

A Linear-Time Approach for Image Segmentation Using Graph-Cut Measures^{*}

Alexandre X. Falcão, Paulo A.V. Miranda, and Anderson Rocha

Institute of Computing – State University of Campinas (UNICAMP)
CEP 13084-851, Campinas, SP – Brazil
{afalcao, paulo.miranda, anderson.rocha}@ic.unicamp.br

Abstract. Image segmentation using graph cuts have become very popular in the last years. These methods are computationally expensive, even with hard constraints (seed pixels). We present a solution that runs in time proportional to the number of pixels. Our method computes an ordered region growing from a set of seeds inside the object, where the *propagation order* of each pixel is proportional to the cost of an *optimum path* in the image graph from the seed set to that pixel. Each pixel defines a region which includes it and all pixels with lower propagation order. The boundary of each region is a possible cut boundary, whose cut measure is also computed and assigned to the corresponding pixel on-the-fly. The object is obtained by selecting the pixel with minimum-cut measure and all pixels within its respective cut boundary. Approaches for graph-cut segmentation usually assume that the desired cut is a global minimum. We show that this can be only verified within a reduced search space under certain hard constraints. We present and evaluate our method with three cut measures: normalized cut, mean cut and an energy function.

1 Introduction

We consider the problem of segmenting an image in object and background by graph-cut measures. The image is interpreted as an undirected graph, whose nodes are the image pixels and whose arcs are weighted and defined by an adjacency relation between pixels. We wish to assign weights to the arcs and define an objective function (a graph-cut measure), such that its minimum corresponds to the desired segmentation (i.e., a *cut boundary* whose arcs connect the nodes between object and background).

Approaches for graph-cut segmentation usually aim at assigning higher weights to arcs inside object and background, and lower weights otherwise. Their objective functions measure some global property of the object's boundary from this weight assignment. Wu and Leahy [1] were the first to introduce a solution for graph cut using as measure the sum of the arc weights in the cut boundary.

^{*} The authors thank the financial support of FAPESP (Procs. 03/13424-1, 05/58103-3, and 05/59808-0) and CNPq (Proc. 302427/04-0).

Their cut measure has the bias toward small boundaries and other objective functions, such as *average cut* [2], *mean cut* [3], *average association* [4], *normalized cut* [5], *ratio cut* [6], and *energy functions* [7, 8, 9] have been proposed to circumvent this problem.

The problem of finding a minimum of an objective function through graph cut is NP-hard for a generic graph and very often solutions require hard constraints. Heuristic solutions have been proposed in polynomial time [10], but with poor computational performance, and the results are sometimes far from the desired segmentation [11]. Indeed we have verified that even in a reduced search space that includes the desired cut, it does not always correspond to the minimum cut. This suggests that hard constraints are really needed in practice. For example, two terminal nodes (*source* and *sink*) can be added to the image graph, representing object and background respectively [7, 8]. Additionally to the weight assignment between pixels, this approach aims at assigning lower arc-weights between source and object pixels, higher arc-weights between sink and object pixels, lower arc-weights between sink and background pixels, and higher arc-weights between source and background pixels. A min-cut/max-flow algorithm from source to sink [12, 13] is used to compute the minimum-cut boundary. If the method fails the detection of the desired boundary, the user can impose the arc weights with source and sink by selecting seed pixels inside and outside the object [7]. The running time of these algorithms is still polynomial [8] (i.e., typically $O(mn^2)$ where m is the number of arcs and n is the number of *nodes*).

We present a solution that runs in linear time (i.e., in $O(n)$). Our method computes an ordered region growing from a set of seeds inside the object, where the *propagation order* of each pixel is proportional to the cost of an *optimum path* in the image graph from the seed set to that pixel. Each pixel defines a region which includes it and all pixels with lower propagation order. The boundary of each region is a possible cut boundary, whose cut measure is also computed and assigned to the corresponding pixel on-the-fly. The object is obtained by selecting the pixel with minimum-cut measure and all pixels within its respective cut boundary.

Our method essentially reduces the search space by ordering possible cuts from inside to outside the object. It requires lower arc weights across the object's boundary than inside it in order to include the desired cut in the reduced space. When this weight assignment is not achieved, the method can still work by adding more seeds. A problem, however, has been the sensitivity of some cut measures with respect to the heterogeneity (arc weights) outside the object. We evaluate this aspect with normalized cut [5], mean cut [3], and an energy function [7, 9].

We could use the same adjacency relation, weight assignment between pixels, and energy function to compare our method with the one by Boykov and Jolly [7] in the context of interactive segmentation. However, under the same conditions, both methods are likely to produce similar results except to the fact that our algorithm is more efficient. Instead of that, we prefer to verify the accuracy of our approach in a real application that represents the worst case for the aforementioned cut measures.

Section 2 presents the image graph, weight functions, and cut measures used in this paper. We present our method for the 2D case, but its extension to 3D is straightforward. The method and its algorithm are presented in Section 3. Section 4 evaluates it using three cut measures and our conclusions are stated in Section 5.

2 Image Graphs and Cut Measures

Consider an undirected graph where the pixels are the nodes and the arcs are defined by an irreflexive 4-adjacency relation between pixels. There are many ways of exploiting image features to compute arc weights [5, 7, 14]. We suggest to assign a membership value for each pixel with respect to the object based on image features (texture, color, gradients), which may be different depending on the application. The idea is to improve the weight assignment by reducing inhomogeneities inside the object.

Let \mathbf{x}_p be a feature vector computed at a given pixel p ; μ_p and Σ_p be mean and covariance matrices of the feature vectors \mathbf{x}_q computed at all pixels q within an adjacency radius around p ; and T be a set of training pixels, selected in object regions that have different image features. For a given pixel $s \in T$, we compute a membership value $R_s(p)$ for every image pixel p .

$$R_s(p) = \exp \left(-\frac{1}{2d} (\mathbf{x}_p - \mu_s)^t \Sigma_s^{-1} (\mathbf{x}_p - \mu_s) \right) \quad (1)$$

where $d > 1$ takes into account the absence of statistical information (e.g., we use $d = 10$). We also set a distinct adjacency radius for each pixel $s \in T$, making it as largest as possible, in order to compute the best estimation for μ_s and Σ_s inside the object region that includes s . A region map R is obtained as

$$R(p) = \max_{\forall s \in T} \{R_s(p)\}. \quad (2)$$

We also apply a median filter on R to make it more homogeneous. The weight $w(p, q)$ for any arc (p, q) is given by

$$w(p, q) = \exp \left(-\frac{(R(p) - R(q))^2}{2d} \right). \quad (3)$$

Figures 1a–c show three original images, where the training pixels and their adjacency radii are indicated by circles. The respective region maps are shown in Figures 1d–f. We used two normalized attributes within $[0, 1]$ for the feature vectors of Equation 1 in each case: brightness and gradient magnitude (Figure 1a); and red and green values (Figures 1b and 1c). Note that the choice of these attributes is a separate problem, and the segmentation can not be generally solved by thresholding the region map and extracting the binary components, which are hard-connected to internal seeds (e.g., Figure 1d).

Due to the heterogeneity of the background, it is very difficult to obtain higher arc weights outside the object. This affects some graph-cut measures more than

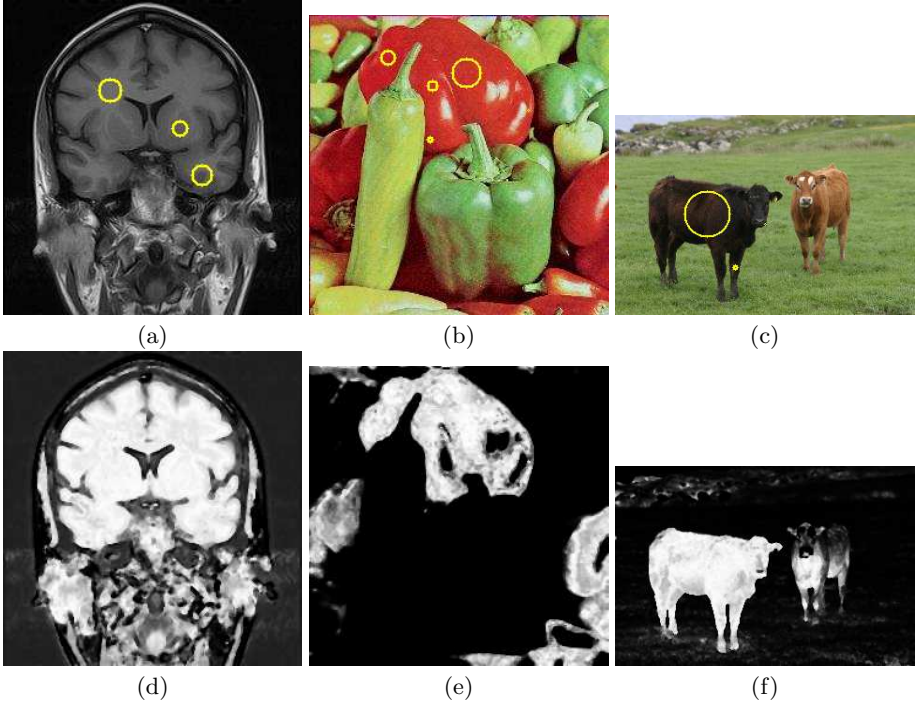


Fig. 1. (a) A Magnetic Resonance (MR) image of a brain with three training pixels (the circles indicate their adjacency radii). (b) A colored image of peppers with four training pixels. (c) A colored image of two cows with two training pixels. (d–f) The respective region maps of (a),(b), and (c).

others. Therefore, we will consider the normalized cut [5], mean cut [3], and an energy function [7, 9] to evaluate this aspect in Section 4.

Let I and E be the interior and exterior of a cut boundary IE , which consists of a set of arcs (p, q) where $p \in I$ and $q \in E$. The normalized cut is defined as

$$\frac{cut(I, E)}{asso(I) + cut(I, E)} + \frac{cut(I, E)}{asso(E) + cut(I, E)} \quad (4)$$

where

$$cut(I, E) = \sum_{\forall (p, q) | p \in I, q \in E} w(p, q) \quad (5)$$

$$asso(I) = \sum_{\forall (p, q) | p \in I, q \in I} w(p, q) \quad (6)$$

$$asso(E) = \sum_{\forall (p, q) | p \in E, q \in E} w(p, q). \quad (7)$$

The mean cut is defined as

$$\frac{cut(I, E)}{|IE|} \quad (8)$$

where $|IE|$ is the number of arcs in IE .

We have chosen an energy function similar to that proposed in [7] and consistent with the general formulation described in [9].

$$\lambda \left(\sum_{\forall p \in I} (1 - Ro(p)) + \sum_{\forall q \in E} (1 - Rb(q)) \right) + cut(I, E) \quad (9)$$

where Ro and Rb are region maps computed by Equation 2 using training pixels inside object and background, respectively; and $\lambda > 0$ represents the importance of the first term (a normalization factor) with respect to the second one.

3 Region Growing by Ordered Propagation with Graph Cut

Let $A_4(p)$ be the set of the 4-adjacent pixels of p , excluding it. A *path* π in the image graph (Section 2) is a sequence $\langle p_1, p_2, \dots, p_n \rangle$, such that $p_{i+1} \in A_4(p_i)$, for $i = 1, 2, \dots, n - 1$.

First, assume that Equation 3 assigns lower arc weights across the object's boundary than inside it. These arc weights are inversely proportional to the dissimilarities $\delta(p, q)$ between 4-adjacent pixels of the region map. For a given set S of internal seeds, we define the *cost* c of a path π as:

$$c(\pi) = \begin{cases} \max_{i=1,2,\dots,n-1} \{\delta(p_i, p_{i+1})\} & \text{if } p_1 \in S \\ +\infty & \text{otherwise} \end{cases} \quad (10)$$

where $\delta(p, q) = K(1 - w(p, q))$ for an integer K that represents the maximum dissimilarity between pixels (e.g., $K = 1023$). The reason for using an integer K will be explained later.

A path from a seed set S to a pixel p is *optimum* when its cost is minimum as compared to the cost of any other path from S to p . Under the above conditions, it is enough to have a single seed in S (we will discuss later the case of multiple seeds), and the optimum paths from S to object pixels will have costs strictly less than the costs of optimum paths with terminus at background pixels. The object could be detected by thresholding the costs of the optimum paths from S , but this threshold is unknown. Thus, we grow a region from S by aggregating one adjacent pixel at a time in order proportional to the cost of an optimum path from S to that pixel; such that the object pixels will be aggregated before the background pixels.

Each pixel defines a region which includes it and all pixels with lower propagation order. The boundary of each region is a possible cut boundary, whose cut measure is also computed and assigned to the corresponding pixel on-the-fly. The desired cut boundary consists of arcs between object and background pixels,

and the object is defined by the pixel with minimum-cut measure and all pixels within its respective cut boundary.

This region growing process creates a reduced search space that includes the desired cut boundary. Now it is expected that the objective function be able to detect it as the one with minimum cut. This is certainly not a problem when Equation 3 assigns lower arc weights across the object's boundary than inside **and outside** it.

If Equation 3 assigns low arc weights inside the object, the method may require one seed for each part of the object that satisfies the above conditions. The cut boundaries from each seed will merge into the desired cut boundary before the optimum paths reach the background pixels.

3.1 Algorithm

Our method uses the *Image Foresting Transform* (IFT)— a tool for the design of image processing operators based on connectivity [15]. The IFT algorithm essentially reproduces the aforementioned process by assigning an optimum path from S to every pixel in a non-decreasing order of cost. Its bottleneck is a priority queue Q , which selects a path of minimum cost $C(p)$ at each iteration by removing its last pixel p from Q . Ties are broken in Q using first-in-first-out policy. The algorithm runs in linear time if $\delta(p, q)$ is an integer in $[0, K]$ and Q is implemented as described in [16].

We need to modify the IFT algorithm as follows. When a pixel p is removed from Q , p receives a propagation order $O_d(p) \in [1, n]$, for an image with n pixels. At this moment, p and all pixels with lower propagation order define a region I and the algorithm has found the optimum paths from S to every pixel in I [15]. The remaining pixels define a region E ; the cut IE is defined by arcs between pixels of I and its 4-adjacent pixels in Q ; and the cut measure $M(p)$ for IE is computed on-the-fly. We first illustrate these modifications for normalized cut.

Algorithm 1 COMPUTATION OF THE PROPAGATION ORDER MAP O_d AND NORMALIZED CUT MAP M

INPUT: An image and adjacency A_4 .

OUTPUT: Maps O_d and M .

AUXILIARY: A priority queue Q and variables o , ai , ie , and ae that store the order and values of the Equations 5- 7 for the cut IE .

1. Set $o \leftarrow 1$, $ai \leftarrow 0$, $ie \leftarrow 0$, and $ae \leftarrow 0$.
2. **For** every image pixel p , **do**
3. Set $C(p) \leftarrow +\infty$ and $O_d(p) \leftarrow +\infty$.
4. **For** every pixel $q \in A_4(p)$ **do**
5. Set $ae \leftarrow ae + w(p, q)/2$.
6. **For** every pixel $p \in S$ **do**
7. Set $C(p) \leftarrow 0$ and insert p in Q .
8. **While** Q is not empty **do**
9. Remove p from Q such that $C(p)$ is minimum.
10. **For** every pixel $q \in A_4(p)$ **do**

```

11.   |   |   If  $O_d(q) < O_d(p)$ , then
12.   |   |   |    $\text{Set } ie \leftarrow ie - w(p, q) \text{ and } ai \leftarrow ai + w(p, q).$ 
13.   |   |   Else
14.   |   |   |    $\text{Set } ie \leftarrow ie + w(p, q) \text{ and } ae \leftarrow ae - w(p, q).$ 
15.   |   |   |    $\text{Set } cst \leftarrow \max\{C(p), \delta(p, q)\}.$ 
16.   |   |   |   If  $cst < C(q)$  then
17.   |   |   |   |   If  $C(q) \neq +\infty$  then
18.   |   |   |   |   |    $\text{Remove } q \text{ from } Q.$ 
19.   |   |   |   |    $\text{Set } C(q) \leftarrow cst \text{ and insert } q \text{ in } Q.$ 
20.   |   |    $\text{Set } O_d(p) \leftarrow o \text{ and } o \leftarrow o + 1.$ 
21.   |    $\text{Set } M(p) \leftarrow \frac{ie}{ie + ai} + \frac{ie}{ie + ae}.$ 

```

Lines 1–7 initialize maps, variables and insert seed pixels in Q . The division by 2 in Line 5 takes into account that the graph is undirected (i.e., $w(p, q) = w(q, p)$ should be considered only once). Thus, variable ae is initialized with the sum of all arc weights in the graph. Lines 8–21 compute the maps M and O_d during the IFT. When p is removed from Q (line 9), it leaves E and goes to I . At this moment, all arcs that contain p need to be evaluated. The condition stated in Line 11 indicates that $q \in I$, then arc (p, q) is being removed from IE and its weight must be considered to update ie and ai . Otherwise $q \in E$, then arc (p, q) is being inserted in IE and its weight must be used to update ie and ae . Lines 15–19 evaluate if the path that reaches q through p is better than the current path with terminus q and update Q and $C(q)$ accordingly. Finally, lines 20–21 compute the propagation order of p and the measure of its corresponding cut IE . After Algorithm 1, the object is obtained by selecting a pixel m with minimum-cut measure and thresholding O_d at values less than or equal to $O_d(m)$.

The above algorithm can be easily modified for mean cut if we set a variable n_{ie} to 0 in line 1 (where n_{ie} stores the size of IE); compute ie as above; insert $n_{ie} \leftarrow n_{ie} + 1$ in line 12 and $n_{ie} \leftarrow n_{ie} - 1$ in line 14; and set $M(p)$ to ie/n_{ie} in line 21. In the case of the energy function, we substitute lines 4 and 5 by $ae \leftarrow ae + (1 - R_b(p))$; compute ie as above; remove the computation of ai and ae from lines 12 and 14; and insert $ai \leftarrow ai + (1 - R_o(p))$ and $ae \leftarrow ae - (1 - R_b(p))$ between lines 20 and 21. In line 21, we set $M(p)$ to $\lambda(ai + ae) + ie$. Note that, we can do the same for many other graph-cut measures (e.g., [2, 4, 9]).

4 Results and Evaluation

Figures 1d and 1e show that the cut boundary may contain multiple contours due to “holes” (dark regions) inside the region map. The holes may be part of the object (Figure 1e) or not (Figure 1d). This problem may occur in any graph-cut segmentation approach. In our method, we close the holes in the resulting binary image and consider only the external contour as object boundary. Some results using the region maps of Figure 1 are presented in Figure 2 for normalized cut, mean cut, and energy function. In the latter, we also used training pixels outside the object to compute the background region map R_b of Equation 9.

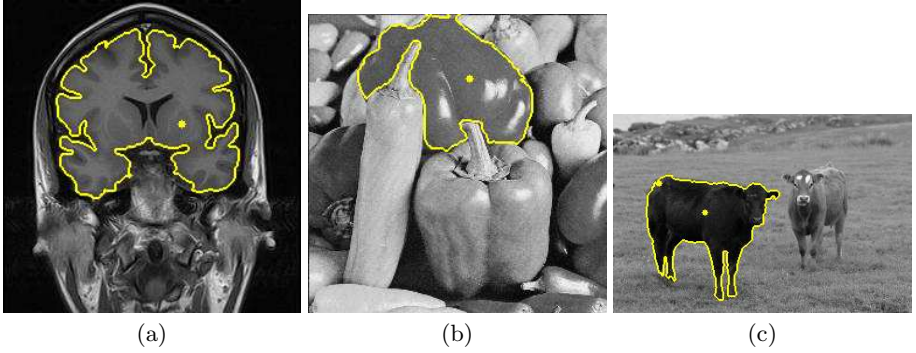


Fig. 2. Segmentation results where the seeds are indicated by dots, using (a) normalized cut, (b) mean cut, and (c) energy function

Figure 3 shows the cut measure versus the pixel propagation order for mean cut, normalized cut, and energy function using the region map of Figure 1d. In the case of the energy function, we also created a background map and set λ to 80 in Equation 9. In all cases the IFT parameters are the same and the desired cut occurs at order 13,340 of the reduced search space. However, it corresponds to the minimum cut only for the energy function (Figure 3c). Mean cut and normalized cut fail because of the weight assignment outside the object (Figures 3a–b). On the other hand, both cut measures can work if we add more hard constraints, such as limiting the search up to some propagation order o , for $o < n$ and greater than the object’s size (e.g., $o = 0.7n$ in this case).

This shows that any approach to separate object and background using graph cut is likely to require some hard constraints, because the problem can not be simply reduced to finding a minimum of an objective function in the entire search space. Since false-cut boundaries due to similarities between object and background are very common in practice, we have chosen a real application representing the worst case in this respect to evaluate our method.

4.1 Experiments for Evaluation

We have selected 6 images of archaeological fragments, similar to the one shown in Figure 4a. In this application, the boundary of each fragment has to be perfectly detected to reassemble the original object [17]. Thus, any failure in the detected boundary is considered a segmentation error. The similarities between object and background and touching fragments fail segmentation by thresholding.

The images have 512×384 pixels ($n = 196,608$) and a total of 211 fragments. We applied morphological operations to reduce internal noise, eliminate the grid pattern in the background, and estimate one seed pixel inside each fragment. This approach was able to find seeds inside 201 out of the 211 fragments automatically. Therefore, our experiments consist of using the method to detect the boundary of 201 seeded fragments in the filtered images.

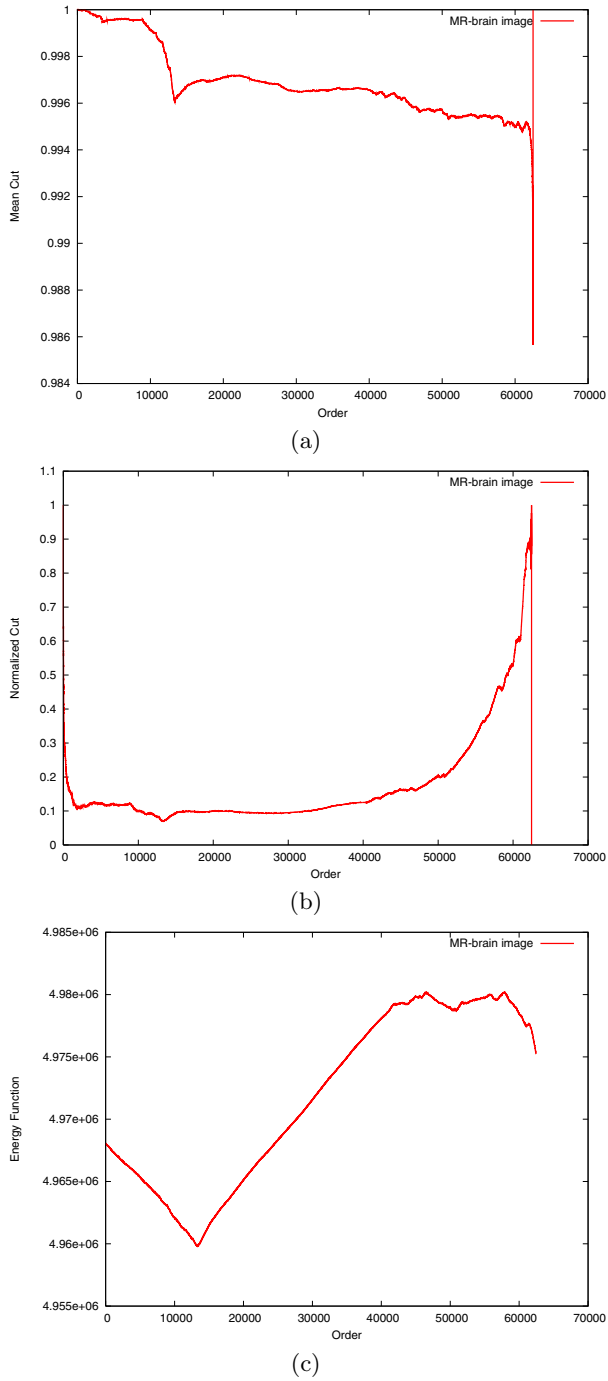


Fig. 3. The cut measure versus the pixel propagation order for (a) mean cut, (b) normalized cut, and (c) energy function using the MR-brain image

A suitable region map for each fragment would require seed selection on the shadow region that appears on most fragments. Since this is impractical in an automatic fashion, we decided to use dissimilarity and weight functions based on differences of brightness, as usually done in graph-cut segmentation [5, 3, 7].

$$\delta(p, q) = |f(p) - f(q)| \quad (11)$$

$$w(p, q) = 1.0 - \frac{\delta(p, q)}{K} \quad (12)$$

where $f(p)$ is the brightness of pixel p and K is the maximum brightness value in the filtered image. However, the region maps Ro and Rb were computed for the entire image (taking into account that fragments and non-fragments have dissimilar features) and used in the following energy function.

$$\lambda \left(\sum_{\forall p \in I} (1 - Ro(p)) + \sum_{\forall q \in E} (1 - Rb(q)) \right) + \sum_{\forall (p, q) | p \in I, q \in E} \alpha(p, q) w(p, q) \quad (13)$$

where

$$\alpha(p, q) = \begin{cases} 0 & \text{if } Ro(p) > Rb(p) \text{ and} \\ & Ro(q) < Rb(q) \\ 1 & \text{otherwise.} \end{cases} \quad (14)$$

and $\lambda = 40$. Note that Equation 14 uses Ro and Rb to restrict the computation of $w(p, q)$ inside uncertainty regions, as suggested in [7].

Our strategy is to assign a distinct number for each seed, detect each fragment separately, and label it with its corresponding number (see examples in Figures 4b–c). Some fragments touch each other, but the algorithm can separate them. When the algorithm fails, it usually outputs the union of two touching fragments twice, one for each seed. This situation is automatically detected and the fragments are separated by watershed transform restricted to their union [15].

The method with normalized cut correctly detected only 52 fragments (25.87%). In order to confirm that this bad result was not due to the IFT, we repeated the experiment with normalized cut and mean cut, but we limited the search for the minimum-cut value up to order $o = 0.05n$. The method with normalized cut correctly detected 104 (51.74%) fragments, while the method with mean cut detected 190 (94.53%) fragments correctly.

We also performed the experiments with the energy function. In this case, the method correctly detected 182 (90.50%) fragments. Although the number of correct detections was lower than using mean cut with $o = 0.05n$, we have observed that energy functions are usually more robust than the other two cut measures, when it is possible to devise a suitable normalization factor in Equation 9.

Finally, the mean running time to execute the method over images with 512×384 pixels was 161 milliseconds, using a 2.8GHz Pentium IV PC.

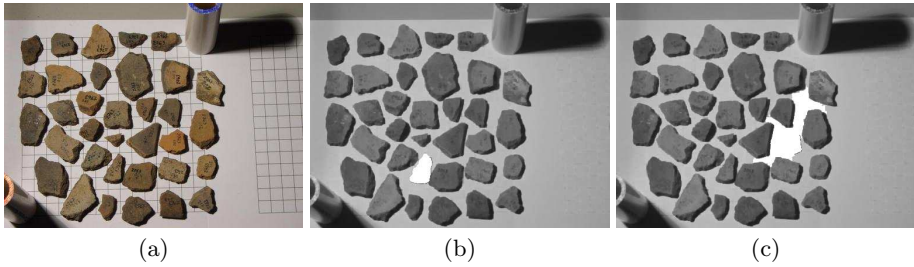


Fig. 4. Detection of archaeological fragments. (a) the original image. (b-c) Examples of correct and incorrect detections.

5 Conclusion

We showed that the segmentation by graph cut usually requires hard constraints to find the desired cut as minimum cut. We proposed a linear-time solution where the desired cut is included in a reduced search space under certain hard constraints applied to arc weight assignment and seed selection. We presented and evaluated our method for three cut measures: normalized cut, mean cut and an energy function.

The method requires proper weight assignment and/or more seeds inside the object, such that the IFT can reproduce its boundary during the region growing process. Under this condition, the problem is reduced to the sensitivity of the cut measures with respect to the weight assignment outside the object. The experiments evaluated this aspect in the worst case (i.e., when object and background parts have similar image properties). Even so, the results show accuracy greater than 90% for some cut measures. Therefore, we may conclude that our approach is a significant contribution in graph-cut segmentation.

In interactive segmentation, the IFT allows competition among internal and external seeds [18]. The combination of external seeds (to reduce the search space) and cut measures (to reduce user intervention) may provide more effective solutions than using [18, 7]. We are currently investigating this variant.

References

1. Wu, Z., Leahy, R.: An optimal graph theoretic approach to data clustering: theory and its applications to image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **15** (1993) 1101–1113
2. Cox, I.J., Rao, S.B., Zhong, Y.: Ratio regions: a technique for image segmentation. In: *Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*. (1996) 557–564
3. Wang, S., Siskind, J.M.: Image segmentation with minimum mean cut. In: *Intl. Conf. on Computer Vision (ICCV)*. Volume 1. (2001) 517–525
4. Sarkar, S., Boyer, K.L.: Quantitative measures of change based on feature organization: eigenvalues and eigenvectors. In: *Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*. (1996) 478–483

5. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **22** (2000) 888–905
6. Wang, S., Sinkind, J.M.: Image segmentation with ratio cut. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **25** (2003) 675–690
7. Yuri Y. Boykov, M.P.J.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: *Intl. Conf. on Computer Vision (ICCV)*. Volume 1. (2001) 105–112
8. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **26** (2004) 1124–1137
9. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **26** (2004) 147–159
10. Fowlkes, C., Belongie, S., Malik, J.: Efficient spatiotemporal grouping using the nystrom method. In: *Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*. (2001) 231–238
11. Carballido-Gamio, J., Belongie, S.J., Majumdar, S.: Normalized cuts in 3D for spinal MRI segmentation. *IEEE Trans. on Medical Imaging* **23** (2004) 36–44
12. Ford, L., Fulkerson, D.: *Flows in networks*. Princeton University Press (1962)
13. Greig, D., Porteous, B., Seheult, A.: Exact maximum a posteriori estimation for binary images. *J. Royal Statistical Society, series B* **51** (1989) 271–279
14. Kohli, P., Torr, P.H.: Efficiently solving dynamic markov random fields using graph cuts. In: *Intl. Conf. on Computer Vision (ICCV)*. Volume II. (2005) 922–929
15. Falcão, A., Stolfi, J., Lotufo, R.: The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **26** (2004) 19–29
16. Falcão, A., Udupa, J., Miyazawa, F.: An ultra-fast user-steered image segmentation paradigm: Live-wire-on-the-fly. *IEEE Trans. on Medical Imaging* **19** (2000) 55–62
17. Leitao, H., Stolfi, J.: A multiscale method for the reassembly of two-dimensional fragmented objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **24** (2002) 1239–1251
18. Falcão, A., Bergo, F.: Interactive volume segmentation with differential image foresting transforms. *IEEE Trans. on Medical Imaging* **23** (2004) 1100–1108