Fast Approximate Energy Minimization via Graph Cuts

Yuri Boykov, Olga Veksler and Ramin Zabih*

Abstract

Many tasks in computer vision involve assigning a label (such as disparity) to every pixel. A common constraint is that the labels should vary smoothly almost everywhere while preserving sharp discontinuities that may exist, e.g., at object boundaries. These tasks are naturally stated in terms of energy minimization. In this paper we consider a wide class of energies with various smoothness constraints. Global minimization of these energy functions is NP-hard even in the simplest discontinuity-preserving case. Our focus is therefore on efficient approximation algorithms. We present two algorithms based on graph cuts that efficiently find a local minimum with respect to two types of large moves, namely *expansion* moves and *swap* moves. These moves can simultaneously change the labels of arbitrarily large sets of pixels. In contrast, many standard algorithms (including simulated annealing) use small moves where only one pixel changes its label at a time. Our expansion algorithm finds a labeling within a known factor of the global minimum, while our swap algorithm handles more general energy functions. Both algorithms allow important cases of discontinuity preserving energies. We experimentally demonstrate the effectiveness of our approach for image restoration, stereo and motion. On real data with ground truth we achieve 98% accuracy.

Index Terms — Energy minimization, graph, minimum cut, maximum flow, stereo, motion, image restoration, Markov Random Fields, Potts model, multiway cut.

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;

*Yuri Boykov is with Siemens Research, Princeton, NJ, yuri@scr.siemens.com. Olga Veksler is with NEC Research, Princeton, NJ, olga@research.nj.nec.com. Ramin Zabih is with the Computer Science Department, Cornell University, Ithaca, NY 14853, rdz@cs.cornell.edu Note that most of this work was done while Yuri Boykov and Olga Veksler were at Cornell University.

they vary smoothly on the surface of an object, but change dramatically at object boundaries. Every pixel $p \in \mathcal{P}$ must be assigned a label in some finite 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 well label f_p fits pixel p given the observed data. In image restoration, for example, $D_p(f_p)$ is normally $(f_p - I_p)^2$, where I_p is the observed intensity of p.

The choice of E_{smooth} is a critical issue, and many different functions have been proposed. For example, in some regularization-based approaches [22, 34], 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 [21, 29, 42]).

The major difficulty with energy minimization 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.

The energy functions that we consider in this paper arise in a variety of different contexts, including the Bayesian labeling of first-order Markov Random Fields (see [30] for details). We consider energies of the form

$$E(f) \quad = \quad \sum_{\{p,q\} \in \mathcal{N}} V_{p,q}(f_p, f_q) + \sum_{p \in \mathcal{P}} D_p(f_p), \tag{1}$$

where $\mathcal{N}$ is the set of interacting pairs of pixels. Typically $\mathcal{N}$ consists of adjacent pixels, but it can be arbitrary. We allow D_p to be nonnegative but otherwise arbitrary. In our choice of E_{smooth} only pairs of pixels interact directly.[1] Note that each pair of pixels $\{p, q\}$ can have

[1] Pair-wise interactions $V_{p,q}$ introduce (long range) dependence between all image pixels. This is a dramatic improvement over models assuming pixel independence. Higher order direct interactions (e.g. between triples of pixels) can potentially yield even better models but have largely been ignored due to tractability issues.

its own distinct penalty $V_{p,q}$. This turns out to be important in many applications, as shown in Section 8.2. However, to simplify the notation, we will frequently write V instead of $V_{p,q}$.

We develop algorithms that approximately minimize the energy $E(f)$ for an arbitrary finite set of labels $\mathcal{L}$ under two fairly general classes of interaction penalty V: metric and semi-metric. V is called a *metric* on the space of labels $\mathcal{L}$ if it satisfies

$$V(\alpha, \beta) = 0 \quad \Leftrightarrow \quad \alpha = \beta, \tag{2}$$

$$V(\alpha, \beta) = V(\beta, \alpha) \geq 0, \tag{3}$$

$$V(\alpha, \beta) \leq V(\alpha, \gamma) + V(\gamma, \beta), \tag{4}$$

for any labels $\alpha, \beta, \gamma \in \mathcal{L}$. If V satisfies only (2) and (3), it is called a *semi-metric.*[2]

Note that both semi-metrics and metrics include important cases of discontinuity preserving interaction penalties. Informally, a discontinuity preserving interaction term should have a bound on the largest possible penalty. This avoids overpenalizing sharp jumps between the labels of neighboring pixels; see [46, 30] and our experimental results in Section 8.6. Examples of discontinuity preserving interaction penalties for a one-dimensional label set $\mathcal{L}$ include the truncated quadratic $V(\alpha, \beta) = min(K, |\alpha - \beta|^2)$ (a semi-metric) and the truncated absolute distance $V(\alpha, \beta) = min(K, |\alpha - \beta|)$ (a metric), where K is some constant. If $\mathcal{L}$ is multidimensional, we can replace $|\cdot|$ by any norm, e.g. $||\cdot||_{L_2}$. These models encourage labelings consisting of several regions where pixels in the same region have similar labels, and therefore we informally call them piecewise smooth models.

Another important discontinuity preserving function is given by the Potts model $V(\alpha, \beta) = K \cdot T(\alpha \neq \beta)$ (a metric), where $T(\cdot)$ is 1 if its argument is true, and otherwise 0. This model encourages labelings consisting of several regions where pixels in the same region have equal labels, and therefore we informally call it a piecewise constant model.

We begin with a review of previous work on energy minimization in early vision. In Section 3 we give an overview of our energy minimization algorithms. Our first algorithm, described in Section 4, is based on α-β-swap moves and works for any semi-metric V. Our second algorithm, described in Section 5, is based on the more interesting α-expansion moves but requires V to be a metric. Optimality properties of our algorithms are discussed in Section 6. For example, we show that our expansion algorithm produces a solution within a known factor of the global minimum of E. In Section 7 we describe an important special case of our energy which arises from the Potts interaction penalty. This is a very simple

[2]In fact, we only assume $V(\alpha, \beta) = V(\beta, \alpha)$ in order to simplify the presentation. We can easily generalize all results in this paper to allow $V(\alpha, \beta) \neq V(\beta, \alpha)$. This generalization requires the use of directed graphs.

type of discontinuity preserving smoothness penalty, yet we prove that computing the global minimum is NP-hard. Experimental data is presented in Section 8.

2 Related work

The energy functions that we are interested in, given in equation (1), arise quite naturally in early vision. Energy based methods attempt to model some global image properties that can not be captured, for example, by local correlation techniques. The main problem, however, is that interesting energies are often difficult to minimize. We show in the appendix that one of the simplest discontinuity preserving cases of our energy function minimization is NP-hard; it is therefore impossible to rapidly compute the global minimum unless P=NP.

Due to the inefficiency of computing the global minimum, many authors have opted for a local minimum. However, in general a local minimum can be arbitrarily far from the optimum. It thus may not convey any of the global image properties that were encoded in the energy function. In such cases it is difficult to determine the cause of an algorithm's failures. When an algorithm gives unsatisfactory results, it may be due either to a poor choice of the energy function, or to the fact that the answer is far from the global minimum. There is no obvious way to tell which of these is the problem.[3] Another common issue is that local minimization techniques are naturally sensitive to the initial estimate.

In general, a labeling f is a local minimum of the energy E if

$$E(f) \leq E(f') \quad \text{for any } f' \text{ "near to" } f. \tag{5}$$

In case of discrete labeling, the labelings near to f are those that lie within a single *move* of f. Many local optimization techniques use what we will call *standard* moves, where only one pixel can change its label at a time (see Fig. 2(b)). For standard moves, equation (5) can be read as follows: if you are at a local minimum with respect to standard moves then you cannot decrease the energy by changing a single pixel's label. In fact, this is a very weak condition. As a result, optimization schemes using standard moves frequently generate low quality solutions. For instance, consider the local minimum with respect to standard moves shown in Fig. 1(c).

An example of a local method using standard moves is Iterated Conditional Modes (ICM),

[3] In special cases where the global minimum can be rapidly computed, it is possible to separate these issues. For example, [20] points out that the global minimum of a special case of Ising energy function is not necessarily the desired solution for image restoration. [9, 20] analyze the performance of simulated annealing in cases with a known global minimum.

which is a greedy technique introduced in [4]. For each pixel, the label which gives the largest decrease of the energy function is chosen, until convergence to a local minimum.

Another example of an algorithm using standard moves is simulated annealing, which was popularized in computer vision by [19]. Annealing is popular because it is easy to implement, and 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. Theoretically, simulated annealing should eventually find the global minimum if run for long enough. As a practical matter, it is necessary to decrease the algorithm's temperature parameter faster than required by the theoretically optimal schedule. Once annealing's temperature parameter is sufficiently low, the algorithm will converge to a local minimum with respect to standard moves. In fact, [20] demonstrate that practical implementations of simulated annealing give results that are very far from the global optimum even in the relatively simple case of binary labelings.

Trying to improve the rate of convergence of simulated annealing [39, 3] developed sampling algorithms for the Potts model that can make larger moves similar to our α-β-swaps. The main difference is that we find the best move among all possible α-β-swaps, while [39, 3] randomly select connected subsets of pixels that change their label from α to β. Like simulated annealing, these algorithms have only convergence "at infinity" optimality properties. The quality of the solutions that these algorithms produce in practice under realistic cooling schedules is not clear.

If the energy minimization problem is phrased in continuous terms, variational methods can be applied. These methods were popularized by [22]. Variational techniques use the Euler equations, which are guaranteed to hold at a local minimum.[4] To apply these algorithms to actual imagery, of course, requires discretization.

Another alternative is to use discrete relaxation labeling methods; this has been done by many authors, including [12, 36, 41]. In relaxation labeling, combinatorial optimization is converted into continuous optimization with linear constraints. Then some form of gradient descent which gives the solution satisfying the constraints is used. Relaxation labeling techniques are actually more general than energy minimization methods, see [23] and [32].

There are also methods that have optimality guarantees in certain cases. Continuation methods, such as graduated non-convexity [8], are an example. These methods involve approximating an intractable (non-convex) energy function by a sequence of energy functions, beginning with a tractable (convex) approximation. There are circumstances where these

[4]Note that in continuous cases the labels near to f in equation (5) are normally defined as $||f - f'|| \leq \epsilon$ where ϵ is a positive constant and $||\cdot||$ is a norm, e.g. L_2, over some appropriate functional space.

methods are known to compute the optimal solution (see [8] for details). Continuation methods can be applied to a large number of energy functions, but except for these special cases nothing is known about the quality of their output.

Mean field annealing is another popular minimization approach. It is based on estimating the partition function, from which the minimum of the energy can be deduced. However computing the partition function is computationally intractable, and saddle point approximations [31] are used. [17] provides an interesting connection between mean field approximation and other minimization methods like graduated non-convexity.

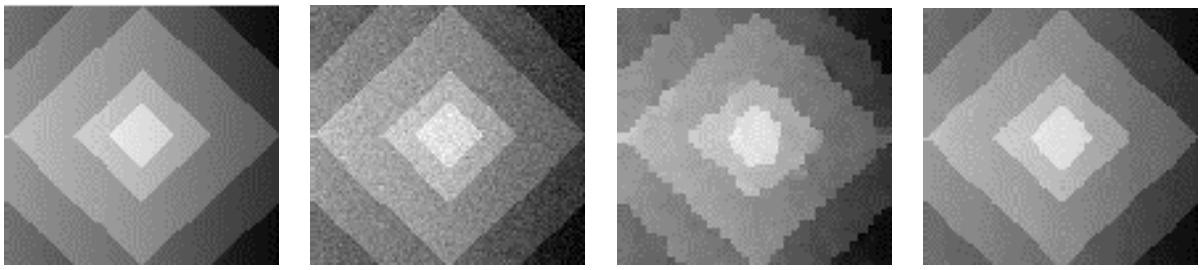
There are a few interesting energy functions where the global minimum can be rapidly computed via dynamic programming [2]. However, dynamic programming is restricted essentially to energy functions in one-dimensional settings. This includes some important cases, such as snakes [26]. In general, the two-dimensional energy functions that arise in early vision cannot be solved efficiently via dynamic programming.

Graph cut techniques from combinatorial optimization[5] can be used to find the global minimum for some multidimensional energy functions. When there are only 2 labels, equation (1) is a special case of the *Ising* model. Greig, Porteous, and Seheult [20] showed how to find the global minimum in this case by a single graph cut computation. Note that the Potts model we discuss in Section 7 is a natural generalization of the Ising model to the case of more than 2 labels. [14] develop a method optimal to within a factor of two for the Potts model; however the data energy they use is very restrictive. Recently [37, 24, 11] used graph cuts to find the exact global minimum of a certain type of energy functions. However, these methods apply only if the labels are one-dimensional. Most importantly they require V to be convex [25], and hence their energies are not discontinuity preserving, see Section 8.6.

Note that graph cuts have also been used for segmentation based on clustering [47, 16, 44]. Unlike clustering, we assume that there is a natural set of labels (e.g. intensities or disparities), and a data penalty function $D_p(\cdot)$ which makes some pixel-label assignments more likely than others.

The main contribution of this paper are two new algorithms for multidimensional energy minimization that use graph cuts iteratively. We generalize the previous results by allowing arbitrary label sets, arbitrary data terms D_p and a very wide class of pair-wise interactions V that includes discontinuity preserving cases. We achieve approximate solutions to this NP hard minimization problem with guaranteed optimality bounds.

[5]Throughout this paper we informally use *graph cuts* to refer to the min-cut/max-flow algorithms that are standard in combinatorial optimization [1]. See Section 3.3 for more details on graph cuts.



(a) original image (b) observed image (c) local min w.r.t. standard moves (d) local min w.r.t. α-expansion moves

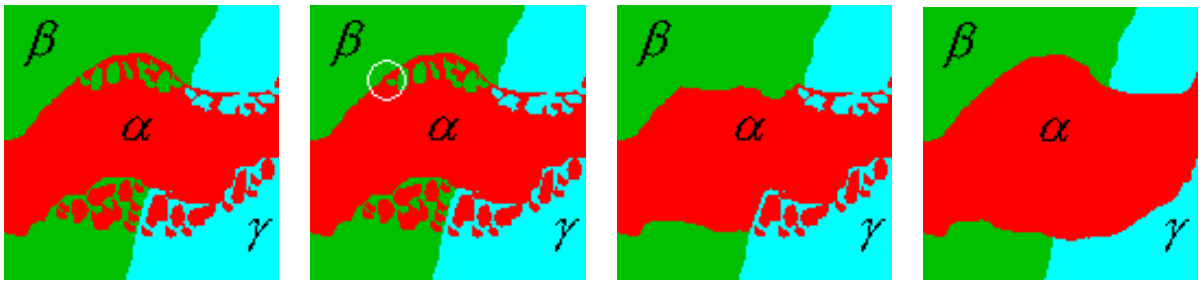
Figure 1: Comparison of local minima with respect to standard and large moves in case of image restoration. We use energy (1) with quadratic data terms penalizing deviations from the observed intensities (b). The smoothness term is truncated L_2 metric. Both local minima in (c) and (d) were obtained using labeling (b) as an initial solution.

3 Overview of our algorithms

The NP-hardness result given in the appendix effectively forces us to compute a local minimum. However, our methods generate a local minimum with respect to very large moves. We show that this approach overcomes many of the problems associated with local minima.

The algorithms introduced in this section generate a labeling that is a local minimum of the energy in (1) for two types of large moves: α-expansion and α-β-swap. In contrast to the standard moves described in Section 2 these moves allow large number of pixels to change their labels simultaneously. This makes the set of labelings within a single move of a locally optimal f exponentially large, and the condition in (5) very demanding. For example, α-expansion moves are so strong that we are able to prove that any labeling locally optimal with respect to these moves is within a known factor of the global minimum (see Section 6). Fig. 1 compares local minima for standard moves (c) and for α-expansion moves (d) obtained from the same initial solution (b). This and other experiments also show that in practice our solutions do not change significantly by varying the initial labelings. In most cases starting from a constant labeling (where all pixels have the same label) is good enough.

In Section 3.1 we discuss the moves we allow, which are best described in terms of partitions. In Section 3.2 we sketch the algorithms and list their basic properties. The main computational step of our algorithms is based on graph cut techniques from combinatorial optimization, which we summarize in Section 3.3.



(a) initial labeling (b) standard move (c) α-β-swap (d) α-expansion

Figure 2: Examples of standard and large moves from a given labeling (a). The number of labels is $|\mathcal{L}| = 3$. A standard move (b) changes a label of a single pixel (in the circled area). Strong moves (c-d) allow large number of pixels to change their labels simultaneously.

3.1 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 $\mathbf{P}'$. A special case of an α-β swap is a move that gives the label α to some set of pixels previously labeled β. One example of α-β swap move is shown in Fig. 2(c).

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. An example of an α-expansion move is shown in Fig. 2(d).

Recall that ICM and annealing use *standard* moves allowing only one pixel to change its intensity. An example of a standard move is given in Fig. 2(b). Note that a move which assigns a given label α to a single pixel is both an α-β swap and an α-expansion. As a consequence, a standard move is a special case of both a α-β swap and an α-expansion.

1. Start with an arbitrary labeling f
2. Set success := 0
3. For each pair of labels $\{\alpha, \beta\} \subset \mathcal{L}$
   3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α-β swap of f
   3.2. If $E(\hat{f}) < E(f)$, set f := $\hat{f}$ 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 $\alpha \in \mathcal{L}$
   3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α-expansion of f
   3.2. If $E(\hat{f}) < E(f)$, set f := $\hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return f

Figure 3: Our swap algorithm (top) and expansion algorithm (bottom).

3.2 Algorithms and properties

We have developed two minimization algorithms. The swap algorithm finds a local minimum when swap moves are allowed and the expansion algorithm finds a local minimum when expansion moves are allowed. Finding such a local minimum is not a trivial task. Given a labeling f, there is an exponential number of swap and expansion moves. Therefore, even checking for a local minimum requires exponential time if performed naïvely. In contrast checking for a local minimum when only the standard moves are allowed is easy since there is only a linear number of standard moves given any labeling f.

We have developed efficient graph based methods to find the optimal α-β-swap or α-expansion given a labeling f (see Sections 4 and 5). This is the key step in our algorithms. Once these methods are available, it is easy to design variants of the "fastest descent" technique that can efficiently find the corresponding local minima. Our algorithms are summarized in Fig. 3.

The two algorithms are quite similar in their structure. 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 algorithm) or for every pair of labels (swap 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 algorithms stop after the first unsuccessful cycle since no further improvement is possible. Obviously, a cycle in the swap algorithm takes $|\mathcal{L}|^2$ iterations, and a cycle in the expansion algorithm takes $|\mathcal{L}|$ iterations.

These algorithms are guaranteed to terminate in a finite number of cycles. In fact, under the assumptions that V and D_p in equation (1) are constants independent of the image size $\mathcal{P}$ we can easily prove termination in $O(|\mathcal{P}|)$ cycles [43]. These assumptions are quite reasonable in practice. However, in the experiments we report in Section 8, the algorithm stops after a few cycles, and most of the improvements occur during the first cycle.

We use graph cuts to efficiently find $\hat{f}$ for the key part of each algorithm in step 3.1. Step 3.1 uses a single graph cut computation. At each iteration the corresponding graph has $O(|\mathcal{P}|)$ pixels. The exact number of pixels, topology of the graph and its edge weights vary from iteration to iteration. The details of the graph are quite different for the swap and the expansion algorithms, and are described in details in Sections 4 and 5.

3.3 Graph cuts

Before describing the key step 3.1 of the swap and the expansion algorithms, we will review graph cuts. 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 cheapest cut among all cuts separating the terminals. Note that we use standard terminology from the combinatorial optimization community.[6]

Sections 4 and 5 show that step 3.1 in Fig. 3 is equivalent to solving the minimum cut problem on an appropriately defined two-terminal graph. Minimum cuts can be efficiently found by standard combinatorial algorithms with different low-order polynomial complexities [1]. For example, a minimum cut can be found by computing the maximum flow between the terminals, according to a theorem due to Ford and Fulkerson [15]. Our experimental results make use of a new max-flow algorithm that has the best speed on our graphs over many modern algorithms [10]. The running time is nearly linear in practice.

[6]To avoid confusion, we would like to mention that some clustering based segmentation techniques in vision use different graph cut terminology. For example, [47] computes a *globally minimum cut*. The minimum is computed among all cuts that sever the graph into two non-empty parts. The terminals need not be specified. [38] introduces *normalized cuts* by proposing a new definition of the cut cost. Although normalized cuts are formulated as a graph partitioning problem the actual approximate optimization is performed via non-combinatorial methods.

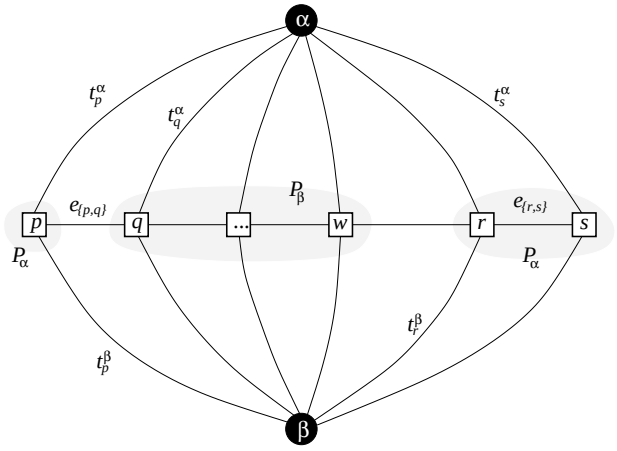


Figure 4: An example of the graph $\mathcal{G}_{\alpha\beta}$ for a 1D image. The set of pixels in the image is $\mathcal{P}_{\alpha\beta} = \mathcal{P}_\alpha \cup \mathcal{P}_\beta$ where $\mathcal{P}_\alpha = \{p, r, s\}$ and $\mathcal{P}_\beta = \{q, \ldots, w\}$.

4 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 swap move algorithm given at the top of Fig. 3. Our technique is based on computing a labeling corresponding to a minimum cut on a 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 4.4 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 Fig. 4. For legibility, this figure shows the case of a 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	$D_p(\alpha) + \sum_{\substack{q \in \mathcal{N}_p \\ q \notin \mathcal{P}_{\alpha\beta}}} V(\alpha, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
t_p^β	$D_p(\beta) + \sum_{\substack{q \in \mathcal{N}_p \\ q \notin \mathcal{P}_{\alpha\beta}}} V(\beta, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
$e_{\{p,q\}}$	$V(\alpha, \beta)$	$\{p,q\} \in \mathcal{N}$ $p,q \in \mathcal{P}_{\alpha\beta}$

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 $\mathcal{G}_{\alpha\beta}$,

$$f_p^{\mathcal{C}} = \begin{cases} \alpha & \text{if } t_p^\alpha \in \mathcal{C} \text{ for } p \in \mathcal{P}_{\alpha\beta} \\ \beta & \text{if } t_p^\beta \in \mathcal{C} \text{ for } p \in \mathcal{P}_{\alpha\beta} \\ f_p & \text{for } p \in \mathcal{P}, p \notin \mathcal{P}_{\alpha\beta}\ . \end{cases} \tag{6}$$

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 4.1 *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 4.2 *For any cut $\mathcal{C}$ and for any n-link $e_{\{p,q\}}$:*

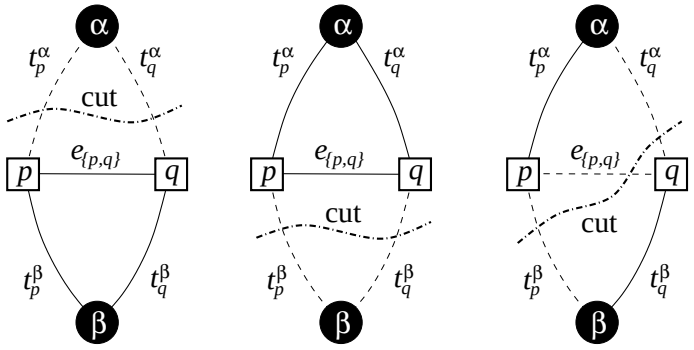
$$\begin{aligned} &(a) \quad \textit{If} \quad t_p^\alpha, t_q^\alpha \in \mathcal{C} \quad \textit{then} \quad e_{\{p,q\}} \notin \mathcal{C}. \\ &(b) \quad \textit{If} \quad t_p^\beta, t_q^\beta \in \mathcal{C} \quad \textit{then} \quad e_{\{p,q\}} \notin \mathcal{C}. \\ &(c) \quad \textit{If} \quad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

Properties (a) and (b) follow from the requirement that no proper subset of $\mathcal{C}$ should separate the terminals. Properties (c) and (d) also use the fact that a cut has to separate the terminals. These properties are illustrated in Fig. 5.

The next lemma is a consequence of property 4.2 and equation 6.

Lemma 4.3 *For any cut $\mathcal{C}$ and for any n-link $e_{\{p,q\}}$*

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



Property 4.2(a) Property 4.2(b) Property 4.2(c,d)

Figure 5: 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 $\mathcal{G}(\mathcal{C}) = \langle \mathcal{V}, \mathcal{E} - \mathcal{C} \rangle$.

PROOF: There are four cases with similar structure; we will illustrate the case where $t_p^\alpha, t_q^\beta \in \mathcal{C}$. In this case, $e_{\{p,q\}} \in \mathcal{C}$ and, therefore, $|\mathcal{C} \cap e_{\{p,q\}}| = |e_{\{p,q\}}| = V(\alpha, \beta)$. As follows from equation (6), $f_p^{\mathcal{C}} = \alpha$ and $f_q^{\mathcal{C}} = \beta$. ■

Note that this proof assumes that V is a semi-metric, i.e. that equations 2 and 3 hold. Lemmas 4.1 and 4.3 plus property 4.2 yield

Theorem 4.4 *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 the n-links that must be cut. We now compute the cost of a cut $\mathcal{C}$, which is

$$|\mathcal{C}| = \sum_{p \in \mathcal{P}_{\alpha\beta}} |\mathcal{C} \cap \{t_p^\alpha, t_p^\beta\}| + \sum_{\substack{\{p,q\} \in \mathcal{N} \\ \{p,q\} \subset \mathcal{P}_{\alpha\beta}}} |\mathcal{C} \cap e_{\{p,q\}}|. \tag{7}$$

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

$$|\mathcal{C} \cap \{t_p^\alpha, t_p^\beta\}| = \begin{cases} |t_p^\alpha| & \text{if } t_p^\alpha \in \mathcal{C} \\ |t_p^\beta| & \text{if } t_p^\beta \in \mathcal{C} \end{cases} = D_p(f_p^{\mathcal{C}}) + \sum_{\substack{q \in \mathcal{N}_p \\ q \notin \mathcal{P}_{\alpha\beta}}} V(f_p^{\mathcal{C}}, f_q).$$

Lemma 4.3 gives the second term in (7). Thus, the total cost of a cut $\mathcal{C}$ is

$$\begin{aligned} |\mathcal{C}| &= \sum_{p \in \mathcal{P}_{\alpha\beta}} D_p(f_p^{\mathcal{C}}) + \sum_{p \in \mathcal{P}_{\alpha\beta}} \sum_{\substack{q \in \mathcal{N}_p \\ q \notin \mathcal{P}_{\alpha\beta}}} V(f_p^{\mathcal{C}}, f_q) + \sum_{\substack{\{p,q\} \in \mathcal{N} \\ \{p,q\} \subset \mathcal{P}_{\alpha\beta}}} V(f_p^{\mathcal{C}}, f_q^{\mathcal{C}}) \\ &= \sum_{p \in \mathcal{P}_{\alpha\beta}} D_p(f_p^{\mathcal{C}}) + \sum_{\substack{\{p,q\} \in \mathcal{N} \\ p \text{ or } q \in \mathcal{P}_{\alpha\beta}}} V(f_p^{\mathcal{C}}, f_q^{\mathcal{C}}). \end{aligned}$$

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

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

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

Corollary 4.5 *The lowest energy labeling within a single α-β swap move from f is $\hat{f} = f^{\mathcal{C}}$, where $\mathcal{C}$ is the minimum cut on $\mathcal{G}_{\alpha\beta}$.*

5 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 expansion move algorithm given at the bottom of Fig. 3. In this section we describe a technique that solves the problem assuming that (each) V is a metric, and thus satisfies the triangle inequality (4). 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. 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 5.4 shows that elementary cuts are in one to one correspondence with those 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 $\hat{f}$ is $f^{\mathcal{C}}$ where $\mathcal{C}$ is a minimum cut on $\mathcal{G}_\alpha$.

The structure of the graph is illustrated in Fig. 6. For legibility, this figure shows the case of a 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

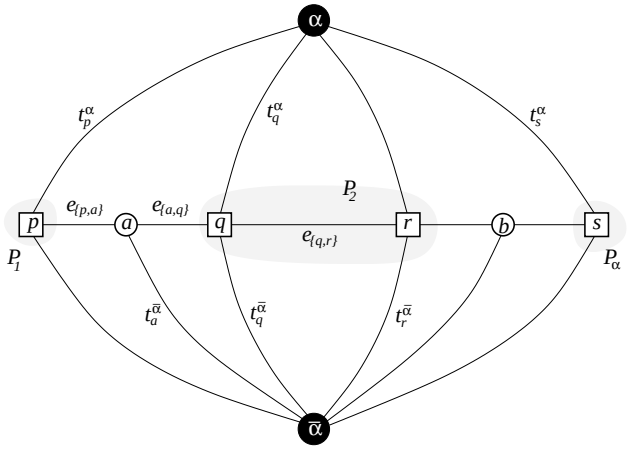


Figure 6: An example of $\mathcal{G}_\alpha$ for a 1D image. The set of pixels in the image is $\mathcal{P} = \{p, q, r, s\}$ and the current partition is $\mathbf{P} = \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_\alpha\}$ where $\mathcal{P}_1 = \{p\}$, $\mathcal{P}_2 = \{q, r\}$, and $\mathcal{P}_\alpha = \{s\}$. 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 $\mathcal{P}_l$.

the current partition (i.e. such that $f_p \neq f_q$), we create an *auxiliary node* $a_{\{p,q\}}$. 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 $t_p^{\bar\alpha}$, respectively. Each pair of neighboring pixels $\{p, q\} \in \mathcal{N}$ which are not separated by the partition $\mathbf{P}$ (i.e. such that $f_p = f_q$) 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 $\mathcal{E}_{\{p,q\}} = \left\{e_{\{p,a\}},\; e_{\{a,q\}},\; t_a^{\bar\alpha}\right\}$ 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}	$D_p(\alpha)$	$p \in \mathcal{P}$
$e_{\{p,a\}}$	$V(f_p, \alpha)$	$\{p,q\} \in \mathcal{N},\ f_p \neq f_q$
$e_{\{a,q\}}$	$V(\alpha, f_q)$	
$t_a^{\bar{\alpha}}$	$V(f_p, f_q)$	
$e_{\{p,q\}}$	$V(f_p, \alpha)$	$\{p,q\} \in \mathcal{N},\ f_p = f_q$

As in Section 4, 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,

$$f_p^{\mathcal{C}} = \begin{cases} \alpha & \text{if } \ t_p^{\alpha} \in \mathcal{C} \\ f_p & \text{if } \ t_p^{\bar{\alpha}} \in \mathcal{C} \end{cases} \qquad \forall p \in \mathcal{P}. \tag{8}$$

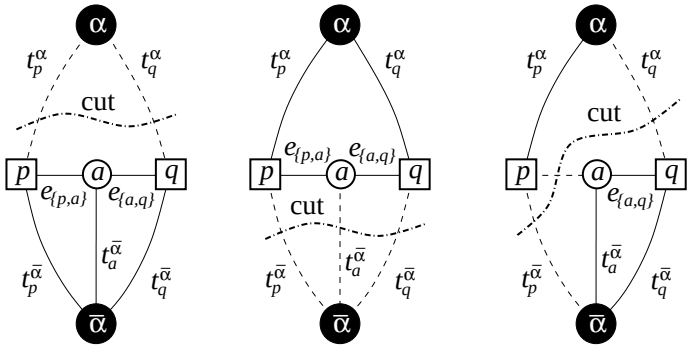
In other words, a pixel p is assigned label α if the cut $\mathcal{C}$ separates p from the terminal α, while p is assigned its old label f_p if $\mathcal{C}$ separates p from $\bar{\alpha}$. Note that for $p \notin \mathcal{P}_\alpha$ the terminal $\bar{\alpha}$ represents labels assigned to pixels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

Lemma 5.1 *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 4.2 holds when we substitute "$\bar{\alpha}$" for "β". We will refer to this as property 4.2($\bar{\alpha}$). Analogously, we can show that property 4.2 and equation (8) establish lemma 4.3 for the n-links $e_{\{p,q\}}$ in $\mathcal{G}_\alpha$.

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 5.2 below. Assume that $a = a_{\{p,q\}}$ is an auxiliary node between the corresponding pair of neighboring pixels.



Property 5.2(a) Property 5.2(b) Property 5.2(c,d)

Figure 7: 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 $\mathcal{G}(\mathcal{C}) = \langle \mathcal{V}, \mathcal{E} - \mathcal{C} \rangle$.

Property 5.2 *If $\{p, q\} \in \mathcal{N}$ and $f_p \neq f_q$ then a minimum cut $\mathcal{C}$ on $\mathcal{G}_\alpha$ satisfies:*

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

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 Fig. 7.

Lemma 5.3 *If $\{p, q\} \in \mathcal{N}$ and $f_p \neq f_q$ then the minimum cut $\mathcal{C}$ on $\mathcal{G}_\alpha$ satisfies*

$$|\mathcal{C} \cap \mathcal{E}_{\{p,q\}}| = V(f_p^{\mathcal{C}}, f_q^{\mathcal{C}}).$$

PROOF: The equation follows from property 5.2, equation (8), and the edge weights. For example, if $t_p^{\bar\alpha}, t_q^{\bar\alpha} \in \mathcal{C}$ then $|\mathcal{C} \cap \mathcal{E}_{\{p,q\}}| = |t_a^{\bar\alpha}| = V(f_p, f_q)$. At the same time, (8) implies that $f_p^{\mathcal{C}} = f_p$ and $f_q^{\mathcal{C}} = f_q$. Note that the right penalty V is imposed whenever $f_p^{\mathcal{C}} \neq f_q^{\mathcal{C}}$, due to the auxiliary pixel construction. ■

Property 4.2($\bar\alpha$) holds for any cut, and property 5.2 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 4.2($\bar\alpha$) and 5.2.

Theorem 5.4 *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 4.2($\bar{\alpha}$) 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 5.2 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

$$|\mathcal{C}| \;=\; \sum_{p\in\mathcal{P}} |\mathcal{C} \cap \{t_p^\alpha, t_p^{\bar{\alpha}}\}| \;+\; \sum_{\substack{\{p,q\}\in\mathcal{N}\\ f_p=f_q}} |\mathcal{C} \cap e_{\{p,q\}}| \;+\; \sum_{\substack{\{p,q\}\in\mathcal{N}\\ f_p\neq f_q}} |\mathcal{C} \cap \mathcal{E}_{\{p,q\}}|. \qquad (9)$$

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 4.3 and 5.3 hold for elementary cuts, since they were based on properties 4.2 and 5.2. Thus, the total cost of a elementary cut $\mathcal{C}$ is

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

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 5.5 *The lowest energy labeling within a single α expansion move from f is $\hat{f} = f^{\mathcal{C}}$, where $\mathcal{C}$ is the minimum cut on $\mathcal{G}_\alpha$.*

6 Optimality properties

Here we discuss optimality properties of our algorithms. In Section 6.1 we show that any local minimum generated by our expansion moves algorithm is within a known factor of the global optimum. This algorithm works in case of metric V. The swap move algorithm can be applied to a wider class of semi-metric V's but, unfortunately, it does not have any (similar) guaranteed optimality properties. In Section 6.2 we show that a provably good solution can be obtained even for semi-metric V by approximating such V's with a simple Potts metric.

6.1 The expansion move algorithm

We now prove that a local minimum when expansion moves are allowed is within a known factor of the global minimum. This factor, which can be as small as 2, will depend on V. Specifically, let

$$c = \frac{\max_{\alpha \neq \beta \in \mathcal{L}} V(\alpha, \beta)}{\min_{\alpha \neq \beta \in \mathcal{L}} V(\alpha, \beta)}$$

be the ratio of the largest non zero value of V to the smallest non zero value of V. Note that c is well defined since $V(\alpha, \beta) \neq 0$ for $\alpha \neq \beta$ according to the metric properties (2) and (3). If $V_{p,q}$'s are different for neighboring pairs p, q then $c = \max_{p,q \in \mathcal{N}} \left(\frac{\max_{\alpha \neq \beta \in \mathcal{L}} V(\alpha,\beta)}{\min_{\alpha \neq \beta \in \mathcal{L}} V(\alpha,\beta)} \right)$.

Theorem 6.1 *Let $\hat{f}$ be a local minimum when the expansion moves are allowed and f^* be the globally optimal solution. Then $E(\hat{f}) \leq 2cE(f^*)$.*

PROOF: Let us fix some $\alpha \in \mathcal{L}$ and let

$$\mathcal{P}_\alpha = \left\{ p \in \mathcal{P} \mid f_p^* = \alpha \right\}. \tag{10}$$

We can produce a labeling f^α within one α-expansion move from $\hat{f}$ as follows:

$$f_p^\alpha = \begin{cases} \alpha & \text{if } p \in \mathcal{P}_\alpha \\ \hat{f}_p & \text{otherwise} \end{cases} \tag{11}$$

The key observation is that since $\hat{f}$ is a local minimum if expansion moves are allowed,

$$E(\hat{f}) \leq E(f^\alpha). \tag{12}$$

Let S be a set consisting of any number of pixels in $\mathcal{P}$ and any number of pairs of neighboring pixels in $\mathcal{N}$. We define $E(f|S)$ to be a restriction of the energy of labeling f to the set S:

$$E(f|S) = \sum_{p \in S} D_p(f_p) + \sum_{\{p,q\} \in S} V(f_p, f_q).$$

Let I^α be the set of pixels and pairs of neighboring pixels contained *inside* $\mathcal{P}_\alpha$. Also, let B^α be the set of pairs of neighboring pixels on the *boundary* of $\mathcal{P}_\alpha$ and O^α be the set of pixels and pairs of neighboring pixels contained *outside* of $\mathcal{P}_\alpha$. Formally,

$$\begin{aligned} I^\alpha &= \mathcal{P}_\alpha \cup \{\{p,q\} \in \mathcal{N} \,:\, p \in \mathcal{P}_\alpha,\ q \in \mathcal{P}_\alpha\}, \\ B^\alpha &= \{\{p,q\} \in \mathcal{N} \,:\, p \in \mathcal{P}_\alpha,\ q \notin \mathcal{P}_\alpha\}, \\ O^\alpha &= (\mathcal{P} - \mathcal{P}_\alpha) \cup \{\{p,q\} \in \mathcal{N} \,:\, p \notin \mathcal{P}_\alpha,\ q \notin \mathcal{P}_\alpha\}. \end{aligned}$$

The following three facts hold:

$$E(f^\alpha|O^\alpha) = E(\hat{f}|O^\alpha), \quad (13)$$

$$E(f^\alpha|I^\alpha) = E(f^*|I^\alpha), \quad (14)$$

$$E(f^\alpha|B^\alpha) \leq cE(f^*|B^\alpha). \quad (15)$$

Equations (13) and (14) are obvious from the definitions in (11) and (10). Equation (15) holds because for any $\{p,q\} \in B^\alpha$ we have $V(f_p^\alpha, f_q^\alpha) \leq cV(f_p^*, f_q^*) \neq 0$.

Since $I^\alpha \cup B^\alpha \cup O^\alpha$ includes all pixels in $\mathcal{P}$ and all neighboring pairs of pixels in $\mathcal{N}$, we can expand both sides of (12) to get:

$$E(\hat{f}|I^\alpha) + E(\hat{f}|B^\alpha) + E(\hat{f}|O^\alpha) \leq E(f^\alpha|I^\alpha) + E(f^\alpha|B^\alpha) + E(f^\alpha|O^\alpha)$$

Using (13), (14) and (15) we get from the equation above:

$$E(\hat{f}|I^\alpha) + E(\hat{f}|B^\alpha) \leq E(f^*|I^\alpha) + cE(f^*|B^\alpha). \quad (16)$$

To get the bound on the total energy, we need to sum equation (16) over all labels $\alpha \in \mathcal{L}$:

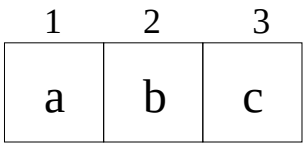
$$\sum_{\alpha\in\mathcal{L}} \left(E(\hat{f}|I^\alpha) + E(\hat{f}|B^\alpha)\right) \leq \sum_{\alpha\in\mathcal{L}} (E(f^*|I^\alpha) + cE(f^*|B^\alpha)) \quad (17)$$

Let $B = \bigcup_{\alpha\in\mathcal{L}} B^\alpha$. Observe that for every $\{p,q\} \in B$, the term $V(\hat{f}_p, \hat{f}_q) = E(\hat{f}|\{p,q\})$ appears twice on the left side of (17), once in $E(\hat{f}|B^\alpha)$ for $\alpha = f_p^*$ and once in $E(\hat{f}|B^\alpha)$ for $\alpha = f_q^*$. Similarly every $V(f_p^*, f_q^*) = E(f^*|\{p,q\})$ appears $2c$ times on the right side of (17). Therefore equation (17) can be rewritten to get the bound of $2c$:

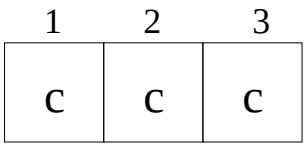
$$E(\hat{f}) + E(\hat{f}|B) \leq E(f^*) + (2c-1)E_B(f^*) \leq 2cE(f^*).$$

■

Note that Kleinberg and Tardos [27] develop an algorithm for minimizing E which also has optimality properties. For the Potts model V discussed in the next section, their algorithm has a bound of 2. This is the same bound as we obtain in Theorem 6.1 for the Potts model.[7] For a general metric V, they have a bound of $O(\log k \log\log k)$ where k is the number of labels. However, their algorithm uses linear programming, which is impractical for the large number of variables occurring in early vision.



(a) Local minimum



(b) Global minimum

	1	2	3
a	0	K	K
b	K	0	K
c	2	2	0

(c) Values of D_p

Figure 8: The image consists of three pixels $\mathcal{P} = \{1, 2, 3\}$. There are two pairs of neighbors $\mathcal{N} = \{\{1, 2\}, \{2, 3\}\}$. The set of labels is $\mathcal{L} = \{a, b, c\}$. D_p is shown in (c). $V(a, b) = V(b, c) = \frac{K}{2}$ and $V(a, c) = K$. It is easy to see that configuration in (a) is a local minimum with the energy of K, while the optimal configuration (b) has energy 4.

6.2 Approximating a semi-metric

A local minimum when the swap moves are allowed can be arbitrarily far from the global minimum. This is illustrated by an example in Fig. 8.

In fact, we can use the expansion algorithm to get an answer within a factor of $2c$ from the optimum of energy (1) even when V is a semi-metric. Here c is the same as in Theorem 6.1. This c is still well defined for a semi-metric. Suppose that penalty V inside the definition of energy E in (1) is a semi-metric. Let r be any real number in the interval $[m, M]$ where

$$m = \min_{\alpha \neq \beta \in \mathcal{L}} V(\alpha, \beta) \qquad \text{and} \qquad M = \max_{\alpha \neq \beta \in \mathcal{L}} V(\alpha, \beta).$$

Define a new energy based on the Potts interaction model

$$E_P(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{\{p,q\} \in \mathcal{N}} r \cdot T(f_p \neq f_q).$$

Theorem 6.2 *If $\hat{f}$ is a local minimum of E_P given the expansion moves and f^* is the global minimum of $E(f)$ then $E(\hat{f}) \leq 2cE(f^*)$.*

PROOF: Suppose f^o is the global minimum of E_P. Then

$$\frac{r}{M} E(\hat{f}) \leq E_P(\hat{f}) \leq 2E_P(f^o) \leq 2E_P(f^*) \leq 2\frac{r}{m} E(f^*)$$

where the second inequality follows from Theorem 6.1. Note that $c = M/m$. ■

[7] In fact, it can be shown that any algorithm that is within a factor of 2 for the Potts model is within a factor of $2c$ for an arbitrary metric V.

Thus to find an answer within a fixed factor from the global minimum for a semi-metric V, one can take a local minimum $\hat{f}$ given the expansion moves for E_P as defined above. Note that such an $\hat{f}$ is not a local minimum of $E(f)$ given the expansion moves. In practice however we find that local minimum given the swap moves gives empirically better results than using $\hat{f}$. In fact, the estimate $\hat{f}$ can be used as a good starting point for the swap algorithm. In this case the swap move algorithm will also generate a local minimum whose energy is within a known factor from the global minimum.

7 The Potts model

An interesting special case of the energy in equation (1) arises when V is given by the Potts model [35]

$$E_P(f) \quad = \quad \sum_{\{p,q\} \in \mathcal{N}} u_{\{p,q\}} \cdot T(f_p \neq f_q) + \sum_{p \in \mathcal{P}} D_p(f_p). \tag{18}$$

Geman *et al.* [18] were the first to use this model in computer vision. In this case, discontinuities between any pair of labels are penalized equally. This is in some sense the simplest discontinuity preserving model and it is especially useful when the labels are unordered or the number of labels is small. The Potts interaction penalty $V_{p,q} = u_{\{p,q\}} \cdot T(f_p \neq f_q)$ is a metric; in this case $c = 1$ and our expansion algorithm gives a solution that is within a factor of 2 of the global minimum. Note that by definition $c \geq 1$, so this is the energy function with the best bound.

Interestingly, the Potts model energy minimization problem is closely related to a known combinatorial optimization problem called the multiway cut problem. In this section we investigate this relationship and its consequences. We will first show (Section 7.1) that the Potts model energy minimization problem can be reduced to the multiway cut problem. More precisely, we prove that the global minimum of the Potts model energy E_P can be computed by finding the minimum cost multiway cut on an appropriately constructed graph. We prove (in the appendix) that if we could efficiently compute the global minimum of E_P we could also solve a certain class of multiway cut problems that are known to be NP-hard. This in turn implies that minimizing E_P is NP-hard, and so is minimizing the energy in (1).

The multiway cut problem is defined on a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ with non-negative edge weights, with a set of terminal vertices $\mathcal{L} \subset \mathcal{V}$. A subset of the edges $\mathcal{C} \subset \mathcal{E}$ is called a *multiway cut* if the terminals are completely separated in the induced graph $\mathcal{G}(\mathcal{C}) = \langle \mathcal{V}, \mathcal{E} - \mathcal{C} \rangle$. We will also require that no proper subset of $\mathcal{C}$ separates the terminals in $\mathcal{G}(\mathcal{C})$. The cost of the multiway cut $\mathcal{C}$ is denoted by $|\mathcal{C}|$ and equals the sum of its edge weights. The *multiway cut problem* is to find the minimum cost multiway cut [13]. In [13] they also show

that the multiway cut problem is NP-complete. Note that the multiway cut problem is a generalization of the standard two-terminal graph cut problem described in Section 3.3.

7.1 The Potts model and the multiway cut problem

We now show that the problem of minimizing the Potts energy $E_P(f)$ can be solved by computing a minimum cost multiway cut on a certain graph. We take $\mathcal{V} = \mathcal{P} \cup \mathcal{L}$. This means that $\mathcal{G}$ contains two types of vertices: *p-vertices* (pixels) and *l-vertices* (labels). Note that l-vertices will serve as terminals for our multiway cut problem. Two p-vertices are connected by an edge if and only if the corresponding pixels are neighbors in the neighborhood system $\mathcal{N}$. The set $\mathcal{E}_{\mathcal{N}}$ consists of the edges between p-vertices, which we will call *n-links*. Each n-link $\{p, q\} \in \mathcal{E}_{\mathcal{N}}$ is assigned a weight $w_{\{p,q\}} = u_{\{p,q\}}$.

Each p-vertex is connected by an edge to each l-vertex. An edge $\{p, l\}$ that connects a p-vertex with a terminal (an l-vertex) will be called a *t-link* and the set of all such edges will be denoted by $\mathcal{E}_{\mathcal{T}}$. Each t-link $\{p, l\} \in \mathcal{E}_{\mathcal{T}}$ is assigned a weight $w_{\{p,l\}} = K_p - D_p(l)$, where $K_p > \max_l D_p(l)$ is a constant that is large enough to make the weights positive. The edges of the graph are $\mathcal{E} = \mathcal{E}_{\mathcal{N}} \cup \mathcal{E}_{\mathcal{T}}$. Fig. 9(a) shows the structure of the graph $\mathcal{G}$.

It is easy to see that there is a one-to-one correspondence between multiway cuts and labelings. A multiway cut $\mathcal{C}$ corresponds to the labeling $f^{\mathcal{C}}$ which assigns the label l to all pixels p which are t-linked to the l-vertex in $\mathcal{G}(\mathcal{C})$. An example of a multiway cut and the corresponding image partition (labeling) is given in Fig. 9(b).

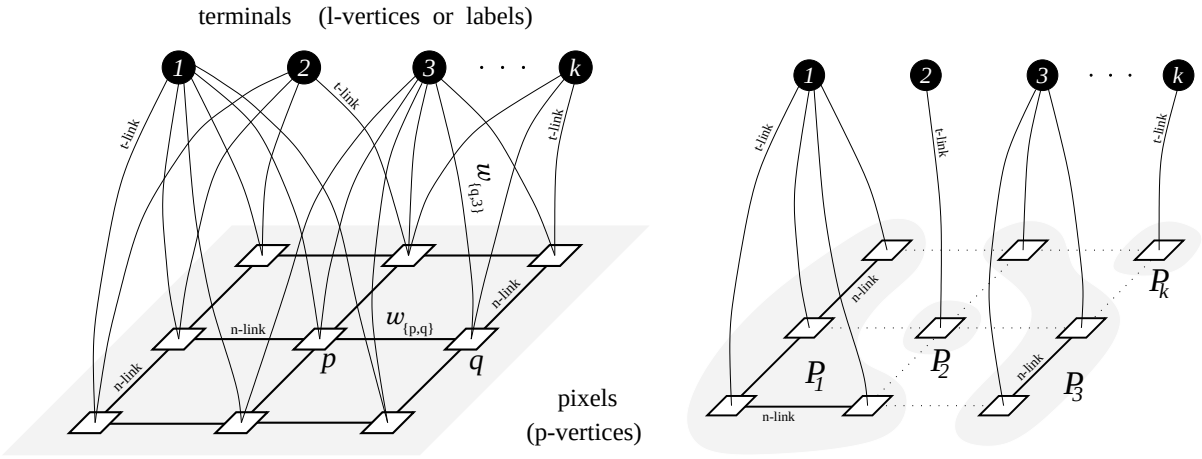
Theorem 7.1 *If $\mathcal{C}$ is a multiway cut on $\mathcal{G}$, then $|\mathcal{C}| = E_P(f^{\mathcal{C}})$ plus a constant.*

The proof of theorem 7.1 is given in [11].

Corollary 7.2 *If $\mathcal{C}$ is a minimum cost multiway cut on $\mathcal{G}$, then $f^{\mathcal{C}}$ minimizes E_P.*

While the multiway cut problem is known to be NP-complete if there are more than 2 terminals, there is a fast approximation algorithm [13]. This algorithm works as follows. First, for each terminal $l \in \mathcal{L}$ it finds an *isolating* two-way minimum cut $\mathcal{C}(l)$ that separates l from all other terminals. This is just the standard graph cut problem. Then the algorithm generates a multiway cut $\mathcal{C} = \cup_{l \neq l_{max}} \mathcal{C}(l)$ where $l_{max} = \arg\max_{l \in \mathcal{L}} |\mathcal{C}(l)|$ is the terminal with the largest cost isolating cut. This "isolation heuristic" algorithm produces a cut which is optimal to within a factor of $2 - \frac{2}{|\mathcal{L}|}$. However, the isolation heuristic algorithm suffers from two problems that limits its applicability to our energy minimization problem.

- The algorithm will assign many pixels a label that is chosen essentially arbitrarily. Note that the union of all isolating cuts $\cup_{l \in \mathcal{L}} \mathcal{C}(l)$ may leave some vertices disconnected



(a) graph $\mathcal{G}$ (b) multiway cut $\Leftrightarrow$ labeling

Figure 9: An example of the graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ with terminals $\mathcal{L} = \{1, \ldots, k\}$ is given in (a). The pixels $p \in \mathcal{P}$ are shown as white squares. Each pixel has an n-link to its four neighbors. Each pixel is also connected to all terminals by t-links (some of the t-links are omitted from the drawing for legibility). The set of vertices $\mathcal{V} = \mathcal{P} \cup \mathcal{L}$ includes all pixels and terminals. The set of edges $\mathcal{E} = \mathcal{E}_{\mathcal{N}} \cup \mathcal{E}_{\mathcal{T}}$ consists of all n-links and t-links. In (b) we show an induced graph $\mathcal{G}(\mathcal{C}) = \langle \mathcal{V}, \mathcal{E} - \mathcal{C} \rangle$ corresponding to some multiway cut $\mathcal{C}$. A multiway cut corresponds to a unique partition (labeling) of image pixels.

from any terminal. The multiway cut $\mathcal{C} = \cup_{l \neq l_{max}} \mathcal{C}(l)$ connects all those vertices to the terminal l_{max}.

- While the multiway cut $\mathcal{C}$ produced is close to optimal, this does not imply that the resulting labeling $f^{\mathcal{C}}$ is close to optimal. Formally, let us write theorem 7.1 as $|\mathcal{C}| = E_P(\mathcal{C}) + K$ (the constant K results from the K_p's, as described in [11]). The isolation heuristic gives a solution $\hat{\mathcal{C}}$ such that $|\hat{\mathcal{C}}| \leq 2|\mathcal{C}^*|$, where $\mathcal{C}^*$ is the minimum cost multiway cut. Thus, $E_P(\hat{\mathcal{C}}) + K \leq 2(E_P(\mathcal{C}^*) + K)$, so $E_P(\hat{\mathcal{C}}) \leq 2E_P(\mathcal{C}^*) + K$. As a result, the isolation heuristic algorithm does not produce a labeling whose energy is within a constant factor of optimal. Note that the K used in the construction given in [11] is so large that this bound is nearly meaningless.

8 Experimental results

In this section we present experimental results on visual correspondence for stereo, motion and image restoration. In image restoration we observe an image corrupted by noise. The task is to restore the original image. Thus the labels are all possible intensities or colors. The restored intensity is assumed to lie around the observed one, and the intensities are expected to vary smoothly everywhere except at object boundaries.

In visual correspondence we have two images taken at the same time from different view points for stereo, and at different times for motion. For most pixels in the first image there is a corresponding pixel in the second image which is a projection along the line of sight of the same real world scene element. The difference in the coordinates of the corresponding points is called the disparity. In stereo the disparity is usually one-dimensional because corresponding points lie along epipolar lines. In motion the disparity is usually two-dimensional. Thus for correspondence the label set is a discretized set of all possible disparities, and the task is to estimate the disparity label for each pixel in the first image.[8] Note that here $\mathcal{P}$ contains the pixels of the first image. The disparity varies smoothly everywhere except at object boundaries, and corresponding points are expected to have similar intensities.

We can formulate the image restoration (Section 8.6) and correspondence problems (Sections 8.3-8.5) as energy minimization problem of the type in equation (1). We describe our

[8]This simple approach does not treat the images symmetrically and allows inconsistent disparities. For example, two pixels in the first image may be assigned to one pixel in the second image. Occlusions are also ignored. [28] presents a stereo algorithm based on expansion moves that addresses these problems.

data terms $D_p(f_p)$ in Section 8.1. We use different interactions $V_{p,q}(f_p, f_q)$ and we state them for each example. Section 8.2 explains static cues that help to set $V_{p,q}$'s.

The corresponding energies are minimized using our swap and expansion algorithms given in Fig. 3. Optimal swap and expansion moves (step 3.1 in Fig. 3) are found by computing minimum cost cuts on graphs designed in Sections 4 and 5. Our implementation computes minimum cuts using a new max-flow algorithm [10]. Running times presented below were obtained on a 333MHz Pentium III.

8.1 Data term

For image restoration our data term is straightforward. Suppose I is the observed image, and I_p is the intensity observed at pixel $p \in \mathcal{P}$. Then $D_p(f_p) = \min(|f_p - I_p|^2, const)$, which says that the restored intensity label f_p should be close to the observed intensity I_p. We set parameter $const = 20$, and it is used to make the data penalty more robust against outliers, i.e. pixels which do not obey the assumed noise model. The algorithm is very stable with respect to *const* which simply helps to smooth out the few outlying pixels. For example if we set *const* to infinity, the results are mostly the same except they become speckled by a few noisy pixels.

Now we turn to the data term for the stereo correspondence problem. Suppose the first image is I and the second is I'. If the pixels p and q correspond, they are assumed to have similar intensities I_p and I'_q. However there are special circumstances when corresponding pixels have very different intensities due to the effects of image sampling. Suppose that the true disparity is not an integer, and the disparity range is discretized to one pixel accuracy, as we do here. If a pixel overlaps a scene patch with high intensity gradient, then the corresponding pixels may have significantly different intensities.

For stereo we use the technique of [6] to develop a D_p that is insensitive to image sampling. First we measure how well p fits into the real valued range of disparities $(d - \frac{1}{2}, d + \frac{1}{2})$ by

$$C_{fwd}(p, d) = \min_{d-\frac{1}{2} \le x \le d+\frac{1}{2}} |I_p - I'_{p+x}|.$$

We get fractional values I'_{p+x} by linear interpolation between discrete pixel values. For symmetry we also measure

$$C_{rev}(p, d) = \min_{p-\frac{1}{2} \le x \le p+\frac{1}{2}} |I_x - I'_{p+d}|.$$

$C_{fwd}(p, d)$ and $C_{rev}(p, d)$ can be computed with just a few comparisons. The final measure is $C(p, d) = (\min\{C_{fwd}(p, d), C_{rev}(p, d), const\})^2$. We set $const = 20$ for all experiments, and its purpose and effect is the same as those described for the image restoration.

For motion we developed $D_p(f_p)$ similar to stereo, except interpolation is done in two dimensions since labels are now two dimensional. Details are given in [43].

8.2 Static cues

In the visual correspondence there is contextual information which we can take advantage of. For simplicity we will consider the case of the Potts model, i.e. $V_{p,q} = u_{\{p,q\}} \cdot T(f_p \neq f_q)$. The intensities of pixels in the first image contain information that can significantly influence our assessment of disparities without even considering the second image. For example, two neighboring pixels p and q are much more likely to have the same disparity if we know that $I(p) \approx I(q)$. Most methods for computing correspondence do not make use of this kind of contextual information. Some exceptions include [5, 33, 45].

We can easily incorporate contextual information into our framework by allowing $u_{\{p,q\}}$ to vary depending on the intensities I_p and I_q. Let

$$u_{\{p,q\}} = U(|I_p - I_q|). \tag{19}$$

Each $u_{\{p,q\}}$ represents a penalty for assigning different disparities to neighboring pixels p and q. The value of the penalty $u_{\{p,q\}}$ should be smaller for pairs $\{p,q\}$ with larger intensity differences $|I_p - I_q|$. In practice we found the following simple function to work well:

$$U(|I_p - I_q|) = \begin{cases} 2K & \text{if } |I_p - I_q| \leq 5 \\ K & \text{if } |I_p - I_q| > 5 \end{cases} \tag{20}$$

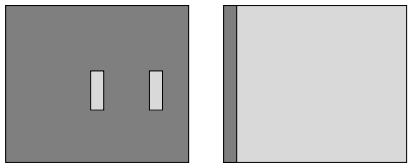
Here K is the Potts model parameter. Note that instead of (19) we could also set the coefficients $u_{\{p,q\}}$ according to an output of an edge detector on the first image. For example, $u_{\{p,q\}}$ can be made small for pairs $\{p,q\}$ where an intensity edge was detected and large otherwise. Segmentation results can also be used.

The following example shows the importance of contextual information. Consider the pair of synthetic images below, with a uniformly white rectangle in front of a black background.

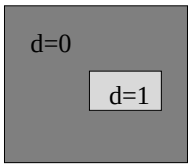
First image Second image

There is a one pixel horizontal shift in the location of the rectangle, and there is no noise. Without noise, the problem of estimating f is reduced to minimizing the smoothness term $E_{smooth}(f)$ under the constraint that pixel p can be assigned disparity d only if $I_p = I'_{p+d}$.

If $u_{\{p,q\}}$ is the same for all pairs of neighbors $\{p, q\}$ then $E_{smooth}(f)$ is minimized at one of the labeling shown in the picture below. Exactly which labeling minimizes $E_{smooth}(f)$ depends on the relationship between the height of the square and the height of the background.



Suppose now that the penalty $u_{\{p,q\}}$ is much smaller if $I_p \neq I_q$ than it is if $I_p = I_q$. In this case the minimum of $E_{smooth}(f)$ is achieved at the disparity configuration shown in the picture below. This result is much closer to human perception.

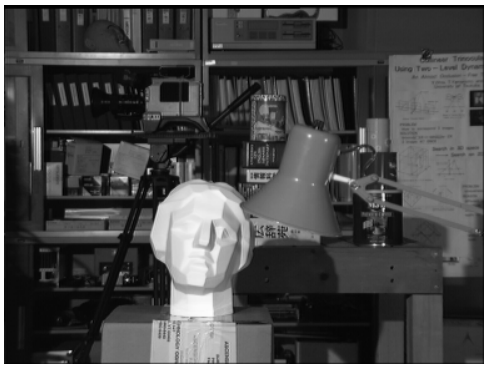


Static cues help mostly in areas of low texture. Application on real images show that the static cues give improvement, but not as extreme as the example above. See Section 8.3 for the improvements that the static cues give on real images.

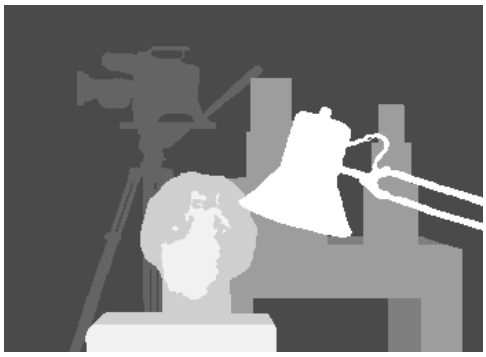
8.3 Real stereo imagery with ground truth

In Fig. 10 we show results from a real stereo pair with known ground truth, provided by Dr. Y. Ohta and Dr. Y. Nakamura from the University of Tsukuba. The left image is in Fig. 10(a), and the ground truth is in Fig. 10(b). The maximum disparity for this stereo pair is 14, so our disparity label set is $\{0, 1, \ldots, 14\}$. The ground truth image actually has only 7 distinct disparities. The objects in this scene are fronto-parallel to the camera, so the Potts model, i.e. $V_{p,q}(f_p, f_q) = u_{\{p,q\}} \cdot T(f_p \neq f_q)$ works well. Since there are textureless regions in the scene, the static cues help, and the coefficients $u_{\{p,q\}}$ are given by equations (19) and (20).

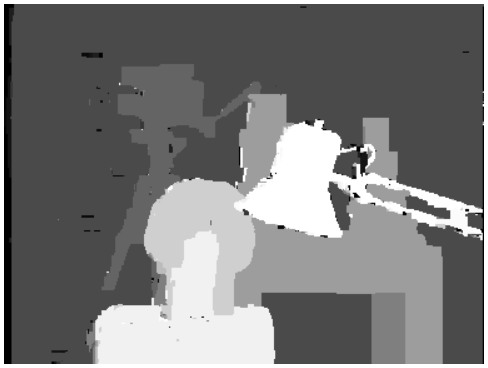
We compared our results against annealing and normalized correlation. For normalized correlation we chose parameters which give the best statistics. We implemented several different annealing variants, and used the one that gave the best performance. This was the Metropolis sampler with a linearly decreasing temperature schedule. To give it a good starting point, simulated annealing was initialized with the results from normalized correlation. In contrast for our algorithms the starting point is unimportant. The results differ by less than 1% of image pixels from any starting point that we have tried. We also run 100 tests with randomly generated initial labelings. Final solutions produced by our expansion and



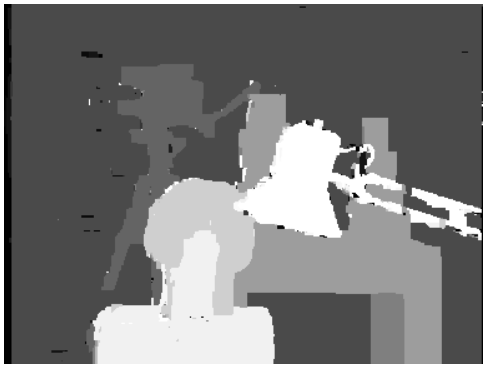
(a) Left image: 384x288, 15 labels



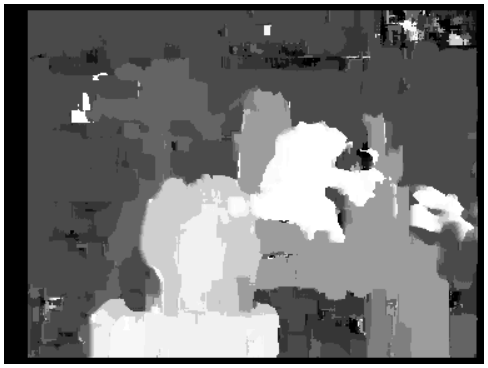
(b) Ground truth



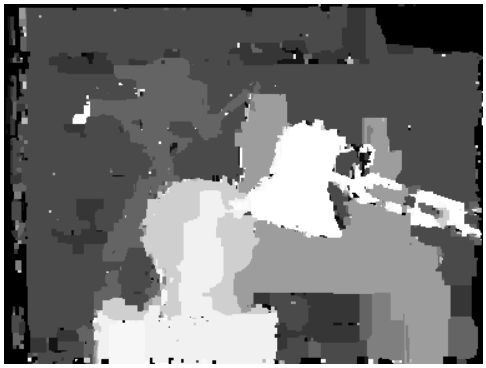
(c) Swap algorithm



(d) Expansion algorithm



(e) Normalized correlation



(f) Simulated annealing

Figure 10: Real imagery with ground truth

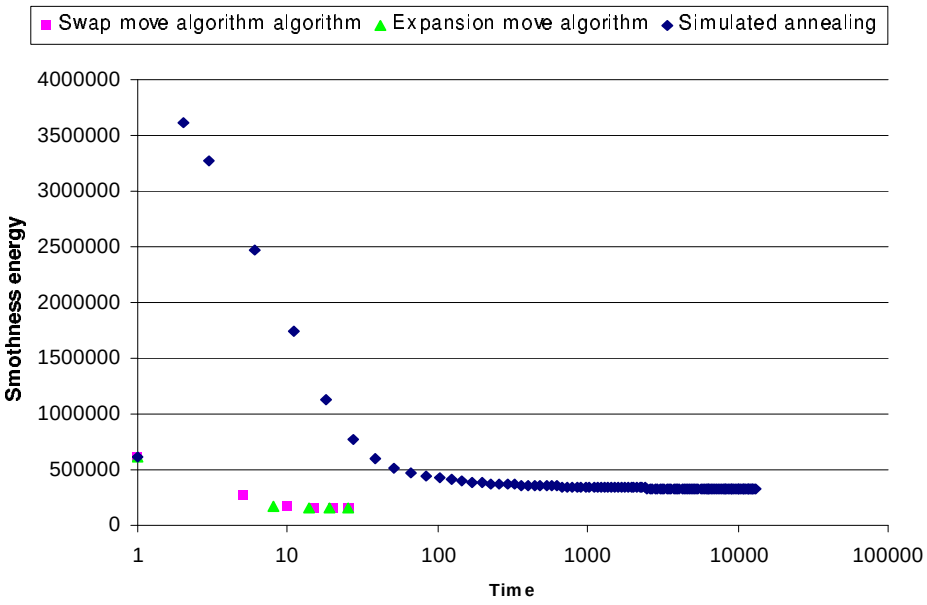


Figure 11: Energy versus time (in seconds) of expansion, swap, and simulated annealing algorithms for the problem in Fig. 10(a). Starting energy is equal for all algorithms.

swap algorithms had the average energy of 252, 157 and 252, 108, correspondingly, while the standard deviations were only 1308 and 459.

Figs. 10(c), and (d) show the results of the swap and expansion algorithms for $K = 20$, where K is the parameter in equation (20). Figs. 10(e) and (f) show the results of normalized correlation and simulated annealing. Comparisons with other algorithms can be found in [40]. Note, however, that [40] confirms that for this imagery the best previous algorithm is simulated annealing, which outperforms (among others) correlation, robust estimation, scanline-based dynamic programming, and mean-field techniques.

The table below summarizes the errors made by the algorithms. In approximately 20 minutes simulated annealing reduces the total errors normalized correlation makes by about one fifth and it cuts the number of ± 1 errors in half. It makes very little additional progress in the rest of 4 hours. Our expansion and swap algorithms make approximately 5 times fewer ± 1 errors and approximately 3 times fewer total errors compared to normalized correlation.

The expansion and swap algorithms perform similarly to each other. The observed difference in errors is insignificant, less than 1%. At each cycle the order of labels to iterate over is chosen randomly. Another run of the algorithm might give slightly different results, and on average about 1% of pixels change their labels between different runs. The expansion

algorithm	**% total errors**	**% of errors $> \pm 1$**	**time**
expansion algorithm (after iteration)	7.9	2.7	8 sec
expansion algorithm (at convergence)	7.2	2.1	25 sec
expansion algorithm (no static cues)	7.6	2.5	25 sec
swap algorithm (at convergence)	7.0	2.0	35 sec
simulated annealing	20.3	5.0	1200 sec
normalized correlation	24.7	10.0	2 sec

Figure 12: Comparison of accuracy and running times.

algorithm	total E	E_{smooth}	E_{data}
expansion algorithm	253,700	157,740	95,960
swap algorithm	251,990	158,660	93,330
simulated annealing	442,000	332,100	109,900

Figure 13: Energies at convergence for our algorithms and simulated annealing.

algorithm converges 1.4 times faster than the swap algorithm, on average.

Fig. 11 shows the graph of E_{smooth} versus time (in seconds) for our algorithms and simulated annealing. Note that the time axis is on a logarithmic scale. We do not show the graph for E_{data} because the difference in the E_{data} among all algorithms is insignificant, as expected from the following argument. Most pixels in real images have nearby pixels with similar intensities. Thus for most pixels p there are several disparities d for which $D_p(d)$ is approximately the same and small. For the rest of d's, $D_p(d)$ is quite large. This latter group of disparities is essentially excluded from consideration by energy minimizing algorithms. The remaining choices of d are more or less equally likely. Thus the E_{data} term of the energy function has very similar values for our methods and simulated annealing. Our methods quickly reduce the smoothness energy to around $160,000$, while the best simulated annealing can produce in 4 hours is around $330,000$, which is twice as bad. The expansion algorithm gives a convergence curve significantly steeper than the other curves. In fact the expansion algorithm makes 99% of the progress in the first iteration which takes 8 seconds. Final energies are given in Fig. 13.

Static cues help in the upper right textureless corner of the image. Without the static cues, a corner of size approximately 800 pixels gets broken off and is assigned to the wrong disparity. This is reflected in the error count shown in Fig. 12, which worsens without the static cues. The percentage improvement may not seem too significant, however visually it

K	% of total errors	% of errors $> \pm 1$	Absolute average error
5	13.0	4.5	0.27
10	7.0	2.3	0.15
20	7.6	2.1	0.15
30	7.9	2.3	0.17
50	8.8	2.3	0.18
100	10.4	2.9	0.21
500	16.3	8.2	0.37

Figure 14: Table of errors for the expansion algorithm for different values of K.

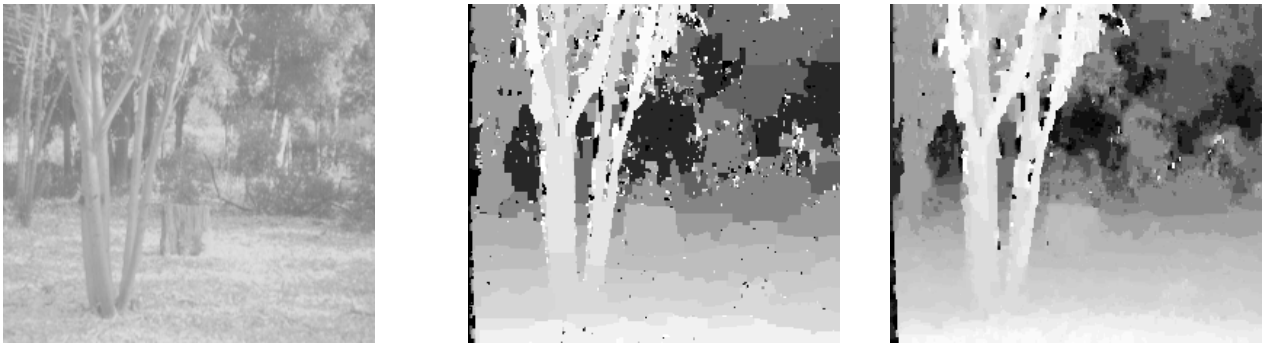
# labels	1 iteration	convergence	% of total errors	% of errors $> \pm 1$
15	8 sec	27 sec	7.3	2.1
30	19 sec	78 sec	7.5	2.3
45	25 sec	78 sec	7.9	2.5
60	31 sec	101 sec	7.4	2.4
75	35 sec	122 sec	8.3	2.3

Figure 15: Dependence of the running time and accuracy on different number of labels (disparities) for the expansion algorithm. Error percentages are given at the convergence.

is very noticeable, since without the static cues a large block of pixels is misplaced. We omit the actual image due to space constraints.

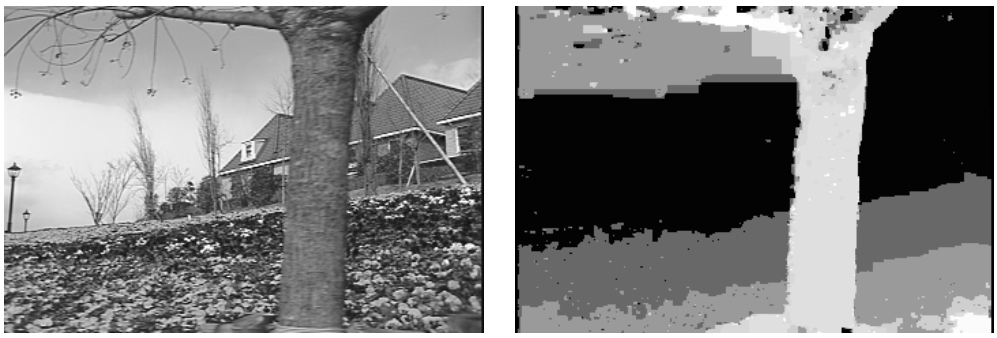
The only parameter of the this energy function is K in equation (20). The algorithms appear stable in the choice of K. The table in Fig. 14 gives the errors made by the expansion algorithm for different K's. For small K there are many errors because the data term is overemphasized, for large K there are many errors because the smoothness term is overemphasized. However for a large interval of K values the results are good.

Another important test is to increase the number of labels and evaluate the effects on the running time and the accuracy of our algorithms. Fig. 15 summarizes the test results for the expansion algorithm (those for the swap algorithm are similar). The first column shows the number of integer disparities that we use. The second and third columns show the time it took to complete one iteration and to converge, correspondingly. The last two columns give the error counts at convergence. The second and third columns confirm that the running time is linear on average. Note that the number of cycles to convergence varies, explaining higher variability in the third column. The last two columns show that the accuracy worsens slightly with the increase in the number of labels.



(a) Left image: 256x233, 29 labels (b) Piecewise constant model (c) Piecewise smooth model

Figure 16: Tree stereo pair.

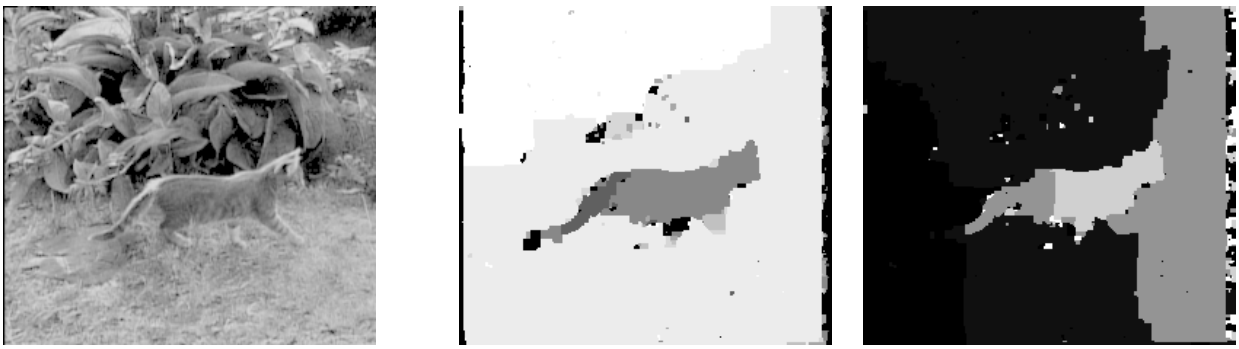


(a) First image, 352x240, 8 labels (b) Horizontal movement

Figure 17: Flower garden sequence

8.4 SRI tree stereo pair

In the SRI stereo pair whose left image is shown in Fig. 16(a) the ground is a slanted surface, and therefore a piecewise constant model (Potts model) does not work as well. For this image pair, we choose $V_{p,q}(f_p, f_q) = 15 \cdot \min(3, |f_p - f_q|)$, which is a piecewise smooth model. It is a metric and so we use the expansion algorithm for minimization. This scene is well textured, so static cues are not used. Fig. 16(b) and (c) compares the results of minimizing with the Potts and piecewise smooth model. The running times to convergence are 94 seconds and 79 seconds respectively. Notice that there are fewer disparities found in Fig. 16(b), since the Potts model tends to produce large regions with the same disparity.



(a) First image: 256x223, 40 labels (b) Horizontal movement (c) Vertical movement

Figure 18: Moving cat

8.5 Motion

Fig. 17 shows the output from the well-known flower garden sequence. Since the camera motion is nearly horizontal, we have simply displayed the camera motion. The motion in this sequence is large, with the foreground tree moving 6 pixels in the horizontal direction. We used the Potts model in this example because the number of labels is small. This image sequence is relatively noisy, so we took $K = 80$. Determining the motion of the sky is a very hard problem in this sequence. Even static cues do not help, so we didn't use them. The running time is 15 seconds to convergence.

Fig. 18(a) shows one image of a motion sequence where a cat moves against moving background. The motion is large, with maximum horizontal displacement of 4 pixels and maximum vertical displacement of 2 pixels. We tested 8 horizontal and 5 vertical displacements, thus the label set has size 40. This is a difficult sequence because the cat's motion is non-rigid. The scene is well-textured, so the static cues are not used. In this case we chose $V_{p,q}(f_p, f_q) = 40 \cdot \min(8, (f_p^h - f_q^h)^2 + (f_p^v - f_q^v)^2)$, where f_p^h and f_p^v are horizontal and vertical components of the label f_p (recall that the labels have two dimensions for motion). This is not a metric, so we used the swap algorithm for minimization. Figs. 18(b) and (c) show the horizontal and vertical motions detected with our swap algorithm. Notice that the cat has been accurately localized. Even the tail and parts of the legs are clearly separated from the background motion. The running time was 24 seconds to convergence.

(a) noisy image (b) truncated abs. diff. (c) abs. diff.

Figure 19: Image restoration. The results in (b,c) are histogram equalized to reveal oversmoothing in (c), which does not happen in (b).

8.6 Image Restoration

In this section we illustrate the importance of discontinuity preserving energy functions on the task of image restoration. Fig. 19 shows image consisting of several regions with constant intensities after it was corrupted by $N(0, 100)$ noise. Fig. 19(b) shows our image restoration results for the truncated absolute difference model $V(f_p, f_q) = 80 \cdot \min(3, |f_p - f_q|)$, which is discontinuity preserving. Since it is a metric, we used the expansion algorithm. For comparison, Fig. 19(c) shows the result for the absolute difference model $V(f_p, f_q) = 15 \cdot |f_p - f_q|$, which is not discontinuity preserving. For the absolute difference model we can find the exact solution using the graph-cut method in [37, 24, 11]. For both models we chose parameters which minimize the average absolute error from the original image intensities. These average errors were 0.34 for the truncated and 1.8 for the absolute difference model, and the running times were 38 and 237 seconds, respectively. The results in Fig. 19(b,c) were histogram equalized to reveal oversmoothing in (c), which does not happen in (b). Similar oversmoothing for the absolute difference model occurs in stereo, see [43, 7].

9 Conclusions

We consider a wide class of energy functions with various discontinuity preserving smoothness constraints. While it is NP-hard to compute the exact minimum, we developed two algorithms based on graph cuts that efficiently find a local minimum with respect to two large moves, namely α-expansion and α-β-swap. Our α-expansion algorithm finds a labeling within a known factor of the global minimum, while our α-β-swap algorithm handles more general energy functions. Empirically, our algorithms performs well on a variety of computer

vision problems such as image restoration, stereo, and motion. We believe that combinatorial optimization techniques, such as graph cuts, will prove to be powerful tools for solving many computer vision problems.

Appendix: Minimizing the Potts energy is NP-hard

In Section 7 we showed that the problem of minimizing the energy in equation (18) over all possible labelings f can be solved by computing a minimum multiway cut on a certain graph. Now we make the reduction in the opposite direction. Let $E_P(f)$ denote the energy in equation (18). For an arbitrary fixed graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ we will construct an instance of minimizing $E_P(f)$ where the optimal labeling f^* determines a minimum multiway cut on $\mathcal{G}$. This will prove that a polynomial-time method for finding f^* would provide a polynomial-time algorithm for finding the minimum cost multiway cut, which is known to be NP-hard [13]. This NP-hardness proof is based on a construction due to Jon Kleinberg.

The energy minimization problem we address takes as input a set of pixels $\mathcal{P}$, a neighborhood relation $\mathcal{N}$ and a label set $\mathcal{L}$, as well as a set of weights $u_{\{p,q\}}$ and a function $D_p(l)$. The problem is to find the labeling f^* that minimizes the energy $E_P(f)$ given in equation (18).

Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ be an arbitrary weighted graph with terminal vertices $\{t_1, \ldots, t_k\} \subset \mathcal{V}$ and edge weights $w_{\{p,q\}}$. We will do the energy minimization using $\mathcal{P} = \mathcal{V}$, $\mathcal{N} = \mathcal{E}$, and $u_{\{p,q\}} = w_{\{p,q\}}$. The label set will be $\mathcal{L} = \{1, \ldots, k\}$. Let K be a constant such that $K > E_P(f^*)$; for example, we can select K to be the sum of all $w_{\{p,q\}}$. Our function $D_p(l)$ will force $f^*(t_j) = j$; if $p = t_j$ is a terminal vertex,

$$D_p(l) = \begin{cases} 0 & l = j, \\ K & \text{otherwise.} \end{cases}$$

For a non-terminal vertex p we set $D_p(l) = 0$ for all l, which means all labels are equally good. We define a labeling f to be *feasible* if the set of pixels labeled j by f forms a connected component that includes t_j. Feasible labelings obviously correspond one-to-one with multiway cuts.

Theorem 9.1 *The labeling f^* is feasible, and the cost of a feasible labeling is the cost of the corresponding multiway cut.*

PROOF: To prove that f^* is feasible, suppose that there were a set S of pixels that f^* labeled j which were not part of the component containing t_j. We could then obtain a labeling with lower energy by switching this set to the label of some pixel on the boundary of S. The energy of a feasible labeling f is $\sum_{\{p,q\} \in \mathcal{N}} u_{\{p,q\}} \cdot T(f(p) \neq f(q))$, which is the cost of the multiway cut corresponding to f. ■

This shows that minimizing the Potts model energy $E_P(f)$ on an arbitrary $\mathcal{P}$ and $\mathcal{N}$ is intractable. It is possible to extend this proof to the case when $\mathcal{P}$ is a planar grid, see [43].

Acknowledgements

We thank J. Kleinberg, D. Shmoys and E. Tardos for providing important input on the content of the paper. This research has been supported by DARPA under contract DAAL01-97-K-0104, by NSF awards CDA-9703470 and IIS-9900115, and by a grant from Microsoft.

References

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

[2] Amir Amini, Terry Weymouth, and Ramesh Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867, September 1990.

[3] S.A. Barker and P.J.W. Rayner. Unsupervised image segmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 2757–2760, 1998.

[4] J. Besag. On the statistical analysis of dirty pictures (with discussion). *Journal of the Royal Statistical Society, Series B*, 48(3):259–302, 1986.

[5] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. *IJCV*, 35(3):1–25, December 1999.

[6] Stan Birchfield and Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, April 1998.

[7] Stanley Birchfield. *Depth and Motion Discontinuities.* PhD thesis, Stanford Universiy, June 1999. Available from http://vision.stanford.edu/~birch/publications/.

[8] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, 1987.

[9] Andrew Blake. Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):2–12, January 1989.

[10] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, number 2134 in LNCS, pages 359–374, Sophia Antipolis, France, September 2001. Springer-Verlag.

[11] 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.

[12] P.B. Chou and C.M. Brown. The theory and practice of Bayesian image labeling. *International Journal of Computer Vision*, 4(3):185–210, 1990.

[13] E. Dahlhaus, D. S. Johnson, C.H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiway cuts. In *ACM Symposium on Theory of Computing*, pages 241–251, 1992.

[14] P. Ferrari, A. Frigessi, and P. de Sá. Fast approximate maximum a posteriori restoration of multicolour images. *Journal of the Royal Statistical Society, Series B*, 57(3):485–500, 1995.

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

[16] Yoram Gdalyahu, Daphna Weinshall, and Michael Werman. Self-organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10), October 2001.

[17] Davi Geiger and Alan Yuille. A common framework for image segmentation. *International Journal of Computer Vision*, 6(3):227–243, 1991.

[18] D. Geman, S. Geman, C. Graffigne, and P. Dong. Boundary detection by constrained optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):609–628, July 1990.

[19] 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.

[20] 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.

[21] W. Eric L. Grimson and Theo Pavlidis. Discontinuity detection for visual surface reconstruction. *Computer Vision, Graphics and Image Processing*, 30:316–330, 1985.

[22] B. K. P. Horn and B. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[23] R. Hummel and S. Zucker. On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:267–287, 1983.

[24] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *European Conference on Computer Vision*, pages 232–248, 1998.

[25] Hiroshi Ishikawa. *Global Optimization Using Embedded Graphs.* PhD thesis, New York University, May 2000. Available from http://cs1.cs.nyu.edu/phd_students/ishikawa/index.html.

[26] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[27] Jon Kleinberg and Eva Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov Random Fields. In *IEEE Symposium on Foundations of Computer Science*, pages 14–24, 1999.

[28] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions via graph cuts. In *International Conference on Computer Vision*, volume II, pages 508–515, 2001.

[29] David Lee and Theo Pavlidis. One dimensional regularization with discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):822–829, November 1988.

[30] S. Li. *Markov Random Field Modeling in Computer Vision.* Springer-Verlag, 1995.

[31] G. Parisi. *Statistical field theory.* Addison-Wesley, Reading MA, 1988.

[32] M. Pelillo. The dynamics of nonlinear relaxation labeling processes. *Journal of Mathematical Imaging and Vision*, 7:309–323, 1997.

[33] T. Poggio, E. Gamble, and J. Little. Parallel integration of vision modules. *Science*, 242:436–440, October 1988. See also E. Gamble and T. Poggio, MIT AI Memo 970.

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

[35] R. Potts. Some generalized order-disorder transformation. *Proceedings of the Cambridge Philosophical Society*, 48:106–109, 1952.

[36] A. Rosenfeld, R.A. Hummel, and S.W. Zucker. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(6):420–433, June 1976.

[37] S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *International Conference on Computer Vision*, 1998.

[38] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

[39] R.H. Swendson and J. Wang. Nonuniversal critical dynamics in monte carlo simulations. *Physical Review Letters*, 58(2):86–88, 1987.

[40] Richard Szeliski and Ramin Zabih. An experimental comparison of stereo algorithms. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, number 1883 in LNCS, pages 1–19, Corfu, Greece, September 1999. Springer-Verlag.

[41] R.S. Szeliski. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 5(3):271–302, December 1990.

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

[43] Olga Veksler. *Efficient Graph-based Energy Minimization Methods in Computer Vision.* PhD thesis, Cornell University, July 1999. Available from www.neci.nj.nec.com/homepages/olga.

[44] Olga Veksler. Image segmentation by nested cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 339–344, 2000.

[45] Y. Weiss and E.H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 321–326, 1996.

[46] Gerhard Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods.* Springer-Verlag, 1995.

[47] Z. Wu and R. Leahy. 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, November 1993.