Fast Approximate Energy Minimization via Graph Cuts

Yuri Boykov   Olga Veksler   Ramin Zabih
Computer Science Department
Cornell University
Ithaca, NY 14853

Abstract

In this paper we address the problem of minimizing a large class of energy functions that occur in early vision. The major restriction is that the energy function's smoothness term must only involve pairs of pixels. We propose two algorithms that use graph cuts to compute a local minimum even when very large moves are allowed. The first move we consider is an α-β-swap: for a pair of labels α, β, this move exchanges the labels between an arbitrary set of pixels labeled α and another arbitrary set labeled β. Our first algorithm generates a labeling such that there is no swap move that decreases the energy. The second move we consider is an α-expansion: for a label α, this move assigns an arbitrary set of pixels the label α. Our second algorithm, which requires the smoothness term to be a metric, generates a labeling such that there is no expansion move that decreases the energy. Moreover, this solution is within a known factor of the global minimum. We experimentally demonstrate the effectiveness of our approach on image restoration, stereo and motion.

1 Energy minimization in early vision

Many early vision problems require estimating some spatially varying quantity (such as intensity or disparity) from noisy measurements. Such quantities tend to be piecewise smooth; they vary smoothly at most points, but change dramatically at object boundaries. Every pixel $p \in \mathcal{P}$ must be assigned a label in some set $\mathcal{L}$; for motion or stereo, the labels are disparities, while for image restoration they represent intensities. The goal is to find a labeling f that assigns each pixel $p \in \mathcal{P}$ a label $f_p \in \mathcal{L}$, where f is both piecewise smooth and consistent with the observed data.

These vision problems can be naturally formulated in terms of energy minimization. In this framework, one seeks the labeling f that minimizes the energy

$$E(f) \quad = \quad E_{smooth}(f) \quad + \quad E_{data}(f).$$

Here E_{smooth} measures the extent to which f is not

piecewise smooth, while E_{data} measures the disagreement between f and the observed data. Many different energy functions have been proposed in the literature. The form of E_{data} is typically

$$E_{data}(f) = \sum_{p \in \mathcal{P}} D_p(f_p),$$

where D_p measures how appropriate a label is for the pixel p given the observed data. In image restoration, for example, $D_p(f_p)$ is typically $(f_p - i_p)^2$, where i_p is the observed intensity of the pixel p.

The choice of E_{smooth} is a critical issue, and many different functions have been proposed. For example, in standard regularization-based vision [6], E_{smooth} makes f smooth everywhere. This leads to poor results at object boundaries. Energy functions that do not have this problem are called *discontinuity-preserving*. A large number of discontinuity-preserving energy functions have been proposed (see for example [7]). Geman and Geman's seminal paper [3] gave a Bayesian interpretation of many energy functions, and proposed a discontinuity-preserving energy function based on Markov Random Fields (MRF's).

The major difficulty with energy minimization for early vision lies in the enormous computational costs. Typically these energy functions have many local minima (i.e., they are non-convex). Worse still, the space of possible labelings has dimension $|\mathcal{P}|$, which is many thousands. There have been numerous attempts to design fast algorithms for energy minimization. Simulated annealing was popularized in computer vision by [3], and is widely used since it can optimize an arbitrary energy function. Unfortunately, minimizing an arbitrary energy function requires exponential time, and as a consequence simulated annealing is very slow. In practice, annealing is inefficient partly because at each step it changes the value of a single pixel.

The energy functions that we consider in this paper arise in a variety of different contexts, including the Bayesian labeling of MRF's. We allow D_p to be

arbitrary, and consider smoothing terms of the form

$$E_{smooth} \quad = \quad \sum_{\{p,q\} \in \mathcal{N}} V_{\{p,q\}}(f_p, f_q), \qquad (1)$$

where $\mathcal{N}$ is the set of pairs of adjacent pixels. In special cases such energies can be minimized exactly. If the number of possible labels is $|\mathcal{L}| = 2$ then the exact solution can be found in polynomial time by computing a minimum cost cut on a certain graph [4]. If $\mathcal{L}$ is a finite 1D set and the interaction potential is $V(f_p, f_q) = |f_p - f_q|$ then the exact minimum can also be found efficiently via graph cuts [5, 2]. In general, however, the problem is NP-hard [8].

In this paper we develop algorithms that approximately minimize energy $E(f)$ for an arbitrary finite set of labels $\mathcal{L}$ under two fairly general classes of interaction potentials V: semi-metric and metric. V is called a *semi-metric* on the space of labels $\mathcal{L}$ if for any pair of labels $\alpha, \beta \in \mathcal{L}$ it satisfies two properties: $V(\alpha, \beta) = V(\beta, \alpha) \geq 0$ and $V(\alpha, \beta) = 0 \Leftrightarrow \alpha = \beta$. If V also satisfies the triangle inequality

$$V(\alpha, \beta) \leq V(\alpha, \gamma) + V(\gamma, \beta) \qquad (2)$$

for any α, β, γ in $\mathcal{L}$ then V is called a *metric*. Note that both semi-metric and metric include important cases of discontinuity-preserving interaction potentials. For example, the truncated L_2 distance $V(\alpha, \beta) = min(K, ||\alpha - \beta||)$ and the Potts interaction penalty $V(\alpha, \beta) = \delta(\alpha \neq \beta)$ are both metrics.

The algorithms described in this paper generalize the approach that we originally developed for the case of the Potts model [2]. In particular, we compute a labeling which is a local minimum even when very large moves are allowed. We begin with an overview of our energy minimization algorithms, which are based on graph cuts. Our first algorithm, described in section 3, is based on α-β-swap moves and works for any semimetric $V_{\{p,q\}}$'s. Our second algorithm, described in section 4, is based on more interesting α-expansion moves but works only for metric $V_{\{p,q\}}$'s (i.e., the additional triangle inequality constraint is required). Note that α-expansion moves produce a solution within a known factor of the global minimum of E. A proof of this can be found in [8].

2 Energy minimization via graph cuts

The most important property of these methods is that they produce a local minimum even when large moves are allowed. In this section, we discuss the moves we allow, which are best described in terms of partitions. We sketch the algorithms and list their basic properties. We then formally introduce the notion of a graph cut, which is the basis for our methods.

```
1.   Start with an arbitrary labeling f
2.   Set success := 0
3.   For each pair of labels {α,β} ⊂ L
     3.1.   Find f̂ = arg min E(f') among f' within
            one α-β swap of f (see Section 3)
     3.2.   If E(f̂) < E(f), set f := f̂
            and success := 1
4.   If success = 1 goto 2
5.   Return f
```

```
1.   Start with an arbitrary labeling f
2.   Set success := 0
3.   For each label α ∈ L
     3.1.   Find f̂ = arg min E(f') among f' within
            one α-expansion of f (see Section 4)
     3.2.   If E(f̂) < E(f), set f := f̂
            and success := 1
4.   If success = 1 goto 2
5.   Return f
```

Figure 1: Our swap move algorithm (top) and expansion move algorithm (bottom).

2.1 Partitions and move spaces

Any labeling f can be uniquely represented by a partition of image pixels $\mathbf{P} = \{\mathcal{P}_l \mid l \in \mathcal{L}\}$ where $\mathcal{P}_l = \{p \in \mathcal{P} \mid f_p = l\}$ is a subset of pixels assigned label l. Since there is an obvious one to one correspondence between labelings f and partitions $\mathbf{P}$, we can use these notions interchangingly.

Given a pair of labels α, β, a move from a partition $\mathbf{P}$ (labeling f) to a new partition $\mathbf{P}'$ (labeling f') is called an α-β *swap* if $\mathcal{P}_l = \mathcal{P}_l'$ for any label $l \neq \alpha, \beta$. This means that the only difference between $\mathbf{P}$ and $\mathbf{P}'$ is that some pixels that were labeled α in $\mathbf{P}$ are now labeled β in $\mathbf{P}'$, and some pixels that were labeled β in $\mathbf{P}$ are now labeled α in $\mathbf{P}'$.

Given a label α, a move from a partition $\mathbf{P}$ (labeling f) to a new partition $\mathbf{P}'$ (labeling f') is called an α-*expansion* if $\mathcal{P}_\alpha \subset \mathcal{P}_\alpha'$ and $\mathcal{P}_l' \subset \mathcal{P}_l$ for any label $l \neq \alpha$. In other words, an α-expansion move allows any set of image pixels to change their labels to α.

Note that a move which gives an arbitrary label α to a single pixel is both an α-β swap and an α-expansion. As a consequence, the standard move space used in annealing is a special case of our move spaces.

2.2 Algorithms and properties

We have developed two energy minimization algorithms, which are shown in figure 1. The structure of

the algorithms is quite similar. We will call a single execution of steps 3.1–3.2 an *iteration*, and an execution of steps 2–4 a *cycle*. In each cycle, the algorithm performs an iteration for every label (expansion move algorithm) or for every pair of labels (swap move algorithm), in a certain order that can be fixed or random. A cycle is successful if a strictly better labeling is found at any iteration. The algorithm stops after the first unsuccessful cycle since no further improvement is possible. Obviously, a cycle in the swap move algorithm takes $|\mathcal{L}|^2$ iterations, and a cycle in the expansion move algorithm takes $|\mathcal{L}|$ iterations.

These algorithms have several important properties. First, the algorithms are guaranteed to terminate in a finite number of cycles; in fact, under fairly general assumptions we can prove termination in $O(|\mathcal{P}|)$ cycles [8]. However, in the experiments we report in section 5, the algorithm stops after a few cycles and most of the improvements occur during the first cycle. Second, once the algorithm has terminated, the energy of the resulting labeling is a local minimum with respect to a swap or an expansion move. Finally, the expansion move algorithm produces a labeling f such that $E(f^*) \leq E(f) \leq 2k \cdot E(f^*)$ where f^* is the global minimum and $k = \frac{max\{V(\alpha,\beta) : \alpha \neq \beta\}}{min\{V(\alpha,\beta) : \alpha \neq \beta\}}$ (see [8]).

2.3 Graph cuts

The key part of each algorithm is step 3.1, where graph cuts are used to efficiently find $\hat{f}$. Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ be a weighted graph with two distinguished vertices called the terminals. A *cut* $\mathcal{C} \subset \mathcal{E}$ is a set of edges such that the terminals are separated in the induced graph $\mathcal{G}(\mathcal{C}) = \langle \mathcal{V}, \mathcal{E} - \mathcal{C} \rangle$. In addition, no proper subset of $\mathcal{C}$ separates the terminals in $\mathcal{G}(\mathcal{C})$. The cost of the cut $\mathcal{C}$, denoted $|\mathcal{C}|$, equals the sum of its edge weights.

The minimum cut problem is to find the cut with smallest cost. There are many algorithms for this problem with low-order polynomial complexity [1]; in practice they run in near-linear time for our graphs.

Step 3.1 uses a single minimum cut on a graph whose size is $O(|\mathcal{P}|)$. The graph is dynamically updated after each iteration. The details of this minimum cut are quite different for the swap move and the expansion move algorithms, as described in the next two sections.

3 Finding the optimal swap move

Given an input labeling f (partition $\mathbf{P}$) and a pair of labels α, β, we wish to find a labeling $\hat{f}$ that minimizes E over all labelings within one α-β swap of f. This is the critical step in the algorithm given at the top of Figure 1. Our technique is based on computing a labeling corresponding to a minimum cut on a
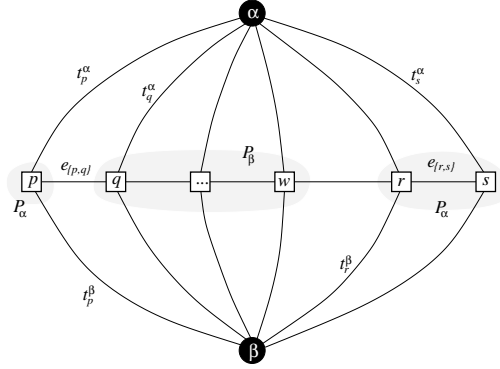

Figure 2: An example of the graph $\mathcal{G}_{\alpha\beta}$ for a 1D image. The set of pixels in the image is $\mathcal{P}_{\alpha\beta} = \mathcal{P}_\alpha \cup \mathcal{P}_\beta$ where $\mathcal{P}_\alpha = \{p, r, s\}$ and $\mathcal{P}_\beta = \{q, \ldots, w\}$.

graph $\mathcal{G}_{\alpha\beta} = \langle \mathcal{V}_{\alpha\beta}, \mathcal{E}_{\alpha\beta} \rangle$. The structure of this graph is dynamically determined by the current partition $\mathbf{P}$ and by the labels α, β.

This section is organized as follows. First we describe the construction of $\mathcal{G}_{\alpha\beta}$ for a given f (or $\mathbf{P}$). We show that cuts $\mathcal{C}$ on $\mathcal{G}_{\alpha\beta}$ correspond in a natural way to labelings $f^{\mathcal{C}}$ which are within one α-β swap move of f. Theorem 1 shows that the cost of a cut is $|\mathcal{C}| = E(f^{\mathcal{C}})$ plus a constant. A corollary from this theorem states our main result that the desired labeling $\hat{f}$ equals $f^{\mathcal{C}}$ where $\mathcal{C}$ is a minimum cut on $\mathcal{G}_{\alpha\beta}$.

The structure of the graph is illustrated in Figure 2. For legibility, this figure shows the case of 1D image. For any image the structure of $\mathcal{G}_{\alpha\beta}$ will be as follows. The set of vertices includes the two terminals α and β, as well as image pixels p in the sets $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$ (that is $f_p \in \{\alpha, \beta\}$). Thus, the set of vertices $\mathcal{V}_{\alpha\beta}$ consists of α, β, and $\mathcal{P}_{\alpha\beta} = \mathcal{P}_\alpha \cup \mathcal{P}_\beta$. Each pixel $p \in \mathcal{P}_{\alpha\beta}$ is connected to the terminals α and β by edges t_p^α and t_p^β, respectively. For brevity, we will refer to these edges as *t-links* (terminal links). Each pair of pixels $\{p, q\} \subset \mathcal{P}_{\alpha\beta}$ which are neighbors (i.e. $\{p, q\} \in \mathcal{N}$) is connected by an edge $e_{\{p,q\}}$ which we will call an *n-link* (neighbor link). The set of edges $\mathcal{E}_{\alpha\beta}$ thus consists of $\bigcup_{p \in \mathcal{P}_{\alpha\beta}} \{t_p^\alpha, t_p^\beta\}$ (the *t*-links) and $\bigcup_{\substack{\{p,q\} \in \mathcal{N} \\ p,q \in \mathcal{P}_{\alpha\beta}}} e_{\{p,q\}}$ (the *n*-links). The weights assigned to the edges are

edge	weight	for
t_p^α	$D_p(\alpha) + \sum_{\substack{q \in \mathcal{N}_p \\ q \notin \mathcal{P}_{\alpha\beta}}} V_{\{p,q\}}(\alpha, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
t_p^β	$D_p(\beta) + \sum_{\substack{q \in \mathcal{N}_p \\ q \notin \mathcal{P}_{\alpha\beta}}} V_{\{p,q\}}(\beta, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
$e_{\{p,q\}}$	$V_{\{p,q\}}(\alpha, \beta)$	$\{p,q\} \in \mathcal{N}$ $p,q \in \mathcal{P}_{\alpha\beta}$

Any cut $\mathcal{C}$ on $\mathcal{G}_{\alpha\beta}$ must sever (include) exactly one t-link for any pixel $p \in \mathcal{P}_{\alpha\beta}$: if neither t-link were in $\mathcal{C}$, there would be a path between the terminals; while if both t-links were cut, then a proper subset of $\mathcal{C}$ would be a cut. Thus, any cut leaves each pixel in $\mathcal{P}_{\alpha\beta}$ with exactly one t-link. This defines a natural labeling $f^{\mathcal{C}}$ corresponding to a cut $\mathcal{C}$ on $\mathcal{G}_{\alpha\beta}$,

$$f_p^{\mathcal{C}} = \begin{cases} \alpha & \text{if } t_p^{\alpha} \in \mathcal{C} \text{ for } p \in \mathcal{P}_{\alpha\beta} \\ \beta & \text{if } t_p^{\beta} \in \mathcal{C} \text{ for } p \in \mathcal{P}_{\alpha\beta} \\ f_p & \text{for } p \in \mathcal{P}, p \notin \mathcal{P}_{\alpha\beta} . \end{cases} \quad (3)$$

In other words, if the pixel p is in $\mathcal{P}_{\alpha\beta}$ then p is assigned label α when the cut $\mathcal{C}$ separates p from the terminal α; similarly, p is assigned label β when $\mathcal{C}$ separates p from the terminal β. If p is not in $\mathcal{P}_{\alpha\beta}$ then we keep its initial label f_p. This implies

Lemma 1 *A labeling $f^{\mathcal{C}}$ corresponding to a cut $\mathcal{C}$ on $\mathcal{G}_{\alpha\beta}$ is one α-β swap away from the initial labeling f.*

It is easy to show that a cut $\mathcal{C}$ severs an n-link $e_{\{p,q\}}$ between neighboring pixels on $\mathcal{G}_{\alpha\beta}$ if and only if $\mathcal{C}$ leaves the pixels p and q connected to different terminals. Formally

Property 1 *For any cut $\mathcal{C}$ and for any n-link $e_{\{p,q\}}$:*

$$\begin{array}{llll} a) & \text{If} & t_p^{\alpha}, t_q^{\alpha} \in \mathcal{C} & \text{then} & e_{\{p,q\}} \notin \mathcal{C}. \\ b) & \text{If} & t_p^{\beta}, t_q^{\beta} \in \mathcal{C} & \text{then} & e_{\{p,q\}} \notin \mathcal{C}. \\ c) & \text{If} & t_p^{\beta}, t_q^{\alpha} \in \mathcal{C} & \text{then} & e_{\{p,q\}} \in \mathcal{C}. \\ d) & \text{If} & t_p^{\alpha}, t_q^{\beta} \in \mathcal{C} & \text{then} & e_{\{p,q\}} \in \mathcal{C}. \end{array}$$

These properties are illustrated in figure 3. The next lemma is a consequence of property 1 and equation 3.

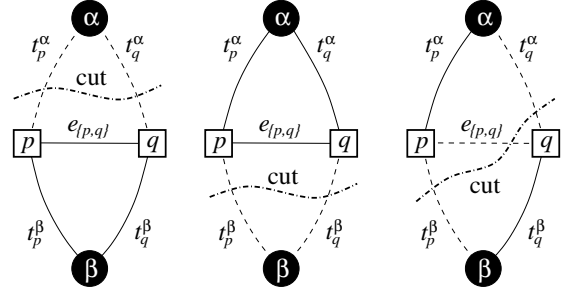**Lemma 2** *For any cut $\mathcal{C}$ and for any n-link $e_{\{p,q\}}$*

$$|\mathcal{C} \cap e_{\{p,q\}}| = V_{\{p,q\}}(f_p^{\mathcal{C}}, f_q^{\mathcal{C}}).$$

Lemmas 1 and 2 plus property 1 yield

Theorem 1 *There is a one to one correspondence between cuts $\mathcal{C}$ on $\mathcal{G}_{\alpha\beta}$ and labelings that are one α-β swap from f. Moreover, the cost of a cut $\mathcal{C}$ on $\mathcal{G}_{\alpha\beta}$ is $|\mathcal{C}| = E(f^{\mathcal{C}})$ plus a constant.*

PROOF: The first part follows from the fact that the severed t-links uniquely determine the labels assigned to pixels p and n-links that must to be cut. We now compute the cost of a cut $\mathcal{C}$, which is

$$|\mathcal{C}| = \sum_{p \in \mathcal{P}_{\alpha\beta}} |\mathcal{C} \cap \{t_p^{\alpha}, t_p^{\beta}\}| + \sum_{\substack{\{p,q\} \in \mathcal{N} \\ \{p,q\} \subset \mathcal{P}_{\alpha\beta}}} |\mathcal{C} \cap e_{\{p,q\}}|. \quad (4)$$

Property 1(a)   Property 1(b)   Property 1(c,d)

Figure 3: Properties of a cut $\mathcal{C}$ on $\mathcal{G}_{\alpha\beta}$ for two pixels $p, q \in \mathcal{N}$ connected by an n-link $e_{\{p,q\}}$. Dotted lines show the edges cut by $\mathcal{C}$ and solid lines show the edges remaining in the induced graph $\mathcal{G}(\mathcal{C}) = \langle \mathcal{V}, \mathcal{E} - \mathcal{C} \rangle$.

Note that for $p \in \mathcal{P}_{\alpha\beta}$ we have

$$|\mathcal{C} \cap \{t_p^{\alpha}, t_p^{\beta}\}| = D_p(f_p^{\mathcal{C}}) + \sum_{\substack{q \in \mathcal{N}_p \\ q \notin \mathcal{P}_{\alpha\beta}}} V_{\{p,q\}}(f_p^{\mathcal{C}}, f_q).$$

Lemma 2 gives the second term in (4). Thus, the total cost of a cut $\mathcal{C}$ is

$$|\mathcal{C}| = \sum_{p \in \mathcal{P}_{\alpha\beta}} D_p(f_p^{\mathcal{C}}) + \sum_{\substack{\{p,q\} \in \mathcal{N} \\ p \text{ or } q \in \mathcal{P}_{\alpha\beta}}} V_{\{p,q\}}(f_p^{\mathcal{C}}, f_q^{\mathcal{C}}).$$

This can be rewritten as $|\mathcal{C}| = E(f^{\mathcal{C}}) - K$ where

$$K = \sum_{p \notin \mathcal{P}_{\alpha\beta}} D_p(f_p) + \sum_{\substack{\{p,q\} \in \mathcal{N} \\ \{p,q\} \cap \mathcal{P}_{\alpha\beta} = \emptyset}} V_{\{p,q\}}(f_p, f_q)$$

is the same constant for all cuts $\mathcal{C}$. ∎

Corollary 1 *The optimal α-β swap from f is $\hat{f} = f^{\mathcal{C}}$ where $\mathcal{C}$ is the minimum cut on $\mathcal{G}_{\alpha\beta}$.*

4 Finding the optimal expansion move

Given an input labeling f (partition $\mathbf{P}$) and a label α, we wish to find a labeling $\hat{f}$ that minimizes E over all labelings within one α-expansion of f. This is the critical step in the algorithm given at the bottom of Figure 1. In this section we describe a technique that solves the problem assuming that each $V_{\{p,q\}}$ is a metric, and thus satisfies the triangle inequality (2). Some important examples of metrics are given in the introduction. Our technique is based on computing a labeling corresponding to a minimum cut on a graph $\mathcal{G}_{\alpha} = \langle \mathcal{V}_{\alpha}, \mathcal{E}_{\alpha} \rangle$. The structure of this graph is determined by the current partition $\mathbf{P}$ and by the label α.
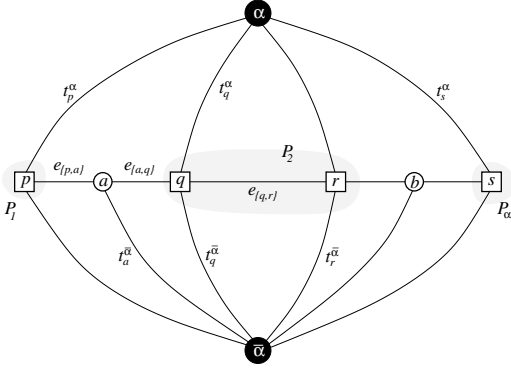
Figure 4: An example of $\mathcal{G}_\alpha$ for a 1D image. The set of pixels in the image is $\mathcal{P} = \{p, q, r, s\}$ and the current partition is $\mathbf{P} = \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_\alpha\}$ where $\mathcal{P}_1 = \{p\}$, $\mathcal{P}_2 = \{q, r\}$, and $\mathcal{P}_\alpha = \{s\}$. Two auxiliary nodes $a = a_{\{p,q\}}$, $b = a_{\{r,s\}}$ are introduced between neighboring pixels separated in the current partition. Auxiliary nodes are added at the boundary of sets $\mathcal{P}_l$.

As before, the graph dynamically changes after each iteration.

This section is organized as follows. First we describe the construction of $\mathcal{G}_\alpha$ for a given f (or $\mathbf{P}$) and α. We show that cuts $\mathcal{C}$ on $\mathcal{G}_\alpha$ correspond in a natural way to labelings $f^{\mathcal{C}}$ which are within one α-expansion move of f. Then, based on a number of simple properties, we define a class of *elementary* cuts. Theorem 2 shows that elementary cuts are in one to one correspondence with labelings that are within one α-expansion of f, and also that the cost of an elementary cut is $|\mathcal{C}| = E(f^{\mathcal{C}})$. A corollary from this theorem states our main result that the desired labeling $\hat{f}$ equals $f^{\mathcal{C}}$ where $\mathcal{C}$ is a minimum cut on $\mathcal{G}_\alpha$.

The structure of the graph is illustrated in Figure 4. For legibility, this figure shows the case of 1D image. The set of vertices includes the two terminals α and $\bar{\alpha}$, as well as all image pixels $p \in \mathcal{P}$. In addition, for each pair of neighboring pixels $\{p, q\} \in \mathcal{N}$ separated in the current partition (i.e. $f_p \neq f_q$) we create an *auxiliary vertex* $a_{\{p,q\}}$. Auxiliary nodes are introduced at the boundaries between partition sets $\mathcal{P}_l$ for $l \in \mathcal{L}$. Thus, the set of vertices is

$$\mathcal{V}_\alpha = \{\ \alpha\ ,\ \bar{\alpha}\ ,\ \mathcal{P}\ ,\ \bigcup_{\substack{\{p,q\} \in \mathcal{N} \\ f_p \neq f_q}} a_{\{p,q\}}\ \}.$$

Each pixel $p \in \mathcal{P}$ is connected to the terminals α and $\bar{\alpha}$ by t-links t_p^α and $t_p^{\bar{\alpha}}$, correspondingly. Each pair of neighboring pixels $\{p, q\} \in \mathcal{N}$ which are not sepa-

rated by the partition $\mathbf{P}$ (i.e. $f_p = f_q$) is connected by an n-link $e_{\{p,q\}}$. For each pair of neighboring pixels $\{p, q\} \in \mathcal{N}$ such that $f_p \neq f_q$ we create a triplet of edges $\mathcal{E}_{\{p,q\}} = \{e_{\{p,a\}}, e_{\{a,q\}}, t_a^{\bar{\alpha}}\}$ where $a = a_{\{p,q\}}$ is the corresponding auxiliary node. The edges $e_{\{p,a\}}$ and $e_{\{a,q\}}$ connect pixels p and q to $a_{\{p,q\}}$ and the t-link $t_a^{\bar{\alpha}}$ connects the auxiliary node $a_{\{p,q\}}$ to the terminal $\bar{\alpha}$. So we can write the set of all edges as

$$\mathcal{E}_\alpha = \{\ \bigcup_{p \in \mathcal{P}} \{t_p^\alpha, t_p^{\bar{\alpha}}\}\ ,\ \bigcup_{\substack{\{p,q\} \in \mathcal{N} \\ f_p \neq f_q}} \mathcal{E}_{\{p,q\}}\ ,\ \bigcup_{\substack{\{p,q\} \in \mathcal{N} \\ f_p = f_q}} e_{\{p,q\}}\ \}.$$

The weights assigned to the edges are

edge	weight	for
$t_p^{\bar{\alpha}}$	∞	$p \in \mathcal{P}_\alpha$
$t_p^{\bar{\alpha}}$	$D_p(f_p)$	$p \notin \mathcal{P}_\alpha$
t_p^α	$D_p(\alpha)$	$p \in \mathcal{P}$
$e_{\{p,a\}}$	$V_{\{p,q\}}(f_p, \alpha)$	
$e_{\{a,q\}}$	$V_{\{p,q\}}(\alpha, f_q)$	$\{p,q\} \in \mathcal{N},\ f_p \neq f_q$
$t_a^{\bar{\alpha}}$	$V_{\{p,q\}}(f_p, f_q)$	
$e_{\{p,q\}}$	$V_{\{p,q\}}(f_p, \alpha)$	$\{p,q\} \in \mathcal{N},\ f_p = f_q$

As in section 3, any cut $\mathcal{C}$ on $\mathcal{G}_\alpha$ must sever (include) exactly one t-link for any pixel $p \in \mathcal{P}$. This defines a natural labeling $f^{\mathcal{C}}$ corresponding to a cut $\mathcal{C}$ on $\mathcal{G}_\alpha$. Formally,

$$f_p^{\mathcal{C}} = \begin{cases} \alpha & \text{if } t_p^\alpha \in \mathcal{C} \\ f_p & \text{if } t_p^{\bar{\alpha}} \in \mathcal{C} \end{cases} \qquad \forall p \in \mathcal{P}. \qquad (5)$$

In other words, a pixel p is assigned label α if the cut $\mathcal{C}$ separates p from the terminal α and, p is assigned its old label f_p if $\mathcal{C}$ separates p from $\bar{\alpha}$. Note that for $p \notin \mathcal{P}_\alpha$ the terminal $\bar{\alpha}$ represents labels assigned to pixels in the initial labeling f. Clearly we have

Lemma 3 *A labeling $f^{\mathcal{C}}$ corresponding to a cut $\mathcal{C}$ on $\mathcal{G}_\alpha$ is one α-expansion away from the initial labeling f.*

It is also easy to show that a cut $\mathcal{C}$ severs an n-link $e_{\{p,q\}}$ between neighboring pixels $\{p, q\} \in \mathcal{N}$ such that $f_p = f_q$ if and only if $\mathcal{C}$ leaves the pixels p and q connected to different terminals. In other words, Property 1 holds when we substitute "$\bar{\alpha}$" for "β". We will refer to this as Property 1($\bar{\alpha}$). Analogously, we can show that Property 1($\bar{\alpha}$) and equation (5) establish Lemma 2 for the n-links $e_{\{p,q\}}$ on $\mathcal{G}_\alpha$.
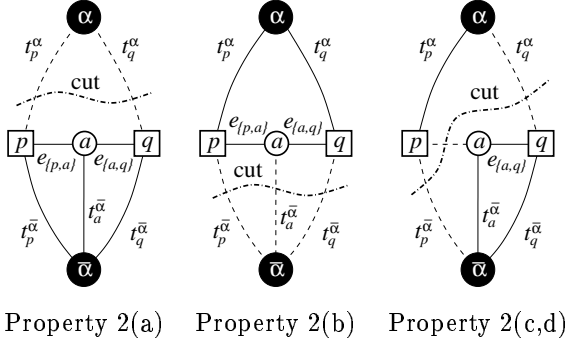
Property 2(a)   Property 2(b)   Property 2(c,d)

Figure 5: Properties of a minimum cut $\mathcal{C}$ on $\mathcal{G}_\alpha$ for two pixel $p, q \in \mathcal{N}$ such that $f_p \neq f_q$. Dotted lines show the edges cut by $\mathcal{C}$ and solid lines show the edges in the induced graph $\mathcal{G}(\mathcal{C}) = \langle \mathcal{V}, \mathcal{E} - \mathcal{C} \rangle$.

Consider now the set of edges $\mathcal{E}_{\{p,q\}}$ corresponding to a pair of neighboring pixels $\{p, q\} \in \mathcal{N}$ such that $f_p \neq f_q$. In this case, there are several different ways to cut these edges even when the pair of severed t-links at p and q is fixed. However, a minimum cut $\mathcal{C}$ on $\mathcal{G}_\alpha$ is guaranteed to sever the edges in $\mathcal{E}_{\{p,q\}}$ depending on what t-links are cut at the pixels p and q.

The rule for this case is described in Property 2 below. Assume that $a = a_{\{p,q\}}$ is an auxiliary node between the corresponding pair of neighboring pixels.

Property 2 *A minimum cut $\mathcal{C}$ on $\mathcal{G}_\alpha$ satisfies:*

a)   If $t_p^\alpha, t_q^\alpha \in \mathcal{C}$ then $\mathcal{C} \cap \mathcal{E}_{\{p,q\}} = \emptyset$.
b)   If $t_p^{\bar{\alpha}}, t_q^{\bar{\alpha}} \in \mathcal{C}$ then $\mathcal{C} \cap \mathcal{E}_{\{p,q\}} = t_a^{\bar{\alpha}}$.
c)   If $t_p^{\bar{\alpha}}, t_q^\alpha \in \mathcal{C}$ then $\mathcal{C} \cap \mathcal{E}_{\{p,q\}} = e_{\{p,a\}}$.
d)   If $t_p^\alpha, t_q^{\bar{\alpha}} \in \mathcal{C}$ then $\mathcal{C} \cap \mathcal{E}_{\{p,q\}} = e_{\{a,q\}}$.

Property (a) results from the fact that no subset of $\mathcal{C}$ is a cut. The others follow from the minimality of $|\mathcal{C}|$ and the fact that $|e_{\{p,a\}}|$, $|e_{\{a,q\}}|$ and $|t_a^{\bar{\alpha}}|$ satisfy the triangle inequality so that cutting any one of them is cheaper than cutting the other two together. These properties are illustrated in Figure 5.

Lemma 4 *If $\{p, q\} \in \mathcal{N}$ and $f_p \neq f_q$ then the minimum cut $\mathcal{C}$ on $\mathcal{G}_\alpha$ satisfies $|\mathcal{C} \cap \mathcal{E}_{\{p,q\}}| = V_{\{p,q\}}(f_p^\mathcal{C}, f_q^\mathcal{C})$.*

PROOF: The equation follows from property 2, equation (5), and the edge weights.   ∎

Property $1(\bar{\alpha})$ holds for any cut, and Property 2 holds for a minimum cut. However, there can be other cuts besides the minimum cut that satisfy both properties. We will define an *elementary* cut on $\mathcal{G}_\alpha$ to be a cut that satisfies Properties $1(\bar{\alpha})$ and 2.

Theorem 2 *Let $\mathcal{G}_\alpha$ be constructed as above given f and α. Then there is a one to one correspondence between elementary cuts on $\mathcal{G}_\alpha$ and labelings within one α-expansion of f. Moreover, for any elementary cut $\mathcal{C}$ we have $|\mathcal{C}| = E(f^\mathcal{C})$.*

PROOF: We first show that an elementary cut $\mathcal{C}$ is uniquely determined by the corresponding labeling $f^\mathcal{C}$. The label $f_p^\mathcal{C}$ at the pixel p determines which of the t-links to p is in $\mathcal{C}$. Property $1(\bar{\alpha})$ shows which n-links $e_{\{p,q\}}$ between pairs of neighboring pixels $\{p, q\}$ such that $f_p = f_q$ should be severed. Similarly, Property 2 determines which of the links in $\mathcal{E}_{\{p,q\}}$ corresponding to $\{p, q\} \in \mathcal{N}$ such that $f_p \neq f_q$ should be cut.

The cost of an elementary cut $\mathcal{C}$ is

$$|\mathcal{C}| = \sum_{p \in \mathcal{P}} |\mathcal{C} \cap \{t_p^\alpha, t_p^{\bar{\alpha}}\}| \qquad (6)$$
$$+ \sum_{\substack{\{p,q\} \in \mathcal{N} \\ f_p = f_q}} |\mathcal{C} \cap e_{\{p,q\}}| + \sum_{\substack{\{p,q\} \in \mathcal{N} \\ f_p \neq f_q}} |\mathcal{C} \cap \mathcal{E}_{\{p,q\}}|.$$

It is easy to show that for any pixel $p \in \mathcal{P}$ we have $|\mathcal{C} \cap \{t_p^\alpha, t_p^{\bar{\alpha}}\}| = D_p(f_p^\mathcal{C})$. Lemmas 2 and 4 hold for elementary cuts, since they are based on Properties $1(\bar{\alpha})$ and 2. Thus, the total cost of a elementary cut $\mathcal{C}$ is

$$|\mathcal{C}| = \sum_{p \in \mathcal{P}} D_p(f_p^\mathcal{C}) + \sum_{\{p,q\} \in \mathcal{N}} V_{\{p,q\}}(f_p^\mathcal{C}, f_q^\mathcal{C}) = E(f^\mathcal{C}).$$

Therefore, $|\mathcal{C}| = E(f^\mathcal{C})$.   ∎

Our main result is a simple consequence of this theorem, since the minimum cut is an elementary cut.

**Corollary 2** *The optimal α expansion from f is $\hat{f} = f^\mathcal{C}$ where $\mathcal{C}$ is the minimum cut on $\mathcal{G}_\alpha$.*

5   Experimental results

For our experiments, we used three energy functions, each with a quadratic D_p. The first energy function, called E_1, uses the truncated quadratic $V_{\{p,q\}}(f_p, f_q) = min(K, (f_p - f_q)^2)$ (for some constant K) as its smoothness term. This choice of V does not obey the triangle inequality, so we minimized E_1 using our swap move method. The second (E_2) and the third (E_3) energy functions use, correspondingly, the Potts model and the truncated L_2 distance as their smoothness penalty V. Both of these obey the triangle inequality and we minimized E_2 and E_3 with our expansion move method. We compared against annealing; we implemented several different annealing variants, and used the one that gave the best performance. This was the Metropolis sampler with a linearly decreasing temperature schedule.

Image Restoration. To illustrate the importance of different choices of V, consider the image restoration problem shown in the top row of figure 7. The original image contains large constant-intensity regions (the diamonds) which are gradually shaded, as if there were a light source to the left of the image. This image is corrupted with normally-distributed noise to produce the input image shown. This example demonstrates the need for non-Potts energy functions, as minimizing E_2 gives significant "banding" problems (shown in the second image). By selecting an energy function with a truncated quadratic $V_{\{p,q\}}$, we obtain the improved results shown at right.

The energy computed by our swap move method is shown below as a function of time. Note that we produce a very low energy after the first iteration, while annealing decreases the energy very slowly.
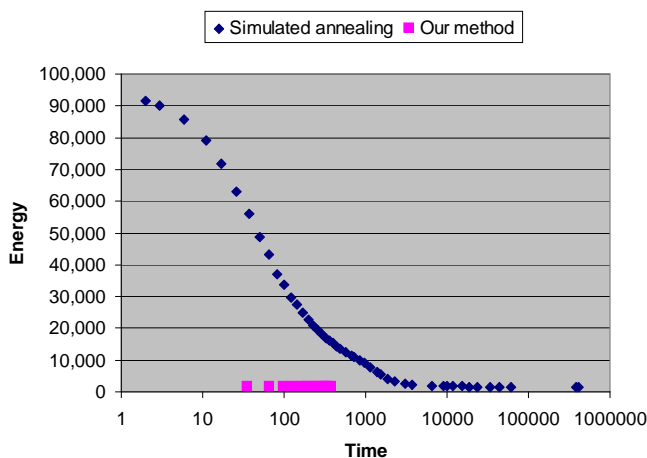
The energy values that we obtain for this and two more examples are shown in figure 6. The energy curves as a function of time are very similar to the diamond example shown above, but are omitted to save space. We also include the ratio between annealing's energy and ours. The third row for each image gives the best energy that annealing eventually achieves, when run until it is making very minimal progress. In this case, annealing eventually achieves a small improvement.

It is worthwhile to analyze E_{smooth}, since in our experience this correlated much more strongly with overall image quality than E. This is partly due to the fact that D_p rises so rapidly; as a result, most labels can be easily eliminated for a given pixel.

Motion and stereo. We also did energy minimization on several standard images, including the SRI tree sequence (taken from a camera moving along a rail) and the rock stereo pair. We compared our swap move and expansion move methods (for E_1 and E_3, correspondingly) with simulated annealing. We initialized both methods with the results of normalized correlation, which are also shown in the figure.

For both images, the energy that annealing achieves after more than 15 hours is significantly worse than the energy we obtain in around 200 seconds. We have experimented with a number of other images and obtained similar results.

Acknowledgements

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993.

[2] Yuri Boykov, Olga Veksler, and Ramin Zabih. Markov random fields with efficient approximations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–655, 1998.

[3] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

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

[5] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 125–131, 1998.

[6] Tomaso Poggio, Vincent Torre, and Christof Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[7] Demetri Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):413–424, 1986.

[8] Olga Veksler. *Efficient Graph-based Energy Minimization Methods in Computer Vision.* PhD thesis, Cornell University, July 1999.

	E			E_{smooth}		
	Our results	Annealing	Ratio	Our results	Annealing	Ratio
Diamond (image restoration, E_1)						
First cycle ($t = 36$)	1,577	55,892	**35.5**	637	9,658	**15.2**
Last cycle ($t = 389$)	1,472	15,215	**10.3**	576	8,475	**14.7**
Best annealing ($t = 417,317$)	—	1,458	—	—	571	—
Tree image (motion, E_1)						
First cycle ($t = 29$)	2,591	3,604	**1.4**	974	2,146	**2.2**
Last cycle ($t = 183$)	2,255	2,449	**1.1**	768	1,078	**1.4**
Best annealing ($t = 56,170$)	—	2,418	—	—	1,050	—
Rock image (stereo, E_3)						
First cycle ($t = 204$)	780	1,635	**2.1**	270	699	**2.6**
Last cycle ($t = 431$)	776	1,557	**2.0**	270	626	**2.3**
Best annealing ($t = 56,645$)	—	1,480	—	—	577	—

Figure 6: Comparative energy minimization results for our methods and simulated annealing

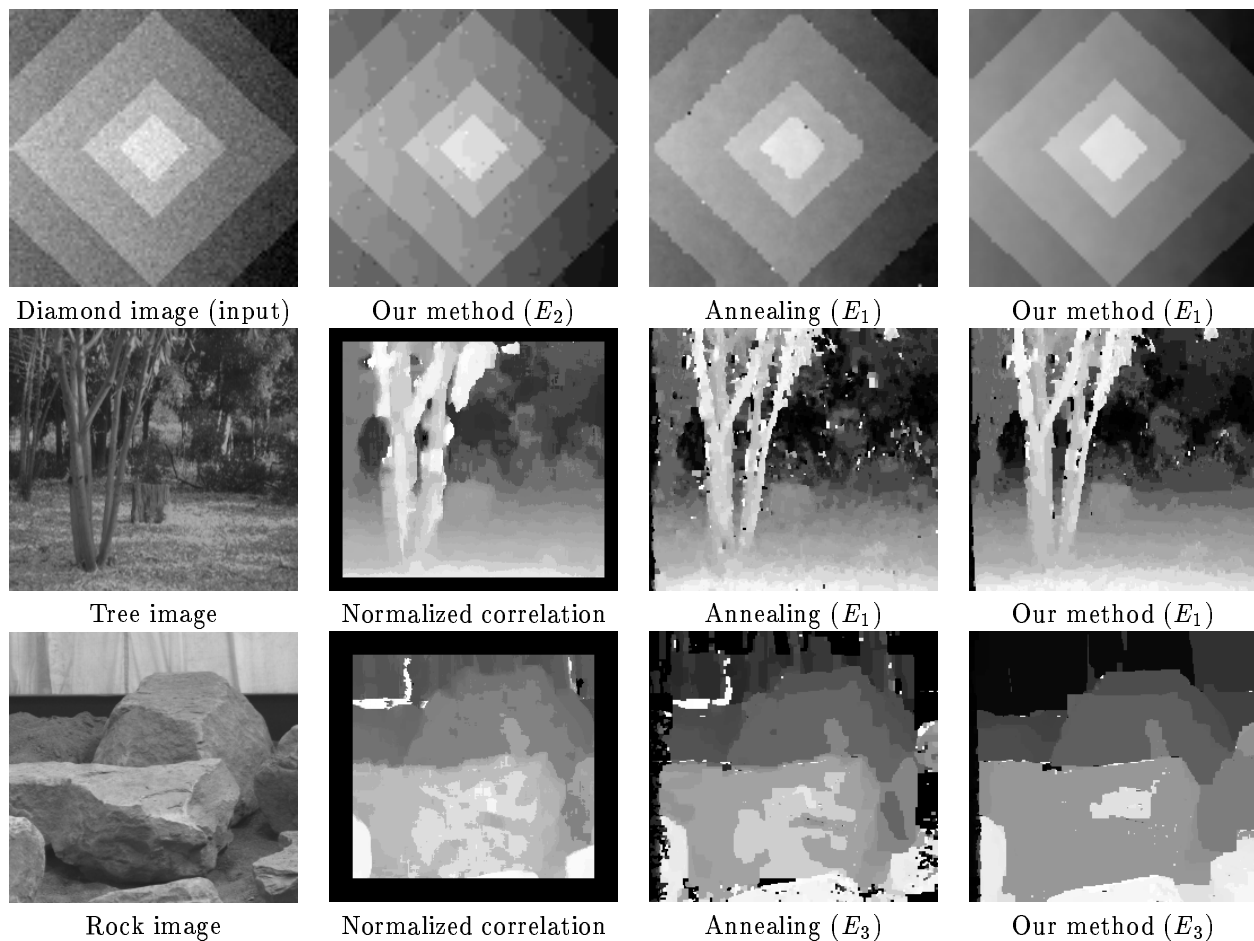| Diamond image (input)	Our method (E_2)	Annealing (E_1)	Our method (E_1)
Tree image	Normalized correlation	Annealing (E_1)	Our method (E_1)
Rock image	Normalized correlation	Annealing (E_3)	Our method (E_3)

Figure 7: Comparative results