Star Shape Prior for Graph-Cut Image Segmentation

Olga Veksler

University of Western Ontario
London, Canada
olga@csd.uwo.ca

Abstract. In recent years, segmentation with graph cuts is increasingly used for a variety of applications, such as photo/video editing, medical image processing, etc. One of the most common applications of graph cut segmentation is extracting an object of interest from its background. If there is any knowledge about the object shape (i.e. a shape prior), incorporating this knowledge helps to achieve a more robust segmentation. In this paper, we show how to implement a *star* shape prior into graph cut segmentation. This is a generic shape prior, i.e. it is not specific to any particular object, but rather applies to a wide class of objects, in particular to convex objects. Our major assumption is that the center of the star shape is known, for example, it can be provided by the user. The star shape prior has an additional important benefit - it allows an inclusion of a term in the objective function which encourages a longer object boundary. This helps to alleviate the bias of a graph cut towards shorter segmentation boundaries. In fact, we show that in many cases, with this new term we can achieve an accurate object segmentation with only a single pixel, the center of the object, provided by the user, which is rarely possible with standard graph cut interactive segmentation.

1 Introduction

In the last decade, two important trends in image segmentation are the introduction of various user interaction techniques, and the development and increased reliance on global optimization methods. Interactive segmentation ([1, 2, 3, 4, 5, 6, 7]) became popular because in different domains, user interaction is available, and it can greatly reduce the ambiguity of segmentation caused by complex object appearance, weak edges, etc. Global optimization ([5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]), often formulated as a graph problem, became popular because it is more robust compared to the local methods.

In this paper, we address the segmentation of an object from its background in the graph cut framework [5, 7]. The advantage of this framework is that it guarantees a globally optimal solution for a wide family of energy functions [17], allows incorporation of regional and boundary constraints, and provides a simple user interaction interface. The user has to mark some pixels as object and some pixels as background. Such pixels are usually called “seeds”.

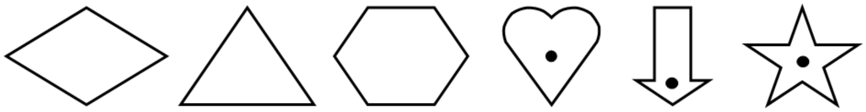


Fig. 1. Star shape examples. First three shapes are convex and therefore are stars with respect to any inside point as the center. Last three shapes are stars with respect to the specified center, although there are multiple other valid centers.

If one has prior knowledge about the shape of an object (or a "shape prior"), incorporating this knowledge makes segmentation more robust. Shape prior reduces ambiguity by ruling out all segments inconsistent with the prior. Using shape priors to improve segmentation has been investigated in the level set and curve evolution frameworks [18, 19, 20, 21]. Level set methods are usually not numerically stable and are prone to getting stuck in a local minimum.

There has been some work on shape priors for graph cuts. The authors in [22] use an elliptical prior, which is implemented only approximately within an iterative refinement process. In [23], a prior which encourages the object to be a convex blob centered around a certain point is implemented. Another example of a blob like prior is in [24]. The above shape prior assumptions are useful, but are quite restrictive on the shape of the object. In [25], an interesting "connectivity" prior is used, that is they enforce the object region to be connected. In [26, 27], an object specific shape prior is used. However, a shape model has to be registered to an image, which is challenging and computationally expensive.

In this paper, we investigate a generic shape prior for graph cut segmentation. Our prior is generic because it is not based on a shape of a specific object class (like a "cow" class), but rather it is based on simple geometric properties of an object, similar to the ellipse assumption in [22]. Our shape prior is much more general than an ellipse though. We call it a *star* shape prior, defined as follows. A *star* shape is defined with respect to a center point c. An object has a *star* shape if for any point p inside the object, all points on the straight line between the center c and p also lie inside the object. Some star shapes are in Fig. 1.

We assume that the user marks the star center. In many cases this information is enough to accurately segment the object, see Sec. 5.

Star shaped objects are abundant in the environment. A special case is a convex shape, and in this case an additional advantage is that the user can choose any point inside the object as the center, since a convex shape is a star with respect to any inside point. For many other shapes there are multiple valid centers, so, in general, the user does not have to be too careful in choosing the center. For example, for the heart shape in Fig. 1, most points, except the ones in approximately the top fifth part of the shape, make a valid center.

The advantage of using a generic star shape prior is that it can be directly incorporated in the optimization procedure, no expensive registration between the model and the image, like in [26, 27] is required. The disadvantage is that only a shape obeying a generic star shape is extracted, we cannot guarantee that the extracted shape will be a circle, or a rectangle, etc.

An important positive side effect of the star shape prior is that we can include in the objective function a length-based "ballooning" term that encourages a larger object segment. This term helps to counterbalance the known bias of a graph cut to small segments. It is not as aggressive as the previously used area-based "ballooning" terms. With the new term, it is frequently enough for a user to provide just the object center, additional information about the object may be unnecessary, making segmentation very undemanding for user interaction. Note that [23, 24] also support a single-click segmentation with graph cuts.

The paper is organized as follows. Sec. 2 reviews graph cut segmentation, Sec. 3 explains how to incorporate the star shape prior, Sec. 4 explains how to implement bias toward larger segments, and Sec. 5 has the experiments.

2 Graph Cut Segmentation

We now briefly review the graph cut segmentation algorithm of [5].

2.1 Graph Cut

Let $G = (V, E)$ be a graph with vertices V and edges E. Each edge $e \in E$ has a non-negative cost w_e. There are two special vertices called *terminals*: the source, s and the sink, t. A cut $C \subset E$ is a subset of edges, such that if C is removed from G, then V is partitioned into two disjoint sets S and $T = V - S$ such that $s \in S$ and $t \in T$. The cost of the cut C is the sum its edge weights: $|C| = \sum_{e \in C} w_e$.

The minimum cut is the cut with smallest cost. The max-flow/mincut algorithm [28] can be used to find the minimum cut in polynomial time. We use the max-flow algorithm of [29], which has linear time performance in practice [29].

2.2 Object/Background Segmentation with a Graph Cut

Segmenting an object from its background is formulated as a binary labeling problem, i.e. each pixel in the image has to be assigned a label from the label set $\mathcal{L} = \{0, 1\}$, where 0 and 1 stand for the background and the object, respectively.

Let $\mathcal{P}$ be the set of all pixels in the image, and let $\mathcal{N}$ be the standard 4 or 8-connected neighborhood system on $\mathcal{P}$, consisting of ordered pixel pairs (p, q) where $p < q$. Let $f_p \in \mathcal{L}$ be the label assigned to pixel p, and $f = \{f_p | p \in \mathcal{P}\}$ be the collection of all label assignments. The energy function commonly used for segmentation is as follows:

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \lambda \sum_{(p,q) \in \mathcal{N}} V_{pq}(f_p, f_q). \tag{1}$$

In Eq. (1), the first term is called the *regional* or *data* term because it incorporates regional constraints. Specifically, it measures how well pixels fit into the object or background models. $D_p(f_p)$ is the penalty for assigning label f_p to pixel p. The more likely f_p is for p, the smaller is $D_p(f_p)$. The object/background models could be known beforehand, or modeled from the seeds provided by the user. To insure that the seeds are segmented correctly, for any object seed p, one sets $D_p(0) = \infty$, and for any background seed p, one sets $D_p(1) = \infty$.

The second sum in Equation (1) is called the *boundary* term because it incorporates the boundary constraints. A segmentation boundary occurs whenever two neighboring pixels are assigned different labels. $V_{pq}(f_p, f_q)$ is the penalty for assigning labels f_p and f_q to neighboring pixels. Most nearby pixels are expected to have the same label, therefore there is no penalty if neighboring pixels have the same label and a penalty otherwise. Typically, $V_{pq}(f_p, f_q) = w_{pq} \cdot I(f_p \neq f_q)$, where $I(\cdot)$ is 1 if $f_p \neq f_q$ and 0 otherwise.

To align the segmentation boundary with intensity edges, w_{pq} is typically a non-increasing function of $|I_p - I_q|$, where I_p is the intensity of pixel p. For example, the following is frequently used [5]:

$$w_{pq} = e^{-\frac{(I_p - I_q)^2}{2\sigma^2}}. \tag{2}$$

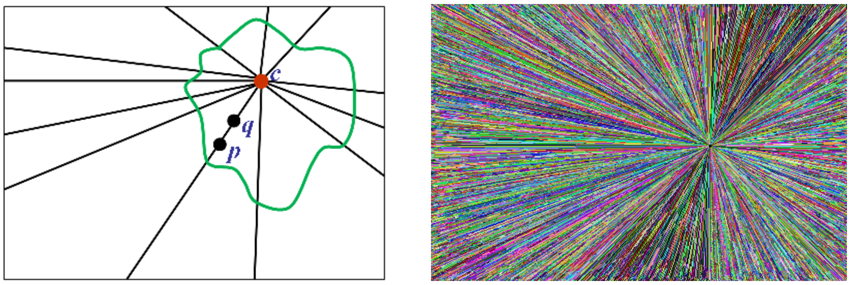
Parameter $\lambda \geq 0$ in Eq. (1) weights the relative importance between the regional and boundary terms. Smaller λ makes regional terms more important.

In [5] they show how to construct a graph such that the labeling corresponding to the minimum cut is the one optimizing the energy in Eq. (1). In general, [17] shows which binary energies can be optimized exactly with a graph cut.

3 Implementing the Star Shape Prior

We now show how to implement the star shape prior in the graph cut segmentation. We assume that the center of the star shape c is known. In interactive segmentation it is provided by the user. In certain restricted domains, such as in medical imaging, it may be possible to calculate the center automatically.

Consider Fig. 2(a). The center of the star shape is marked with a black dot c, and an example of a star shape is outlined in green. Some of the straight lines



(a) (b)

Fig. 2. (a) A star shape is in green, its center is marked with a red dot c. Let p and q be pixels on the line passing through c. If p is labeled as the object, then q must be also labeled as the object; (b) Discretized lines are displayed with random colors.

passing through c are shown in black. Let 1 and 0 be the object label and the background labels, respectively. To get an object segment of a star shape, for any point p inside the object, we have to insure that every single point q on the straight line connecting c and p is also inside the object. This implies that if p is assigned label 1, then every point between c to p (on a straight line) is also assigned 1. The following pairwise *shape constraint* term S_{pq} implements this:

$$S_{pq}(f_p, f_q) = \begin{cases} 0 & \text{if } f_p = f_q, \\ \infty & \text{if } f_p = 1 \text{ and } f_q = 0, \\ \beta & \text{if } f_p = 0 \text{ and } f_q = 1 \end{cases} \tag{3}$$

Eq. (3) assumes that q is between c and p. A segmentation with a finite cost never violates the star shape constraints. Parameter β is discussed later.

In discrete implementation, c, p, and q are pixels. Observe that the shape constraint term S_{pq} in Eq. (3) does not need to be placed between all pairs of pixels p, q that lie on a line passing through c. It is enough to put an S_{pq} only between neighboring pixels p and q. Indeed, if the star shape is violated along some line passing through c, then there may be several pairs of pixels p and q, (with q in between c and p) that violate the constraint. There will be a pair of pixels p and q with the smallest distance between them, and such two pixels must be neighbors. Conversely, if the star shape constraints are not violated between all the neighboring pixels pairs, they are not violated between pairs of pixels that are not neighbors, and therefore the shape is a star. Thus the neighborhood system for incorporating the star constraints is the same as for the boundary constraints, making the efficiency overhead for the shape prior negligible. Also note that using the star shape constraints is equivalent to adding a flux field [30].

In practice we have to discretize the set of lines passing through the center c. We consider all the lines that pass through the center pixel c and any other image pixel p. This is the finest possible discretization at the given image resolution.

We have to be careful when implementing the shape constraints on discrete lines. Continuous lines intersect only at the center c. Discrete lines can "intersect" at more than one pixel. Consider Fig. 3(a). One discretized line is shown in red, and another line with a larger slope is shown in black. These two lines first intersect at pixel p, and then at pixels q and r. After pixel p, these two lines

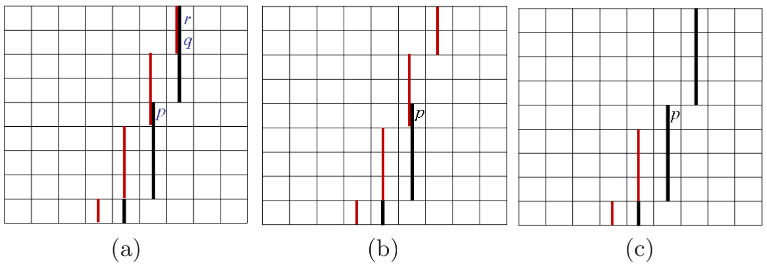


Fig. 3. (a) the red and black discrete lines "intersect" at more than one point; (b) the black line is merged into the red line; (c) the red line is merged into the black line

become essentially indistinguishable at image precision. Therefore at the first detected intersection pixel, in this case pixel p, we merge either the black line into the red one (Fig. 3 (b)) or vice versa (Fig. 3 (c)), chosen at random.

Fig. 2(b) shows with random colors the discrete merged lines that are used for star shape constraints (generated from a particular example). Closer to the center of the star shape more lines have to be merged together, therefore the density of lines is smaller compared to the density towards the image borders.

With the shape constraints, the our energy function becomes:

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \lambda \sum_{(p,q) \in \mathcal{N}} V_{pq}(f_p, f_q) + \sum_{(p,q) \in \mathcal{N}} S_{pq}(f_p, f_q). \qquad (4)$$

In Eq. (4), the V_{pq} terms are as defined in Sec. 2, and the shape constraint S_{pq} terms are as defined in Eq. (3). According to [17], the energy in Eq. (4) can be optimized exactly with a graph cut if all the pairwise terms are submodular, where a binary function g of two variables is submodular if $g(0,0) + g(1,1) \leq g(1,0) + g(0,1)$. Both the V_{pq} and S_{pq} terms are clearly submodular, and what is more interesting, the S_{pq} terms are submodular for any finite choice of β. If we set $\beta = 0$, then the labeling minimizing the energy in Eq. (4) is the same as the one optimizing the standard energy in Eq. (1), except the optimal object segment is star shaped. However, we can do more interesting things. Notice that β can be set to a negative value. This enables a bias towards a longer segmentation boundary, as explained in the next section.

4 Bias Toward Longer Segment Boundaries

4.1 Boundary Based Ballooning

A graph cut has a well known bias towards shorter boundaries. When a reliable model for the object and background is available, the data term in Eq. (4) can be given a large weight relative to the boundary term, by setting λ to a relatively smaller value. In this case, the bias to shorter boundaries is actually helpful to the segmentation process, since it serves to regularize the data terms. The data term can be known beforehand or it can be estimated from the seeds [7].

In the absence of a reliable model for the foreground/background, the data term has to be weighted low relative to the boundary term. In such a case, bias towards shorter boundaries is not helpful. The extreme case is when nothing about the appearance is known, and therefore the only non-zero data terms are those for the background/foreground seeds. If the user marks only a few seeds, then in most cases the result will consist of most pixels assigned to the same label. By marking enough seeds, the correct segmentation can always be achieved, but the amount of user interaction may be excessive.

If a user enters only a few seeds, estimating a reliable appearance model may be impossible. Furthermore, in the case when the background and foreground objects have similar appearance, it may be difficult or impossible to construct reliable appearance models. Consider the image in Fig. 4(a). The heart object

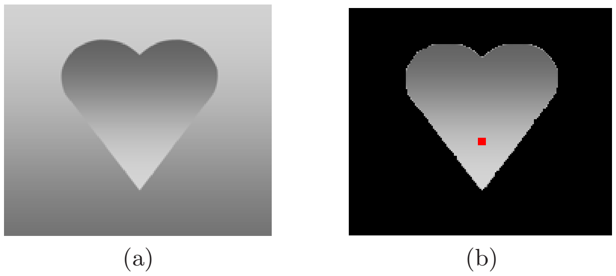


Fig. 4. (a) The heart object and its background have identical histograms; (b) Our result, the seed point is in red, only one object seed pixel is provided by the user, the border of the image is assumed to be the background

and its background have identical intensity histograms. If the appearance model is based only on the intensity histogram, it cannot distinguish between the foreground and the background. A user has to provide a significant number of seeds to segment this object. Notice that this image is not simple to segment with local algorithms because of intensity variation and weak boundaries.

To prevent the shrinking bias of a graph cut in the absence of a strong data term, a bias towards a longer boundary is needed, or, in other words, a "ballooning" force. We can easily incorporate such bias by setting β in Eq. (4) to a negative value. The last summation term in Eq. (4) is roughly proportional to the length of the boundary, and setting β to a negative value implies that longer segmentation boundaries decrease the energy function more as compared to the shorter boundaries[1]. The question is how to choose an appropriate β value, since the best value is likely to be different for each image.

In the related work on ratio cycles and regions [9], [12], [31], a ratio energy $E_{ratio}(f) = f_{cost1} + \beta \cdot f_{cost2}$ is considered. Here f_{cost1} is usually related to the cost of the object boundary, and f_{cost2} is related to the object area or boundary length. A minimum ratio region is found by searching for β that s.t. the optimum value of E_{ratio} is 0. Usually binary search is used to find such β, and the energy E_{ratio} is repeatedly optimized for different β values. The optimum region has the smallest normalized f_{cost1}, where normalization is by length or by area, depending on f_{cost2}. Typically f_{cost1} is related to the contrast on the boundary, and therefore the region with highest normalized contrast is found.

Our energy in Eq. (4) is basically the same as the ratio energy. Ignoring the data terms, our energy is approximately $f_{weight} + \beta \cdot f_{length}$, where f_{weight} is the sum of w_{pq} weights on the boundary between the object and the background segments, and f_{length} is the length of the boundary, or the sum of all S_{pq}'s. Therefore we could follow the strategy similar to the ratio regions by finding the highest contrast boundary. However, we observe that the highest contrast boundary may not be what the user wants. For example, if every image is placed in a "frame" with high contrast, this frame would always be extracted.

[1] This is due to the merging of discrete lines, discussed in the previous section.

Instead, we pursue a different strategy. We find the largest β such that the object segment is at least some minimum specified size, which we set to 100 in all the experiments. Let $\beta_1 < \beta_2 < 0$, and let f^1 be the labeling minimizing the energy in Eq. 4 with $\beta = \beta_1$ and f^2 be the labeling minimizing the energy in Eq. 4 with $\beta = \beta_2$. It is easy to see that $f^1_{weight} \geq f^2_{weight}$ and $f^1_{length} \geq f^2_{length}$. That is a smaller (or large negative) value of β results in a larger object segment with a larger sum of boundary weights w_{pq}. The sum of the boundary weights w_{pq} is just the standard cost of a labeling in Eq. (1), without ballooning (and ignoring the data terms). Therefore our strategy is equivalent to searching for a minimum cost labeling (without ballooning) that gives the object segment of size at least 100. To find such β, we use binary search, in the range from 0 to 50.

To test the effectiveness of our approach, we do not use background/foreground models for all the experiments in this paper. We set the pixels on the image border to be the background seeds, and the user provides the center of the star shape, which is the single object seed. We could also incorporate the data term, but it makes it harder to evaluate the effectiveness of the shape prior and the parameter search strategy. Fig. 4(b) shows our segmentation result on the image in Fig. 4(a). The smallest β that gave the first large object segment is -1.97.

Notice that we do not need to rerun the graph cut algorithm from scratch when searching for the value of β. We can use the idea of [32] to reuse the flow computation from the previous run. Thus the overhead we pay for the search is minimal, on average, the algorithm is 2.3 times slower with the search for β than without[2]. The algorithm in [32], while performing well in practice, has no guarantees on the computational efficiency, in general. The parametric max-flow algorithm of [31] does have theoretical guarantees, but unfortunately their method has certain restrictions that are not applicable to our approach.

4.2 Relation to Other "Ballooning" Methods

To encourage a larger object segment, we balloon the boundary, i.e. have a bias to a longer boundary. Our ballooning is effectively equivalent to the ratio cycle method in [12]. The difference is that we work in the graph cut framework, and can easily implement user interaction, background models, and all the other advantages of graph cuts. Another difference that instead of finding the "cycle" with the best ratio (or best average) contrast, we find a large enough "cycle" with a good contrast, which has certain advantages, as already mentioned above.

There are other ways to add a "ballooning" force. For example, uniform area based ballooning [33] can be used, which is implemented by adding a constant bonus to each pixel in the image if it is assigned the foreground label. The disadvantage of uniform ballooning is that the object region is not guaranteed to be connected, as is guaranteed with our method. A bigger disadvantage is that area ballooning is more aggressive compared to the boundary ballooning, in the sense that it may prefer a larger region to a smaller, but also a reasonable

[2] The graph cut code for "recycling" the flow was downloaded from http://www.adastral.ucl.ac.uk/~vladkolm/software.html

cost region. This may also happen with length ballooning, but it is less likely. We can roughly show that if a region can be extracted with the area ballooning, then it can be extracted with length ballooning, but not vice versa.

Let $E_{length}(f) = f_{cost} + \beta \cdot f_{length}$ be the energy with length ballooning and $E_{area}(f) = f_{cost} + \beta \cdot f_{area}$ be the energy area ballooning, where f_{cost} is the cost of the boundary related to its contrast, f_{length} is the length of the object segment and f_{area} is the area of the object segment. Suppose $\bar{f}$ can be extracted with area ballooning, that is there is a $\bar{\beta}$ s.t. $\bar{f}_{cost} + \bar{\beta} \cdot \bar{f}_{area} \le f_{cost} + \bar{\beta} \cdot f_{area}$ for all labelings f. Then $\bar{\beta} \le \frac{f_{cost} - \bar{f}_{cost}}{\bar{f}_{area} - f_{area}}$ for all f s.t. $f_{area} \le \bar{f}_{area}$ and $\bar{\beta} \ge \frac{f_{cost} - \bar{f}_{cost}}{\bar{f}_{area} - f_{area}}$ for all f s.t. $f_{area} > \bar{f}_{area}$. Let f^s be s.t. $f^s_{area} \le \bar{f}_{area}$ and $\frac{f^s_{cost} - \bar{f}_{cost}}{\bar{f}_{area} - f^s_{area}}$ is the smallest possible out of $f_{area} \le \bar{f}_{area}$. We can safely assume that $f^s_{cost} - \bar{f}_{cost} < 0$, since if $f^s_{cost} > \bar{f}_{cost}$, then any labeling with smaller area than $\bar{f}$ has a cost higher than $\bar{f}$, and therefore no labeling smaller than $\bar{f}$ can be extracted and we would not even have to consider labelings with smaller area than $\bar{f}$. Similarly, let f^g be s.t. $\frac{f^g_{cost} - \bar{f}_{cost}}{\bar{f}_{area} - f^g_{area}}$is as large as possible out of all $f^g_{area} > \bar{f}_{area}$. We must have $\frac{f^g_{cost} - \bar{f}_{cost}}{\bar{f}_{area} - f^g_{area}} \le \bar{\beta} \le \frac{f^s_{cost} - \bar{f}_{cost}}{\bar{f}_{area} - f^s_{area}}$. Since area scales quadratically in terms of length, approximately we have $f_{area} = (f_{length})^2$. Therefore we can approximate $\frac{f^g_{cost} - \bar{f}_{cost}}{(\bar{f}_{length} - f^g_{length})(\bar{f}_{length} + f^g_{length})} \le \bar{\beta} \le \frac{f^s_{cost} - \bar{f}_{cost}}{(\bar{f}_{length} - f^s_{length})(\bar{f}_{length} + f^s_{length})}$. Rearranging the terms we get: $\frac{f^g_{cost} - \bar{f}_{cost}}{\bar{f}_{length} - f^g_{length}} \le \bar{\beta}(\bar{f}_{length} + f^g_{length}) \le \frac{(f^s_{cost} - \bar{f}_{cost})(\bar{f}_{length} + f^g_{length})}{(\bar{f}_{length} - f^s_{length})(\bar{f}_{length} + f^s_{length})}$. Now $\frac{\bar{f}_{length} + f^g_{length}}{\bar{f}_{length} + f^s_{length}} \ge 1$ and $f^s_{cost} - \bar{f}_{cost} < 0$. Therefore $\frac{f^g_{cost} - \bar{f}_{cost}}{\bar{f}_{length} - f^g_{length}} \le \bar{\beta}(\bar{f}_{length} + f^g_{length}) \le \frac{f^s_{cost} - \bar{f}_{cost}}{\bar{f}_{length} - f^s_{length}}$. We need to make a couple of reasonable assumptions. Let us assume that f^s is also s.t. $f^s_{length} \le \bar{f}_{length}$ and $\frac{f^s_{cost} - \bar{f}_{cost}}{\bar{f}_{length} - f^s_{length}}$ is the smallest possible out of $f_{length} \le \bar{f}_{length}$. Similarly, let f^g be also s.t. $\frac{f^g_{cost} - \bar{f}_{cost}}{\bar{f}_{length} - f^g_{length}}$is as large as possible out of all $f^g_{length} > \bar{f}_{length}$. Rolling back the arguments in the beginning of this paragraph, we get that $\bar{f}_{cost} + \bar{\beta}(\bar{f}_{length} + f^g_{length}) \cdot \bar{f}_{length} \le f_{cost} + \bar{\beta}(\bar{f}_{length} + f^g_{length}) \cdot f_{length}$ for all labelings f, and therefore $\bar{f}$ can be extracted with length based ballooning for $\beta = \bar{\beta}(\bar{f}_{length} + f^g_{length})$.

The reverse is not true, that is, for some choices of $\bar{f}$, there is a $\bar{\beta}$ that allows extraction with length ballooning but not area ballooning. Instead of proving this, we show in example. In Fig. 5, the original image is in (a), our results (with boundary ballooning) are in (b), and the results with uniform area ballooning are in (c). For area ballooning, just as for our algorithm, we chose the smallest β giving the object segment of size at least 100 pixels. Since for larger β segments get larger, with uniform area ballooning, for no choice of β is it possible to extract the seal (with only a single seed provided by the user).

Another option is non-uniform area ballooning. For example, in [34] the ballooning decreases at the rate of $1/r$ where r is the distance from the pixel to the center. The problem is that such ballooning is biased towards including pixels

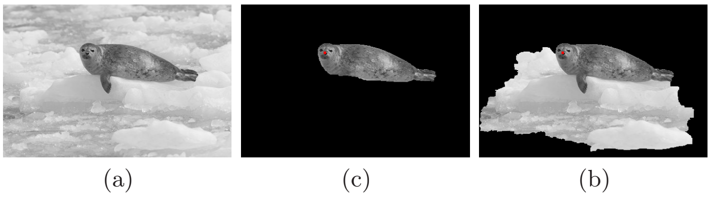


Fig. 5. (a) Original image; (b) Our algorithm; (c) Area "ballooning"

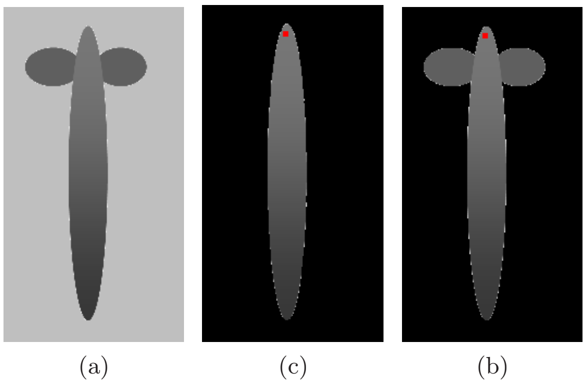


Fig. 6. (a) Original image; (b) Our results; (c) Non-uniform area "ballooning"

closer to the center, since such pixels get "ballooned" more. Consider Fig 6. It shows an oval object front of two smaller oval objects. The contrast between the larger ovals and two smaller ones is weak. Because of the star shape, we are able to extract the oval, shown in (b). With non-uniform ballooning, all three ovals are extracted because the smaller ovals get stronger "ballooning" than the lower two thirds of the large oval, since they are closer to the seed.

5 Experimental Results

First we summarize the experimental setup. The borders of the image are fixed to be the background seeds. The user provides a single object seed pixel, which is assumed to be the center of the star shape. Using binary search, we find the smallest β resulting in object segment larger than 100 pixels. Although we could make further corrections for the boundary with the help of user interaction, we choose not to do so to make the evaluation of the star shape effects easier.

We set $\lambda = 20$, although its particular choice is not important, since changing β changes the relative weight between the regional and boundary terms. The only remaining parameter is σ in Eq. (2). Typically [5] one sets σ as an average absolute intensity difference between neighboring pixels. Such choice for σ is motivated by the following. If the difference in intensities between two pixels is twice larger than the typical (average) difference, then the weight of connection w_{pq} between these pixels is very small, allowing to place a boundary between

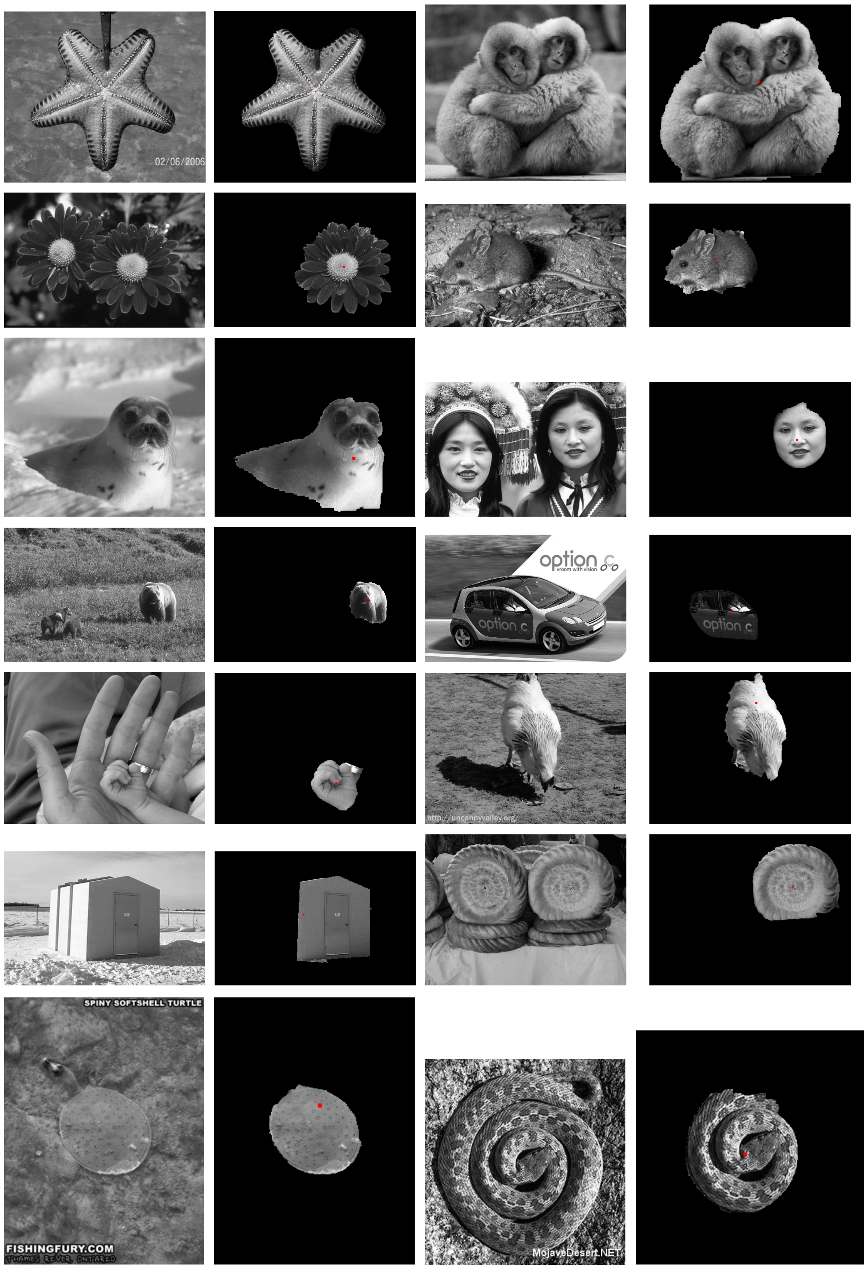


Fig. 7. Some results

them at a very small cost. If the intensity difference is smaller than the average, then w_{pq} is large. When the difference is in the range from the average to twice larger than the average, the weights w_{pq} decrease at the exponential rate. We use the same idea, except for pixels p and q we set σ to be the absolute average intensity difference in the box around p and q, not the whole image. The box size is set to be 20 by 20. This approach is helpful to encourage the boundary along the edges that may be relatively weak as far as the whole image is concerned, but are significantly strong if one looks only at their local neighborhood.

Fig. 7 shows results. Some of the images are from the Berkeley database [35], and others are from the Web. The odd columns show the original image, and the even columns show segmentation results. All the images are gray scale, which makes the segmentation problem much more challenging compared to the color images. The object segment is shown with its original intensities, and the background is shown in black. The seed pixel is a red square. For $\beta = 0$, for all the images shown, the object segment consists of either the single seed pixel or just a few pixels. Therefore the standard graph cut algorithm without bias to longer object boundaries fails without further user interaction.

The results are very promising, especially considering that they were obtained with a single user click. In many cases, the extracted segment is a meaningful object or a collection of objects. We are able to deal with weak boundaries, and complex foreground and background. In some cases, a meaningful part of the object is found. For example, in the forth row of Fig. 7, the door part is segmented even though the whole car is a star shape. This is due to a strong intensity edge around the door part.

In cases when only a part of the object has been segmented, the extracted part can be used for improving segmentation without further user interaction. For example, for the snake image in the bottom row of Fig. 7, we can get a reliable texture model from the extracted part, and then rerun the graph cut with this data term without the star shape constraints. This would make the algorithm similar to the Grab Cut [36]. In Grab Cut, an initial region containing the object is specified by the user as a box. However, this box usually contains a significant part of the background, making appearance modeling less reliable. Our method can find a large significant part of the foreground, mostly without the background pixels, making it more appropriate for modeling the object appearance.

The running times on a 2.66 GHz computer with 2.0 GB or RAM were between a fraction of a second to less than 2 seconds, depending on the image size, which was in the range from 100 by 100 to 512 by 512.

Acknowledgments

We thank Daniel Cremers for suggesting the name to our shape prior. We also thank the reviewers for numerous comments which helped to improve the paper.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. IJCV 2, 321–331 (1998)
2. Falaco, A.X., Udupa, J., Samarasekara, S., Sharma, S.: User-steered image segmentation paradigms: Live wire and live lane. In: Graphical Models and Image Processing, vol. 60, pp. 233–260 (1998)

3. Mortensen, E.N., Barrett, W.A.: Interactive segmentation with intelligent scissors. In: Graphical Models and Image Processing (GMIP), vol. 60, pp. 349–384 (1998)
4. Osher, S., Sethian, J.: Fronts propagating with curvature dependent speed: Algorithm based on hamilton jacobi formulations. Journal of Computational Physics 79, 12–49 (1988)
5. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation. In: ICCV 2001, vol. I, pp. 105–112 (2001)
6. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive Image Segmentation Using an Adaptive GMMRF Model. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 428–441. Springer, Heidelberg (2004)
7. Boykov, Y., Funka Lea, G.: Graph cuts and efficient n-d image segmentation. International Journal of Computer Vision 69(2), 109–131 (2006)
8. Wu, Z., Leahy, R.: An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. PAMI 15(11), 1101–1113 (1993)
9. Cox, I., Rao, S.B., Zhong, Y.: "ratio regions": A technique for image segmentation. In: ICPR 1996, pp. 557–565 (1996)
10. Shi, J., Malik, J.: Normalized cuts and image segmentation. In: IEEE Conference on Computer Vision(ICCV), pp. 731–737 (1997)
11. Veksler, O.: Image segmentation by nested cuts. In: CVPR 2000, vol. I, pp. 339–344 (2000)
12. Jermyn, I., Ishikawa, H.: Globally optimal regions and boundaries as minimum ratio weight cycles. PAMI 23(10), 1075–1088 (2001)
13. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. IJCV 59(2), 167–181 (2004)
14. Wang, S., Kubota, T., Siskind, J., Wang, J.: Salient closed boundary extraction with ratio contour. PAMI 27(4), 546–561 (2005)
15. Grady, L., Schwartz, E.L.: Isoperimetric graph partitioning for image segmentation. PAMI 28(3), 469–475 (2006)
16. Schoenemann, T., Cremers, D.: Globally optimal image segmentation with an elastic shape prior. In: ICCV 2007, pp. 1–6 (2007)
17. Kolmogorov, V., Zabih, R.: What energy function can be minimized via graph cuts? IEEE Transaction on PAMI 26(2), 147–159 (2004)
18. Leventon, M., Grimson, W., Faugeras, O.: Statistical shape influence in geodesic active contours. In: CVPR 2000, vol. I, pp. 316–323 (2000)
19. Tsai, A., Yezzi Jr., A., Wells III, W., Tempany, C., Tucker, D., Fan, A., Grimson, W., Willsky, A.: Model-based curve evolution technique for image segmentation. In: CVPR, vol. I, pp. 463–468 (2001)
20. Rousson, M., Paragios, N.: Shape priors for level set representations. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2351, pp. 78–92. Springer, Heidelberg (2002)
21. Cremers, D., Osher, S., Soatto, S.: Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. IJCV 69(3), 335–351 (2006)
22. Slabaugh, G., Unal, G.: Graph cuts segmentation using an elliptical shape prior. In: ICIP 2005, vol. II, pp. 1222–1225 (2005)
23. Funka-Lea, G., Boykov, Y., Florin, C., Jolly, M., Moreau-Gobard, R., Ramaraj, R., Rinck, D.: Automatic heart isolation for ct coronary visualization using graph-cuts. In: ISBI 2006, pp. 614–617 (2006)
24. Das, P., Veksler, O., Zavadsky, S., Boykov, Y.: Semiautomatic segmentation with compact shapre prior. In: CRV 2006, pp. 28–36 (2006)
25. Vicente, S., Kolmogorov, V., Rother, C.: Graph cut based image segmentation with connectivity priors. In: CVPR (2008)

26. Freedman, D., Zhang, T.: Interactive graph cut based segmentation with shape priors. In: CVPR, vol. I, pp. 755–762 (2005)
27. Kumar, M., Torr, P., Zisserman, A.: Obj cut. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. I, pp. 18–25 (2005)
28. Ford, L., Fulkerson, D.: Flows in Networks. Princeton University Press, Princeton (1962)
29. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. PAMI 26(9), 1124–1137 (2004)
30. Kolmogorov, V., Boykov, Y.: What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In: ICCV, vol. I, pp. 564–571 (2005)
31. Kolmogorov, V., Boykov, Y., Rother, C.: Applications of parametric maxflow in computer vision. In: ICCV, pp. 1–8 (2007)
32. Kohli, P., Torr, P.: Dynamic graph cuts for efficient inference in markov random fields. PAMI 29(12), 2079–2088 (2007)
33. Cohen, L., Cohen, I.: Finite-element methods for active contour models and balloons for 2-d and 3-d images. PAMI 15(11), 1131–1147 (1993)
34. Appleton, B., Talbot, H.: Globally optimal geodesic active contours. JMIV 23(1), 67–86 (2005)
35. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV, vol. 2, pp. 416–423 (July 2001)
36. Rother, C., Kolmogorov, V., Blake, A.: "grab-cut"- interactive foreground extraction using iterated graph cuts. ACM Transaction on Graphics 23(3), 309–314 (2004)