Multi-label Moves for MRFs with Truncated Convex Priors

Olga Veksler

Computer Science Department
University of Western Ontario
London, Canada
olga@csd.uwo.ca

Abstract. Optimization with graph cuts became very popular in recent years. As more applications rely on graph cuts, different energy functions are being employed. Recent evaluation of optimization algorithms showed that the widely used swap and expansion graph cut algorithms have an excellent performance for energies where the underlying MRF has Potts prior. Potts prior corresponds to assuming that the true labeling is piecewise constant. While surprisingly useful in practice, Potts prior is clearly not appropriate in many circumstances. However for more general priors, the swap and expansion algorithms do not perform as well. Both algorithms are based on moves that give each pixel a choice of only two labels. Therefore such moves can be referred to as binary moves. Recently, range moves that act on multiple labels simultaneously were introduced. As opposed to swap and expansion, each pixel has a choice of more than two labels in a range move. Therefore we call them multi-label moves. Range moves were shown to work better for problems with truncated convex priors, which imply a piecewise smooth labeling. Inspired by range moves, we develop several different variants of multi-label moves. We evaluate them on the problem of stereo correspondence and discuss their relative merits.

1 Introduction

Energy optimization with graph cuts [1,2,3] is increasingly used for different applications in computer vision and graphics. Some examples are image restoration [2], stereo and multi-view reconstruction [4,5,2,6,7], motion segmentation [8,9,10], texture synthesis [11], segmentation [12,13,14,15], digital photomontage [16]. Optimization with graph cuts either results in an exact minimum or an approximate minimum with non-trivial quality guarantees. This frequently translates into a result of high accuracy, given that the energy function is appropriate for the application.

A typical energy function to be minimized is as follows:

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

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. Frequently $\mathcal{P}$ is set of image pixels. The label assigned to pixel $p \in \mathcal{P}$ is denoted

D. Cremers et al. (Eds.): EMMCVPR 2009, LNCS 5681, pp. 1–13, 2009.

by f_p, and f is the collection of all pixel-label assignments. The first sum in Eq. (1) is the data term. In the data term, $D_p(f_p)$ is the penalty for pixel p to be assigned label f_p. The data term usually comes from the observed data. The second sum in Eq. (1) is the smoothness term, and it uses the prior knowledge about what the likely labelings f should be like. The sum is over ordered pixel pairs $(p, q) \in \mathcal{N}$. Often $\mathcal{N}$ is the 4 or 8 connected grid, however longer range interactions are also useful [5]. Without loss of generality, we assume that if $(p, q) \in \mathcal{N}$ then $p < q$.

Any choice for D_p is easy to handle. The choice of V_{pq} determines whether the energy function can be efficiently minimized. The V_{pq}'s often 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 $V_{pq}(f_p, f_q) = w_{pq} \cdot \min\{1, |f_p - f_q|\}$. The coefficients w_{pq}'s can be different for each pair of neighboring pixels. Potts model penalizes any difference between f_p and f_q by the same amount. Intuitively, it corresponds to the prior knowledge that f should be piecewise constant, that is it consists of several pieces where pixels inside the same piece share the same label.

Another common choice is $V_{pq}(f_p, f_q) = w_{pq} \cdot \min\{T, |f_p - f_q|^a\}$. If $a = 1$ the model is called a truncated linear, and if $a = 2$, it is called a truncated quadratic. 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 vary "smoothly"[1]. Parameter T is called a *truncation* constant. 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 [17], 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 [2].

Recently, Szeliski et.al. [18] performed an experimental evaluation of several optimization methods popular for minimizing energies in Eq. (1) : the graph cut based expansion and swap algorithms [2], sequential tree-reweighted message passing (TRW-S) [19], and loopy belief propagation (LBP) [20]. They show that for Potts model, both expansion and swap algorithms have an excellent performance, they find an answer within a small percentage of the global minimum. 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. Szeliski et.al. [18] studied only the case when $\mathcal{N}$ is the 4-connected grid. Kolmogorov and Rother [21] performed a comparison between graph cuts and TRW-S when longer range interactions are present, 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.

Recently [22] developed a new type of moves, called the *range* moves for optimizing energies with truncated convex priors. A truncated quadratic and linear are examples of truncated convex prior. Informally, truncated convex priors correspond to assuming that f is piecewise smooth. The insight in [22] is that both expansion and swap algorithms give a pixel a choice of only two labels, but for problems where piecewise smoothness

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

assumptions are appropriate, to obtain a good approximation, a pixel should have a choice among more than two labels. The range moves that they develop act on a larger set of labels than the expansion and swap moves. Because of this property, we call the range moves *multi-label* moves. In [23] they use a similar idea to develop multi-label moves for an energy function useful for single-view scene reconstruction. The energy function in [23] is neither Potts nor truncated convex, and thus not directly related to our work.

We further explore the idea of multi-label moves for truncated convex priors. One can regard the range moves developed in [22] as a generalization of the swap move. In this paper, we develop a multi-label move that is can be regarded as a generalization of the expansion move. The optimal multi-label expansion move can be found only approximately. We explore an additional move that we call multi-label smooth swap. Note that simultaneously but independently, [24] developed a move similar to our multi-label expansion [25]. Their graph construction is very similar, with some minor differences in edge weights. They also do not find an optimal multi-label expansion move, but its approximation. The ideas that lay behind our multi-label moves (as, indeed, the ideas behind any move-making optimization algorithm) are related to the framework of majorization-minimization [26].

We evaluate our new multi-label moves on the energy functions arising in stereo correspondence, and discuss their relative merits as well as compare them with the range moves.

2 Prior Work

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 $\{0, 1, ...k-1\}$, which is necessary since the construction is based on that in [17].

2.2 Convex Priors

Ishikawa [17] develops a method to find the exact minimum of the energy function in Eq. (1) when V_{pq} are convex functions of the label differences. Specifically, $V_{pq}(l_1, l_2) = w_{pq} \cdot g(l_1 - l_2)$ 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]. In [27,28] they extend the results in [17] to handle a more general definition of convexity.

We follow [22] to describe the work in [17]. 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 $p_0, p_1, ..., p_k$, see Fig. 1 Node p_0 is connected with the source s with an edge of infinite capacity. Similarly, we connect p_k with the sink t with an edge of infinite capacity. This way p_0 is essentially identified with the source, and p_k with the sink. We connect node p_i to node p_{i+1} with a

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

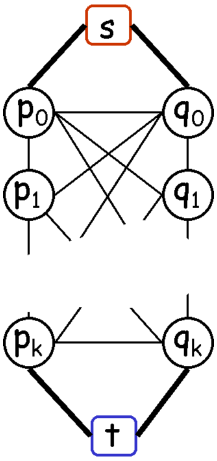

Fig. 1. Graph construction for convex V_{pq}. Thick links indicate edges of infinite capacity.

directed edge e_i^p for $i = 0, 1, ..., k-1$. In addition, for $i = 1, ..., k$, node p_i is connected to node p_{i-1} 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. If an edge e_i^p is cut, then pixel p is assigned label i, thus any cut C induces a labeling f^C.

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 $i = 0, ..., k$ and $j = 0, ..., k$. The weight of this edge is

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

The edge weight defined by Eq. (2) is non-negative, since $g(x)$ is convex.

The weights e_i^p are defined 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, and $h(i) = -\frac{1}{2}[g(k+1-i) + g(i+1)]$ Under this edge weights assignment, the cost of any finite cut C is exactly $E(f^C)$ plus a constant, see [22]. Therefore the minimum cut gives the optimal labeling.

Note that [29] 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. [2] develop the expansion and swap algorithms. These methods can be applied when V_{pq} is Potts, truncated linear or quadratic, but the answer is only approximate, since the energy is NP-hard to optimize in these cases [2].

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 $E(f') \geq E(f)$ for any $f' \in M(f)$.

Given a labeling f and a label pair (α, β), a move from f to f' is called an α-β swap if $f_p \neq f'_p \Rightarrow f_p, f'_p \in \{\alpha, \beta\}$. $M(f)$ is then defined as the collection of α-β swaps for all pairs of labels $\alpha, \beta \in \mathcal{L}$.

Given a labeling f and a label α, a move f' is called an α-expansion if $f_p \neq f'_p \Rightarrow f'_p = \alpha$. $M(f)$ is then defined as the collection of α-expansions for all labels $\alpha \in \mathcal{L}$.

The optimal α-expansion and the optimal α-β swap can be found as a minimum cut in a certain graph [2]. Thus the expansion and swap algorithms find a local minimum with respect to expansion or swap moves, correspondingly. Starting with an initial labeling f, optimal swap (or expansion) moves are found until convergence.

The energy with truncated linear V_{pq} can be optimized by both expansion and swap algorithms, whereas for truncated quadratic V_{pq}, only the swap algorithm applies directly. In practice, however, it is possible to apply the expansion algorithm with a "truncation trick" [30]. 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.

2.4 Range Moves for Truncated Convex Priors

In this section we review the range moves of [22]. Based on the notion of convexity in [17], V_{pq} is truncated convex if there exists a symmetric function $g(x)$ such that $g(x+1) - 2g(x) + g(x-1) \geq 0$ and

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

Throughout the rest of the paper, we assume truncated convex V_{pq}'s.

Recall that $\mathcal{L} = \{0, 1, ..., k-1\}$. Let $\mathcal{L}_{\alpha\beta} = \{\alpha, \alpha+1, ..., \beta\}$, where $\alpha < \beta \in \mathcal{L}$. 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 $\{\alpha, \alpha+1, .., \beta\}$.

In [22], they show how to find an optimal α-β range move if $|\alpha - \beta| \leq T$[3]. The basic idea is as follows. 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 $p, q \in \mathcal{T}$, $V_{pq}(f_p, f_q) = w_{pq} g(f_p - f_q) \leq w_{pq} \cdot T$. Non-convex term arise only on the boundary of $\mathcal{T}$, but they can be arranged in a graph construction by adding appropriate constants to edges e^p_i, see Section 2.2.

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'.

The α-β range move can be slightly generalized. As previously, let $|\alpha - \beta| = T$ and, as before, let $\mathcal{T} = \{p | \alpha \leq f_p \leq \beta\}$. Let

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

[3] If $|\alpha - \beta| > T$, α-β range move is NP-hard to find.

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 $M^{\alpha\beta}(f) \subset M^{\alpha\beta t}(f)$. It is not possible to find the optimal move in $M^{\alpha\beta t}(f)$, but [22] shows how to 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 one is lucky, $E(\hat{f})$ could be significantly better than the optimal move in $M^{\alpha\beta}(f)$. In practice, t is set to a small constant. Let us call this generalized range move as α-β-t-range move.

3 Multi-label Moves

The key idea of the range moves in [22] is to allow a pixel to choose among several labels in a single move. This is in contrast to the swap and expansion moves, which allow each pixel a choice between only two labels. We are going to refer to moves that allow a choice of more than two labels as *multi-label*. Multi-label moves have already proven successful in [22,23,24]. There is a multitude of such moves possible. In this paper, we develop several different multi-label moves for truncated convex priors and compare their performance. To have a clear terminology, we are going to rename to the generalized α-β-t-range move with as multi-label α-β-t-swap. There is no need to rename the α-β-range move since it is a special case of α-β-t-range move with $t = 0$.

In [22], the idea was to find a subset of pixels $\mathcal{P}'$ and a subset of labels $\mathcal{L}'$ s.t. when the V_{pq} terms are restricted to $\mathcal{P}'$ and $\mathcal{L}'$, they are convex. The boundary terms are easy to implement, as shown in [22]. Throughout the remainder of this section, we are going to exploit different ways of selecting $\mathcal{P}'$ and $\mathcal{L}'$. The two new moves that we develop are called *multi-label expansion* and *multi-label smooth swap*.

In order to perform iterative energy optimization that reduces the energy of the current labeling f, it seems necessary to ensure that the labels of pixels in $\mathcal{P}'$ under labeling f are contained in $\mathcal{L}'$. This ensures that the current labeling f is also within the set of allowed moves, and the lowest energy move is not worse than the current labeling. For the multi-label smooth swap, we are able to enforce this condition. For multi-label expansion, we are not able to always enforce it. We will still guarantee though that the energy goes down at each iteration by simply rejecting any move whose energy is higher than that of the current labeling.

3.1 Multi-label Smooth Swap

Let f be a current labeling. Let $\mathcal{P}'$ be a subset of pixels of $\mathcal{P}$. We call $\mathcal{P}'$ a smooth subset under labeling f, if for any $(p, q) \in \mathcal{N}$, whenever $\{p, q\} \subset \mathcal{P}'$, then $|f_p - f_q| \leq T$, where T is the truncation constant in Eq. (3). In words, if a subset $\mathcal{P}'$ is smooth under f, then the label difference for any two pixels contained in $\mathcal{P}'$ is not larger than the truncation constant.

Let f be the current labeling and $\mathcal{P}'$ be a smooth subset under f. Let $\mathcal{L}(\mathcal{P}', f) = \{f_p | p \in \mathcal{P}'\}$, that is $\mathcal{L}(\mathcal{P}')$ is the collection of labels that pixels in $\mathcal{P}'$ have under labeling f.

Given a smooth subset $\mathcal{P}'$ under f, let $M_{smooth}(f, \mathcal{P}') = \{f' | f'_p \neq f_p \Rightarrow p \in \mathcal{P}' \text{ and } f'_p \in \mathcal{L}(\mathcal{P}', f)\}$. $M_{smooth}(f, \mathcal{P}')$ describes exactly the set of all multi-label smooth swap moves. In words, a smooth swap move takes a smooth set of pixels under f, collects their labels, and reassigns their labels among them.

Just as it was possible to generalize the multi-label swap move by extending the range of labels, it is possible to generalize the multi-label smooth swap. Let t be a constant for extending the range of labels $\mathcal{L}'(\mathcal{P}', f)$. Let us define the extended range of labels as

$$\mathcal{L}'(\mathcal{P}', f, t) = \{l \in \mathcal{L} | \exists l' \in \ \mathcal{L}'(\mathcal{P}', f) \text{ s.t. } |l - l'| \leq t\} \cap \mathcal{L}.$$

In words, to get $\mathcal{L}'(\mathcal{P}', f, t)$ we add to $\mathcal{L}'(\mathcal{P}', f)$ labels that are at distance no more than t from some label already in $\mathcal{L}'(\mathcal{P}', f)$. The intersection with $\mathcal{L}$ is performed to make sure that after the "padding", the augmented set is still contained in $\mathcal{L}$. Let the set of smooth swap moves augmented by t be denoted by $M_{smooth}(f, \mathcal{P}', t)$.

A multi-label smooth swap is naturally related to a multi-label swap. In a multi-label swap move, the pixels participating in a move have labels in a range limited by truncation, i.e. all the labels are between some α and β with $|\alpha - \beta| < T$. In a multi-label smooth swap, the domain of pixels participating in a move can be larger than that compared to the multi-label swap. That is the pixels participating in the move can have labels between some α and β with $|\alpha - \beta| > T$. The restriction is that in the pixels participating in a smooth swap must form a "smooth" component in the current labeling f, that is the labels of any two neighbors cannot differ by more than T.

There are two questions that remain to be answered: how to choose the smooth subsets $\mathcal{P}'$ and how to optimize with smooth swap moves. Let us first consider the question of optimization.

In general, it is not possible to find the optimal smooth swap move, given a smooth subset $\mathcal{P}'$ and the current labeling f. However, we are able to find a good swap move, the one that improves the current labeling f.

Let $\mathcal{S}$ be a subset of pixels in $\mathcal{P}$ and let us define:

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

In words, $E_{\mathcal{S}}(f)$ is the sum all the terms of the energy function which depend on pixels in $\mathcal{S}$. Let us further define:

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

In words, $E^{open}_{\mathcal{S}}(f)$ is the sum of all the terms of the energy function which depend *only* on pixels in $\mathcal{S}$. It is clear that for any $\mathcal{S} \in \mathcal{P}$, $E(f) = E_{\mathcal{S}}(f) + E^{open}_{\mathcal{P}-\mathcal{S}}(f)$.

Let f be the current labeling, and let $\mathcal{P}'$ be a smooth subset under f. Let f' be a smooth swap move from f, i.e. $f' \in M_{smooth}(f, \mathcal{P}', t)$.

We use basically the same construction as in Section 2.4. We construct a graph for pixels in $\mathcal{P}'$. However, the label range is $\mathcal{L}'(\mathcal{P}', f, t)$, and we identify it with label set

$\{0, 1, ..., |\mathcal{L}'(\mathcal{P}', f, t)| - 1\}$. Otherwise, the graph construction is identical to that in Section 2.4.

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 $\mathcal{P}'$. Let f^C be the labeling corresponding to the cut C, which we define as follows: $f_p^C = f_p$ for $p \notin \mathcal{P}'$, and for $p \in \mathcal{P}'$, f_p^C 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, $w(C) = \tilde{E}_{\mathcal{P}'}(f^C) + K$, where K is a constant and $\tilde{E}(f)$ is the same energy as $E(f)$, except there is no truncation in V_{pq} terms for $p, q \in \mathcal{P}'$. That is for $p, q \in \mathcal{P}'$, $V_{pq}(f_p, f_q) = w_{pq} \cdot g(f_p - f_q)$ in the energy $\tilde{E}$.

For any f, $E_{\mathcal{P}'}(f) \leq \tilde{E}_{\mathcal{P}'}(f)$, since the only difference between E and $\tilde{E}$ is that the V_{pq} terms are not truncated in $\tilde{E}$ for $p, q \in \mathcal{P}'$. Recall that for any f, $E(f) = E_{\mathcal{P}'}(f) + E^{open}_{\mathcal{P}-\mathcal{P}'}(f)$. Also, $E^{open}_{\mathcal{P}-\mathcal{P}'}(f) = \tilde{E}^{open}_{\mathcal{P}-\mathcal{P}'}(f)$, since $\tilde{E}$ is not different from E outside of set $\mathcal{P}'$.

Let f be the current labeling. Notice that $E_{\mathcal{P}'}(f) = \tilde{E}_{\mathcal{P}'}(f)$, since V_{pq} terms in f do not need to be truncated on the set $\mathcal{P}'$. Let $\hat{C}$ be the minimum cost cut, and let $\hat{f}$ be its corresponding labeling, defined as above. Let f be the current labeling (notice that $f \in M_{smooth}(f, \mathcal{P}', t)$), and let C be the cut which corresponds to it in the graph. We have that $\tilde{E}_{\mathcal{P}'}(\hat{f}) + K = w(\hat{C}) \leq w(C) = \tilde{E}_{\mathcal{P}'}(f) + K$. Since $E_{\mathcal{P}'}(\hat{f}) \leq \tilde{E}_{\mathcal{P}'}(\hat{f})$ and $E_{\mathcal{P}'}(f) = \tilde{E}_{\mathcal{P}'}(f)$, we get that $E_{\mathcal{P}'}(\hat{f}) \leq E_{\mathcal{P}'}(f)$. Now, for any labeling f'', $E(f'') = E_{\mathcal{P}'}(f'') + E^{open}_{\mathcal{P}-\mathcal{P}'}(f'')$. We have that $E^{open}_{\mathcal{P}-\mathcal{P}'}(\hat{f}) = E^{open}_{\mathcal{P}-\mathcal{P}}(f')$, and therefore we get that $E(\hat{f}) \leq E(f)$. This shows that the minimum cut gives a labeling $\hat{f}$ with energy not larger than the current labeling f. So if we cannot find the optimal smooth swap move, we can at least guarantee smooth swap move does not increase energy.

The question remains of how to find smooth subsets $\mathcal{P}'$. In general, given a current labeling f, we can partition it into a set of $\mathcal{P}_1, \mathcal{P}_2 ... \mathcal{P}_d$, s.t. $\bigcap_i \mathcal{P}_i = \mathcal{P}$ and each $\mathcal{P}_i$ is smooth. This partition can be performed by computing connected components. This partition is not unique, however. To remove bias due to visitation order, we compute connected components in random order. That is we pick a pixel p at random, compute a maximal smooth subset $\mathcal{P}_1$ containing p, then choose another pixel $q \notin \mathcal{P}_1$, compute a maximal smooth subset $\mathcal{P}_2$ containing q, and so on, until all pixels are partitioned into smooth subsets. Then we compute smooth swap moves for each $\mathcal{P}_i$. This is not the only way to proceed, but we found it to be effective. Computing all smooth swap moves for a partition $\mathcal{P}_1, \mathcal{P}_2 ... \mathcal{P}_d$ constitutes one iteration of the algorithm. We perform iterations until convergence.

The advantage of the multi-label smooth swap move over the multi-label swap is that it converges faster. If we start from a good solution (typically we start from the results of the binary expansion algorithm), the number of smooth subsets in a partition of $\mathcal{P}$ is small, so the number of moves is smaller compared to the multi-label swap. The disadvantage is that it gives energies that are slightly higher in practice.

3.2 Multi-label Expansion

We now develop a multi-label expansion move. Let α and β be two labels s.t. $\alpha < \beta$. The idea behind multi-label expansion move is similar to that of a binary expansion move. We wish to construct a move in which each pixel can either stay with its old

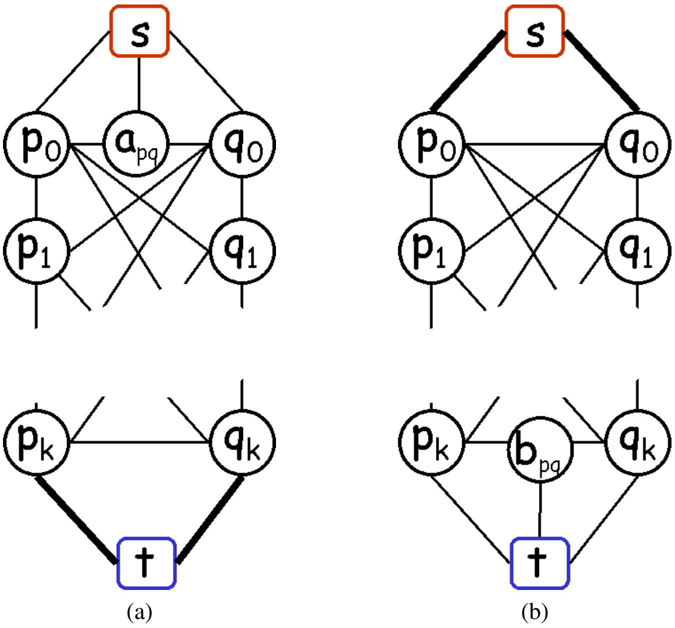

Fig. 2. Graph construction for multi-label expansion

label, or switch to a label in the set $\{\alpha, \alpha+1, ..., \beta\}$. The name "expansion", as before, reflects the fact that labels in the set $\{\alpha, \alpha+1, ..., \beta\}$ expand their territory.

Let $M^{\alpha\beta}(f) = \{f' | f'_p \neq f_p \Rightarrow f'_p \in \mathcal{L}_{\alpha\beta}\}$. That is $M^{\alpha\beta}(f)$ is exactly the set of all α-β multi-label expansion moves from labeling f. Unfortunately, the optimal expansion move cannot be computed exactly, so we are forced to approximate it.

Suppose that we are given a labeling f and we wish to approximate the optimal α-β expansion move, where $|\alpha - \beta| = T$. The construction is similar to that in Section 2.2. We identify label set $\{\alpha, \alpha+1, ..., \beta\}$ with set $\{0, 1, ..., T\}$. One of the differences is that now all pixels participate in a move. First we build a graph exactly like in Sec. 2.4, except the links between the source and p_0 are not set to infinite, for all pixels p. We create an auxiliary pixel a_{pq} between each pair of neighboring pixels (p, q)[4]. We connect p_0 to a_{pq}, q_0 to a_{pq}, and s to a_{pq}, as illustrated in Fig. 2 (a). If the minimum cut severs edge between s and p_0, then p is assigned its old label in the move. Otherwise, the label assignment is exactly like in Sec. 2.4.

For the construction in 2.4 if we sever links e^p_i and e^q_j, then the cost of all the links e^{pq}_{ij} that have to be severed adds up to $C + V_{pq}(i, j)$. The costs of the new links that we create for the expansion algorithm are as in Fig. 3.

This construction insures that if a links between s and p_0 and between q_0 and q_1 are broken, then the cost of all edges severed corresponds exactly to $V_{pq}(f_p, \alpha)$ plus a constant, which is exactly what is needed. Similarly the correct thing happens if links between s and q_0 and between p_0 and p_1 are broken, and if the link between s and a_{pq}

[4] Note that auxiliary pixel is not necessary, see [3], but it clarifies the explanation.

link	weight
p_0 to a_{pq}	$V_{pq}(f_p, \alpha) + C/2$
q_0 to a_{pq}	$V_{pq}(\beta, f_q) + C/2$
s to a_{pq}	$V_{pq}(f_p, f_q)$+C
s to p_0	$D_p(f_p)$
s to q_0	$D_p(f_q)$

Fig. 3. Weights of the new links

is broken. Unfortunately in other cases, as long as the new links in Fig. 3 are involved, the V_{pq} value can be underestimated or overestimated. The minimum graph cut is not even guaranteed to reduce the energy from that of the current labeling f. Still in practice we found that many minimum cuts correspond to an assignment with a lower energy. Therefore, to make sure that the energy never goes up, if f' is the assignment returned by our approximate multi-label expansion, we first test if $E(f') < E(f)$, where f is the current labeling. If yes, we accept f' as the new current labeling. If no, we reject it.

As with the multi-label swap, the range of labels involved in multi-label expansion can be extended by some t. The construction changes appropriately, similar to what is done when extending the range of multi-label swap moves see Sec. 2.4.

In practice, we found the following version of the multi-label expansions to work better. Let $\mathcal{T} = \{p \in \mathcal{P} | f_p \leq \beta\}$ and let $\mathcal{B} = \{p \in \mathcal{P} | f_p \geq \alpha\}$. We perform the multi-label expansion on pixels in set $\mathcal{T}$ using the graph like in Fig. 2(a), and an expansion on pixel in set $\mathcal{B}$ using the graph like in Fig. 2(b), with symmetrically modified weights in Fig. 3 for the second case. The weights also have to be corrected because there are pixels not participating in the move, so the "border" conditions resulting from such pixels have to be incorporated into edge weights e_i^p, just like in Sec. 2.4. The improvement is probably due to the fact that more V_{pq}'s are correctly represented by this split graph construction. Another improvement is probably due to the fact that pixels on the border not participating in the move pull the energy in the right direction by having their V_{pq} terms correctly modeled through the edge weights e_i^p.

4 Experimental Results

In this section, we present our results on stereo correspondence for the Middlebury database stereo images[5]. We took four pairs of stereo images for evaluation, namely: *Venus,Sawtooth,Teddy,Cones*. This database was constructed by D. Scharstein and R. Szeliski, and these images are the top benchmark in evaluating the performance of stereo algorithms [31,32].

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

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

	Venus	Sawtooth	Teddy	Cones
Swap	7,871,677	9,742,107	16,376,181	21,330,284
Expansion	8,131,203	9,418,529	15,829,221	21,020,174
α-β-2 swap	**7,188,393**	**9,371,745**	15,421,437	**20,490,753**
Multi-Label Smooth Range	7,193,823	9,373,126	15,616,999	20,515,493
α-β-2 Expansion	7,188,404	9,377,494	**15,408,234**	20,626,809

Fig. 4. Energies on Middlebury database. The minimum in each column is highlighted.

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.

We use the truncated quadratic $V_{pq}(f_p, f_q) = 100 \cdot \min\{(f_p - f_q)^2, 25\}$. 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 and compare the multi-label moves, and not to come up with the best stereo algorithm. Fig. 4 compares the energies obtained with the expansion algorithm, swap algorithm, multi-label swap moves (or range moves in terminology of [22]), multi-label expansion moves, and smooth swap moves.

From the Table 4, we can make the following conclusions. First let us consider the "binary" swap and expansion moves. The swap and expansion algorithms are clearly inferior when it comes to truncated convex priors. Even though the swap algorithm is guaranteed to find a best swap move and the expansion algorithm is not guaranteed to find the best move under the truncated quadratic model, expansion algorithm does performs better for all scenes except "Venus". This is probably explained by the fact that expansion moves are more powerful than the swap moves. Even if we do not find the optimal expansion, a good expansion may be better than the optimal swap.

Now let us discuss the multi-label moves. First of all, the running times for the multi-label swap move was on the order of minutes (from 5 to 10 minutes). The smooth range move achieved the energy very close to that of the multi-label swap, but its running time is about 2 or 3 times faster. The multi-label expansion move is almost always slightly worse that the multi-label swap, it is better only on the "Teddy sequence". One would expect a better performance from the expansion move, but since we cannot find the optimal one, only an approximate one, these results are not entirely surprising. The running time for the expansion move is much worse than for other multi-label moves, since the graphs are much bigger. Multi-label expansion takes about 9-10 times longer than multi-label swap.

We should mention that the running times of our algorithms can be significantly improved using the ideas in [33]. They employ techniques such as good initialization, reducing the number of unknown variables by computing partially optimal solutions, and recycling flow. All of these are directly transferable to the implementation of our multi-label moves. Their speed ups are around a factor of 10 or 15.

4.1 Discussion

In this paper develop and compare two new multi-label moves for energies with truncated convex prior, as well compare the new moves with the previously known multi-label moves called range moves. Clearly, there are more interesting multi-label moves that can be developed for multi-label energies. An interesting question is whether it is possible to discover automatically new multi-label moves with good properties for a given energy, rather than develop them by hand.

References

1. Ishikawa, H., Geiger, D.: Occlusions, discontinuities, and epipolar lines in stereo. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, p. 232. Springer, Heidelberg (1998)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI 23(11), 1222–1239 (2001)
3. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 65–81. Springer, Heidelberg (2002)
4. Boykov, Y., Veksler, O., Zabih, R.: Markov random fields with efficient approximations. In: CVPR, pp. 648–655 (1998)
5. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions via graph cuts. In: ICCV, vol. II, pp. 508–515 (2001)
6. Kolmogorov, V., Zabih, R.: Multi-camera scene reconstruction via graph cuts. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 82–96. Springer, Heidelberg (2002)
7. Lempitsky, V., Boykov, Y., Ivanov, D.: Oriented visibility for multiview reconstruction. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 226–238. Springer, Heidelberg (2006)
8. Wills, J., Agarwal, S., Belongie, S.: What went where. In: CVPR, vol. I, pp. 37–44 (2003)
9. Xiao, J., Shah, M.: Motion layer extraction in the presence of occlusion using graph cuts. PAMI 27(10), 1644–1659 (2005)
10. Schoenemann, T., Cremers, D.: High resolution motion layer decomposition using dual-space graph cuts. In: CVPR, pp. 1–7 (2008)
11. Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: Image and video synthesis using graph cuts. ACM Transactions on Graphics, SIGGRAPH 2003 22(3), 277–286 (2003)
12. Boykov, Y., Jolly, M.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: ICCV, vol. I, pp. 105–112 (2001)
13. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive image segmentation using an adaptive GMMRF model. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 428–441. Springer, Heidelberg (2004)
14. Rother, C., Minka, T., Blake, A., Kolmogorov, V.: Cosegmentation of image pairs by histogram matching: Incorporating a global constraint into mrfs. In: CVPR, vol. I, pp. 993–1000 (2006)
15. Kolmogorov, V., Criminisi, A., Blake, A., Cross, G., Rother, C.: Probabilistic fusion of stereo with color and contrast for bilayer segmentation. PAMI 28(9), 1480–1492 (2006)
16. Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., Cohen, M.: Iteractive digital photomontage. In: ACM Transactions on Graphics, SIGGRAPH (2004)

17. Ishikawa, H.: Exact optimization for markov random fields with convex priors. PAMI 25(10), 1333–1336 (2003)
18. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A comparative study of energy minimization methods for markov random fields with smoothness-based priors. IEEE Transacions on Pattern Analysis and Machine Intellegence 30(6), 1068–1080 (2008)
19. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. PAMI 28(10), 1568–1583 (2006)
20. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Francisco (1988)
21. Kolmogorov, V., Rother, C.: Comparison of energy minimization algorithms for highly connected graphs. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 1–15. Springer, Heidelberg (2006)
22. Veksler, O.: Graph cut based optimization for mrfs with truncated convex priors. In: CVPR, pp. 1–8 (2007)
23. Liu, X., Veksler, O., Samarabandu, J.: Graph cut with ordering constraints on labels and its applications. In: CVPR, pp. 1–8 (2008)
24. Kumar, M.P., Torr, P.H.S.: Improved moves for truncated convex models. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems 21, pp. 889–896 (2009)
25. Torr, P.H.S.: In: Personal communication (2008)
26. Hunter, D.R., Lange, K.: A tutorial on MM algorithms. The American Statistician (58) (2004)
27. Schlesinger, D., Flach, B.: Transforming an arbitrary minsum problem into a binary one. Technical Report TUD-FI06-01, Dresden University of Technology (2006)
28. Darbon, J.: Global optimization for first order markov random fields with submodular priors. In: Brimkov, V.E., Barneva, R.P., Hauptman, H.A. (eds.) IWCIA 2008. LNCS, vol. 4958, pp. 229–237. Springer, Heidelberg (2008)
29. Kolmogorov, V.: Primal-dual algorithm for convex markov random fields. Technical Report MSR-TR-2005-117, Microsoft (2005)
30. Rother, C., Kumar, S., Kolmogorov, V., Blake, A.: Digital tapestry. In: CVPR, vol. I, pp. 589–596 (2005)
31. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV 47(1-3), 7–42 (2002)
32. Scharstein, D., Szeliski, R.: High-accuracy stereo depth maps using structured light. In: CVPR, vol. I, pp. 195–202 (2003)
33. Alahari, K., Kohli, P., Torr, P.: Reduce, reuse, recycle: Efficiently solving multi-label mrfs. In: CVPR, pp. 1–8 (2008)