Image Segmentation with Minimum Mean Cut

Song Wang
Department of ECE and Beckman Institute
University of Illinois at Urbana-Champaign
Urbana IL 61802
songwang@mri.beckman.uiuc.edu

Jeffrey Mark Siskind
NEC Research Institute, Inc.
4 Independence Way
Princeton NJ 08540
qobi@research.nj.nec.com

Abstract

We introduce a new graph-theoretic approach to image segmentation based on minimizing a novel class of 'mean cut' cost functions. Minimizing these cost functions corresponds to finding a cut with minimum mean edge weight in a connected planar graph. This approach has several advantages over prior approaches to image segmentation. First, it allows cuts with both open and closed boundaries. Second, it guarantees that the partitions are connected. Third, the cost function does not introduce an explicit bias, such as a preference for large-area foregrounds, smooth or short boundaries, or similar-weight partitions. This lack of bias allows it to produce segmentations that are better aligned with image edges, even in the presence of long thin regions. Finally, the global minimum of this cost function is largely insensitive to the precise choice of edge-weight function. In particular, we show that the global minimum is invariant under a linear transformation of the edge weights and thus insensitive to image contrast. Building on algorithms by Ahuja, Magnanti, and Orlin (1993), we present a polynomial-time algorithm for finding a global minimum of the mean-cut cost function and illustrate the results of applying that algorithm to several synthetic and real images.

1. Introduction

Image segmentation is often formulated as the problem of partitioning an image into regions such that a cost function of the regions and/or the boundary between those regions is minimized. Two fundamental issues motivate the choice of cost function. First, the cost function should lead to perceptually salient segmentations. Second, the cost function should be amenable to global optimization to give confidence that the segmentations found result from the cost function itself and not from artifacts of the optimization procedure. Research in image segmentation over the past 15 years has explored the space of cost functions and the tradeoffs between these two issues. Early work, such as the active contour method (Kass, Witkin, & Terzopolous, 1987), used gradient descent to find local minima of a boundary cost function. Amini, Weymouth, and Jain (1990) introduced a dynamic-programming method for finding a global minimum to this boundary cost function within a local neighborhood of an initial boundary and iterated this method in a hill-climbing fashion to find local minima. Geiger, Gupta, Costa, and Vlontzos (1995) extended this dynamic-programming method to find a global minimum in the entire image under the constraint that the boundary pass through specified pixels. Motivated by the desire to find a cost function amenable to global optimization over the entire image, without constraint, Wu and Leahy (1993) introduced a graph-theoretic approach to image segmentation. This approach takes the image pixels to be vertices in an undirected graph $G = (V, E)$ with weighted edges between neighboring pixels. Throughout this paper, we only consider neighborhoods that lead to connected planar graphs. The non-negative real-valued edge-weight function $w(u, v)$ is selected to reflect the similarity between pixels u and v and is often chosen as a decreasing function of the difference in intensity I of adjacent pixels, such as the following Gaussian edge-weight function:

$$w(u, v) = e^{-\frac{(I_u - I_v)^2}{\sigma^2}} \tag{1}$$

A 2-way *cut* of G partitions V into two subsets A and B such that $A \cup B = V$ and $A \cap B = \emptyset$. Let us define the cut *boundary* as the set of edges that cross from A to B. Let us further define:

$$c(A, B|f) = \sum_{u \in A, v \in B, (u,v) \in E} f$$

The cut *cost* is defined as $c(A, B|w(u, v))$. Wu and Leahy (1993) perform image segmentation by finding a minimum-cost cut in G.

This approach suffers from a bias toward cuts with short boundaries and thus small regions, especially when the

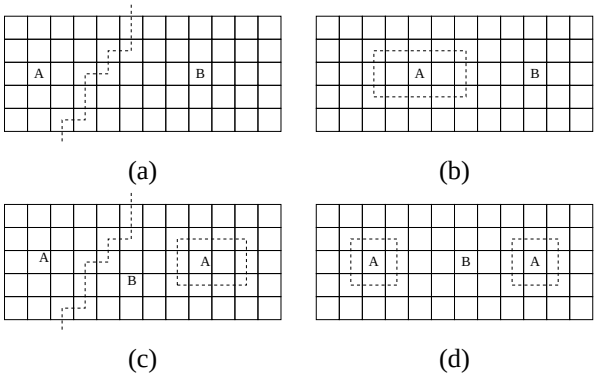


Figure 1. (a) Cut with open boundary. (b) Cut with closed boundary. (a) & (b) Cuts with connected partitions. (c) & (d) Cuts with unconnected partitions.

edge-weight function is not carefully selected. To address this problem, several researchers (Cox, Rao, & Zhong, 1996; Shi & Malik, 1997; Jermyn & Ishikawa, 1999; Veksler, 2000) have proposed a variety of normalized cost functions. Each cost function and optimization method has its advantages and disadvantages. Some allow cuts with both open (figure 1a) and closed (figure 1b) boundaries, while others only allow cuts with closed boundaries. Some produce cuts with only connected partitions (figures 1a and 1b), while others can produce cuts with unconnected partitions (figures 1c and 1d). Further, the different normalization methods induce different biases in the segmentation process. For some cost functions, this bias can drive the boundary away from image edges, while for others, it does not. The different cost functions and their properties are summarized in table 1.

This paper introduces a novel graph-theoretic image-segmentation cost function, mean cut, along with a novel polynomial-time global optimization algorithm for this cost function. This new method has several advantages over prior methods. First, it allows cuts with both open and closed boundaries. Supporting open boundaries is allows segmenting an object that is partially out of the field of view and also allows segmenting two objects whose boundary is partially occluded by a third object as shown in figure 4(b). Second, it guarantees that partitions are connected. Unconnected partitions may lead to spatially incoherent segmentations. Third, the cost function does not introduce an explicit bias, such as a preference for large-area foregrounds, smooth or short boundaries, or similar-weight partitions. This lack of bias allows it to produce segmentations that are better aligned with image edges, even in the presence of long thin regions. Finally, the global minimum of this cost function is largely insensitive to the precise choice of edge-weight function. In particular, we show that the global minimum of this cost function is invariant under a linear transformation of the edge weights and thus insensitive to image contrast.

Our method uses algorithmic techniques that bear superficial resemblance to those used by Jermyn and Ishikawa (1999). Their method finds minimum mean cycles in directed graphs. Part of our method involves finding minimum mean *simple* cycles in *undirected* graphs. These are different graph-theoretic problems. Furthermore, these two approaches use different classes of cost functions that have different image-segmentation properties.

The remainder of this paper is organized as follows. Section 2 introduces our novel mean-cut cost function along with an algorithm for finding the global minimum to that cost function. Section 3 presents an experimental evaluation of our method on several synthetic and real images. Section 4 concludes with a discussion of future work.

2. Minimum mean cut

The mean-cut cost function is:

$$\bar{c}(A,B) = \frac{c(A,B|w(u,v))}{c(A,B|1)}$$

which normalizes the cut cost by the cut boundary length. In contrast to approaches to image segmentation based on min-cut (Wu & Leahy, 1993) which find cuts that minimize the sum of the edge weights in the cut boundary, our approach finds cuts that minimize the average edge weight in the cut boundary. This normalization avoids the bias towards cuts with short boundaries that is inherent in using the cut cost alone.

In the next three subsections, we present a polynomial-time algorithm for finding a global minimum to the mean-cut cost function when applied to connected planar graphs. We do not know whether it is possible to find the minimum mean cut of an arbitrary graph in polynomial time and have not found any references to this problem in the literature. Our method, limited to connected planar graphs, consists of three reductions: minimum mean cut to minimum mean simple cycle, minimum mean simple cycle to negative simple cycle, and negative simple cycle to minimum-weight perfect matching. The above reductions all operate on undirected graphs. The second reduction, from minimum mean simple cycle to negative simple cycle was motivated by a similar reduction, discussed in Ahuja et al. (1993, pp. 150–157), that is used to find the minimum mean cycle in a directed graph. The third reduction, from negative simple cycle to minimum-weight perfect matching, was also motivated by a similar reduction, discussed in Ahuja et al. (1993, pp. 496–497), that is used to find shortest paths in undirected graphs.

method	cost function	optimization method*	open boundaries	connected partitions	bias	edge aligned
Kass et al.	$\int_\Omega \alpha\|\|v'(s)\|\|^2 + \beta\|\|v''(s)\|\|^2 + P(v(s))ds$	local 2-way	no	yes	smooth boundary	no
Wu & Leahy	$c(A, B\|w(u,v))$	global k-way	yes	yes	short boundary	‡
Cox et al.	$\frac{c(A, B\|w(u,v))}{\|A\|}$	global 2-way	no	no	smooth boundary, A is foreground, B is background, foreground has large area	no
Shi & Malik	$\frac{c(A, B\|w(u,v))}{c(A, V\|w(u,v))} + \frac{c(A, B\|w(u,v))}{c(B, V\|w(u,v))}$	†	yes	no	similar weight partition	no
Jermyn & Ishikawa	$\frac{\int_{\partial\mathcal{R}} g ds}{\left\|\int_{\mathcal{R}} f dx dy\right\|}$	global 2-way	no	yes§	depends on f and g	no
Veksler	no explicit cost function	n/a	yes	yes	depends on threshold	yes
MMC	$\frac{c(A, B\|w(u,v))}{c(A, B\|1)}$	global 2-way	yes	yes	none	yes

* All of the optimization methods listed as global in this table find a global optimum of the stated cost function in polynomial time.
† Finding the global minimum to the discrete problem is NP-complete. Spectral graph theory (Chung, 1997) can be used to find a global minimum to a continuous approximation to this discrete problem in polynomial time.
‡ depends on edge-weight function
§ Jermyn and Ishikawa (1999) do not discuss whether their algorithm can produce non-simple cycles. If it can, regions can contain partitions that are connected only by a single pixel.

Table 1. Comparison between different image-segmentation cost functions.

2.1. Reducing minimum mean cut to minimum mean simple cycle

We reduce the problem of finding a minimum mean cut (MMC) to the problem of finding a minimum mean simple cycle (MMSC), i.e. a simple cycle with minimum mean edge weight. A cycle is simple if it does not traverse a vertex more than once. Our reduction assumes that G is a connected planar graph. Initially, this graph will correspond to the entire image. However, recursive application of the algorithm, as will be described in section 2.4, applies this algorithm to graphs corresponding to image segments. The reduction constructs a *dual* graph $\hat{G} = (\hat{V}, \hat{E})$ from G. For simplicity, we limit our discussion here to grid-like graphs, though we have generalized the reduction to arbitrary connected planar graphs. The limited reduction, illustrated in figure 2, is as follows:

1. For every grid in G, $\hat{G}$ contains a corresponding vertex located in the center of this grid. These vertices are called *basic* vertices and form a new grid system.

2. $\hat{G}$ contains a distinct vertex for each border of G. These vertices are called *auxiliary* vertices. Conventional images are compact (i.e. do not contain holes). Graphs constructed from such images will have a single border and thus the corresponding $\hat{G}$ will have a single auxiliary vertex. However, recursive application of MMC can yield non-compact image segments which will have multiple boundaries and corresponding auxiliary vertices.

3. Each non-border edge $e \in E$ is mapped to a corresponding edge $\hat{e} \in \hat{E}$, with the same weight, that crosses e, in the grid system of $\hat{G}$, as shown in figure 2.

4. Each border edge $e \in E$ is mapped to a corresponding edge $\hat{e} \in \hat{E}$, with the same weight, that crosses e and connects a border vertex to the auxiliary vertex for that border, as shown in figure 2.

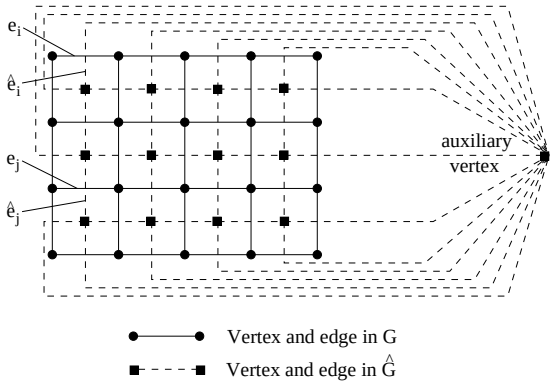


Figure 2. An illustration of the method for constructing $\hat{G}$ from G to reduce MMC to MMSC for the case of a rectangular compact image. The mapping from e_i to $\hat{e}_i$ illustrates the transformation of border edges while the mapping from e_j to $\hat{e}_j$ illustrates the transformation of non-border edges.

For any simple cycle $\hat{c} = \{\hat{e}_1, \hat{e}_2, ..., \hat{e}_l\}$ in $\hat{G}$, removing the edges $c = \{e_1, e_2, ..., e_l\}$ from E partitions G into two connected components and thus corresponds to a cut in G

with boundary c, and vice versa. Furthermore, if $\hat{c}$ traverses an auxiliary vertex, then c is an open boundary for that cut. Otherwise, c is a closed boundary. Thus we have the following lemma:

Lemma 1 *The MMC in G corresponds to the MMSC in $\hat{G}$.*

2.2. Reducing minimum mean simple cycle to negative simple cycle

One can find a minimum mean cycle in a *directed* graph using dynamic programming (Karp, 1978). This has been applied to image segmentation by Jermyn and Ishikawa (1999). However, we need to find the minimum mean simple cycle in an *undirected* graph. To do this, we first show the following:

Lemma 2 *Applying a linear transformation $w' = a \cdot w - b$ with $a > 0$ to the edge weights of an undirected graph does not change its MMSC.*

Proof: Assume $c_{min} = \{e_1, e_2, ..., e_l\}$ is the MMSC. Then for any other simple cycle $c = \{g_1, g_2, ..., g_m\}$, we have

$$\frac{1}{l}\sum_{i=1}^{l} w(e_i) \leq \frac{1}{m}\sum_{i=1}^{m} w(g_i)$$

Then, we have

$$\begin{aligned}\frac{1}{l}\sum_{i=1}^{l} w'(e_i) &= a \cdot \frac{1}{l}\sum_{i=1}^{l} w(e_i) - b \\ &\leq a \cdot \frac{1}{m}\sum_{i=1}^{m} w(g_i) - b = \frac{1}{m}\sum_{i=1}^{m} w'(g_i)\end{aligned}$$

Thus c_{min} is still the MMSC. □

A corollary of this lemma is that a linear transformation of the edge weights does not change the MMC in the original graph G. A second corollary of this lemma is that we can remove the non-negativity constraint on edge weights that applies to other graph-based segmentation algorithms. This property of linear transformation invariance, along with the following lemma, can be used to reduce the problem of finding a MMSC to the problem of finding a negative simple cycle (NSC). The reduction, adapted from Ahuja et al. (1993, pp. 150–157), relies on the following lemma:

Lemma 3 *A graph $\hat{G}$ has a MMSC c with mean weight b^* if and only if $\hat{G}$ has a zero-weight minimum simple cycle when its edge weights are transformed by $w' = w - b^*$.*

If $\hat{G}$ has a NSC when its edge weights are transformed by $w' = w - b$, then $b^* < b$. Likewise, if $\hat{G}$ does not have a NSC under that transformation, then $b^* \geq b$. If we have an algorithm for determining whether a graph has a NSC, b^* can be found by binary search. Let $\underline{w}$ and $\overline{w}$ be the smallest and largest edge weights in $\hat{G}$ respectively. Initialize $\underline{b}$ to $\underline{w}$ and $\overline{b}$ to $\overline{w}$. We know that $\underline{b} \leq b^* \leq \overline{b}$. Repeatedly let b' be the mean of $\underline{b}$ and $\overline{b}$. If $\hat{G}$ has a NSC when its edge weights are transformed by $w' = w - b'$ then set $\overline{b}$ to b'. Otherwise set $\underline{b}$ to b'.

Unfortunately, binary search is not guaranteed to terminate with real-valued edge weights. Even limiting edge weights to integers can yield rational mean values and lead to nontermination. However, given the mean-cut cost function and integral edge weights, the mean value will be rational with a denominator in the range $1, \ldots, |E|$. Thus the denominator will always be divisible by $|E|!$. By lemma 2, we can multiply all edge weights by $|E|!$ without changing the MMSC or MMC. Under such a scale transformation, the MMSC and MMC will have integral value. Since the search range is now from $|E|! \cdot \underline{w}$ to $|E|! \cdot \overline{w}$, and since $|E|! < |E|^{|E|}$, binary search is now guaranteed to terminate in at most $|E| \lg |E| + \lg(\overline{w} - \underline{w})$ iterations.

The above binary search algorithm yields only b^*, the mean value of the MMSC, but not the MMSC itself. To recover the MMSC, we can find one NSC for $\hat{G}$ when its edge weights are transformed by $w' = w - (b^* + \frac{1}{2})$ using the algorithm discussed in the next section. This NSC is the desired MMSC.

While the above binary search algorithm yields a polynomial-time algorithm for MMC, since, as shown in the next section, NSC can be reduced to minimum-weight perfect matching, which is polynomial time (Edmonds, 1965a, 1965b), it might not be practical because $|E|!$ can be large. Accordingly, our implementation uses a different technique for finding b^* and the corresponding MMSC. While we have not yet been able to prove that the implemented technique is polynomial time, it works well in practice, typically converging in a few iterations. Furthermore, while the binary-search technique works only for rational-valued edge weights, the implemented technique works for real-valued edge weights.

Our implemented technique works as follows: Like the binary search algorithm, we start by initializing $\overline{b}$ to $\overline{w}$. We know that $b^* \leq \overline{b}$. We use the algorithm discussed in the next section to find a set $S = \{c_1, \ldots, c_k\}$ of NSCs for $\hat{G}$ when its edge weights are transformed by $w' = w - \overline{b}$. Let a_i denote the (negative) mean value of each $c_i \in S$. We know that b^* must be less than each $a_i + \overline{b}$. Thus we set $\overline{b}$ to the smallest of $a_i + \overline{b}$ and repeat until no NSCs are found for some $\overline{b}$. This $\overline{b}$ is b^*. The desired MMSC is the NSC with the minimum mean value that was found in the penultimate iteration.

Both the binary and linear search techniques described above are similar to techniques described in Ahuja et al. (1993, pp. 150–157), where they are used to find the min-

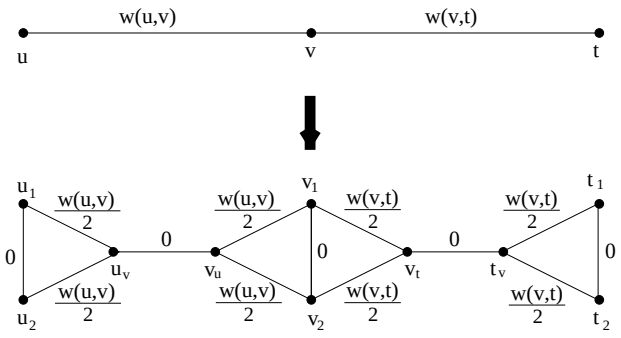


Figure 3. An illustration of the method for constructing G' from $\hat{G}$ to reduce NSC to minimum-weight perfect matching. Adapted from Ahuja et al. (1993, figure 12.22).

imum mean cycle in a directed graph. These techniques do not apply directly to undirected graphs because finding negative cycles in undirected graphs is more difficult than in directed graphs. It is this problem that we address in the next section.

2.3. Reducing negative simple cycle to minimum weight perfect matching

We can determine whether a graph $\hat{G} = (\hat{V}, \hat{E})$ has a NSC by asking whether a new graph $G' = (V', E')$ has a negative-weight perfect matching. A perfect matching in a graph is a subset of the edges such that each vertex has one incident edge from that subset. The reduction, adapted from Ahuja et al. (1993, pp. 496–497) and illustrated in figure 3, constructs G' from $\hat{G}$ as follows:

1. For each vertex u in $\hat{G}$, G' contains two corresponding vertices, u_1 and u_2, and one corresponding zero-weight edge (u_1, u_2).

2. For each edge (u, v) in $\hat{G}$, G' contains two corresponding vertices, u_v and v_u, and five corresponding edges with weights as follows: $w(u_1, u_v) = w(u_2, u_v) = w(v_1, v_u) = w(v_2, v_u) = \frac{1}{2}w(u, v)$ and $w(u_v, v_u) = 0$.

A graph has a negative perfect matching if and only if its minimum-weight perfect matching is negative. Edmonds (1965a, 1965b) gives a polynomial-time algorithm for finding a minimum-weight perfect matching. Our implementation of MMC uses the `blossom4` implementation of minimum-weight perfect matching (Cook & Rohe, 1998). We can show that G' always contains a perfect matching by constructing a trivial perfect matching containing all the edges of the form (u_1, u_2) and (u_v, v_u) for each $u \in \hat{V}$ and $(u, v) \in \hat{E}$. Since the total weight of this perfect matching is zero, the minimum-weight perfect matching in G' must have non-positive weight. The next lemma, adapted from an argument in Ahuja et al. (1993, pp. 496–497), addresses the relation between the existence of a NSC in $\hat{G}$ and the existence of a negative-weight perfect matching in G'.

Lemma 4 *$\hat{G}$ contains a NSC if and only if G' has a negative-weight perfect matching.*

Proof: To prove the forward direction, assume $\hat{G}$ has a NSC c. Select an arbitrary direction for this cycle and direct the edges according to this cycle direction. We can construct a perfect matching M in G' from c as follows:

1. M contains the two edges (u_1, u_v) and (v_u, v_2) for each directed edge $(u, v) \in c$. Note that the total weight of these edges is the same as that of c.

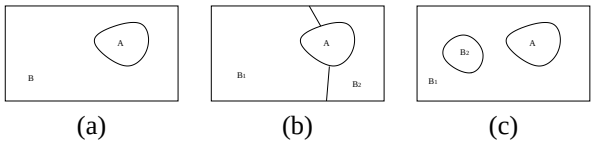
2. M contains the zero-weight edge (u_v, v_u) for each edge $(u, v) \notin c$.

3. M contains the zero-weight edge (u_1, u_2) for each vertex u that is not mentioned in c.

M is thus a perfect matching with the same (negative) total weight as c.

To prove the converse, we show a mapping from a negative minimum-weight perfect matching M in G' to a set of negative-weight cycles $S = \{c_1, c_2, ..., c_l\}$ in $\hat{G}$. S contains those edges (u, v) for which M contains either (u_1, u_v) or (u_2, u_v). First, we show that S is a set of simple cycles. M can contain at most one of (u_1, u_v) and (u_2, u_v). If M contains either (u_1, u_v) or (u_2, u_v), it must also contain either (v_1, v_u) or (v_2, v_u) and must not contain any of (u_v, v_u), (u_1, u_2), and (v_1, v_2). If M contains neither (u_1, u_v) nor (u_2, u_v), it must contain (u_v, v_u) and must not contain any of (v_1, v_u) and (v_2, v_u). If M contains (u_1, u_v), it must also contain (u_2, t_u), for some other vertex t. Likewise, if M contains (u_2, u_v), it must also contain (u_1, t_u), for some other vertex t. This means that every vertex that appears in S must have precisely two incident edges in S and thus the connected segments in S must be simple cycles. Finally, we show that the cycles in S must be non-positive. The sums of the edge weights in S and M are the same. If S contains some c that has positive total weight, we can remove c from S to yield a corresponding perfect matching in G' with smaller total weight and thus M is not the minimum-weight perfect matching. □

This proof for lemma 4 not only shows how to determine whether $\hat{G}$ has a NSC, by a reduction to minimum-weight perfect matching, it also shows how to construct a set S of NSCs from the negative minimum-weight perfect matching M.



(a) (b) (c)

Figure 4. The number of boundaries in a region can be less than, the same as, or greater than the number of boundaries in its parent.

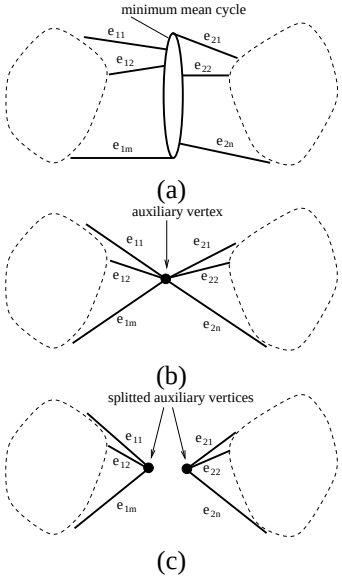


Figure 5. Illustration of the algorithm for splitting $\hat{G}$ into $\hat{G}_1$ and $\hat{G}_2$.

2.4. Graph splitting for recursive segmentation

After MMC is used to segment an initial image into two connected subregions, it can be applied recursively to each subregion. Initial images are typically compact and thus have a single border and corresponding auxiliary vertex in $\hat{G}$. However, recursive application of MMC to image segments may require processing non-compact segments, i.e. segments, such as B in figure 4(a), that have multiple boundaries and thus multiple auxiliary vertices. Sometimes, as for B in figure 4(a) or B_1 in figure 4(c), a subregion may have more boundaries than its parent. Sometimes, as for B_1 or B_2 in figure 4(b), a subregion may have fewer boundaries than its parent. And sometimes, as for A in figures 4(a-c), a subregion may have the same number of boundaries as its parent.

There is an efficient method for constructing $\hat{G}_1$ and $\hat{G}_2$ from $\hat{G}$, after MMC partitions G into G_1 and G_2, in a way that preserves much of the graph structure. This method, illustrated in figure 5, can be summarized as follows:

(a) (b) (c)

Figure 6. Applying MMC to a synthetic binary image. (a) Original image. (b) Segmentation after the first iteration. (c) Segmentation after four iterations.

1. Remove edges in $\hat{G}$ that are in the MMSC c.
2. Label each edge in $\hat{G}$ with i, if the corresponding edge in G is in G_i.
3. Merge all vertices mentioned in c into a single new auxiliary vertex a. Replace any edge that is incident on a vertex mentioned in c with an edge of the same weight and label that is instead incident on a.
4. Label each vertex in $\hat{G}$ with the same label as its incident edge. Note that it is not possible for a vertex to have two incident edges with different labels.
5. Split a into new auxiliary vertices a_1 and a_2, labeled 1 and 2 respectively. Each edge labeled i that is incident on a is replaced with an edge of the same weight and label that is incident on a_i.
6. $\hat{G}_i$ is the collection of vertices and edges with label i.

3. Experiments

Previous authors (Wu & Leahy, 1993; Shi & Malik, 1997; Veksler, 2000) have noted that graph-based image segmentation is often sensitive to the choice of edge-weight function. Prior work typically adopts equation (1) as the edge-weight function. With this function, segmentation is sensitive to the choice of σ. MMC is much less sensitive to the choice of edge-weight function. This is illustrated by the synthetic binary image in figure 6. Choosing any decreasing function of the intensity difference $|I_u - I_v|$ as the edge-weight function with MMC will result in the proper segmentation as shown in figure 6. No other image-segmentation method from table 1 has this property.

To further illustrate the insensitivity of MMC to the choice of edge-weight function, we processed the images in figures 7(a) and 7(e) with two different edge-weight functions. Figures 7(b) and 7(f) show the results of processing these images with the Gaussian edge-weight function from equation (1). Figures 7(c) and 7(g) show the results of processing these images with the edge-weight function

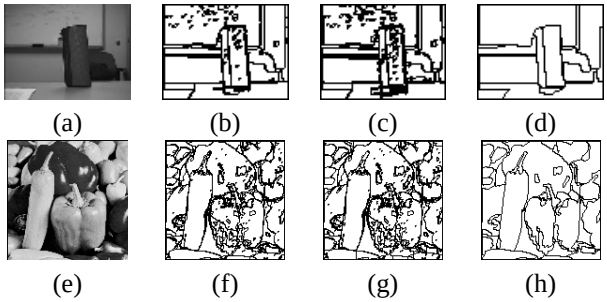


Figure 7. Applying MMC to a desktop scene (80×60) (a-d) and to the 'peppers' image (Veksler, 2000) (128×128) (e-f). (a,e) Original images. (b,f) Segmentation with the Gaussian edge-weight function with $\sigma = 15$. (c,g) Segmentation with the edge-weight function $w(u,v) = -|I_u - I_v|$. (d,h) The final segmentation after post-processing.

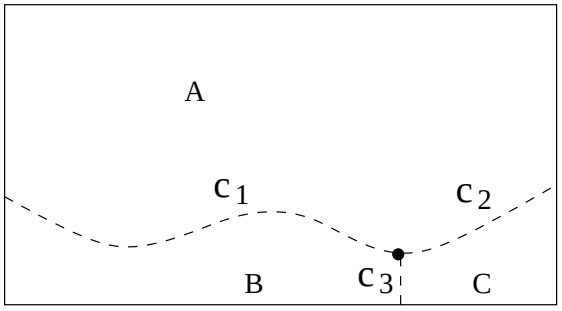


Figure 8. An illustration of the cause behind spurious cuts.

$w(u,v) = -|I_u - I_v|$. Both edge-weight functions produce similar results on both images.

The segmentations shown in figures 7(b), 7(c), 7(f), and 7(g) contain many small regions. Reasons for this include salt and pepper noise in the image and the fact that region boundaries in the image are wider than one pixel. This results in many short cuts inside these region boundaries. We can post-process the segmentation results to merge small adjacent regions (Wu & Leahy, 1993).

A more serious problem is that segmentations sometimes contain spurious cuts that do not correspond to image edges. An example of this phenomenon can be seen in figure 7(c), where the desk surface is segmented into several regions, and in figure 7(g), where the bottom-left pepper is segmented into two regions. Sometimes, these spurious cuts disappear with a different choice of edge weights, as in figure 7(f), and sometimes they do not, as in figure 7(b).

The cause of this phenomenon is illustrated in figure 8. Suppose that the desired cut boundary, which corresponds to image edges, is $c_1 \cup c_2$ with length $l_1 + l_2$. And suppose that the boundary c_3, which does not correspond to image edges, has a large mean value $\frac{w(c_3)}{l_3}$. In this case, MMC will yield the undesired cut boundary $c_1 \cup c_3$ when $\frac{w(c_1)}{l_1} < \frac{w(c_2)}{l_2}$ and $l_3 \ll l_2 < l_1$. Nonetheless, a subsequent recursive application of MMC will likely segment region $A \cup C$ along c_2. This motivates the use of a second region-merging criterion to address this problem.

The region-merging criterion that we use is also based on the mean-cut cost function. Suppose that recursive application of MMC yields the set $\{s_1, s_2, ..., s_n\}$ of regions. Repeatedly merge the neighbors s_i and s_j with the largest $\bar{c}(s_i, s_j)$ so long as $\bar{c}(s_i, s_j) > \delta$. In the experiments described below, we select the threshold δ to obtain a desired number of regions. Alternatively, δ can be varied to yield a multi-scale segmentation.

We applied the first region-merging criterion to the segmentations in figures 7(b) and 7(f), removing regions with less than 35 pixels, and then applied the second region-merging criterion with the target of obtaining 15 and 57 regions respectively. These targets were empirically selected to produce aesthetically pleasing segmentations. The results are shown in figures 7(d) and 7(h). Figure 9 shows the application of MMC along with both region-merging criteria on various images. For this experiment, a Gaussian edge-weight function was used with $\sigma = 15$.

4. Conclusion

We have presented our novel approach to image segmentation. This approach is based on a novel polynomial-time algorithm for finding the minimum mean cut in a connected planar graph. Our approach has four main advantages over prior graph-based approaches to image segmentation: it allows open cut boundaries, it guarantees that the partitions are connected, the mean-cut cost function does not introduce an explicit bias, and the minimum of this cost function is largely insensitive to the choice of edge-weight function. We have implemented our approach and used this implementation to illustrate these advantages on synthetic and real sample images.

Our initial prototype implementation is slow. Furthermore, as discussed in section 3, the approach produces fragmented segmentations when presented with noisy input. We are currently investigating enhancements to the approach that we hope will address these issues and plan on reporting on these results in the future.

Acknowledgments

We would like to thank Tugkan Batu, Lance Fortnow, David Jacobs, Satish Rao, Warren Smith, Olga Veksler,

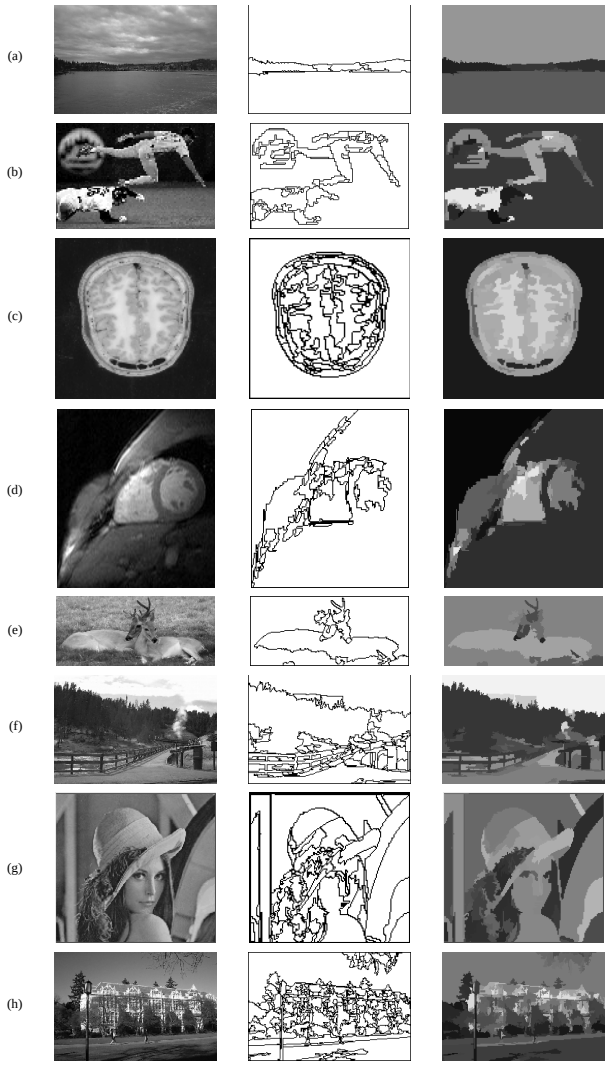


Figure 9. (a) A scenery photo, 226×151. (b) The 'baseball' image, 221×147. (c) A head cryosection image, 128×128. (d) A cardiac MRI image, 171×193. (e) A deer photo, 199×86. (f) A park photo, 226×151. (g) Lenna, 156×146. (h) A building photo, 226×151. For (a)–(h), regions with less than 35, 35, 20, 35, 35, 35, 35, and 20 pixels were merged by the first criterion and δ for the second criterion was selected to obtain 10, 50, 78, 50, 15, 80, 70, and 220 regions respectively. These δ values were empirically selected to produce aesthetically pleasing segmentations.

and Kevin Wayne for useful discussions related to this paper. Wayne pointed us toward Ahuja et al. (1993) and Batu, Fortnow, and Smith helped find the reductions in sections 2.2 and 2.3. This work was supported, in part, by a Beckman Graduate Fellowship 2000–2001 (S.W.).

References

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, & Applications*. Englewood Cliffs, NJ: Prentice-Hall.

Amini, A. A., Weymouth, T. E., & Jain, R. C. (1990). Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(9), 855–867.

Chung, F. R. K. (1997). *Spectral Graph Theory*. American Mathematical Society.

Cook, W., & Rohe, A. (1998). Computing Minimum-Weight Perfect Matchings. http://www.or.unibonn.de/home/rohe/matching.html.

Cox, I. J., Rao, S. B., & Zhong, Y. (1996). Ratio Regions: A Technique for Image Segmentation. In *Proceedings of the International Conference on Pattern Recognition*, pp. 557–564, Santa Barbara, CA.

Edmonds, J. (1965a). Maximum Matching and a Polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards*, *69B*, 125–130.

Edmonds, J. (1965b). Paths, trees and flowers. *Canadian Journal of Mathematics*, *17*, 449–467.

Geiger, D., Gupta, A., Costa, L. A., & Vlontzos, J. (1995). Dynamic programming for detecting, tracking, and matching deformable contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(3), 294–302.

Jermyn, I. H., & Ishikawa, H. (1999). Globally Optimal Regions and Boundaries. In *Proceedings of the* 7^{th} *International Conference on Computer Vision*, pp. 10–27.

Karp, R. (1978). A Characterization of the Minimum Cycle Mean in a Digraph. *Discrete Mathematics*, *23*, 309–311.

Kass, M., Witkin, A., & Terzopolous, D. (1987). Snakes: Active Contour Models. *International Journal of Computer Vision*, *1*(4), 321–331.

Shi, J., & Malik, J. (1997). Normalized Cuts and Image Segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 731–717.

Veksler, O. (2000). Image Segmentation by Nested Cuts. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 339–344.

Wu, Z., & Leahy, R. (1993). An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(11), 1101–1113.