to star-shape prior [23] only in (b) when $\theta = 90$ and if the star center was the octagon's center.

Figure 5 shows a simple foreground/background segmentation experiment illustrating how the choice of θ and vector field affects the segmentation result. As you case see in Fig.5(b) increasing θ leads to a segmentation that better aligns with the levelsets of the distance transform which is beneficial in case of clutter. Also, using a different vector field other than the gradient of the distance transform leads to a different set of allowed shapes by the hedgehog prior. We can see how the choice the vector field affected the segmentation by comparing Fig.5(b) to (c) for the same θ.

3. Multi-hedgehog segmentation energy

Given a set of pixels $\mathcal{P}$, neighborhood system $\mathcal{N}$, and labels $\mathcal{L}$ our multi-labeling segmentation energy is

$$E(\mathbf{f}) = \overbrace{\sum_{p\in\mathcal{P}} D_p(f_p)}^{data} + \overbrace{\lambda \sum_{pq\in\mathcal{N}} V_{pq}(f_p, f_q)}^{smoothness} + \overbrace{H_\theta(\mathbf{f})}^{hedgehogs} \quad (5)$$

where $\mathbf{f} = \{f_p \in \mathcal{L} | \ \forall p \in \mathcal{P}\}$ is a labeling.

The first two terms, namely *data* and *smoothness* terms, are widely used in computer vision, e.g. [5, 3, 20]. The data term commonly referred to as the *regional* term as it measures how well pixels fit into their corresponding labels. To be specific, $D_p(f_p)$ is the penalty for assigning label f_p to pixel p. Similar to [20], a label's probabilistic model, Gaussian Mixture in our case, is found by fitting a probabilistic model to the seeds given by the user.

The smoothness term is a standard pairwise regularizer that discourages segmentation discontinuities between neighboring pixels. A discontinuity occurs whenever two neighboring pixels $pq \in \mathcal{N}$ are assigned to different labels. In its simplest form, $V_{pq}(f_p, f_q) = w_{pq}[f_p \neq f_q]$ where [ ] is *Iverson bracket* and w_{pq} is a non-increasing function of the intensities at p and q. Also, λ is a parameter that weights the importance of the smoothness term.

Third term, our contribution, is the *Hedgehog term*

$$H_\theta(\mathbf{f}) = \sum_{k\in\mathcal{L}} \sum_{(pq)\in\mathcal{E}_k(\theta)} w_\infty \ [f_p = k, f_q \neq k] \quad (6)$$

where $w_\infty = \infty$. Those familiar with graph cuts may prefer to think of it as an ∞-cost arc from p to q, thus prohibiting any cut that statisfy $f_p = k$ and $f_q \neq k$. The Hedgehog term is the sum of the Hedgehog constraints over all the labels and it guarantees that any feasible labeling[2], i.e. $E(\mathbf{f}) < \infty$, will result in a segmentation with surface normals respecting the orientation constraints (1).

3.1. Expansion Moves

In this section we will describe how to extend the binary expansion moves of α-exp [6] to respect the shape constraints, and show that these moves are submodular. The main idea of α-exp algorithm is to maintain a current feasible labeling $\mathbf{f}'$, i.e. $E(\mathbf{f}') < \infty$, and iteratively move to a better labeling until no improvements could be made. To be specific at each iteration, a label $\alpha \in \mathcal{L}$ is chosen and variables f_p for all $p \in \mathcal{P}$ are given a binary choice x_p; 0 to retain their old label $f_p = f'_p$ or 1 switch to α, i.e. $f_p = \alpha$.

[2]We use feasible (and not bound) because there is at least one trivial solution with finite cost. In practice, it is practical to assume that one of the labels, e.g. background label, does not require enforcing shape constraints otherwise the problem could become over-constrained. One trivial solution is to label all pixels as background except those labeled by user scribbles.

The Hedgehog term (6) for a binary α-exp move could be written as

$$H^{\alpha}_{\theta}(\mathbf{x}) = \sum_{(p,q)\in\mathcal{E}_{\alpha}(\theta)} S_{pq}(x_p, x_q) + \sum_{k\in\mathcal{L}\setminus\alpha} \sum_{\substack{f'_p=f'_q=k \\ (pq)\in\mathcal{E}_k(\theta)}} S_{qp}(x_q, x_p) \quad (7)$$

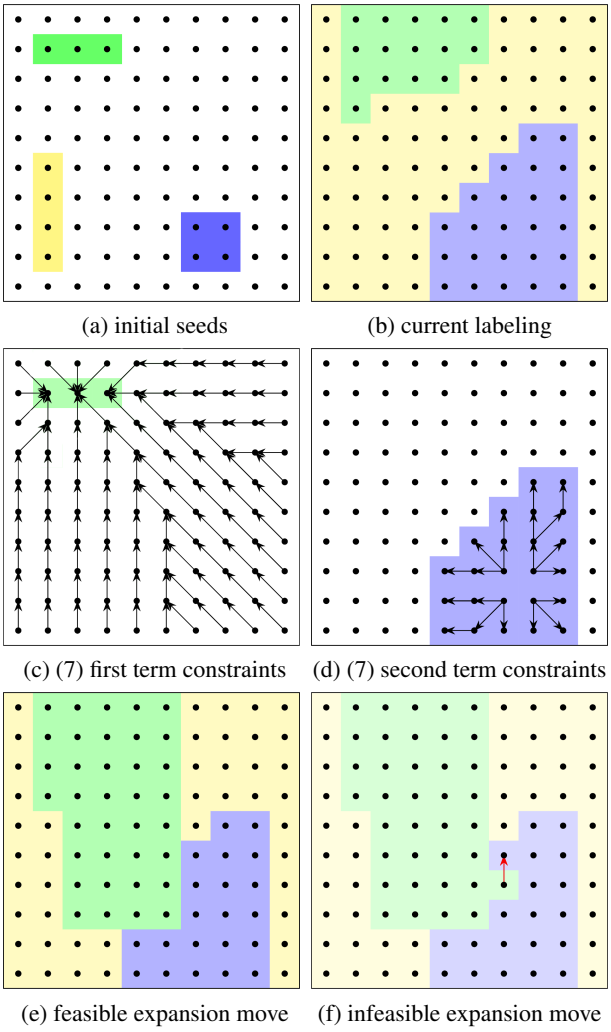
where $\mathbf{x} = \{x_p \in \{0,1\} \mid \forall p \in \mathcal{P}\}$ and

$$S_{pq}(x_p, x_q) = \begin{cases} \infty & \text{if } x_p = 1,\ x_q = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

The first term in (7) guarantees that the resulting labeling respects label α hedgehog constraints. In addition, the second term guarantees that the hedgehog constraints satisfied by the current labeling $\mathbf{f}'$ for all labels in $\mathcal{L}\setminus\alpha$ are not violated by the new labeling $\mathbf{f}$.

According to [16], any first-order binary function could be exactly optimized if all pairwise terms are submodular. A binary function g of two variables is submodular if $g(0,0) + g(1,1) \leq g(1,0) + g(0,1)$. Our energy (7) is submodular as it could be written as the sum of submodular pairwise binary energies over all possible pairs of p and q. Notice that for any given pq pair, $S_{pq}(1,1) = 0$ by construction and $S_{pq}(0,0) = 0$ as long as the current labeling is a feasible one, i.e. it does not cut any of the ∞-cost arcs. Also, $S_{pq}(1,0)$ and $S_{pq}(0,1)$ are both ≥ 0 by construction. Therefore, the submodularity condition is satisfied for all pairs of p and q.

Fig.6 shows an illustrative example of an α-exp move over the green label. In this example we assume that we are enforcing the shape constraints only for the green and purple labels. Fig.6(a) shows the initial seeds for three different labels while (b) shows the current feasible labeling. Fig.6(c-d) show the shape constraints enforced by green and purple labels while expanding the green label. Note, green shape constraints are enforced all over the image while purple shape constraints are enforced inside its current labeling support area, as it is not necessarily to enforce it everywhere. Fig.6(e) shows a feasible move that respects green and purple shape constraints while (f) shows an infeasible move that respects only the green shape constraints.



(a) initial seeds (b) current labeling

(c) (7) first term constraints (d) (7) second term constraints

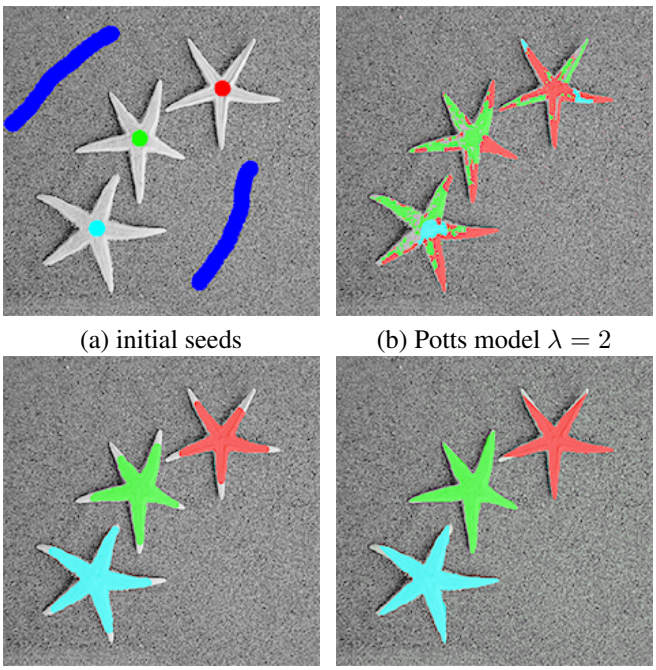
(e) feasible expansion move (f) infeasible expansion move

Figure 6. Illustration of a feasible and an infeasible expansion move for the green label. (a-b) Initial seeds and current labeling, respectively. (c-d) Hedgehog shape constraints (7) enforced by green and purple labels when expanding the green one. (e-f) show a feasible and an infeasible expansion moves, respectively. In (f) severed ∞-cost purple shape edge/constraint is shown in red.

4. Experiments

In the following set of experiments we show the benefit of incorporating our *Hedgehogs term* (6) to the well studied Potts model segmentation energy, i.e. *data term* + *smoothness term*, for multi-object segmentation in 2D and 3D. We will also give an illustrative real life example to show that the hedgehog shape is more general than star-shape [23]. The results shown in this section for our method were generated using $\theta = \frac{\pi}{4}$ when computing the hedgehog shape constraints, also we did not enforce any shape constraints on the background model. Also, the same smoothness weight λ is used when comparing methods unless stated otherwise.

Our optimization framework is similar to [20] where the user marks a set of initial seeds in the form of a scribble for the required labels, e.g. left kidney, right kidney etc. The seeds for each label were used to fit an initial Gaussian Mixture color model, and to generate its hedgehog shape constraints. Similarly to [13, 9], we iteratively optimize our energy (5) (or Potts model) in an EM-style iterative fashion. We alternate between finding a better segmentation and re-estimating the color models using the current segmentation. Finally, the framework terminates when it can not decrease the energy anymore.



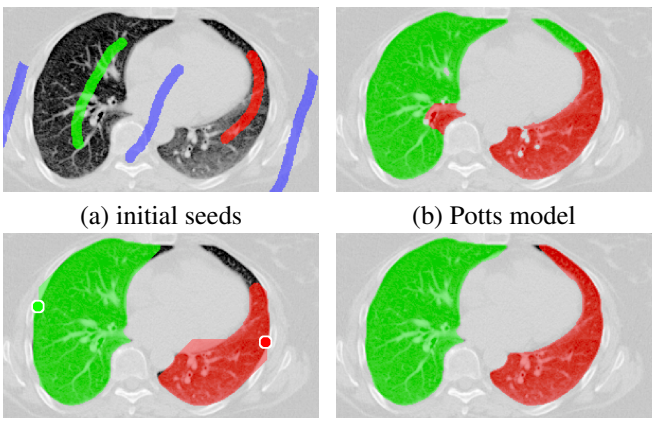
(a) initial seeds (b) Potts model $\lambda = 2$

(c) Potts model $\lambda = 6$ (d) Hedgehogs + Potts $\lambda = 2$

Figure 7. Three hedgehogs one for each star. (a) shows user scribbles. (c-b) and (d) show Potts model results for different λ and our results, respectively. (d) shows that enforcing hedgehogs shape priors (our method) eliminated oversegmented solutions as the one in (b) which is typical for small λ. (c) shows Potts model result for a larger λ at which the stars were not oversegmented, notice star tips were wrongly segmented due to the increase in shrinking bias.

For the example shown in Fig.7(a), (b-c) show Potts model results for $\lambda = 2$ and 6, respectively. It should be noted that 6 is the smallest smoothness weight that did not result in oversegmentation when using Potts. However, the result in Fig.7(c) is biased towards smaller objects (notice star tips) because by increasing the smoothness we are also increasing the shrinking bias. Oversegmented results as in Fig.7(b) could be avoided without increasing the shrinking bias by incorporating multi-shape priors. Our method which incorporates Hedgehog priors with Potts model was able to find a better segmentation, see Fig.7(d).

The objective of the example shown in Fig.8(a) is to segment left and right lungs, and the background. Potts model result shown in Fig.8(b) has holes, i.e. part of the background appears in the middle of the lungs. Furthermore, Potts model converged to biased color models where the right lung preferred brighter colors while the left preferred darker colors. Similar to the previous example, increasing λ for Potts model will increase the shrinking bias and it becomes hard to segment the elongated part of the the right lung. Using multi-star which is a generalization of [23] to multi-object segmentation is not enough because the right lung is not a star-shape. To be specific, there is no point



(a) initial seeds (b) Potts model

(c) Multi-Star + Potts model (d) Hedgehogs + Potts model

Figure 8. Two hedgehogs one for each lung. As can be seen in (b) Potts model resulted in segmentation with holes (background inside lungs), and converged to wrong color models. Segmentation holes could be eliminated by using multi-star shape priors (c)—star centers are the midpoints of the green and red circles. However, multi-star can never properly segment the right lung as it is not a star-shape. Our method (Hedgehogs + Potts) (d) eliminated holes and properly segmented the lungs by enforcing a more general shape constraint derived from the user scribble.

inside the right lung that could act as a center of a star-shape that would include it. Fig.8(d) shows the result for our method, where user scribbles were used to enforce shape constraints compared to using a single pixel per label [10].

We applied our method on PET-CT scans of three different subjects to segment their liver, left kidney, right kidney and the background. Although we applied our method and Potts model on the 3D volumes we only show the results on a few representative slices from each volume in Fig.9. Also, the results of different methods for each subject were computed using the same smoothness. We can see from the last two rows which compare our method to Potts, using Hedgehogs constraints enabled us to avoid geometrically incorrect segmentations, e.g. one liver inside the other (last-row middle), or parts of left kidney is between the right kidney and liver (last-row right). Furthermore, for test subjects 1 and 2 the kidneys and background were poorly segmented by Potts model, e.g. most of the kidneys were segmented as background for test subject 1. Potts poor performance is due to the large overlap between the kidneys and background color models. This overlap resulted in an in-discriminative data term for Potts to properly separate them. This issue becomes worse in iterative frameworks where color models are re-estimated based on current segmentation. To be specific, if at any iteration Potts model resulted in a bad segmentation then re-estimating the color models will bias them towards the bad segmentation and subsequent iterations worsen the results. Comparing our results for subjects 1 and 2 to Potts model shows that our method is less prone

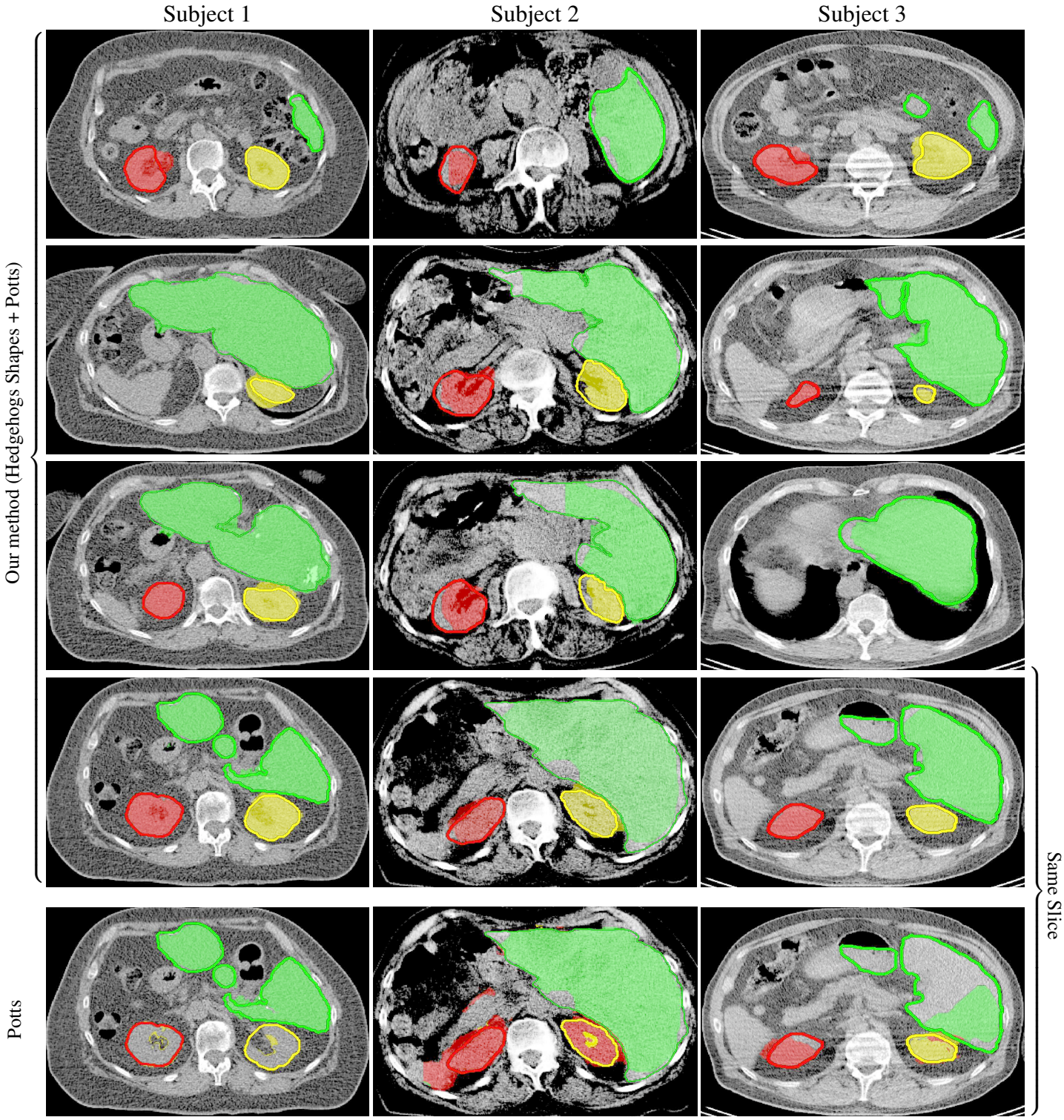


Figure 9. Three hedgehogs for liver and two kidneys, the colored contours represent liver and kidneys ground truth. Each column shows the result of a different test subject. The first four rows show our results, each row represents a different slice. The last row shows Potts model results. Also, the last two rows show results of the same slice for our method and Potts model, respectively. Our method (Hedgehogs+Potts) out performed Potts, results show that enforcing shape constraints avoids/forbids some undesirable segmentations, e.g. for subject 2 Potts segmentation shows that the left kidney surrounds the right kidney, and for subject 3 it shows that part of the left kidney is between the right kidney and liver. In addition, for subjects 1 and 2 Potts model did not properly separate the kidneys from the background.

to the aforementioned issue as we forbid undesirable segmentations, i.e. those that do not respect shape constraints.

For quantitative comparison, Table 1 lists for each organ of a subject the $\mathbf{F}_1$ Score, Precession and Recall measures of our method and Potts model where $\mathbf{F}_1 = 2 * \frac{Precession * Recall}{Precession + Recall}$. For kidneys, our method clearly out performed Potts model, e.g. note Potts model poor precision/recall for subjects 1 and 2. For liver, both methods performed comparably.

	Subject 1		Subject 2		Subject 3	
	Ours	Potts	Ours	Potts	Ours	Potts
	Right Kidney ■					
$\mathbf{F}_1$ score	**0.85**	0.05	**0.69**	0.11	**0.92**	0.85
Prec.	0.77	0.16	0.58	0.13	0.93	0.85
Recall	0.96	0.03	0.84	0.10	0.91	0.87
	Left Kidney ■					
$\mathbf{F}_1$ score	**0.96**	0.08	**0.81**	0.48	**0.93**	0.84
Prec.	0.90	0.97	0.85	0.34	0.95	0.76
Recall	0.95	0.04	0.78	0.80	0.91	0.93
	Liver ■					
$\mathbf{F}_1$ score	0.92	**0.93**	0.90	**0.91**	**0.92**	0.84
Prec.	0.92	0.93	0.97	0.96	0.97	0.96
Recall	0.92	0.93	0.84	0.87	0.88	0.74

Table 1. The table lists the $\mathbf{F}_1$ score, precision and recall measures for each method, individual organ and subject—the closer these values are to 1 the more accurate the segmentation is. For the kidneys where most of the color model overlap occurs, our method was a clear winner. For the liver which has a bigger volume and a more distinct color model compared to the kidneys/background, the two methods performed comparably.

5. Discussion

As illustrated in Section 2.2 the vector field plays an important role in controlling the set of shapes allowed by the hedgehog prior. While this work focuses only on a very basic vector field, i.e. gradient of the distance transform, the obtained results are promising. There are many ways to extend our work in regards of which vector field to use, which constraints to enforce and how to enforce them.

Instead of Euclidean distance transform gradients, hedgehog prior vector field can correspond to the gradient of the geodesic distance transform [19, 2] incorporating information about image colors. In case of binary segmentation with $\theta = \frac{\pi}{2}$ and with such geodesic vector field the hedgehog shape prior is similar to the geodesic-star [11]. Another idea for a hedgehog vector field could be based on a shape template and its *medial axis transform* with a known radius function. One can generate a set of nested shapes by scaling the radius function and compute the hedgehog vector field as the normals to these shapes.

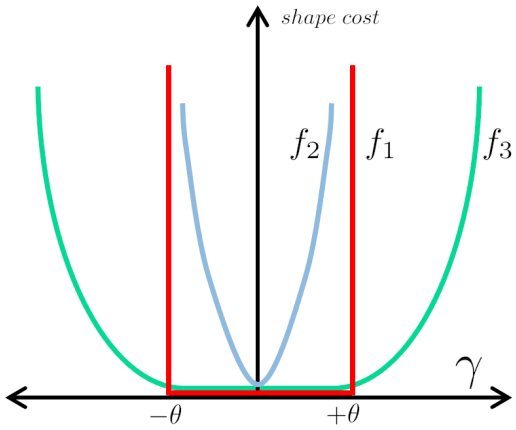


Figure 10. This figure shows a few different shape penalty/cost functions vs. γ, i.e. the angle between a segment's normal and its preferred orientation, but any convex function could be used [14]. In this work we only experimented with f_1.

Our proposed hedgehog utilizes only the angular information of the vector field. Alternatively, one could replace the hard constraints, i.e. ∞-cost edges, with soft constraints proportional to the vector field's magnitude at each pixel. In general, hedgehog prior could utilize different shape penalty functions as in Fig.10. In this work we only explored f_1 in Fig.10, for simplicity.

6. Conclusion

We proposed a novel interactive multi-object segmentation method where objects are independently restricted by hedgehog shape priors. In a nutshell, hedgehog shape prior for a given object uses a vector field derived from the user interaction to restrict the object's surface normals. In addition, we showed how to modify α-expansion moves to optimize our multi-labeling problem with hedgehog constraints. We also proved submodularity of the modified binary expansion moves. Furthermore, we applied our multi-labeling segmentation with hedgehog shapes on 2D images and 3D medical volumes. Our experiments show the significant improvement in segmentation accuracy when using our method over Potts model. Specially in medical data where our method outperformed Potts model in separating multiple organs with similar appearances and weak edges.

Acknowledgments We thank Kaleem Siddiqi of McGill University for his helpful discussion regarding skeleton consistency. We also thank Ipek Oguz of Iowa University for interesting comments regarding LOGISMOS. This work was mainly supported by NIH grant R01-EB004640. We also thank Drs. Sue O'Dorisio and Yusuf Menda for providing the PET-CT liver data (NIH grant U01-CA140206). This work was also supported by NSERC Discovery and RTI grants (Canada) for Y. Boykov and O. Veksler.

References

[1] S. Andrews, C. McIntosh, and G. Hamarneh. Convex multi-region probabilistic segmentation with shape prior in the isometric log-ratio transformation space. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2096–2103. IEEE, 2011.

[2] X. Bai and G. Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *International journal of computer vision*, 82(2):113–132, 2009.

[3] Y. Boykov and M.-P. Jolly. *Interactive graph cuts* for optimal boundary & region segmentation of objects in N-D images. In *ICCV*, volume I, pages 105–112, July 2001.

[4] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *International Conference on Computer Vision*, volume I, pages 26–33, 2003.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *International Conference on Computer Vision*, volume I, pages 377–384, 1999.

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.

[7] J. Damon. Determining the Geometry of Boundaries of Objects from Medial Data. *International Journal of Computer Vision (IJCV)*, 63(1):45–64, 2005.

[8] A. Delong and Y. Boykov. Globally Optimal Segmentation of Multi-Region Objects. In *International Conference on Computer Vision (ICCV)*, 2009.

[9] A. Delong, A. Osokin, H. Isack, and Y. Boykov. Fast Approximate Energy Minization with Label Costs. *International Journal of Computer Vision (IJCV)*, 96(1):1–27, January 2012.

[10] P. F. Felzenszwalb and O. Veksler. Tiered scene labeling with dynamic programming. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[11] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3129–3136. IEEE, 2010.

[12] H. Isack, Y. Boykov, and O. Veksler. A-expansion for multiple" hedgehog" shapes. *arXiv preprint arXiv:1602.01006*, 2016.

[13] H. N. Isack and Y. Boykov. Energy-based Geometric Multi-Model Fitting. *International Journal of Computer Vision (IJCV)*, 97(2):123–147, April 2012.

[14] H. Ishikawa. Exact optimization for markov random fields with convex priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1333–1336, 2003.

[15] V. Kolmogorov and Y. Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *International Conference on Computer Vision*, October 2005.

[16] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. In *7th European Conference on Computer Vision*, volume III of *LNCS 2352*, pages 65–81, Copenhagen, Denmark, May 2002. Springer-Verlag.

[17] K. Li, X. Wu, D. Z. Chen, and M. Sonka. Optimal surface segmentation in volumetric images-a graph-theoretic approach. *IEEE transactions on Pattern Analysis and Pattern Recognition (PAMI)*, 28(1):119–134, January 2006.

[18] S. Pizer *et al.* Deformable M-Reps for 3D Medical Image Segmentation. *International Journal of Computer Vision (IJCV)*, 55(2-3):85–106, November 2003.

[19] A. Protiere and G. Sapiro. Interactive image segmentation via adaptive weighted distances. *Image Processing, IEEE Transactions on*, 16(4):1046–1057, 2007.

[20] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. In *ACM transactions on Graphics (SIGGRAPH)*, August 2004.

[21] K. Siddiqi and S. Pizer. *Medial Representations: Mathematics, Algorithms and Applications*. Springer, December 2008.

[22] S. Stolpner, S. Whitesides, and K. Siddiqi. Sampled medial loci for 3D shape representation. *Computer Vision and Image Understanding (CVIU)*, 115(5):695–706, May 2011.

[23] O. Veksler. Star shape prior for graph-cut image segmentation. In *European Conference on Computer Vision (ECCV)*, 2008.

[24] N. Vu and B. Manjunath. Shape prior segmentation of multiple objects with graph cuts. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.