INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

A linear-time approach for graph-cut object detection

A.X. Falcão *P.A.V. Miranda* *A. Rocha*

Technical Report - IC-05-31 - Relatório Técnico

December - 2005 - Dezembro

The contents of this report are the sole responsibility of the authors.
O conteúdo do presente relatório é de única responsabilidade dos autores.

A linear-time approach for graph-cut object detection

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 usually computationally expensive, even with hard constraints (seed pixels). We present a solution that requires only internal seeds and runs in time proportional to the number of pixels. Our method computes an ordered region growing where the propagation order of each pixel is proportional to the cost of an optimum path 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. The method works with various cut measures and is evaluated using several experiments.

1 Introduction

Image segmentation approaches using graph cuts have become very popular in the last years. These approaches usually interpret an image as an undirected graph, whose nodes are the image pixels and whose arcs are weighted and defined by an adjacency relation between pixels. Arc weights must capture similarity, continuity, and proximity within the segments as a function of image features, such as color, gradient, and texture, and/or based on some prior knowledge about the segmentation problem.

The strategy in these approaches involves two challenges: (i) to assign higher weights to arcs inside the segments and lower weights otherwise, and (ii) to define some objective function based on these weights, such that its minimum corresponds to the desired segmentation. The segmentation is an optimal graph partitioning, where the *cut boundary* is the set of arcs that connect nodes from distinct segments.

Wu and Leahy [20] were the first to introduce these approaches using as objective function the sum of the arc weights in the cut boundary. Their objective function has the bias toward small boundaries and other objective functions, such as *average cut* [4], *mean cut* [19], *average association* [16], *normalized cut* [17], *ratio cut* [18], and *energy functions* [21, 2, 1, 13] 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 [3]. In object detection [21, 2, 1], terminal nodes (*source* and *destination*) are added to the image graph, representing object and background respectively. The arc weight between each pixel and a terminal (source/destination) is a penalty of assigning that pixel to the object/background. A min-cut/max-flow algorithm from source to destination [9, 11] computes a minimum-cut boundary that represents the object's boundary. If the method fails, the user can impose hard constraints to correct segmentation by adding seed pixels that have to be part of the object/background [21]. However, the running time of the min-cut/max-flow algorithms is still high for practical applications [1] (e.g., typically $O(mn^2)$ where m is the number of arcs and n is the number of *nodes*).

For similar image graphs, alternative solutions for object detection can be obtained based on *optimum paths* from seed pixels [7]. The method is known as *Image Foresting Transform* (IFT). It unifies many boundary- and region-based segmentation approaches and its algorithm runs in linear time for sparse image graphs. The IFT computes an optimum-path forest rooted in a seed set, where each seed defines an optimum-path tree composed by pixels that are more strongly connected to that seed than to any other.

For internal and external seeds, the object can be obtained as the union of the optimum-path trees rooted at its internal seeds [5]. This approach has been used in 3D interactive object detection, where the user can add seeds and remove trees, and the forest is recomputed in sublinear time. For only internal seeds, the object can also be detected by exploiting the topology of the forest to eliminate subtrees in the background. This approach is called *tree-pruning* segmentation [6].

In the present work, we investigate a possible combination between graph cut and IFT. The effectiveness of graph-cut segmentation depends on the arc weights and objective function. By adding hard constraints, graph-cut approaches essentially reduce the number of possible cut boundaries in the original search space. We use the IFT in a sparse image graph to reduce the search space to the number n of pixels and solve the graph-cut problem in $O(n)$.

For a set S of internal seeds and suitable *path-cost function*, the IFT computes an ordered region growing where the propagation order of each pixel is proportional to the cost of an optimum path from the seed set to that pixel. In this process, 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 during the process. The object is obtained by selecting the pixel with minimum-cut measure and all pixels within its respective cut boundary.

We show that the method works with various cut measures [17, 19, 21, 13] and evaluate it using several experiments. Section 2 presents the sparse 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 *normalized cut* [17], *mean cut* [19], and an *energy function* [21, 13]). We state conclusion and discuss future work 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. There are many ways of exploiting image features to compute arc weights [17, 21, 12]. Our approach first assigns a membership value for each pixel with respect to the object based on image features. 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 $\mathbf{\Sigma}_p$ 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.

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

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 $\mathbf{\Sigma}_s$ inside the object region that includes s. A region map R is obtained as

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

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

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

Another idea has been presented in [15]. A pixel classifier creates boundary maps B rather than region maps, where $B(p) \in [0, 1]$ represents the likelihood of a pixel p to be on the boundary of any object. In this case, we can compute the arc weight $w(p, q)$ by

$$w(p,q) \quad = \quad 1 - \max\{B(p), B(q)\}. \tag{4}$$

Figure 1 shows some examples of region and boundary maps created by the above procedures. The training pixels and their adjacency radii are indicated by circles on the original images (Figures 1a, 1e, and 1g). The respective region maps are shown in Figures 1b, 1f, and 1h. We used two normalized attributes within $[0, 1]$ for the feature vectors of Equation 1 in each case: brightness and gradient magnitude (Figure 1b); and red and green values (Figures 1e and 1h). 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 1b). Figures 1c and 1d show the original image and boundary map that result from the classifier proposed in [15].

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 others. Therefore, we will consider the normalized cut [17], mean cut [19], and an energy function [21, 13] 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

$$\frac{cut(I,E)}{asso(I) + cut(I,E)} + \frac{cut(I,E)}{asso(E) + cut(I,E)} \tag{5}$$

where

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

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

$$asso(E) \quad = \quad \sum_{\forall (p,q)|\ p \in E, q \in E} w(p,q). \tag{8}$$

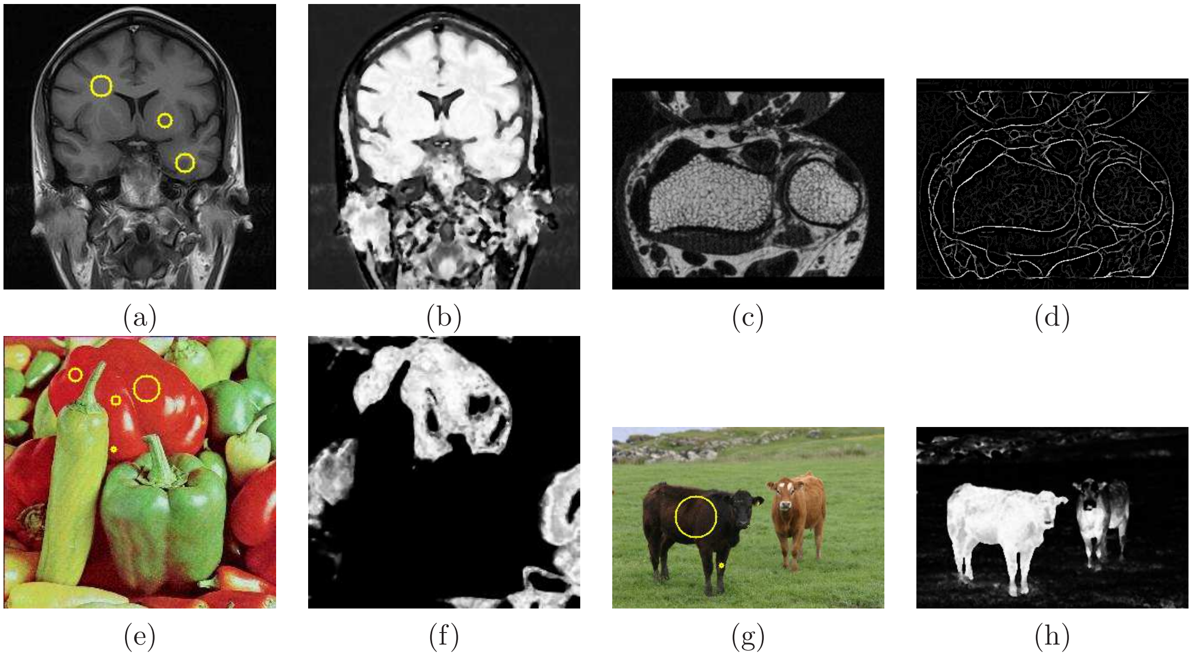


Figure 1: (a) A Magnetic Resonance (MR) image of a brain with three training pixels (the circles indicate their adjacency radii) and (b) the region map of the brain. (c) An MR-image of a wrist and (d) the resulting boundary map. (e) A colored image of peppers with five training pixels and (f) the region map of the central red pepper. (g) A colored image of two cows with three training pixels and (h) the region map of the darker cow.

The mean cut is defined as

$$\frac{cut(I,E)}{|IE|} \tag{9}$$

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

We have chosen an energy function similar to that proposed in [21] and consistent with the general formulation described in [13].

$$\lambda \left(\sum_{\forall p \in I} (1 - Ro(p)) + \sum_{\forall q \in E} (1 - Rb(q)) \right) \quad + \\ \sum_{\forall (p,q) | p \in I, q \in E} w(p,q) \tag{10}$$

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 with respect to the second one (e.g., we use $\lambda = 40$).

3 Region growing by ordered propagation with graph cut

We denote by $A_4(p)$ the set of the 4-adjacent pixels of p. A *path* is a sequence of 4-adjacent pixels and a *path-cost function* c assigns to each path π a *path cost* $c(\pi)$. A path π is *optimum* if $c(\pi) \leq c(\tau)$ for any other path τ with the same destination of π. The path-cost functions use arc weights as *dissimilarity* values $\delta(p,q) = K(1 - w(p,q))$, where K is an integer representing the maximum dissimilarity between pixels (e.g., we use $K = 1023$):

$$c_{\max}(\pi) = \begin{cases} \max_{\forall (p,q)\in\pi}\{\delta(p,q)\} & \text{if } \pi^* \in S \\ +\infty & \text{otherwise} \end{cases} \tag{11}$$

$$c_{sum}(\pi) = \begin{cases} \sum_{\forall (p,q)\in\pi} \delta(p,q) & \text{if } \pi^* \in S \\ +\infty & \text{otherwise} \end{cases} \tag{12}$$

where S is a set of internal seed pixels and π^* is the origin of path π. We will evaluate both path-cost functions.

The IFT computes an optimum path from S to every pixel p in a non-decreasing order of costs and using a priority queue Q. The linear-time efficiency of the process requires that $\delta(p,q)$ be an integer in $[0, K]$ and Q is implemented as proposed in [8]. Therefore, the process is an ordered region growing from S to every pixel p, such that the propagation order $O_d(p)$ is proportional to the cost $C(p)$ of an optimum path from S to p. Ties are broken in Q using first-in-first-out policy. When p is removed from Q, p and all pixels with lower propagation order define a region I. At this moment, the algorithm has found the optimum paths from S to every pixel in I, the remaining pixels define a region E, and the cut measure $M(p)$ (normalized cut, mean cut, or energy function) for IE can be computed on-the-fly. 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 algorithm below shows an instance of the method for c_{sum} and 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 6- 8 for the cut IE.

```
Set $o \leftarrow 1$, $ai \leftarrow 0$, $ie \leftarrow 0$, and $ae \leftarrow 0$.
For every image pixel $p$, do
    Set $C(p) \leftarrow +\infty$ and $O_d(p) \leftarrow +\infty$.
    For every pixel $q \in A_4(p)$ do
        Set $ae \leftarrow ae + w(p,q)/2$.
For every pixel $p \in S$ do
    Set $C(p) \leftarrow 0$ and insert $p$ in $Q$.
While $Q$ is not empty do
    Remove $p$ from $Q$ such that $C(p)$ is minimum.
    For every pixel $q \in A_4(p)$ do
        If $O_d(q) < O_d(p)$, then
            Set $ie \leftarrow ie - w(p,q)$ and $ai \leftarrow ai + w(p,q)$.
        Else
            Set $ie \leftarrow ie + w(p,q)$ and $ae \leftarrow ae - w(p,q)$.
            Set $cst \leftarrow C(p) + \delta(p,q)$.
            If $cst < C(q)$ then
                If $C(q) \neq +\infty$ then
                    Remove $q$ from $Q$.
                Set $C(q) \leftarrow cst$ and insert $q$ in $Q$.
    Set $O_d(p) \leftarrow o$ and $o \leftarrow o + 1$.
    Set $M(p) \leftarrow \frac{ie}{ie + ai} + \frac{ie}{ie + ae}$.
```

Note that, it is easy to modify the above algorithm for any combination between $\{c_{\max}, c_{sum}\}$ and {normalized cut, mean cut, energy function}. This observation is also valid for many other graph-cut measures (e.g., [4, 16, 13]).

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.

We can observe in Figures 1b and 1f that the cut boundary may contain multiple contours due to "holes" (dark regions) inside the region map. In Figure 1b, the holes are lateral ventricles and do not belong to the object. The holes are illumination effects in Figure 1f and belong to the object. Note that, 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.

Figure 2 illustrates some results of combining IFT with cut measures. Figure 2a presents the brain segmentation using one seed, path-cost function $c_{\max}$, mean cut and the region map of Figure 1b. Figure 2b presents the brain segmentation using the same seed and region map, but path-cost function c_{sum} and normalized cut. Note that, c_{sum} usually imposes more regularization to the cut boundary than $c_{\max}$. On the other hand, $c_{\max}$ is more robust to the initial location of the seeds and fits better the cut boundary to protrusions and indentations.

Figure 2c shows the wrist segmentation using the boundary map of Figure 1c, path-cost function c_{sum}, and normalized cut. Unfortunately, our method with boundary maps do not always produce the desired result. This may happen because the method proposed in [15] aims to detect all image boundaries rather than a given object's boundary. Therefore, undesired boundaries usually appear with high probability values inside and outside the object. Besides, very often, pixels on the object's boundary have probability values close or equal to zero.

The segmentation of the red pepper is shown in Figure 2d, where we use $c_{\max}$, mean cut, and the region map of Figure 1f. The region map of Figure 1h is used as Ro to compute $w(p, q)$ in Equation 10. We use background seeds to compute the other region map Rb. Figures 2e and 2f show the segmentation results of Figure 1g for one and two seed pixels, respectively. In both cases, we use the energy function of Equation 10 and $c_{\max}$.

4 Evaluation

It is hard to avoid false cut boundaries outside the object, when the background contains many parts with image features similar to those of the object. The mean-cut measure is the most sensitive to this problem. The results in Figure 2 were obtained by selecting the minimum cut in all cases, except for the mean-cut measure (Figures 2a and 2d). Figure 3 illustrates the mean-cut curve versus the pixel propagation order for the MR-brain image (Figure 2a). The desired minimum is at order 13,251 and it can only be detected when we limit the search up to some propagation order o, such that $o < n$ and greater than the object's size. The results in Figures 2a and 2d were obtained with $o = 0.7n$. For comparison, Figure 4 shows the normalized-cut measure and the energy function

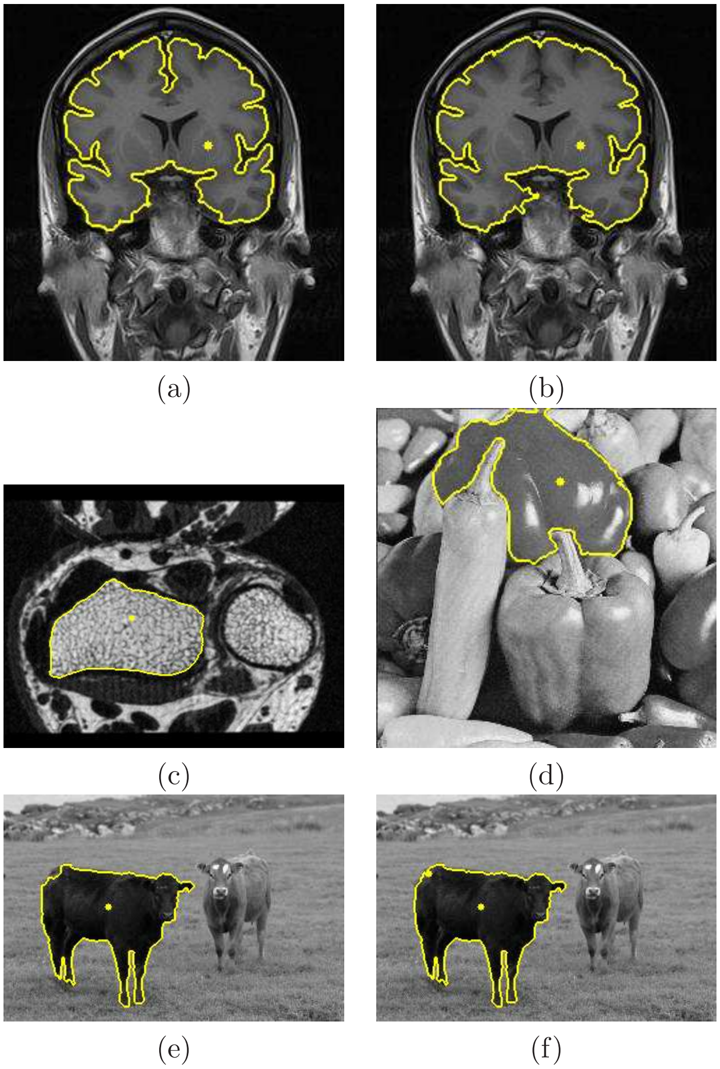


Figure 2: Segmentation results where the seeds are indicated by dots, using (a) $c_{\max}$, $w(p,q)$ by Equation 3 and mean cut, (b) c_{sum}, $w(p,q)$ by Equation 3 and normalized cut, (c) c_{sum}, $w(p,q)$ by Equation 4 and normalized cut, (d) $c_{\max}$, $w(p,q)$ by Equation 3 and mean cut, and (e) $c_{\max}$, $w(p,q)$ by Equation 3 and the energy function of Equation 10.

for the MR-brain image, and their respective segmentation results using the same seed, path-cost function $c_{\max}$, and region map of Figure 1b (an additional background map Rb was created for the energy function and we had to increase λ to 80 in Equation 10).

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 global minimum of an objective function. Since false-cut boundaries due to similarities between object and background are very common in practice, we have chosen an application which represents the worst case to evaluate our method.

We selected 6 images of archaeological fragments, similar to the one shown in Figure 5a. In this application, the boundary of each fragment has to be perfectly detected to reassemble the original object [14]. Thus, any failure in the detected boundary is considered a segmentation error.

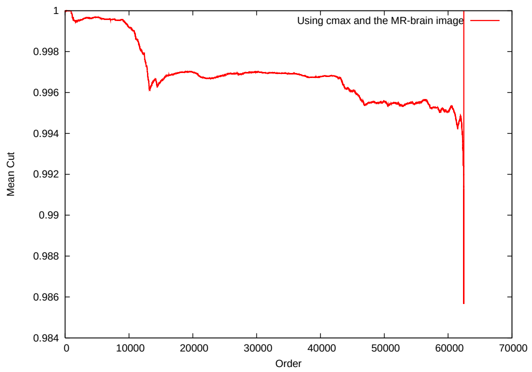
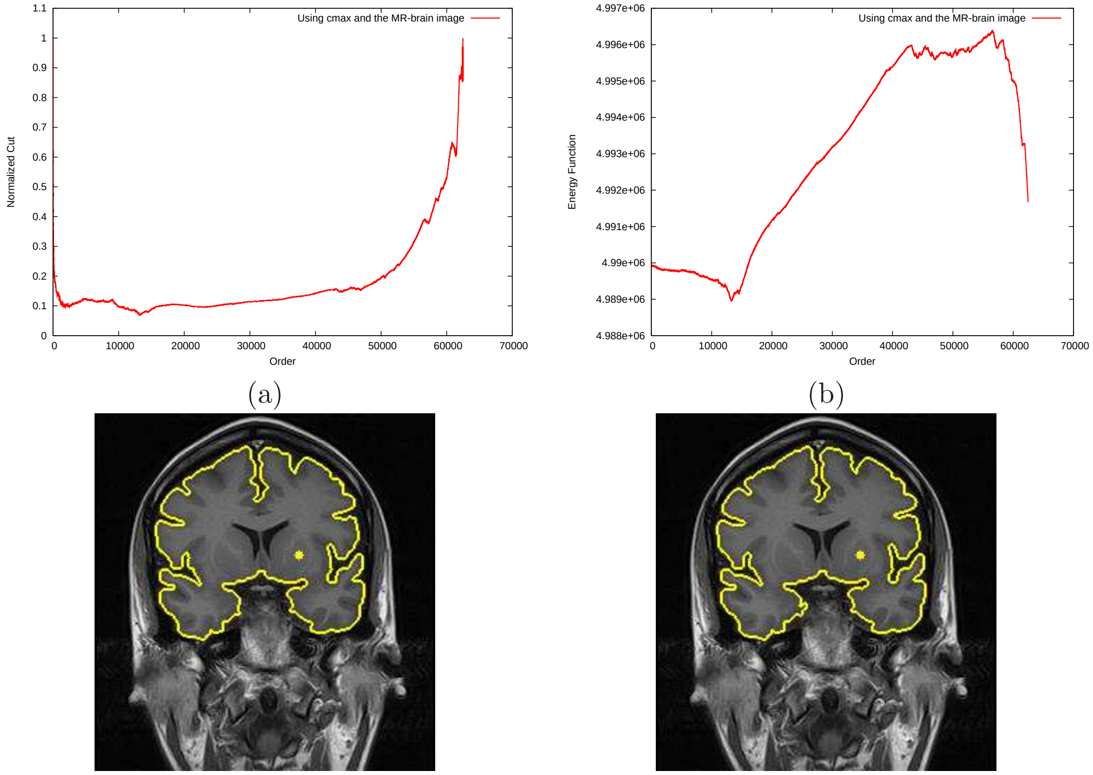


Figure 3: The mean-cut curve versus the pixel propagation order for the MR-brain image (Figure 2a). The desired minimum can only be detected when we limit the search up to some propagation order o, such that $o < n$ and greater than the object's size.



(a)

(b)

(c)

(d)

Figure 4: (a-b) Normalized-cut curve and energy function versus the pixel propagation order for the MR-brain image. (c-d) Their respective segmentation results using the same seed, path-cost function $c_{\max}$, and region map.

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 (Figure 5b). Therefore, our experiments consist of using the method to detect the boundary of 201 seeded fragments in the filtered images.

Instead of computing one region map for each fragment, we also decided to use simpler dissimilarity and weight functions.

$$\begin{aligned} \delta(p,q) &= |f(p) - f(q)| && (13) \\ w(p,q) &= 1.0 - \frac{\delta(p,q)}{K} && (14) \end{aligned}$$

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.

$$\lambda \left(\sum_{\forall p \in I} (1 - Ro(p)) + \sum_{\forall q \in E} (1 - Rb(q)) \right) + \sum_{\forall (p,q) | p \in I, q \in E} \alpha(p,q) w(p,q) \quad (15)$$

where

$$\alpha(p,q) = \begin{cases} 0 & \text{if } Ro(p) > Rb(p) \text{ and} \\ & Ro(q) < Rb(q) \\ 1 & \text{otherwise.} \end{cases} \quad (16)$$

That is, the region maps Ro and Rb can be used to restrict the computation of $w(p,q)$ inside uncertainty regions, as suggested in [21].

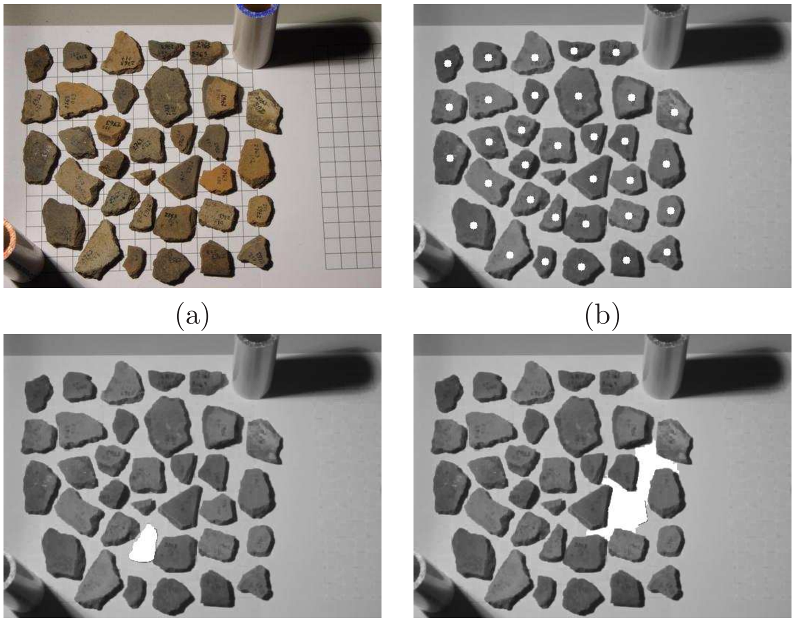
Our strategy is to assign a distinct number for each seed, detect each fragment separately, and label it with its corresponding number. Some fragments touch each other, but the algorithm can separate them. When the algorithm fails, it usually outputs the union of two touching fragments twice, one for each seed. This situation is automatically detected and the fragments are separated by watershed transform restricted to their union [7]. Figures 5c and 5d illustrate examples of correct and incorrect detections, respectively.

The IFT with $c_{\max}$ and 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 $c_{\max}$ and mean cut, but we limited the search for the minimum-cut value up to order $o = 0.7n$. The method correctly detected 172 (85.57%) fragments. Indeed, the size of the fragments are not so large and we could further reduce this threshold to $o = 0.05n$. In this case, the IFT with $c_{\max}$ and mean cut correctly detected 190 (94.53%) fragments. We repeated this last experiment with c_{sum}, but the number of correct detections was reduced to 178 (88.56%).

The IFT with $c_{\max}$ and energy function correctly detected 182 (90.50%) of the seeded 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.

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.

These experiments show that, even in the worst case, the proposed method provides good results (accuracy greater than 90%) under certain hard constraints.



(a)

(b)

(c)

(d)

Figure 5: Detection of archaeological fragments. (a) the original image. (b) The center of each disk represents a seed pixel over a filtered image. (c-d) Examples of correct and incorrect detections, respectively.

5 Conclusion

We have discussed some limitations of graph-cut segmentation; presented an efficient solution that uses the IFT to reduce the general graph-cut problem to $O(n)$ in images with n pixels; and evaluated the method using several cut measures and experiments. The experiments showed that the search for minimum-cut values usually require hard constraints and some prior knowledge about the desired object. The results indicate that the proposed method can be effective for object detection.

Different from other graph-cut approaches [1, 18], our method uses a simpler image graph (working for multidimensional images), allows different objective measures independently of the image graph, runs in linear time, and allows sublinear corrections in the case of interactive segmentation (see [5]). On the other hand, the IFT has to generate the object's boundary during pixel propagation. In this sense, the method is more restrictive than tree pruning [6], because the later only requires that optimum paths to object pixels do not pass through the background.

Our future work includes a comparison with tree-pruning segmentation and a 3D extension that incorporates graph-cut measures in the algorithm presented in [5].

6 Acknowledgments

The authors thank the financial support of FAPESP (Proc. 03/09793-1, 03/13424-1, and 04/02384-1) and CNPq (302427/04-0).

References

[1] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, Sep 2004.

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

[3] J. Carballido-Gamio, S. J. Belongie, and S. Majumdar. Normalized cuts in 3D for spinal MRI segmentation. *IEEE Trans. on Medical Imaging*, 23(1):36–44, Jan 2004.

[4] I. J. Cox, S. B. Rao, and Y. Zhong. Ratio regions: a technique for image segmentation. In *Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 557–564, 1996.

[5] A. Falcão and F. Bergo. Interactive volume segmentation with differential image foresting transforms. *IEEE Trans. on Medical Imaging*, 23(9):1100–1108, 2004.

[6] A. Falcão, F. Bergo, and P. Miranda. Image segmentation by tree pruning. In *XVII Brazilian Symposium of Computer Graphics and Image Processing (SIBGRAPI)*, pages 65–71. IEEE, Oct 2004.

[7] A. Falcão, J. Stolfi, and R. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1):19–29, 2004.

[8] A. Falcão, J. Udupa, and F. Miyazawa. An ultra-fast user-steered image segmentation paradigm: Live-wire-on-the-fly. *IEEE Trans. on Medical Imaging*, 19(1):55–62, 2000.

[9] L. Ford and D. Fulkerson. *Flows in networks*. Princeton University Press, 1962.

[10] C. Fowlkes, S. Belongie, and J. Malik. Efficient spatiotemporal grouping using the nyström method. In *Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 231–238, 2001.

[11] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *J. Royal Statistical Society, series B*, 51(2):271–279, 1989.

[12] P. Kohli and P. H. Torr. Efficiently solving dynamic markov random fields using graph cuts. In *Intl. Conf. on Computer Vision (ICCV)*, volume II, pages 922–929, 2005.

[13] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(2):147–159, Feb 2004.

[14] H. Leitao and J. Stolfi. A multiscale method for the reassembly of two-dimensional fragmented objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(9):1239–1251, Sep 2002.

[15] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect nature image boundaries using local brightness, color, and texture cues. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1):1–20, Jan 2004.

[16] S. Sarkar and K. L. Boyer. Quantitative measures of change based on feature organization: eigenvalues and eigenvectors. In *Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 478–483, 1996.

[17] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug 2000.

[18] S. Wang and J. M. Sinkind. Image segmentation with ratio cut. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(6):675–690, Jun 2003.

[19] S. Wang and J. M. Siskind. Image segmentation with minimum mean cut. In *Intl. Conf. on Computer Vision (ICCV)*, volume 1, pages 517–525, Jul 2001.

[20] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: theory and its applications to image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, Nov 1993.

[21] M.-P. J. Yuri Y. Boykov. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Intl. Conf. on Computer Vision (ICCV)*, volume 1, pages 105–112, 2001.