

Geodesic Graph Cut for Interactive Image Segmentation

Brian L. Price
Brigham Young University
bprice@cs.byu.edu

Bryan Morse
Brigham Young University
morse@byu.edu

Scott Cohen
Adobe Systems
scohen@adobe.com

Abstract

Interactive segmentation is useful for selecting objects of interest in images and continues to be a topic of much study. Methods that grow regions from foreground/background seeds, such as the recent geodesic segmentation approach, avoid the boundary-length bias of graph-cut methods but have their own bias towards minimizing paths to the seeds, resulting in increased sensitivity to seed placement. The lack of edge modeling in geodesic or similar approaches limits their ability to precisely localize object boundaries, something at which graph-cut methods generally excel. This paper presents a method for combining geodesic-distance information with edge information in a graph-cut optimization framework, leveraging the complementary strengths of each. Rather than a fixed combination we use the distinctiveness of the foreground/background color models to predict the effectiveness of the geodesic distance term and adjust the weighting accordingly. We also introduce a spatially varying weighting that decreases the potential for shortcutting in object interiors while transferring greater control to the edge term for better localization near object boundaries. Results show our method is less prone to shortcutting than typical graph cut methods while being less sensitive to seed placement and better at edge localization than geodesic methods. This leads to increased segmentation accuracy and reduced effort on the part of the user.

1. Introduction

Segmentation is one of the most fundamental and well-studied problems in computer vision. Because of the inherent difficulty and ambiguity, many methods use interactive segmentation, which allows a user to supply information regarding the object of interest. Many forms of interaction have been used, ranging from loosely tracing the desired boundary (e.g., [3, 10, 13, 21, 30]) to loosely marking parts of the desired object and/or background (e.g., [2, 5, 6, 11, 18, 22, 23, 27]) to loosely placing a bounding box around the desired object (e.g., [16, 24]). In all forms, the goal is to allow the user to accurately select objects of interest with minimal effort.

We focus here on approaches where the user marks or

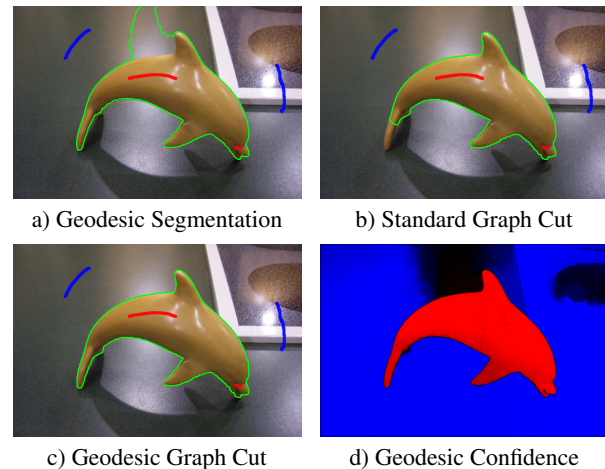


Figure 1. Geodesic graph cut. Without edge information, geodesic segmentation alone can fail in areas where the foreground/background colors are not distinct (a). Graph cut segmentation does a better job of aligning with edges but is susceptible to short-cutting (b). Graph-cut optimization with an automatically tuned geodesic-distance region term leverages the strengths of the approaches and more accurately selects the object (c). In addition to global tuning, spatially adaptive weighting (d) is used to prevent boundary placement in clearly foreground (red) or background (blue) regions, shifting greater control to the edge-finding component when uncertain (black).

“scribbles” on parts of the desired foreground and background regions to seed the segmentation (Figure 1). Such approaches are popular because they generally require less precise input from the user, allowing them to loosely mark broader interior regions instead of more finely tracing near object boundaries, though each approach can sometimes be advantageous. Allowing the user to draw a bounding box [24] is simpler in many cases, though may not provide sufficient control in all cases, in which scribble-based corrections are often employed to refine the results.

Many methods for seeded segmentation expand outward from the seeds to selectively fill the desired region, either explicitly [2, 22, 23, 33] or conceptually [11]. Because these approaches work from the interior of the selected object outwards and do not explicitly consider the object boundary, they are particularly useful for selecting objects with

complex boundaries such as those with long, thin parts. However, because these expansions are monotonic, these approaches suffer from a bias that favors shorter paths back to the seeds. As a result, they can be sensitive to seed placement, as illustrated for geodesic segmentation in Figure 2. Because they lack an explicit edge component, these methods may also fail to accurately localize object boundaries. The stronger the image edges are, the more likely these methods are to make these transitions here, but this is not guaranteed, as illustrated for geodesic segmentation in Figure 3.

The most popular approach to seeded segmentation is currently the graph-cut approach of [5], with numerous proposed variations (e.g., [18]). This method combines explicit edge-finding and region-modeling components, formulated as a weighted combination and optimized by framing the problem as a minimum cut in a weighted graph that partitions foreground seeds from background seeds. (See Section 3 for a more detailed description.) In its original form (and most subsequent forms) the region term uses foreground/background color models inferred from the respective seed pixels. This region/edge combination can be an effective method in many cases, frequently improving on edge- or region-based segmentation methods alone.

However, because the boundary term in graph-cut methods consists of a summation over the boundary of the segmented regions, there is an inherent and well-known bias towards shorter paths, sometimes known as the *length* or *shrinking bias*. (This length bias predates graph-cut methods and is present in earlier least-cost path approaches [21].) This can be especially noticeable when these methods shortcut across the interior of an object to avoid segmenting an appendage as illustrated in Figure 4. This is offset somewhat by the region term, which tries to penalize shortcuts through areas where the labeling is clear from the coloring. However, this is not without tradeoffs—overweighting the region term to compensate for the boundary term’s shrinking bias can result in discontinuous objects with coloring similar to the user’s respective scribbles being incorrectly selected (as also illustrated in Figure 4). This leads to a delicate balance between the weighting of the two terms and the resulting strengths and limitations of each.

Most approaches for otherwise avoiding the shrinking bias in graph-cut and similar approaches involve variations on normalizing the cost of the cut by the size of the resulting object(s). This may be done for grey-level images for which flux may be defined along the boundary of the region [14], but as noted in [29], this does not readily extend to color images. Optimizing general normalized cost functions directly is NP-hard but may be approximated [26, among others]. Alternatively, a subspace of solutions may be explored by varying the relative weighting of the boundary and region terms [15]. (The authors of [15] note, however, that this method is not fast enough for interactive color image segmentation.) More recent work in [25] uses

a curvature-minimizing rather than length-minimizing regularization term to smooth the resulting boundaries while avoiding shortcutting, but this does not use an edge component to localize edges, nor does it run in interactive time.

Because of the complementary strengths and weaknesses of seed-expansion and graph-cut approaches, some have suggested combining them. Work in [28] showed that graph-cuts and random-walkers [11] (a form of seed expansion), along with a new method similar in principle to geodesic segmentation [2], could be placed in a common framework in which the three methods differ by only the (integer) selection of a single parameter, an idea further expanded in [7] by varying a second parameter. They also showed empirically that of these three approaches the new method (analogous to geodesic segmentation) is most sensitive to seed placement while “because of the ‘small cut’ phenomenon, the Graph Cuts segmentation is the least robust to the quantity of seeds.” They went on to suggest a way that the two “could be combined” but did not explore this idea further in that work. More recent work [27] has explored a way to blend the respective strengths of these methods using non-integer selections for the free parameter in [28], determining a suitable parameter selection empirically over a set of sample images with known ground truth.

This paper introduces a new method for interactive segmentation that makes the following three contributions. First, it combines geodesic-distance region information with explicit edge information in a graph-cut optimization framework. This combines the ability of seed-expansion approaches to fill contiguous, coherent regions without regard to boundary length with the ability of edge-based segmentation to accurately localize boundaries. Second, it uses pre-segmentation evaluation of the color models inferred from the user’s seeds to assess the likely effectiveness of the geodesic distance component and weights the terms accordingly. This avoids the tendency for geodesic segmentation to degenerate to simple distance maps when the foreground/background color models are indistinct. Third, it introduces a spatially-varying weighting based on the local confidence of the geodesic component and uses this to further adjust the relative weighting of the terms. This makes cuts even more expensive in object interiors (or exteriors) while transferring more control to the edge-localizing component when near the object’s boundary. The result is a method that adapts both globally and locally to the relative strengths of each approach, providing better boundary placement than geodesic segmentation and stronger region connectivity and less short-cutting than typical graph-cut methods (Figure 1). This leads to less user interaction needed to produce a desired segmentation.

2. Geodesic Segmentation Revisited

Geodesic segmentation [2], like other seed-expansion approaches, can robustly segment long, thin structures with-

out regard to boundary length. By incorporating mixture-based color models inferred from the user's seeds into an inter-pixel distance metric, it can select even multicolored or textured objects. Although we point out here key limitations of this approach in order to address and improve upon it, we refer the reader to the original paper for examples of its successful application.

Geodesic segmentation labels pixels by computing their geodesic distance $D_l(x)$ from the nearest foreground (\mathcal{F}) and background (\mathcal{B}) strokes using

$$D_l(x) = \min_{s \in \Omega_l} d_l(s, x) \quad (1)$$

where Ω_l is the set of seeds with label $l \in \{\mathcal{F}, \mathcal{B}\}$. The pixel is labeled according to the smaller of the two distances.

The geodesic distance from any point to any other according to the color model for the label l is given by

$$d_l(x_0, x_1) = \min_{L_{x_0, x_1}} \int_0^1 |W_l(L_{x_0, x_1}(p)) \cdot \dot{L}_{x_0, x_1}(p)| dp \quad (2)$$

where L_{x_0, x_1} is a path parameterized by $p = [0, 1]$ connecting x_0 to x_1 respectively, and $W_l(x)$ gives the geodesic weight according to the model l . [2] defines $W_l(x) = \nabla P_l(C(x))$, where

$$P_l(c) = \frac{Pr(c|l)}{Pr(c|\mathcal{F}) + Pr(c|\mathcal{B})}, \quad (3)$$

$C(x_i)$ is the color at x_i , and $Pr(c|l)$ is the probability of the color c given by the color model generated from pixels Ω_l .

Geodesic segmentation performs best when the geodesic distance between neighboring pixels inside of (or outside of) the desired object is small relative to the geodesic distance between neighboring pixels across the object boundary. This requires an accurate foreground/background color model that is consistent in assigning probabilities to the pixels. However, even small errors or variations in the probabilities can accumulate over the geodesic paths and lead to incorrect results in two keys ways:

1. If the color models are not distinct, the probabilities $P_l(c)$ may be highly unstable from one pixel to the next due to unavoidable image noise. This neutralizes the color-based distance metric and in the limit causes geodesic segmentation to degenerate to simple (and noisy) distance maps. This causes the segmentation to be highly sensitive to seed placement, as illustrated in Figure 2.
2. Even for more distinct color models, the lack of an explicit edge-finding component can cause geodesic segmentation to come close to but not precisely localize object boundaries (as in Figure 3 for a simple 1-D example). As the transition in the geodesic distances increases the method is more likely to place the boundary correctly, but with softer edges or with even modest noise it can sometimes fail to do so.

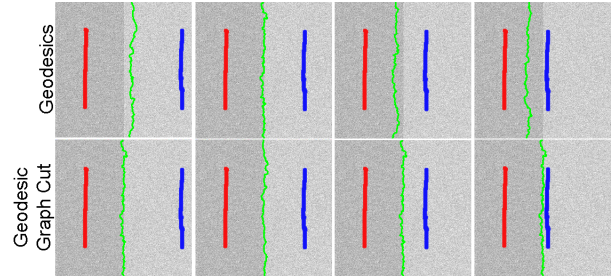


Figure 2. Sensitivity of geodesic segmentation to seed placement. As the background stroke (blue) is translated, the object boundary computed by geodesic segmentation shifts also (top row), while geodesic graph cut (and other graph cut approaches) give a more consistent result (bottom row).

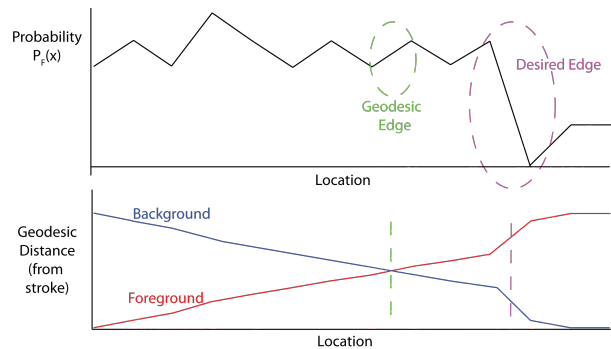


Figure 3. Why geodesic segmentation can miss edges. Because of noise in the probability along the geodesic paths (top), the geodesic boundary (green) between the user's foreground stroke and background stroke misses the true image edge (magenta).

The methods in this paper address these two limitations respectively by 1) globally weighting geodesic-distance component by assessing the relative distinctiveness of the foreground/background color models, and 2) transferring relative control from the geodesic-segmentation component to more explicit edge-finding near object boundaries.

We note here one related approach of interest [8], in which a number of geodesic distance transforms are generated by varying the parameters to generate multiple candidate solutions. The candidate minimizing a cost function that includes an edge-finding component (similar to that used in graph-cut approaches) is then selected as the final result. This method differs from that proposed here in that the region and edge information that graph cut uses are not used while generating geodesic candidates. Because of this, the space of candidate solutions may only approximate the optimal solution.

3. Graph Cut Segmentation Revisited

Graph cut segmentation [5] seeks to minimize a cost function of the form

$$E(\mathcal{L}) = \sum_{x_i \in P} R_{\mathcal{L}_i}(x_i) + \lambda \sum_{(x_i, x_j) \in N} B(x_i, x_j) |\mathcal{L}_i - \mathcal{L}_j| \quad (4)$$

where $\mathcal{L} = (\mathcal{L}_i)$ is a binary vector of labels and \mathcal{L}_i is the label \mathcal{F} or \mathcal{B} for pixel x_i , $R_l(x_i)$ is a region cost term based on the label l , $B(x_i, x_j)$ is a boundary cost-term, λ is a relative weighting of R and B , P is the set of pixels in the image, and N is the set of pairs of neighboring pixels. The terms R and B have been defined in various ways by different researchers. Generally B corresponds to a measure of the similarity between the colors of adjacent pixels and R is based on color models of the foreground and background.

Graph cut methods minimize Eq. 4 by casting the problem as a graph-partitioning one and using the mincut/max-flow graph algorithm, where boundary costs are assigned to graph edges between pixels and region costs are assigned to edges that connect pixel nodes to the source and sink nodes [5].

Graph-cut methods perform well over a variety of images. Because both region and boundary information are explicitly captured in the algorithm, they are capable of both selecting objects consistent with region information and placing object boundaries on image edges. Many variations have sought to improve on this approach, including using watershed regions as primitives in order to reduce the size of the graph and accelerate the computation [18], using tensor-based region terms to model texture [20], and iteratively alternating segmentation and model updating to converge to the solution [24]. Because it can easily operate on multi-dimensional images, the graph-cut approach has also been applied to videos [17,31] and image volumes [1,4,19]. It is important to note that the term “graph cut segmentation” has grown to encompass a wide variety of approaches that minimize cost functions of the form in Eq. 4 by framing the problem as a graph-based min-cut/max-flow one. (See [4] for an excellent discussion of the variety of approaches for which graph-cut optimization has been used.)

As noted in Section 1, a key weakness of graph-cut approaches is that the boundary term in Eq. 4 causes an inherent shrinking bias toward shorter segmentation boundaries. This can be especially noticeable when the algorithm short-cuts across the interior of an object to avoid segmenting an appendage as illustrated in Figure 4. The ideal boundary of the object contains the segment B_2 , but since the segment B_1 is so much shorter, the cost of cutting all the links along B_1 may be less than the cost of cutting the links along B_2 , even if no individual link in B_1 is cheaper than in B_2 . This leaves the region A_{12} out of the selection.

Using this intuition, short-cutting may be prevented by increasing the cost of B_1 relative to B_2 . However, increasing the edge sensitivity may cause even weak gradients to

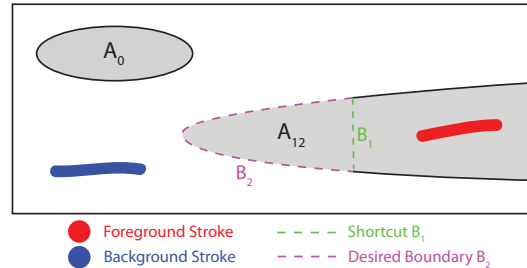


Figure 4. Shortcuts. Graph-cut methods may short-cut across the desired object along B_1 instead of following the true edge B_2 because less cost is accrued transversing the shorter path.

become attractive options. A better strategy is to increase the incurred region cost of the shortcut by increasing the sensitivity of the color model or by increasing its weight relative to the boundary terms by decreasing λ . However, this can have an ill effect when using a global color-similarity model as is common with graph-cut methods. Other background objects with properties (region A_0 in Figure 4) similar to the foreground user stroke are more likely to be incorrectly segmented as foreground in such cases. Problems with short-cutting and selection of disconnected unseeded regions can be reduced by allowing users to explicitly specify that certain regions should stay connected or disconnected [29], but this requires either prior knowledge or further user interaction.

4. Geodesic Graph Cut

As discussed in Section 2, in cases where the color models inferred from the user's strokes are indistinct, geodesic segmentation can be improved by the inclusion of explicit edge information to encourage placement of selection boundaries on edges in the image and allow the user more freedom in placing strokes. In cases where the color models are more distinct, though, the edge information (with its inherent shrinking bias) is not as necessary. The region term alone can often carry the segmentation in such cases, but as discussed in Section 3 global color models without spatial locality information can often select disjoint regions. The use of geodesic distance rather than simple color-similarity alone can avoid this. This section presents how geodesic distances and edge information can be combined in a graph-cut optimization framework, and then presents a way to use the predicted classification accuracy from the inferred color models to automatically tune the tradeoff between the strengths and weaknesses of the two.

4.1. Using Geodesic Distance as a Unary Term

We formulate our algorithm in simplest form as a graph-cut problem using a normalized form of geodesic distance as one of the unary region terms. Using Eq. 4 and minimizing it using graph cut, we compute the unary region term as

follows:

$$R_l(x_i) = s_l(x_i) + M_l(x_i) + G_l(x_i) \quad (5)$$

where $M_l(x_i)$ is based on a global color model as is often used for graph-cut segmentation, $G_l(x_i)$ is based on geodesic distance, and

$$s_l(x_i) = \begin{cases} \infty & \text{if } x_i \in \Omega_{\bar{l}} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

indicates the presence of a user stroke where \bar{l} is the label opposite l (i.e. if $l = \mathcal{F}$, then $\bar{l} = \mathcal{B}$).

We use the Fast Gauss Transform [32] to compute the foreground/background color models $P_l(c)$ (see Eq. 3) for both global similarity and geodesic distances. $M_l(x_i)$ is computed by

$$M_l(x_i) = P_{\bar{l}}(C(x_i)). \quad (7)$$

$G_l(x_i)$ is computed by normalizing the relative foreground/background geodesic distances (see Equation 1):

$$G_l(x_i) = \frac{D_l(x_i)}{D_{\mathcal{F}}(x_i) + D_{\mathcal{B}}(x_i)}. \quad (8)$$

and using the efficient method in [2] to compute the distances $D_l(x_i)$.

For the boundary term we use [18]:

$$B(x_i, x_j) = \frac{1}{1 + \|C(x_i) - C(x_j)\|^2} \quad (9)$$

where $C(x) \in [0, 255]$.

4.2. Global Weighting Based on Color Model Error

The simple form (Eq. 4) in Section 4.1 can give good results in many cases, generally performing better than either geodesic or graph-cut segmentation alone. However, we would like to allow it to provide greater weight to the geodesic-based unary term in cases where this method is known to perform well, specifically when the foreground/background color distributions are well-separated. This increased reliance on geodesic distance for the region term serves to reduce the potential for short-cutting due to the boundary term. But caution must be exercised with this, because over-reliance on the geodesic component can cause increased sensitivity to seed placement when the color models are not distinct.

To allow for global weighting of the relative importance of the region and boundary components, we modify Eq. 4 as follows:

$$E(\mathcal{L}) = \lambda_R \sum_{x_i \in P} R_{\mathcal{L}_i}(x_i) + \lambda_B \sum_{(x_i, x_j) \in N} B(x_i, x_j) |\mathcal{L}_i - \mathcal{L}_j| \quad (10)$$

Although we could fold the separate region (λ_R) and boundary (λ_B) weights into a single weight, we choose to keep

them separate to make their respective purposes clearer. The boundary weight λ_B serves the role of the traditional fixed region/boundary weighting in graph cut methods, and we adjust it to individual images by considering only the size of the image (due to the disproportionate scaling of an object's area (unary term) and perimeter (boundary term)). The region weight λ_R is the relative weighting of the geodesic distance and other region components. While the user could tune λ_R manually, this would require excessive tweaking and is undesirable; instead, we want to automatically tune this parameter on a per-image basis by predicting the segmentation performance of the geodesic distance term.

To do this, we consider that Eq. 3 is the posterior probability of a pixel with color c belonging to foreground (\mathcal{F}) or background (\mathcal{B}) respectively, assuming equal priors. As such, it is essentially functioning as a simple Bayesian classifier, the error in which can be estimated by (using the notation of Eq. 7)

$$\varepsilon = \frac{1}{2} \left[\frac{\sum_{x \in \mathcal{F}} P_{\mathcal{B}}(C(x))}{|\Omega_{\mathcal{F}}|} + \frac{\sum_{x \in \mathcal{B}} P_{\mathcal{F}}(C(x))}{|\Omega_{\mathcal{B}}|} \right] \quad (11)$$

When there is no error ($\varepsilon = 0$), we would like to give the color-based terms (M and G) full weight, and when the color models become indistinct ($\varepsilon \geq 0.5$), we want to give them no weight:

$$\lambda_R = \begin{cases} 1 - 2\varepsilon & \text{if } \varepsilon < 0.5 \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

4.3. Local Weighting Based on Geodesic Confidence

Globally adjusting the relative weighting of the region and boundary terms on a per-image basis can help automatically tune the method to different image types, but it does not account for the properties of different local areas. We further weight the geodesic and boundary terms based on the local confidence $u(x)$ of the geodesic components:

$$u(x_i) = \left| \frac{D_{\mathcal{F}}(x_i) - D_{\mathcal{B}}(x_i)}{D_{\mathcal{F}}(x_i) + D_{\mathcal{B}}(x_i)} \right|^{\gamma} \quad (13)$$

where empirically we have found $\gamma = 2$ to 2.5 to work well. (For the results in Section 5, we use a value of $\gamma = 2.5$.)

We redefine the region terms to weight the geodesic component by $u(x_i)$:

$$R_l(x_i) = s_l(x_i) + M_l(x_i) + u(x_i)G_l(x_i) \quad (14)$$

This maintains the weight of the geodesic distance term when relatively certain that the pixel x_i is clearly in the object's interior or exterior ($u(x_i)$ close to 1) and decreases it near where geodesic segmentation would place boundaries ($u(x_i)$ close to 0).

We also correspondingly spatially adapt the weighting of the boundary costs based on $u(x)$ as follows:

$$B(x_i, x_j) = \frac{1 + (u(x_i) + u(x_j))/2}{1 + \|C(x_i) - C(x_j)\|^2} \quad (15)$$

Note that when the average geodesic certainty of the two pixels is high, this suggests an object interior/exterior, and the cost of placing a cut here is further increased. When this geodesic confidence is low, this suggests that geodesic segmentation alone would consider this to be near a boundary, and we reduce the effect of the geodesic component, shifting control to the more accurate edge-finding term.

The net effect of this spatially adaptive weighting is to both *increase* the relative weighting of the unary geodesic distance term and *increase* the cost of a boundary cut in what are clearly interior/exterior regions, while both *decreasing* the relative weighting of the unary geodesic term and *decreasing* the cost of a boundary cut in areas where we want to more accurately localize the object boundary.

5. Results

Geodesic graph cut with automatic tuning and spatial adaptation works well both in cases suitable for geodesic segmentation and in cases suitable for standard graph cut methods, in many cases outperforming both. While accuracy is an essential element of interactive segmentation, so too is the minimization of user interaction required to achieve that level of accuracy. This section demonstrates the accuracy of geodesic graph cut, and the interaction required for these examples is shown in recorded videos¹. The time for our algorithm on these images ranged from 0.2–2.6 seconds for image sizes from 256×256 to 720×480 , with most computations requiring approximately 1 second or less. *In all cases the unary term weighting (λ_R) and the spatially adaptive weights ($u(x)$) are set automatically by the method, with no per-image manual tuning.*

In Figure 1, geodesic segmentation fails to segment along the dolphin's back due to the specular reflection on the table. Graph cut, because of its explicit edge term, can better segment along the back but shortcuts across the tail. Geodesic graph cut leverages the strengths of each approach and correctly segments both areas. While additional strokes could of course correct either the graph-cut or geodesic segmentation, this increases the required user interaction.

Figure 5 shows examples where geodesic graph cut segmentation automatically adjusts to the distinct color models, exploiting the geodesic distance term to avoid shortcutting. For these images, geodesic segmentation performs comparably to the results shown here for our method.

Figure 6 shows similar examples for images whose color models are less distinct. In these cases, our method recognizes the error in the color models and automatically adjusts to rely less heavily on geodesic distances. In the pyramid (top) and candy (middle) examples, the foreground and background color models overlap considerably. Without distinct color models, geodesic segmentation alone fails noticeably. In particular, the candy (middle) result demonstrates the way geodesic distance can degenerate to noisy

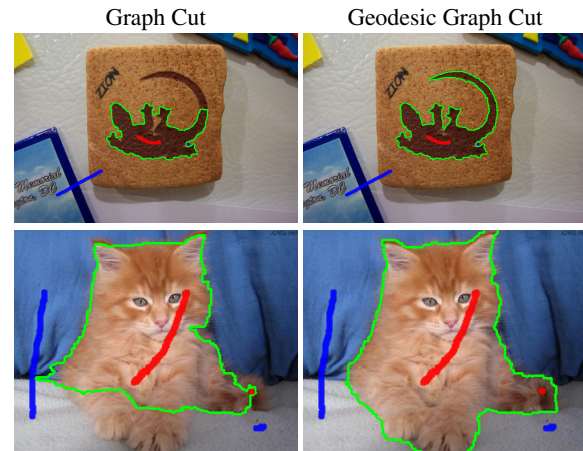


Figure 5. Examples with distinct color models. For these images geodesic graph cut segmentation automatically relies more on geodesic distances ($\lambda_R = 0.97, 0.99$ respectively), avoiding short-cutting common to non-adaptive graph-cut methods.

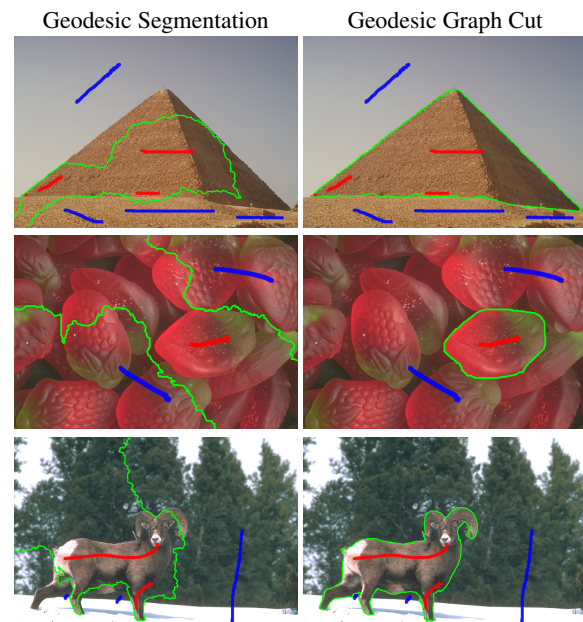


Figure 6. Examples with less-distinct color models. In these cases geodesic graph cut cannot rely on the geodesic distances and automatically adjusts to rely more on explicit edges or a combination of color models and edges ($\lambda_R = 0.40, 0.22, 0.65$ respectively).

distance maps when the colors are not distinct. In the ram (lower) example, the color models are more distinct but are still mixed enough to cause geodesic segmentation mistakenly select the part of the background away from the user-placed background seeds. The results of graph cut segmentation for these examples are comparable to our method.

Figure 7 show examples where our method outperforms both geodesic and graph-cut segmentation. For the rolling-pin example (top), because of the similarity between the foreground and background colors, geodesic segmentation again mistakenly selects part of the background away from

¹<http://vision.cs.byu.edu/papers/ggc>

the seeds. Graph cut segmentation avoids this problem but again exhibits shortcutting. Our method places only a moderate geodesic distance weight for this image ($\lambda_R = 0.72$), avoiding the problems exhibited by geodesic segmentation alone while using sufficient weighting of this term to avoid region shortcutting. For the elephant example (bottom), geodesic segmentation fails to segment along the top of the elephant's back due to the similar background even though there is still an edge there, while graph cut segmentation shortcuts the trunk and part of the back. Geodesic graph cut corrects both of these problems for this image.

To quantitatively evaluate the accuracy of the segmentations produced by geodesic graph cut, we applied it to the Microsoft GrabCut dataset [3]. As noted in [9], this dataset is not well suited to evaluating interactive scribble-based segmentation because it assumes the user loosely traces the contour of the desired object. As such, it provides far more seeds than are typically provided with interactive scribbles, and the seeds are more uniformly placed on either side of the boundary. We believe that interactive scribble-based methods cannot be evaluated with static seeds—once the user places the first scribble the resulting scribbles are dependent on the segmentation result from that and each successive one. But, as also noted in [9], this is the only evaluation database to provide seeds, and we compute our results on this dataset for comparison (Table 1).

Over all 50 images in the database, geodesic graph cut averaged 4.8% error, better than either geodesic segmentation (6.8%) or standard graph-cut segmentation (6.7%) individually. This error was also lower than that of any of the other compared-to methods that do not have a spatial position bias, either explicitly [16] or implicitly [12].² To our knowledge, this is the lowest error rate reported for this dataset by a scribble-based selection method.

For 48 of the 50 images, geodesic graph cut outperformed either graph cut (43 of 50) or geodesic segmentation (39 of 50) alone, typically performing at a level near the better of the two methods for each image. For 34 of the 50 images, it outperformed both.

We also used this dataset with the skeleton-based initialization suggested in [27] as provided by its authors. This provides fewer seeds overall but tends to place more seeds in object protrusions. Graph cut segmentation had an error rate of 6.3% with this form of initialization, while geodesic segmentation had an error rate of 10% and geodesic graph cut had an error rate of 3.6%.

From our observations, geodesic graph cut usually does as well or better than the better of the two individual meth-

²As noted in [9], the adaptive thresholding method of [12] has a strong bias towards placing an object boundary in the middle of the uncertainty band of the trimap. This is because it uses an adaptive window that grows to “include at least one $\alpha = 0$ pixel and one $\alpha = 1$ pixel.” Due to the nature of the GrabCut database, which uses an uncertainty band centered close to the actual boundary, this causes this method to have artificially low error. Although [9] use and report this method for their method and [11], they add this specific disclaimer on the reported error rates.

Segmentation Model	Error Rate
GMMRF [3]	7.9%
Geodesic Segmentation [2]	6.8%
Graph Cut (as reported by [16])	6.7%
Random Walker (s=2) [11]	5.4%
Segmentation by Transduction [9]	5.4%
Geodesic Graph Cut (proposed method)	4.8%
Guan and Qiu [12] with AT*	4.6%
Random Walker (s=2) [11] with AT [12]*	3.3%
Segmentation by Transduction [9] with AT [12]*	3.3%
GrabCut-GC (as reported by [16])	5.9%
Bounding Box Prior (LP-Pinpoint) [16]	5.0%
Bounding Box Prior (GrabCut-Pinpoint) [16]	3.7%

Table 1. Quantitative comparison using the Microsoft GrabCut database [3]. Error rates reported here are either computed by us (our method, [2]), reported by the method's authors ([3, 12]), or were previously reported by [9] or [16]. The first nine were initialized using the “Lasso” form of the trimap provided by the database. The adaptive thresholding (AT) method of [12] is biased towards the middle of the trimap's uncertainty band and is artificially favored by this form of initialization. The lower three used the corresponding bounding box initializations, with the last two using this box as a prior on the spatial extent of the object.

ods. It struggles on inputs for which both individual methods have difficulties. This typically happens when the foreground and background color models are overlapping and either there are weak edges or highly textured regions around the object boundary. Another less severe problem we have observed is that some long thin structures are sometimes still short-cutted near the tip, although our adaptive weighting usually decreases the size of the short-cutted area and thus requires less additional effort from the user to correct than typical graph cut. This problem is most likely to occur when there are strong edges across or texture within the long thin structure and when the color models overlap enough that our algorithm cannot rely on the geodesic information to entirely prevent a shortcut.

6. Conclusion

This paper has presented a way to incorporate both geodesic-distance region information and explicit edge information together in the popular graph-cut optimization framework in a way that leverages the strengths of each. Rather than a simple fixed combination, the method tries to best leverage the respective strengths of the two approaches by adaptively tuning their combination based on pre-segmentation assessment of the classification performance of the color models inferred from the user's input. When the image's foreground/background color models as inferred from the user's marked seeds are distinct, greater weight is given to the geodesic-distance component in order to provide greater region coherence and avoid boundary short-cutting. As the color models become less distinct, the geodesic-distance approach becomes increasingly unreliable and is weighted less accordingly. In addition, a spa-

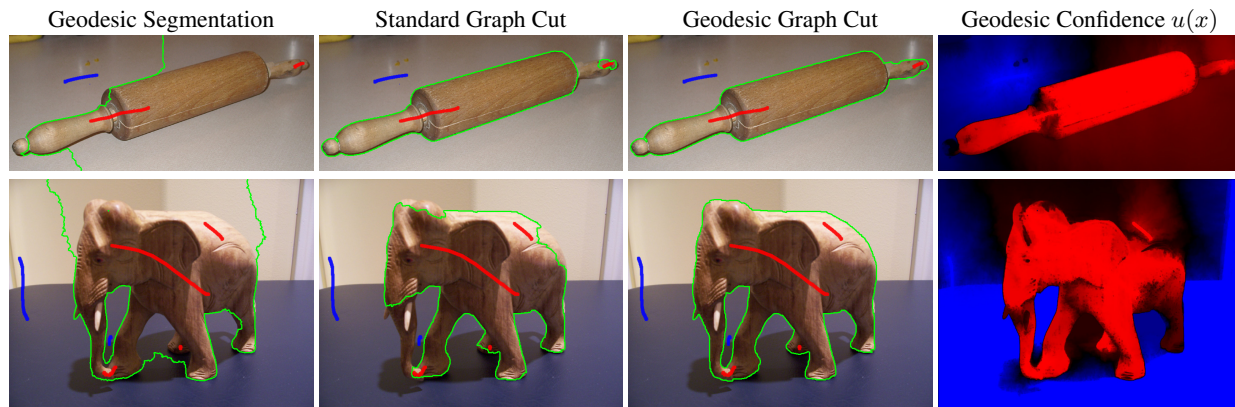


Figure 7. Examples for which geodesic graph cut segmentation outperforms both geodesics segmentation and standard graph cut.

tially adaptive weighting is introduced that makes boundary short-cutting more expensive in object interiors or exteriors while transferring greater control to the edge-finding component to better localize edges near object boundaries. Results demonstrate that geodesic graph cut is able to segment objects in a variety of images, generally performing as well as the better of these two methods for each image, and often outperforming both methods.

References

- [1] C. Armstrong, B. Price, and W. Barrett. Interactive segmentation of image volumes with live surface. *Computers and Graphics*, 31(2), April 2007.
- [2] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. *IEEE ICCV 2007*, pages 1–8, 2007.
- [3] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, pages 428–441, May 2004.
- [4] Y. Boykov and G. Funka-Lea. Graph cuts and efficient N-D image segmentation. *IJCV*, 70(2):109–131, 2006.
- [5] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *IEEE ICCV*, pages 105–112, 2001.
- [6] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *IEEE Trans. PAMI*, volume 26, pages 1124–1137, September 2004.
- [7] C. Couprie, L. Grady, L. Najman, and H. Talbot. Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In *IEEE ICCV*, 2009.
- [8] A. Criminisi, T. Sharp, and A. Blake. GeoS: Geodesic image segmentation. In *ECCV*, 2008.
- [9] O. Duchenne, J.-Y. Audibert, R. Keriven, J. Ponce, and F. Segonne. Segmentation by transduction. In *IEEE CVPR*, June 2008.
- [10] M. Gleicher. Image snapping. *ACM SIGGRAPH*, pages 183–190, 1995.
- [11] L. Grady. Random walks for image segmentation. *IEEE Trans. PAMI*, 28(11):1768–1783, November 2006.
- [12] J. Guan and G. Qiu. Interactive image segmentation using optimization with statistical priors. In *ECCV International Workshop on The Representation and Use of Prior Knowledge in Vision*, 2006.
- [13] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *IEEE ICCV*, pages 259–268, 1987.
- [14] V. Kolmogorov and Y. Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *IEEE ICCV*, volume 1, pages 564–571, Oct. 2005.
- [15] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. *IEEE ICCV*, 2007.
- [16] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *IEEE ICCV*, 2009.
- [17] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. *ACM Trans. Graph.*, 24(3):595–600, 2005.
- [18] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. In *ACM SIGGRAPH 2004*, pages 303–308, 2004.
- [19] H. Lombaert, Y. Sun, L. Grady, and C. Xu. A multilevel banded graph cuts method for fast image segmentation. In *IEEE ICCV*, pages 259–265, 2005.
- [20] J. Malcolm, Y. Rathi, and A. Tannenbaum. A graph cut approach to image segmentation in tensor space. In *IEEE CVPR*, June 2007.
- [21] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *ACM SIGGRAPH 1995*, pages 191–198, 1995.
- [22] A. Protiere and G. Sapiro. Interactive image segmentation via adaptive weighted distances. *IEEE Trans. Imag. Proc.*, 16(4):1046–1057, 2007.
- [23] L. J. Reese and W. A. Barrett. Image editing with intelligent paint. In *Eurographics 2002*, volume 21, pages 714–724, 2002.
- [24] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004*, pages 309–314, 2004.
- [25] T. Schoenemann, F. Kahl, and D. Cremers. Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation. In *IEEE ICCV*, 2009.
- [26] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE CVPR*, pages 731–737, 1997.
- [27] D. Singaraju, L. Grady, and R. Vidal. P-Brush: Continuous valued MRFs with normed pairwise distributions for image segmentation. In *IEEE CVPR*, pages 1303–1310, 2009.
- [28] A. K. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *IEEE ICCV*, 2007.
- [29] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *IEEE CVPR*, 2008.
- [30] J. Wang, M. Agrawala, and M. F. Cohen. Soft scissors: an interactive tool for realtime high quality matting. *ACM Trans. Graph.*, 26(3):9, 2007.
- [31] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. *ACM Trans. Graph.*, 24(3):585–594, 2005.
- [32] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved fast gauss transform and efficient kernel density estimation. In *IEEE ICCV*, pages 464–471, 2003.
- [33] S. Zucker. Region growing: Childhood and adolescence. *Computer graphics and image processing*, 5(3):382–399, 1976.