Graph Cut Based Optimization for MRFs with Truncated Convex Priors*

Olga Veksler
University of Western Ontario
London, Canada
olga@csd.uwo.ca

Abstract

Optimization with graph cuts became very popular in recent years. Progress in problems such as stereo correspondence, image segmentation, etc., can be attributed, in part, to the development of efficient graph cut based optimization. Recent evaluation of optimization techniques shows that the popular expansion and swap graph cut algorithms perform extremely well for energies where the underlying MRF has the Potts prior, which corresponds to the assumption that the true labeling is piecewise constant. For more general priors, however, such as corresponding to piecewise smoothness assumption, both swap and expansion algorithms do not perform as well. We develop several optimization algorithms for truncated convex priors, which imply piecewise smoothness assumption. Both expansion and swap algorithms are based on moves that give each pixel a choice of only two labels. Our insight is that to obtain a good approximation under piecewise smoothness assumption, a pixel should have a choice among more than two labels. We develop new "range" moves which act on a larger set of labels than the expansion and swap algorithms. We evaluate our method on problems of image restoration, inpainting, and stereo correspondence. Our results show that we are able to get more accurate answers, both in terms of the energy, which is the direct goal, and in terms of accuracy, which is an indirect, but more important goal.

1. Introduction

In recent years, energy optimization with graph cuts [6, 10, 7, 18] has become very popular in computer vision and graphics fields. Optimization with graph cuts has been successfully applied to image restoration [7], stereo and multi-view reconstruction [6, 16, 7, 17], motion [28], texture synthesis [19], segmentation [4, 3, 22, 14], digital photomontage [1]. Optimization with graph cuts is successful, in large part, because either the exact minimum or an approximate minimum with certain quality guarantees can be found, unlike the older optimization techniques, such as Simulated Annealing [8] or ICM [2]

Usually, the following energy function is minimized:

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

In Eq. (1), $\mathcal{L}$ is a finite set of labels, representing the property needed to be estimated at each pixel, such as intensity, color, etc. $\mathcal{P}$ is the set of sites that one needs to assign labels to. Often $\mathcal{P}$ is set of image pixels. The label assigned to pixel $p \in \mathcal{P}$ is denoted by f_p, and f is the collection of all assignments: $f = \{f_p | p \in \mathcal{P}\}$. The first sum in Eq. (1) is usually called the data term, because it represents the preferences of individual pixels for their labels, and it is expressed as the sum over all pixels of individual pixel preferences $D_p(f_p)$. The data term usually comes from the observed data. The second sum in Eq. (1) is usually called the smoothness term, and it represents our prior knowledge about what the likely labelings f should be like. The second term is the sum over interacting ordered pixel pairs $(p, q) \in \mathcal{N}$. Usually $\mathcal{N}$ is the 4 or 8 connected grid, however longer range interactions are also useful [16]. Without loss of generality, we assume that if $(p, q) \in \mathcal{N}$ then $p < q$.

The energy function in Eq. (1) arises in Maximum A Posteriori (MAP) estimation in Markov Random Fields (MRF). The labeling f minimizing the energy gives the maximum of the posterior distribution $p(f|D)$, where D is the observed data. In the MRF framework, the data term comes from the likelihood of the data, and the smoothness term comes from the prior distribution $p(f)$.

Usually there are no restrictions on D_p's. The V_{pq}'s (often called the *interaction penalties*) are typically used to specify the smoothness assumptions on the labeling f. Different choices of V_{pq}'s correspond to different smoothness assumptions. A common choice is the Potts model, which is $V_{pq}(f_p, f_q) = w_{pq} \cdot \min\{1, |f_p - f_q|\}$. Here w_{pq}'s are per pixel pair weights that can be different for different pixel pairs. Potts model penalizes any difference between f_p and f_q by the same amount. Intuitively, it corresponds to the expectation that f should be piecewise constant, that is to

*This work was supported in part by NSERC Canada.

consist of several pieces where pixels inside the same piece share the same label. This is because the Potts model treats small or large differences in labels in the same way.

Other common choices for V_{pq} are $V_{pq}(f_p, f_q) = w_{pq} \cdot \min\{T, |f_p - f_q|\}$ and $V_{pq}(f_p, f_q) = w_{pq} \cdot \min\{T, (f_p - f_q)^2\}$, the truncated linear and truncated quadratic interactions, respectively. These V_{pq}'s correspond to the piecewise smooth assumption on f, that is the assumption that f consists of several pieces, where the labels between neighboring pixels inside each piece differ just a little, or vary "smoothly"[1]. This is because smaller differences in labels cause smaller penalties, and larger differences cause larger penalties. Therefore a labeling f is encouraged to have just a few places where nearby pixels change their labels significantly. Most nearby pixels are expected to have similar labels. It is important to limit the penalty from above by a *truncation* constant T. Otherwise $|l_1 - l_2|$ or $(l_1 - l_2)^2$ might be prohibitively large, and assigning labels l_1 and l_2 to neighboring pixels will be too costly, resulting in an oversmoothed labeling f. V_{pq}'s which do not lead to oversmoothing are typically called discontinuity-preserving. Without the truncation, that is if V_{pq} is the absolute linear or quadratic difference, the energy in Eq. (1) can be optimized exactly with a graph cut [9], but the corresponding energies are not discontinuity preserving. Energy in Eq. (1) is NP-hard to optimize if discontinuity preserving Potts, truncated linear or quadratic V_{pq}'s are used [7].

Recently, Szeliski *et al*. [27] performed an experimental evaluation of several energy minimization methods that have been extensively used to for minimizing energies given by Eq. (1) : the graph cut based expansion and swap algorithms [7], sequential tree-reweighted message passing (TRW-S) [13], and loopy belief propagation (LBP) [20]. Their results show that for the Potts model, both expansion and swap algorithms perform extremely well, with the expansion algorithm slightly outperforming the swap algorithm. Both expansion and swap algorithms get the answer within a small percentage of the global minimum (the lower bound on the global minimum is estimated by the TRW-S algorithm [13]). TRW-S performs as well as graph cuts, but takes significantly longer to converge. An additional benefit of graph cuts over TRW-S is when longer range interactions are present in $\mathcal{N}$. Szeliski *et al*. [27] studied only the case when $\mathcal{N}$ is the 4-connected grid. Kolmogorov and Rother [15] performed a comparison between graph cuts and TRW-S when longer range interactions are present in $\mathcal{N}$, and they concluded that graph cuts perform significantly better in terms of energy than TRW-S in this case.

For the truncated linear V_{pq}'s the expansion and swap algorithms still perform relatively well, but for the truncated quadratic V_{pq} the energy value is noticeably worse than that of TRW-S. In this paper, we develop several optimization algorithms for the truncated linear and quadratic V_{pq}. In general, our methods can be used for truncated convex priors, we define what we mean by a truncated convex prior in Section 3. Informally, truncated convex priors correspond to piecewise smoothness assumption on f. Our insight is that both expansion and swap algorithms give a pixel a choice of only two labels, but for problems where piecewise smoothness assumptions are appropriate, to obtain a good approximation, a pixel should have a choice among more than two labels. We develop new "range" moves which act on a larger set of labels than the expansion and swap moves.

Note that Kleinberg and Tardos [11] develop an algorithm for minimization with truncated linear V_{pq}, with an optimality factor $O(\log k \log\log k)$, where $k = |\mathcal{L}|$. However, they use linear programming, which is impractical for the computer vision problems even of moderate size.

We evaluate our method on problems of image restoration, inpainting, and stereo correspondence, and our results show that we are able to get more accurate answers, both in terms of energy (which is the direct goal), and in terms of accuracy (which is an indirect, but more important goal).

2. Energy Optimization with Graph Cuts

In this section, we briefly explain the prior work on optimization with graph cuts.

2.1. Assumptions on the Label Set

For the rest of the paper we assume that the labels can be represented as integers in the range $\{0, 1, ...k\}$, which is necessary since we base our method on the construction in [9]. Assuming integer labels rules out directly using our methods for motion estimation, since in motion, labels are two dimensional. However, there are indirect ways to apply our methods to motion, by fixing one component of a motion vector and letting the other one vary [23].

2.2. Convex Priors

Ishikawa [9] develops a method to find the exact minimum of the energy function in Eq. (1) in the case when the interaction terms V_{pq} are convex functions of the label differences. His definition of convexity in the discrete setting is as follows. An interaction penalty $V_{pq}(l_1, l_2) = w_{pq} \cdot g(l_1 - l_2)$ is said to be convex if and only if for any integer x, $g(x+1) - 2g(x) + g(x-1) \geq 0$. It is assumed that $g(x)$ is symmetric[2], otherwise it can be replaced with $\frac{g(x)+g(-x)}{2}$ without changing the labeling that minimizes it. Note that [26] extend the results in [9] to handle a more general definition of convexity.

Convex V_{pq} include the absolute and squared difference interactions as a special case. While the energy arising from convex priors is not discontinuity preserving, Ishikawa's construction gives us an important tool for energy optimization with truncated convex priors, which do lead to discontinuity preserving energies. Since our construction is based

[1] The term "smoothly" is used informally here.

[2] A function is symmetric if $g(x) = g(-x)$.

on Ishikawa's, we first explain Ishikawa's construction in detail. Note that the construction we give in this section is slightly different from that in [9].

Ishikawa's method is based on computing a minimum cut in a particular graph. There are two special nodes in the graph, the source s and the sink t. For each $p \in \mathcal{P}$, we create a set of nodes $p_0, p_1, ..., p_{k+1}$. We identify p_0 with the source s, and we identify p_{k+1} with the sink t. We connect node p_i to node p_{i+1} with a directed edge e_i^p for $i = 0, 1, ..., k$. In addition, for $i = 0, 1, ..., k$, node p_{i+1} is connected to node p_i with a directed edge of infinite weight. This ensures that for each p, only one of the edges of type e_i^p will be in the minimum cut, see [9]. If an edge e_i^p is cut, then pixel p is assigned label i. Thus a cut C of finite cost corresponds to a labeling f^C in a unique way.

Furthermore, for any $(p, q) \in \mathcal{N}$, an edge e_{ij}^{pq} which connects node p_i to node q_j is created for $i = 0, ..., k+1$ and $j = 0, ..., k+1$. The weight of this edge is

$$w(e_{ij}^{pq}) = \frac{w_{pq}}{2}[g(i-j+1) - 2g(i-j) + g(i-j-1)]. \quad (2)$$

The edge weight defined by Eq. (2) is non-negative, since $g(x)$ is convex. This is important, since graph-cut/max-flow algorithms require non-negative edge weights.

Let C be a cut of finite cost. Let $(p, q) \in \mathcal{N}$. If edges e_i^p and e_j^q are in the cut C, then all the edges in the set S_{pq}^{ij}, defined below, also have to be in the cut C.

$$S_{pq}^{ij} = \{e_{lm}^{pq} | 0 \le l \le i, j+1 \le m \le k+1\} \cup$$
$$\{e_{lm}^{pq} | i+1 \le l \le k+1, 0 \le m \le j\}.$$

When summing up over S_{pq}^{ij}, most of edge weights cancel out, and we are left with

$$\sum_{e \in S_{pq}^{ij}} w(e) = w_{pq}[g(i-j) + g(k+2) + h(i) + h(j)],$$

where $h(i) = -\frac{1}{2}[g(k+1-i) + g(i+1)]$. Recall that the cut C corresponds to a labeling f^C. Except some extra terms, the sum above is almost exactly $V_{pq}(i, j) = V_{pq}(f_p^C, f_q^C) = w_{pq} \cdot g(i-j)$. The term $g(k+2)$ can be ignored since it is a constant and does not change the minimum cut, just its cost. Terms $h(i)$ and $h(j)$ can be subtracted from the costs of edges e_i^p and e_j^p. Therefore we define the weights e_i^p as follows:

$$w(e_i^p) = D_p(i) - \sum_{q \in \mathcal{N}_p} w_{pq} \cdot h(i),$$

where $\mathcal{N}_p$ is the set of neighbors of pixel p. Under this edge weights assignment, the cost of any finite cut C is exactly $E(f^C)$ plus a constant. Therefore the minimum cut gives the optimal labeling.

For the absolute linear V_{pq} this construction leads to a graph with $O(|\mathcal{P}| \cdot |\mathcal{L}|)$[3] vertices and edges, assuming 4-connected grid. This is because the edges of type e_{ij}^{pq} have zero weight unless $i = j$. For more general V_{pq}, for example the squared difference V_{pq}, the number of vertices is still $O(|\mathcal{P}| \cdot |\mathcal{L}|)$, but the number of edges is $O(|\mathcal{P}| \cdot |\mathcal{L}|^2)$.

Note that [12] develops an algorithm for minimizing energy with convex V_{pq} which is more memory and time efficient. However it can be used only when the D_p's are convex.

2.3. Expansion and Swap Algorithms

Boykov *et al*. [7] develop the expansion and swap algorithms based on graph cuts for minimizing the energy in Eq. (1). These methods can be applied when V_{pq} is Potts, truncated linear or quadratic, but the answer is only approximate, which is not surprising, since the energy is NP-hard to optimize under these interaction penalties [7]. However, in case of the Potts V_{pq}, the expansion algorithm gives an answer within a factor of 2 from the optimal [7], although in practice, the answer is much closer to the optimal [27].

Both the expansion and the swap algorithms find a local minimum of the energy function in the following sense. For each f, we define a set of "moves" $M(f)$. We say that f is a local minimum with respect to the set of moves, if $E(f') \ge E(f)$ for any $f' \in M(f)$. A move from f to f' is *standard* if there is at most one pixel p s.t. $f_p \ne f'_p$. The number of standard moves is $O(|\mathcal{P}||\mathcal{L}|)$, therefore an optimal standard move is easy to compute.

Swap moves are defined as follows. Given a labeling f and a label pair (α, β), a move from f to f' is called an α-β swap if $f_p \ne f'_p \Rightarrow f_p, f'_p \in \{\alpha, \beta\}$. That is an α-β swap reassign labels α, β among pixels that are labeled either α or β in f. $M(f)$ is then defined as the collection of α-β swaps for all pairs of labels $\alpha, \beta \in \mathcal{L}$.

The expansion moves are defined as follows. Given a labeling f and a label α, a move f' is called an α-expansion if $f_p \ne f'_p \Rightarrow f'_p = \alpha$. That is the set of pixels assigned α can only expand from f to f'. $M(f)$ is then defined as the collection of α-expansions for all labels $\alpha \in \mathcal{L}$.

The expansion and swap algorithm finds a local minimum with respect to expansion or swap moves, correspondingly. In either case, the number of moves to search for the best one is exponential in the number of pixels, and exhaustive search is ruled out. In [7] they describe how to compute, for a given a labeling f, the optimal α-expansion and the optimal α-β swap with the minimum cut on a certain graph. The conditions on V_{pq} for the expansion or swap algorithm to work were generalized from those in [7] by [18].

According to [18], the swap algorithm may be used whenever $V_{pq}(\alpha, \alpha) + V_{pq}(\beta, \beta) \le V_{pq}(\alpha, \beta) + V_{pq}(\beta, \alpha)$ for all $\alpha, \beta \in \mathcal{L}$. The expansion algorithm may be used whenever $V_{pq}(\alpha, \alpha) + V_{pq}(\beta, \gamma) \le V_{pq}(\alpha, \gamma) + V_{pq}(\beta, \alpha)$ for all $\alpha, \beta, \gamma \in \mathcal{L}$. Therefore the requirements for expansion algorithm are more strict than those for the swap algorithm. For example, the energy with truncated linear V_{pq}

[3] Here $|S|$ denotes the size of set S.

can be optimized by both expansion and swap algorithms, whereas for truncated quadratic V_{pq}, only the swap algorithm applies. In practice, however, it is possible to apply the expansion algorithm with a "truncation trick" [21] to energies which do not satisfy the necessary inequality above. The resulting labeling is no longer guaranteed to be a local minimum with the respect to expansion moves, but the energy is guaranteed to go down.

Unlike the algorithm in [9], the expansion and swap algorithms are iterative. We start with an initial labeling f. Then we iterate until convergence over labels $\alpha \in \mathcal{L}$ for the expansion and over pairs of $\alpha, \beta \in \mathcal{L}$ for the swap algorithm. At each iteration, we find the optimal α-expansion (or α-β-swap) from our current labeling, and then replace our current labeling with it.

3. Graph Cuts for Truncated Convex Priors

In this section, we describe our algorithm for the case when V_{pq} is given by a truncated convex prior. We use the truncated version of Ishikawa's [9] interaction penalty. That is we assume that $V_{pq}(f_p, f_q)$ only depends on the label difference $f_p - f_q$. We define an interaction penalty to be truncated convex if there exists a symmetric function $g(x)$ such that $g(x+1) - 2g(x) + g(x-1) \geq 0$ and

$$V_{pq}(l_1, l_2) = w_{pq} \cdot \min\{g(l_1 - l_2), T\}. \qquad (3)$$

Throughout this section and the rest of the paper, we will assume that the energy in Eq. (1) has truncated convex interaction penalties defined by Eq. (3).

To motivate our multi-label moves, we will prove a theorem below, but first we need some notation and definitions.

Given a subset $\mathcal{T} \subset \mathcal{P}$, let

$$E_{\mathcal{T}}(f) = \sum_{p \in \mathcal{T}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}, \{p,q\} \cap \mathcal{T} \neq \emptyset} V_{pq}(f_p, f_q).$$

In words, $E_{\mathcal{T}}(f)$ is the sum all the terms of the energy function which depend on pixels in $\mathcal{T}$.

Let f and f' be two labelings. Given a subset of pixels $\mathcal{T} \subset \mathcal{P}$, we use notation $comb(f, f', \mathcal{T})$ to denote a labeling s.t. $comb(f, f', \mathcal{T})_p = f_p$ if $p \notin \mathcal{T}$ and $comb(f, f', \mathcal{T})_p = f'_p$ if $p \in \mathcal{T}$. That is $comb(f, f', \mathcal{T})$[4] is equal to f for pixels not in $\mathcal{T}$ and it is equal to f' for pixels in $\mathcal{T}$.

Definition Given a subset of pixels $\mathcal{T} \subset \mathcal{P}$ and a labeling f, we define a *sublabeling* $f_{\mathcal{T}}$ as $f_{\mathcal{T}} = \{f_p | p \in \mathcal{T}\}$, that is $f_{\mathcal{T}}$ is the labeling f restricted to the set of pixels $\mathcal{T}$. □

Definition Given a labeling f and a subset $\mathcal{T} \subset \mathcal{P}$, $f_{\mathcal{T}}$ is called a *smooth sublabeling*, if for every $p \in \mathcal{T}$, if its neighbor q is also in $\mathcal{T}$, then there is a sequence of pixels $p = r_0, r_1, ..., r_m = q$ s.t. $r_i \in \mathcal{T}$ for all i, r_i is a neighbor of r_{i+1} for $i = 0, ..., m-1$ and $|f_{r_i} - f_{r_{i+1}}| \leq T$. □

Theorem Let $\hat{f}$ be a labeling such that for any labeling f' and any $\mathcal{T} \subset \mathcal{P}$ s.t. $f'_{\mathcal{T}}$ is a smooth sublabeling, $E(comb(\hat{f}, f', \mathcal{T})) \geq E(\hat{f})$ (i.e. $\hat{f}$ is a local minimum with respect to smooth sublabelings). Then $E(\hat{f}) \leq 2E(f^*)$, where f^* is the global minimum of the energy.

Proof: Let f^* be the optimal labeling. We can split $\mathcal{P}$ into disjoint subsets $\mathcal{T}_1, \mathcal{T}_2, ...\mathcal{T}_t$ s.t. $\mathcal{P} = \bigcup_{i=1,...,t} \mathcal{T}_i$ and for every i, $f^*_{\mathcal{T}_i}$ is a smooth sublabeling. There are many possible such decompositions of $\mathcal{P}$. Let us take the decomposition which has the smallest number of subsets $\mathcal{T}_i$, that is t is as small as possible. Notice that this implies that for any $i \neq j$, and any $p \in \mathcal{T}_i$, $q \in \mathcal{T}_j$ s.t. p and q are neighbors, $|f^*_p - f^*_q| > T$. Otherwise we could obtain a smaller decomposition by uniting sets $\mathcal{T}_i$ and $\mathcal{T}_j$. Since $\hat{f}$ is a local minimum with respect to smooth sublabelings, $E(comb(\hat{f}, f^*, \mathcal{T}_i)) \geq E(\hat{f})$. To simplify the notation, let us use h_i to denote the labeling $comb(\hat{f}, f^*, \mathcal{T}_i)$. We have that $E_{\mathcal{T}_i}(h_i) \geq E_{\mathcal{T}_i}(\hat{f})$, since labelings $\hat{f}$ and h_i are identical outside the set $\mathcal{T}_i$. Furthermore, $E_{\mathcal{T}_i}(h_i) \leq E_{\mathcal{T}_i}(f^*)$, since h_i and f^* are identical inside the set $\mathcal{T}_i$, and $V_{pq}(f^*_p, f^*_q) = T$, its maximum value, for neighboring pixels p and q s.t. exactly one of p and q is in $\mathcal{T}_i$. Therefore, $E_{\mathcal{T}_i}(\hat{f}) \leq E_{\mathcal{T}_i}(f^*)$. Summing up over all i, we get: $E(\hat{f}) \leq \sum_i E_{\mathcal{T}_i}(\hat{f}) \leq \sum_i E_{\mathcal{T}_i}(f^*) \leq 2E(f^*)$ □

The theorem above implies that to get a good approximation to the optimal energy, we should be trying to find moves that are 'smooth', where a move from f to f' is smooth if $f'_{\mathcal{T}}$ is a smooth sublabeling, where $\mathcal{T} = \{p | f_p \neq f'_p\}$. This means that we should be developing moves where each pixel can choose from one of several labels, and not just one out of two labels, like with the expansion or swap algorithms. In the next 2 sections we develop such moves.

3.1. α-β Range Moves

Recall that the label set is $\mathcal{L} = \{0, 1, ..., k\}$. Let $\mathcal{L}_{\alpha\beta} = \{\alpha, \alpha+1, ..., \beta\}$, where $\alpha < \beta \in \mathcal{L}$. That is $\mathcal{L}_{\alpha\beta}$ is the subset of labels containing consecutive integers. Given a labeling f, we say that f' is an α-β *range* move from f, if $f_p \neq f'_p \Rightarrow \{f_p, f'_p\} \subset \mathcal{L}_{\alpha\beta}$. The α-β range moves can be viewed as a generalization of α-β swap moves. An α-β swap move reassigns labels α, β among pixels that are currently labeled α and β. An α-β range move reassigns the labels in the range $\{\alpha, \alpha+1, .., \beta\}$ among the pixels that currently have labels in the range $\{\alpha, \alpha+1, .., \beta\}$.

Of course, if we knew how to find the best α-β range move for $\alpha = 0$ and $\beta = k$, we would find the global optima of the energy function, which is impossible since the problem is NP-hard, in general. However, we can find the best α-β range move from f if $|\alpha - \beta| \leq T$.

3.2. α-β Range Moves for $|\alpha - \beta| \leq T$

To simplify the description, we will assume that $|\alpha - \beta| = T$, but it is trivial to extend the construction in this section to handle the case when $|\alpha - \beta| < T$. Suppose that we are given a labeling f and we wish to find

[4] Word "comb" stands for "combination"

the optimal α-β range move, where $|\alpha - \beta| = T$. The graph construction is similar to that in Section 2.2. Let $\mathcal{T} = \{p|\alpha \leq f_p \leq \beta\}$. Notice that the truncated convex terms V_{pq} become convex when $p, q \in \mathcal{T}$, since for any $p, q \in \mathcal{T}$, $V_{pq}(f_p, f_q) = w_{pq}g(f_p - f_q) \leq w_{pq} \cdot T$.

We identify label set $\mathcal{L}_{\alpha\beta}$ with label set $\{0, 1, ..., T\}$ and we employ the construction in Section 2.2 but only on the pixels in the subset $\mathcal{T}$ and with a slight twist. First we need more notation. Given a $\mathcal{T} \in \mathcal{P}$, let

$$E^{open}_{\mathcal{T}}(f) = \sum_{p \in \mathcal{T}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}, \{p,q\} \subset \mathcal{T}} V_{pq}(f_p, f_q).$$

In words, $E^{open}_{\mathcal{T}}(f)$ is the sum of all the terms of the energy function which depend *only* on pixels in $\mathcal{T}$. Note that $E^{open}_{\mathcal{T}}(f) \neq E_{\mathcal{T}}(f)$ in most cases.

Let $M^{\alpha\beta}(f) = \{f' | f'_p \neq f_p \Rightarrow f'_p, f_p \in \mathcal{L}_{\alpha\beta}\}$. That is $M^{\alpha\beta}(f)$ is exactly the set of all α-β range moves from labeling f. If we directly use the construction in Section 2.2 on pixels in $\mathcal{T}$ and labels in $\mathcal{L}_{\alpha\beta}$ then we will find the $f' \in M^{\alpha\beta}(f)$ s.t. $E^{open}_{\mathcal{T}}(f')$ is as small as possible. However, we actually want to find $f' \in M^{\alpha\beta}(f)$ that makes $E^{\mathcal{T}}(f')$ is as small as possible. This is since for $f' \in M^{\alpha\beta}(f)$,

$$E(f') = E_{\mathcal{T}}(f') + E^{open}_{\mathcal{P}-\mathcal{T}}(f') = E_{\mathcal{T}}(f') + E^{open}_{\mathcal{P}-\mathcal{T}}(f), \quad (4)$$

where $\mathcal{P} - \mathcal{T}$ denotes set difference. Thus the labeling $f' \in M^{\alpha\beta}(f)$ which minimizes $E_{\mathcal{T}}(f')$ gives the biggest decrease in energy from f to f' among all $f' \in M^{\alpha\beta}(f)$. Notice that since $f \in M^{\alpha\beta}(f)$, $f' \in M^{\alpha\beta}(f)$ and minimizing $E_{\mathcal{T}}(f')$ is guaranteed to have $E(f') \leq E(f)$.

Essentially, the problem is that the construction in Section 2.2 does not consider the effect of terms V_{pq} on the boundary of $\mathcal{T}$, that is those V_{pq} for which we have $|\{p, q\} \cap \mathcal{T}| = 1$. This boundary problem is easy to fix. For each pixel $p \in \mathcal{T}$, if there is a neighboring pixel $q \notin \mathcal{T}$, we add to the weight of edge e^p_i an additional cost which equals to $V_{pq}(i, f_q)$, for all $i = 0, 1, ..., k$. Recall that we identified the label set $\{\alpha, \alpha + 1, ...\beta\}$ with the label set $\{0, 1, ..., T\}$. Therefore $V_{pq}(i, f_q) = V_{pq}(i + \alpha, f_q)$. This additional weight to edges e^p_i makes sure that the terms V_{pq} on the boundary of $\mathcal{T}$ are accounted for. Now this fixed construction will find the $f' \in M^{\alpha\beta}(f)$ which optimizes $E_{\mathcal{T}}(f')$.

Just as with α-β swaps, the algorithm starts at some labeling f. Then it iterates over a set of label ranges $\{\alpha, .., \beta\}$ with $|\alpha - \beta| = T$, finding the best α-β range move f' and switching the current labeling to f'.

3.3. α-β Generalized Range Moves

We can slightly generalize the construction in the previous section as follows. As previously, let $|\alpha - \beta| = T$ (the case of $|\alpha - \beta| < T$ is basically identical) and, as before, let $\mathcal{T} = \{p|\alpha \leq f_p \leq \beta\}$. Let

$$\mathcal{L}_{\alpha\beta t} = \{\alpha - t, \alpha - t + 1, ..., \beta + t - 1, \beta + t\} \cap \mathcal{L},$$

that is $\mathcal{L}_{\alpha\beta t}$ extends the range of $\mathcal{L}_{\alpha\beta}$ by t in each direction, making sure that the resulting range is still a valid range of labels in $\mathcal{L}$. Let

$$M^{\alpha\beta t}(f) = \{f' | f'_p \neq f_p \Rightarrow f_p \in \mathcal{L}_{\alpha\beta}, f'_p \in \mathcal{L}_{\alpha\beta t}\}.$$

That is $M^{\alpha\beta t}(f)$ is a set of moves that change pixels labels from $\mathcal{L}_{\alpha\beta}$ to labels in $\mathcal{L}_{\alpha\beta t}$. Notice that $M^{\alpha\beta}(f) \subset M^{\alpha\beta t}(f)$. We actually will not be able to find the optimal move in $M^{\alpha\beta t}(f)$, but we can find $\hat{f} \in M^{\alpha\beta t}(f)$ s.t. $E(\hat{f}) \leq E(f^*)$, where f^* is the optimal move in $M^{\alpha\beta}(f)$. Thus labeling $\hat{f}$ is not worse than the optimal move in $M^{\alpha\beta}(f)$, and if we are lucky, $E(\hat{f})$ could be significantly better than the optimal move in $M^{\alpha\beta}(f)$.

We use basically the same construction as in Section 3.2. We construct a graph for pixels in $\mathcal{T} = \{p|\alpha \leq f_p \leq \beta\}$. However, the label range is now $\mathcal{L}_{\alpha\beta t}$, and as before, we identify it with label set $\{0, 1, ..., |\mathcal{L}_{\alpha\beta t}| - 1\}$. Otherwise, the graph construction is identical to that in Section 3.2.

Let C be any finite cost cut in our graph. Notice that a cut of finite cost assigns labels (as described in Section 2.2) only to pixels in $\mathcal{T}$. Let f^C be the labeling corresponding to the cut C, which we define as follows: $f^C_p = f_p$ for $p \notin \mathcal{T}$, and for $p \in \mathcal{T}$, f^C_p is equal to the label assigned to pixel p by the cut C. Let $w(C)$ be the cost of cut C. By graph construction, $w(C) = E^{\neg T}_{\mathcal{T}}(f^C) + K$, where K is a constant and $E^{\neg T}(f)$ is the same as $E(f)$, except there is no truncation in V_{pq} terms for $p, q \in \mathcal{T}$. That is for $p, q \in \mathcal{T}$, $V_{pq}(f_p, f_q) = w_{pq} \cdot g(f_p - f_q)$ in the energy $E^{\neg T}$. Note also that for any labeling f', $E_{\mathcal{T}}(f') \leq E^{\neg T}_{\mathcal{T}}(f')$, since the only difference between E and $E^{\neg T}$ is that the V_{pq} terms are not truncated in $E^{\neg T}$ for $p, q \in \mathcal{T}$. Also note that for any labeling $f' \in M^{\alpha\beta}(f)$, $E_{\mathcal{T}}(f') = E^{\neg T}_{\mathcal{T}}(f')$, since V_{pq} terms in $f' \in M^{\alpha\beta}(f)$ do not need to be truncated on the set $\mathcal{T}$.

Let $\hat{C}$ be the minimum cost cut, and let $\hat{f}$ be its corresponding labeling, defined as above. Let f' be a labeling in $M^{\alpha\beta}(f)$, and let C' be the cut which corresponds to it in the graph. Notice that this cut has finite cost. We have that $E^{\neg T}_{\mathcal{T}}(\hat{f}) + K = w(\hat{C}) \leq w(C') = E^{\neg T}_{\mathcal{T}}(f') + K$. Since $E_{\mathcal{T}}(\hat{f}) \leq E^{\neg T}_{\mathcal{T}}(\hat{f})$ and $E_{\mathcal{T}}(f') = E^{\neg T}_{\mathcal{T}}(f')$, we get that $E_{\mathcal{T}}(\hat{f}) \leq E_{\mathcal{T}}(f')$. Now, for any labeling f'', $E(f'') = E_{\mathcal{T}}(f'') + E_{\mathcal{P}-\mathcal{T}}(f'')$. We have that $E_{\mathcal{P}-\mathcal{T}}(\hat{f}) = E_{\mathcal{P}-\mathcal{T}}(f')$, and therefore we get the desired result: $E(\hat{f}) \leq E(f^*)$ where f^* is the optimal move in $M^{\alpha\beta}(f)$.

In practice we found that it is enough to set t to a small constant. This is because the larger is the value of t, the more the graph construction overestimates the interaction penalty $V_{pq}(f_p, f_q)$, and it is too costly to assign labels f_p and f_q under such overestimated cost. Using small t saves computational time, especially for the truncated quadratic model, since the size of the graph is quadratic in the number of labels in the truncated quadratic case.

4. Experimental Results

For all the experiments presented in this section, we use the α-β generalized range moves presented in Section 3.3, with $t = 3$ for all the experiments. For our implementation, we used the max-flow algorithm in [5]. We evaluate the performance of our algorithm on the problems of image restoration, image inpainting, and stereo correspondence.

4.1. Image Restoration

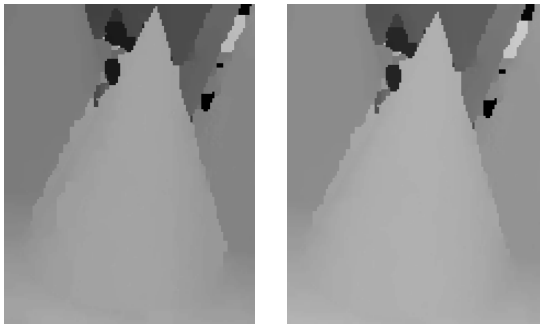
In image restoration, we want to reconstruct the original image from the given noisy one. In this case, $\mathcal{P}$ is the set of all image pixels, $\mathcal{L}$ is the set of all gray levels, that is $\mathcal{L} = \{0, 1, ..., 255\}$. We set $D_p(f_p) = (I_p - f_p)^2$, where I_p is the intensity of pixel p in the given noisy image. We used truncated quadratic $V_{pq}(f_p, f_q) = 8 \cdot \min\{(f_p - f_q)^2, 50\}$.

Fig. 1(a) shows an artificial image we constructed, which consists of circle and square in front of the background, and the intensities inside the circle, square and background vary smoothly. Fig. 1(b) shows image in (a) corrupted by zero mean Gaussian noise with variance 16. Figs. (c) and (d) show the result of the expansion and our algorithm, respectively. We omit the results of the swap algorithm because they are visually similar to the results of the expansion algorithm. The energies of the ground truth, expansion algorithm and our algorithm are listed in the figure. Notice that our algorithm not only produces an answer with a significantly lower energy, but also gives the answer which looks smoother[5]. The expansion algorithm tends to assign the same label to a subset of pixels that is too large, and the resulting answer looks piecewise-constant as opposed to piecewise smooth. This is because expansion moves seeks to change a large subset of pixels to the *same* label, as opposed to our algorithm which can change a subset of pixels to a smooth range of labels. In addition to producing a labeling which is more piecewise smooth, our answer is much closer to the ground truth. This is due not only to the fact that our energy is lower, but also to the fact that truncated quadratic energy is appropriate for piecewise smooth restoration. The absolute average error (compared to the ground truth in (a)) for our answer is 0.82, for the swap algorithm the error is 1.35, and for the expansion algorithm the error is 1.38. Our algorithm does take twice longer to run than the expansion algorithm on this example. Expansion algorithm takes about 40 seconds, and our algorithm takes about 80 seconds.

4.2. Image Inpainting

Image inpainting problem is similar to image restoration, except that some pixels have been "occluded" and

[5]Depending on the printer resolution, the ground truth and our answer in the hard copy version of the paper may actually appear not piecewise smooth but piecewise constant. Zooming in on the electronic version, one will see images that do look piecewise smooth for our answer and the ground truth, and piecewise constant for the expansion algorithm.



(a) expansion algorithm (b) our results G(0,16)

Figure 2. Zoom in on the detail.

therefore they have no preference for any label, that is for an occluded pixel p, $D_p(l) = 0$ for all $l \in \mathcal{L}$. We took an example from [27], available from D. Scharstein's web site[6]. We used the same energy as in [27], namely $V_{pq}(f_p, f_q) = 25 \cdot \min\{(f_p - f_q)^2, 200\}$. The expansion algorithm gives a labeling with energy 16,091,118, the labeling from swap algorithm has energy 16,388,270, and our energy is 15,332,536, which is significantly lower. Note that the energies that swap and expansion algorithms give in our implementation are slightly different from those published in [27], probably because the iteration over labels is performed in random order, and different runs of the swap and expansion algorithms will give slightly different results. TRW-S algorithm does give better results than we get, from [27] we get that the energy of the labeling produced by TRW-S is 15,100,492. Graph cuts, however, perform better than TRW-S when longer range interactions are present [15].

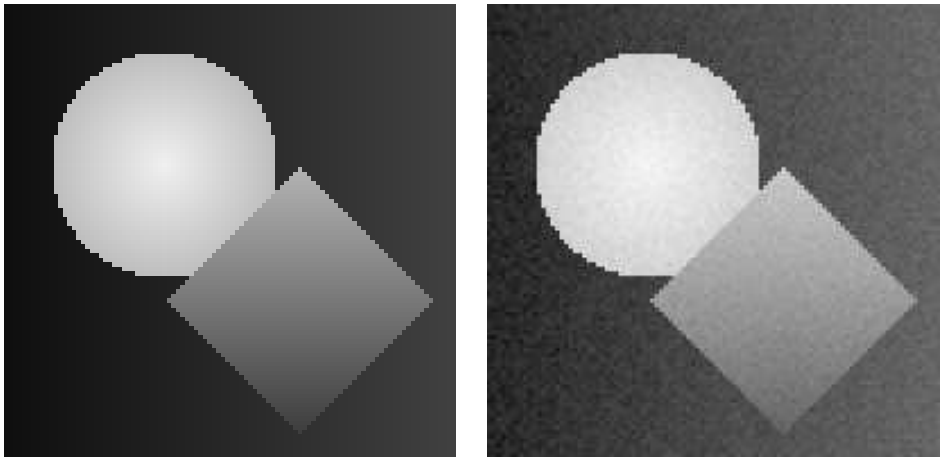
4.3. Stereo Correspondence

In this section, we present our results on stereo correspondence for the Middlebury database stereo images[7]. This database was constructed by D. Scharstein and R. Szeliski, and these images are the top benchmark in evaluating the performance of stereo algorithms [24, 25].

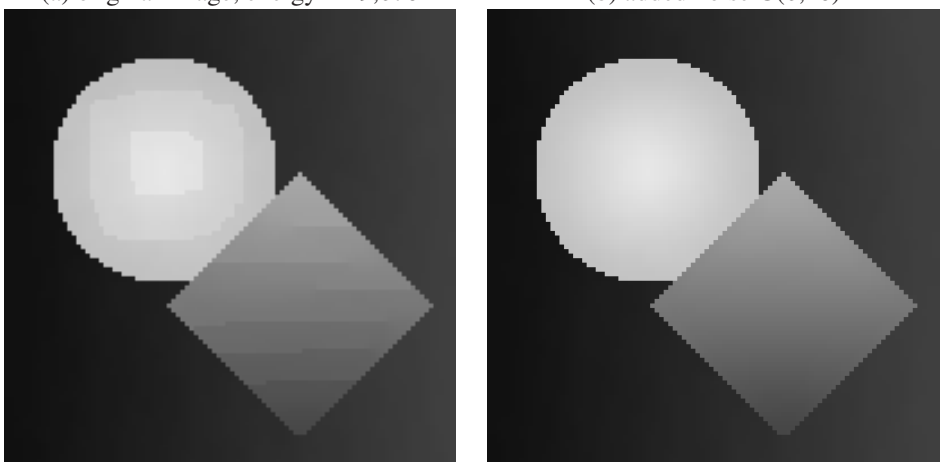
For stereo correspondence, $\mathcal{P}$ is the set of all pixels in the left image, $\mathcal{L}$ is the set of all possible stereo disparities. We take the disparity labels at sub-pixel precision, in quarter of a pixel steps. That is if $|f_p - f_q| = 1$, then the disparities of pixels p and q differ by 0.25 pixels. Let d_l stand for the actual disparity corresponding to the integer label l, for example label 2 corresponds to disparity 0.75. The data costs are $D_p(l) = \left|I_L(p) - [I_R(p - \overline{d_l}) \cdot (d_l - \underline{d_l}) + I_R(p - \underline{d_l})(\overline{d_l} - d_l)]\right|$, where $\underline{x}$ stands for rounding down, $\overline{x}$ stands for rounding up, and $p - x$ stands for the pixel that has the coordinates of pixel p shifted to the left by x.

[6]http://vision.middlebury.edu/MRF/

[7]The images were obtained from www.middlebury.edu/stereo



(a) original Image, energy=419,076 (b) added noise G(0,16)



(c) expansion algorithm, energy=453,994 (d) our results, energy=388,730

Figure 1. Image Restoration Results.

	Tsukuba	Venus	Teddy	Cones
Our Algorithm	1,758,136	2,671,875	6,058,678	7,647,529
Swap	1,804,548	2,702,371	6,099,656	7,706,717
Expansion	1,765,386	2,690,970	6,124,697	7,742,709

Figure 3. Energies on Middlebury database.

We use the truncated quadratic $V_{pq}(f_p, f_q) = 10 \cdot \min\{(f_p - f_q)^2, 16\}$. Using spatially varying weights w_{pq} improves results of stereo correspondence, since it helps to align disparity discontinuities with the intensity discontinuities. We set all $w_{pq} = 10$, since the main purpose of our paper is to evaluate our α-β range moves, and not to come up with the best stereo algorithm. Fig. 3 compares the energies obtained with our method to those obtained with the swap and expansion algorithms. The accuracy of the labelings is summarized in Fig. 4. Each number in Fig. 4 gives the percentage of pixels away from ground truth by more than 0.5 pixels. *Tsukuba, Venus, Teddy, Cones* are the name of the four scenes in the Middlebury stereo database. Notice that our algorithm performs better not only in terms of energy, but also in terms of ground truth. The accuracy improvement is slight, but consistent across all the images in the database. At threshold 0.5, all algorithms are ranked 7th out of 33 other algorithms at the submission time.

Fig. 2 shows a zoom in on the detail in the *Cones* sequence. Notice that our algorithm gives results that look smoother over the surface of the cone than the expansion algorithm.[8]

[8]To see the difference, it may be necessary to zoom in on these images in the electronic version of the paper.

	Tsukuba	Venus	Teddy	Cones
Our Algorithm	6.7	3.25	15.1	6.79
Swap	7.47	4.04	15.8	7.64
Expansion	7.14	4.19	16.0	7.81

Figure 4. Accuracy on the Middlebury database.

5. Acknowledgements

We would like to thank Andrew Delong for contributing the idea of moves that operate on several labels simultaneously. He declined a coauthorship offer. Instead, Andrew chose to receive 13 grams of chocolate per day for the rest of his graduate studies. This amounts to 12kg of chocolate over the next several years. Andrew is a very rare person for having chosen only 12 kg. of chocolate over a CVPR publication. Anyone else would have requested at least 120 kg.

References

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Iteractive digital photomontage. In *ACM Transactions on Graphics, SIGGRAPH*, 2004. 1

[2] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48:259–302, 1986. 1

[3] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. In *ECCV*, pages Vol I: 428–441, 2004. 1

[4] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*, pages I: 105–112, 2001. 1

[5] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, September 2004. 6

[6] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *CVPR*, pages 648–655, 1998. 1, 1

[7] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, November 2001. 1, 1, 1, 2, 2, 3, 3, 3, 3, 3

[8] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *PAMI*, 6:721–741, 1984. 1

[9] H. Ishikawa. Exact optimization for markov random fields with convex priors. *PAMI*, 25(10):1333–1336, October 2003. 2, 2, 2, 2, 3, 3, 4, 4

[10] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *ECCV*, page I:232, 1998. 1

[11] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov Random Fields. In *FOCS*, pages 14–24, 1999. 2

[12] V. Kolmogorov. Primal-dual algorithm for convex markov random fields. Technical Report MSR-TR-2005-117,, Microsoft, 2005. 3

[13] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10):1568–1583, October 2006. 2, 2

[14] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Probabilistic fusion of stereo with color and contrast for bilayer segmentation. *PAMI*, 28(9):1480–1492, September 2006. 1

[15] V. Kolmogorov and C. Rother. Comparison of energy minimization algorithms for highly connected graphs. In *ECCV*, pages II: 1–15, 2006. 2, 6

[16] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions via graph cuts. In *ICCV*, pages II: 508–515, 2001. 1, 1

[17] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV*, page III: 82 ff., 2002. 1

[18] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *ECCV*, 2002. 1, 3, 3

[19] V. Kwatra, A. Schdl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics, SIGGRAPH 2003*, 22(3):277–286, July 2003. 1

[20] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988. 2

[21] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake. Digital tapestry. In *CVPR*, pages I: 589–596, 2005. 4

[22] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching: Incorporating a global constraint into mrfs. In *CVPR*, pages I: 993–1000, 2006. 1

[23] S. Roy and V. Govindu. Mrf solutions for probabilistic optical flow formulations. In *ICPR00*, pages Vol III: 1041–1047, 2000. 2

[24] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, April 2002. 6

[25] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR*, pages I: 195–202, 2003. 6

[26] D. Schlesinger and B. Flach. Transforming an arbitrary minsum problem into a binary one. Technical Report TUD-FI06-01, Dresden University of Technology, 2006. 2

[27] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. In *ECCV*, pages II: 16–29, 2006. 2, 2, 3, 6, 6, 6, 6

[28] J. Wills, S. Agarwal, and S. Belongie. What went where. In *CVPR*, pages I: 37–44, 2003. 1