Making the Right Moves: Guiding Alpha-Expansion using Local Primal-Dual Gaps

Dhruv Batra
Toyota Technological Institute at Chicago
dbatra@ttic.edu

Pushmeet Kohli
Microsoft Research Cambridge
pkohli@microsoft.com

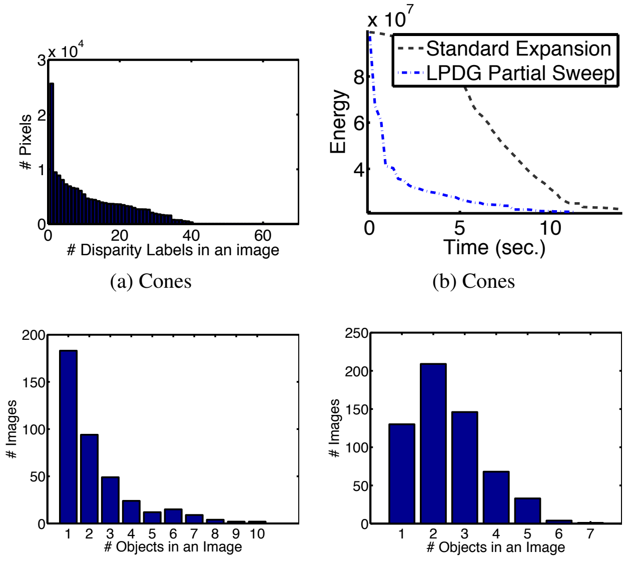
Abstract

This paper presents a new adaptive graph-cut based move-making algorithm for energy minimization. Traditional move-making algorithms such as Expansion and Swap operate by searching for better solutions in some predefined moves spaces around the current solution. In contrast, our algorithm uses the primal-dual interpretation of the Expansion-move algorithm to adaptively compute the best move-space to search over. At each step, it tries to greedily find the move-space that will lead to biggest decrease in the primal-dual gap. We test different variants of our algorithm on a variety of image labelling problems such as object segmentation and stereo. Experimental results show that our adaptive strategy significantly outperforms the conventional Expansion-move algorithm, in some cases cutting the runtime by 50%.

1. Introduction

Graph-cut based move-making algorithms such as Expansion and Swap [4] are extremely popular in computer vision. They enable researchers to efficiently compute approximate maximum *a posteriori* (MAP) solutions of Markov and Conditional Random Fields (MRFs, CRFs), and are used for solving a wide variety of labelling problems such as image segmentation [3, 19, 21], object-specific segmentation [2, 8, 26], geometric labelling [10, 20], image denoising/inpainting [9, 24, 28], stereo [4, 22, 31] and optical flow [4, 6].

Classical move-making algorithms such as Expansion and Swap operate by making a series of changes to the solution. These changes (also called moves) are performed such that they do not lead to an increase in the solution energy. Convergence is achieved when the energy cannot be decreased any further. In each iteration, the algorithm searches for a lower energy solution in a pre-defined neighborhood (also called the move space) around the current solution. It is important to draw a distinction between *moves* and *move-spaces*. For instance, the Expansion-move algorithm has $|\mathcal{L}|$ possible move-spaces (one corresponding to every possible label in $\mathcal{L}$), and 2^n possible moves within each space (one corresponding to a binary choice for every node). An α-expansion move (where $\alpha \in \mathcal{L}$) finds the minimum energy move within the move-space α.

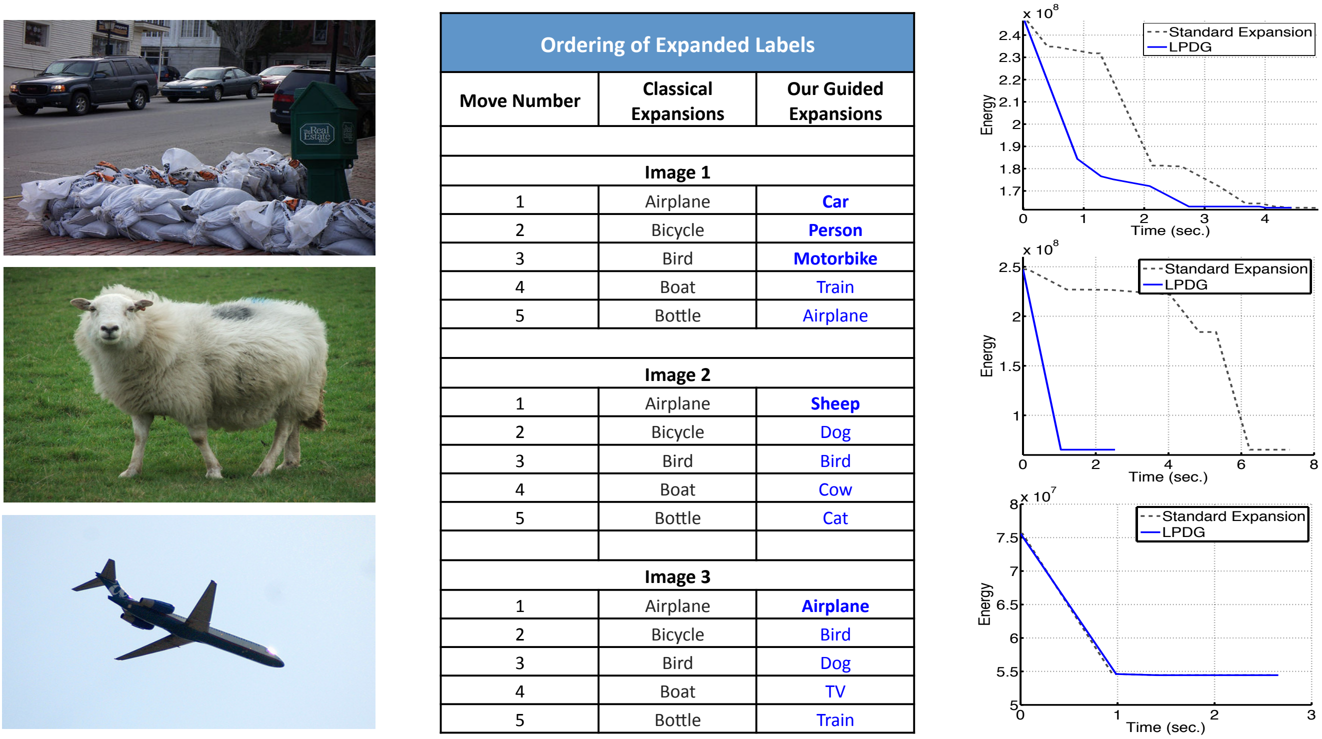


(a) Cones (b) Cones

(c) PASCAL (d) MSRC

Figure 1: Exploiting Label Sparsity: (a) shows the histogram of pixels in stereo sequence 'Cones' with respect to disparity labels. We can see that a lot of labels are never observed; (b) shows that our proposed approach is able to exploit this fact and cut the runtime by more than 50% compared to standard Expansion; (c,d) show histograms of number of labels observed in a single image for PASCAL 2007 and MSRC datasets, both containing 21 categories. We can see that most images contain less than 4 classes, and not a single image contains more than 10 labels.

Clearly, the move-space has an important impact on the performance of the algorithm. In fact, the size and form of the move-space is the key property that defines any move-making algorithms. There has been a lot of research on finding larger or more effective move spaces in which the optimal move (one that decreases the energy of the solu-



Move Number	Classical Expansions	Our Guided Expansions
	Image 1	
1	Airplane	**Car**
2	Bicycle	**Person**
3	Bird	**Motorbike**
4	Boat	Train
5	Bottle	Airplane
	Image 2	
1	Airplane	**Sheep**
2	Bicycle	Dog
3	Bird	Bird
4	Boat	Cow
5	Bottle	Cat
	Image 3	
1	Airplane	**Airplane**
2	Bicycle	Bird
3	Bird	Dog
4	Boat	TV
5	Bottle	Train

Figure 2: The first column of the figure shows three images from the dataset used for the PASCAL visual object category (VOC) challenge 2009. We used the pairwise energy functions for this problem used by [17]. The second column shows the first five labels used for expansion moves by the standard algorithm and our guided variant. It can be seen that our algorithm is able to propose labels relevant to the problem instance (marked in bold). The third column shows graphs of how the energy of the solutions obtained by standard Expansion and our guided variant changes with time. It can be seen that our algorithm consistently outperforms the standard Expansion algorithm.

tion by the most amount) can be computed in polynomial time [7, 16, 18, 29, 32].

However, there has been surprisingly little work on determining what is the *best move space* to search over, given the current solution and a family of move spaces. For instance, in the case of the Expansion algorithm, the standard approach is to iterate over the move spaces corresponding to different labels one after the other in a pre-specified or random order. This strategy is *energy-oblivious* and does not exploit the knowledge that certain labels are more probable to occur in the image and thus expanding them may lead to lower energies. Further, the standard Expansion algorithm makes complete sweeps over the label set. In many labelling problems, only a few labels are assigned in the MAP solution. This is particularly true in the case of object segmentation problems. Fig. 1 shows that individual images in the popular MSRC [26] (21 object labels) and PASCAL Visual Object Category dataset [5] (20 foreground and 1 background labels) contain very few labels. Specifically, even though both dataset contain 21 labels, most images contain less than 4 labels, and no image contains more than 10 labels. This certainly begs the question – why would we want to repeatedly iterate over *all possible* labels within the Expansion-move algorithm?

Contributions. In this paper, we propose an adaptive move-making algorithm that tries to find the best move space to search over in each iteration. Our algorithm tries to greedily find the label α for which the corresponding α-expansion move will lead to the most decrease in the energy. The labels chosen for some object segmentation problems are shown in Fig. 2. We can see that the labels chosen by our algorithm for Expansion are more meaningful and specific to the images. The figure also shows that the guided Expansion algorithm is able to find a lower-energy solution much more quickly compared to the standard method.

Our algorithm is inspired from the primal-dual interpretation of the Expansion algorithm given by Komodakis *et al.* [15, 16]. Komodakis and Tziritas [15] proposed three graph-cut-based primal-dual algorithms namely, PD1, PD2 and PD3 for performing MAP inference. Furthermore, they showed that one of their algorithms (PD3) has the exact same steps as that of the Expansion algorithm. Thus, the Expansion algorithm can be seen as solving a well known Linear Programming (LP) relaxation (and its corresponding dual) of the energy minimization problem.

The energy of the current solution and the achieved objective of the LP-dual provides upper and lower bounds respectively on the energy of the optimal solution. Furthermore, the difference between these values (called the primal-dual gap) provides a quantitative measure of the accuracy of the current solution. In fact, state-of-the-art LP-based methods for MAP inference [27] operate by incrementally tightening the relaxation to reduce the primal-dual gap. In the context of move-making algorithms, the above argument would lead to a strategy where we search for the optimal move in the move space that will lead to the biggest decrease in the primal-dual gap.

Finding the best move space is a difficult problem. For instance, to find out the best move space for Expansion na ively, one may need to try out all possible move spaces, *i.e*. one per label in the label set. This is a computationally expensive operation and would make the algorithm impractical for large labels spaces (*e.g*. in denoising applications). The key observation of this paper is that a good approximation to the relative drop in the primal-dual gap corresponding to different Expansion move spaces can be constructed using the primal and dual variables of the LP formulation of the problem. This approach is extremely efficient and runs in linear time in the number of variables and labels. We test the efficacy of our method on a number of image labelling problems. The experimental results show that our adaptive method significantly outperforms the widely used traditional Expansion-move algorithm.

2. Related Work

The last few years have seen the proposal of a number of sophisticated methods to improve the efficiency of move-making algorithms such as Expansion [1, 16, 18, 32]. These methods can be divided into two broad categories: *energy-oblivious* and *energy-aware*.

Energy-oblivious methods do not use the knowledge of the energy function to reduce computation time. The FAST-PD algorithm proposed by Komodakis *et al.* [16] and a related but simpler algorithm proposed by Alahari *et al.* [1] are two such methods. They use results of initial iterations of the Expansion algorithm to make subsequent iterations faster, and are inspired from the the dynamic graph cuts algorithm proposed by Kohli and Torr [13]. Another example is the fusion-move algorithm [18, 32]. This algorithm uses proposal solutions of the labeling problem obtained from different methods for defining the move space for the move making algorithm.

Our algorithm belongs to the class of energy-aware techniques which use the energy function to guide the search. The only other method which belongs to this class is the gradient-descent fusion-move algorithm proposed by Ishikawa [11]. This method tries to find the most promising move-space to search over by using the gradient of the energy function.

3. Notation and Preliminaries

We start by providing the notation used in the manuscript. For any positive integer n, let $[n]$ be shorthand for the set $\{1, 2, \ldots, n\}$. We consider a set of discrete random variables $\mathbf{x} = \{x_i \,|\, i \in [n]\}$, each taking value in a finite label set $x_i \in \mathcal{L} = [k]$. Let $G = (\mathcal{V}, \mathcal{E})$ be a graph defined over these variables, *i.e.* $\mathcal{V} = [n]$, $\mathcal{E} \subseteq \binom{[n]}{2}$. The goal of MAP inference is to find the labelling $\mathbf{x}$ of the variables which minimize a real-valued energy function associated with this graph:

$$\min_{\mathbf{x}\in\mathcal{L}^n} E(\mathbf{x}) = \min_{\mathbf{x}\in\mathcal{L}^n} \left(\sum_{i\in\mathcal{V}} \theta_i(x_i) + \sum_{(i,j)\in\mathcal{E}} \theta_{ij}(x_i, x_j) \right), \quad (1)$$

where $\theta_i(\cdot), \theta_{ij}(\cdot,\cdot)$ denote node and edge energies.

3.1. The Expansion-Move Algorithm

The Expansion algorithm starts with an initial solution and proceeds by making a series of changes which lead to solutions having lower energy. An α-expansion move over a label $\alpha \in \mathcal{L}$ allows any random variable to either retain its current label or take label α. One *sweep* of the algorithm involves making moves for all $\alpha \in \mathcal{L}$ in some order successively.

The optimal α-expansion move can be computed in polynomial time if the pairwise energy parameters θ_{ij} define a metric, *i.e.*

$$\theta_{ij}(a, b) = 0 \quad \iff \quad a = b, \quad (2a)$$

$$\theta_{ij}(a, b) = \theta_{ij}(b, a) \geq 0 \quad (2b)$$

$$\theta_{ij}(a, c) \leq \theta_{ij}(a, b) + \theta_{ij}(b, c), \qquad \forall a, b, c \in \mathcal{L}. \quad (2c)$$

In this work, we do not assume the pairwise energies to be metrics. We only require: $\theta_{ij}(a, a) = 0,\ \forall a$ and $\theta_{ij}(a, b) \geq 0$.

3.2. MAP Integer Program

MAP inference is typically set up as an integer programming problem over the boolean variables. Let $\mu_i(s), \mu_{ij}(s, t) \in \{0, 1\}$ be indicator variables, such that $\{\mu_i(s) = 1 \Leftrightarrow x_i = s\}$, and $\{\mu_{ij}(s, t) = 1 \Leftrightarrow x_i = s, x_j = t\}$. Moreover, let $\boldsymbol{\mu}_i = \{\mu_i(s) \mid s \in \mathcal{L}\}, \boldsymbol{\theta}_i = \{\theta_i(s) \mid s \in \mathcal{L}\}$ be vectors of indicator-variables and energies for node i. Let $\boldsymbol{\mu}_{ij}$ and $\boldsymbol{\theta}_{ij}$ be defined analogously. Using this notation, the MAP inference integer program can be written as:

$$\min_{\boldsymbol{\mu}_i, \boldsymbol{\mu}_{ij}} \quad \sum_{i\in\mathcal{V}} \boldsymbol{\theta}_i \cdot \boldsymbol{\mu}_i + \sum_{(i,j)\in\mathcal{E}} \boldsymbol{\theta}_{ij} \cdot \boldsymbol{\mu}_{ij} \quad (3a)$$

$$s.t. \quad \sum_{s\in\mathcal{L}} \mu_i(s) = 1 \qquad \forall i \in \mathcal{V} \quad (3b)$$

$$\sum_{s\in\mathcal{L}} \mu_{ij}(s, t) = \mu_j(t) \qquad \forall (i, j), (j, i) \in \mathcal{E} \quad (3c)$$

$$\mu_i(s), \mu_{ij}(s, t) \in \{0, 1\}. \quad (3d)$$

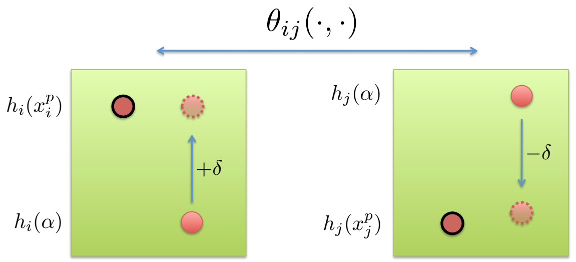


Figure 3: Interpretation of dual variables. See text for details

Problem (3) is known to be NP-hard in general [25]. The standard LP relaxation of this problem, also known as Schlesingers bound [23, 30], is given by relaxing the boolean constraints (3d) to the unit interval, *i.e.* $\mu_i(s), \mu_{ij}(s,t) \geq 0$.

3.3. The Primal-Dual Interpretation of Expansion

Komodakis *et al.* [15, 16] gave a primal-dual interpretation of α-expansion. Since our contribution builds on their interpretation, we briefly review their work here.

The LP-dual of (3) that Komodakis *et al.* [15, 16] chose to work with was:

$$\begin{aligned} \max_{h_i, y_{ij}(t)} \quad & \sum_{i \in \mathcal{V}} h_i && && (4) \\ s.t. \quad & h_i \leq h_i(s) && \forall i \in \mathcal{V}, s \\ & y_{ij}(t) + y_{ji}(s) \leq \theta_{ij}(s,t) && \forall (i,j) \in \mathcal{E}, s, t \\ & h_i, y_{ij}(t) \in \mathbb{R}, \end{aligned}$$

where $h_i(s) \doteq \theta_i(s) + \sum_{j \in \mathcal{N}(i)} y_{ji}(s)$, where $\mathcal{N}(i) = \{j \,|\, (i,j) \in \mathcal{E}\}$ is the set of neighbours of node i.

Every feasible dual solution provides a lower-bound on the MAP value, *i.e.*:

$$E(\mathbf{x}) \geq E(\mathbf{x}^*) \geq \sum_{i \in \mathcal{V}} h_i \tag{5}$$

Moreover, for any primal labelling $\mathbf{x}^p$, the quantity Primal-Dual Gap ($E(\mathbf{x}^p) - \sum_{i \in \mathcal{V}} h_i$) gives an estimate of the tightness of the relaxation. Specifically, a pair of primal-dual solutions $(\mathbf{x}^p, \{h_i, h_i(\cdot), y_{ij}(\cdot)\})$ is an f-approximate solution iff:

$$E(\mathbf{x}^p) = \sum_{i \in \mathcal{V}} \theta_i(x_i^p) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}(x_i^p, x_j^p) \leq f \left(\sum_{i \in \mathcal{V}} h_i \right) \tag{6}$$

Komodakis *et al.* [15, 16] proposed two primal-dual algorithms called PD3 [15] and FastPD [16][1] that were both shown to be equivalent to α-expansion. These algorithms are based on a particular set of relaxed complementary slackness conditions for the primal-dual programs. Specifically, if we define $d_{\max} \doteq \max_{ij,s,t} \theta_{ij}(s,t)$ and $d_{\min} \doteq \min_{ij,s \neq t} \theta_{ij}(s,t)$, a pair of primal-dual solutions $(\mathbf{x}^p, \{h_i, h_i(\cdot), y_{ij}(\cdot)\})$ achieves an $f_{app} = 2 \cdot \frac{d_{\max}}{d_{\min}}$-approximation ratio iff [14, 15]:

$$h_i(x_i^p) = \min_{s \in \mathcal{L}} h_i(s) \tag{7a}$$

$$y_{ij}(x_j^p) + y_{ji}(x_i^p) = \theta_{ij}(x_i, x_j) \tag{7b}$$

$$y_{ij}(t) + y_{ji}(s) \leq 2\theta_{ij}(s,t) \tag{7c}$$

Intuitive Interpretation + Example: An intuitive interpretation of the dual variables helps in understanding these constraints. Fig. 3 shows the dual variables $h_i(x_i^p), h_i(\alpha), h_j(x_j^p), h_j(\alpha)$ associated with an edge (i,j). The cost of each node-label pair is represented by a ball with that *height*. Condition (7a) requires that the height of ball corresponding to primal labelling be the lowest. We can see that this is satisfied for node j but not for node i, where $h_i(\alpha) < h_i(x_i^p)$. Recall that $h_i(s) = \theta_i(s) + \sum_{j \in \mathcal{N}(i)} y_{ji}(s)$. Thus, we may increase the height of α at node i, $h_i(\alpha)$, by increasing the dual variable y_{ji}, *i.e.* by setting $y_{ji}(\alpha) \leftarrow y_{ji}(\alpha) + \delta$. However, constraint (7c) forces:

$$y_{ij}(\alpha) + y_{ji}(\alpha) \leq 2\theta_{ij}(\alpha,\alpha) = 0 \tag{8}$$

$$\Rightarrow \quad y_{ij}(\alpha) \leq -y_{ji}(\alpha). \tag{9}$$

Thus, increasing $h_i(\alpha)$ decreases $h_j(\alpha)$ by at least the same amount, *i.e.*, $h_j(\alpha) \leftarrow h_j(\alpha) - \delta$. In this example, this change increased the dual by $+\delta$, *i.e.* $h_i + h_j \leftarrow h_i + h_j + \delta$.

For a general graph, both algorithms – PD3 [15] and FastPD [16] – operate in a block-coordinate ascent fashion. They loop over labels in a pre-determined order and at each step optimize all dual-variables corresponding to a label α, while holding all other dual variables fixed.

- Loop on label $\alpha \in \mathcal{L}$:
  - α-expansion: Update $\{h_i, h_i(\alpha), y_{ij}(\alpha), y_{ji}(\alpha)\}$; Update $\{x_i^p \,|\, x_i^p = \alpha\}$.

In both algorithms, conditions (7b), (7c) are satisfied by construction, and the goal of these block-updates is to make progress on condition (7a), *i.e.* to extract a primal labelling that has the minimal height at all nodes. We refer the reader to Komodakis *et al.* [15, 16] for details of the update steps.

4. Adaptive Primal-Dual Expansion Moves

In this paper we focus on how to enumerate over the labels for primal-dual Expansion moves. Typical choices are to loop over labels one after the other in a pre-specified or random order. This is exactly the scheme that FastPD follows. We follow an *energy-aware* strategy that uses the primal and dual variables to make adaptive Expansion moves. Specifically, we present a label proposal heuristic that sorts labels according to a scoring function that quantifies how helpful a label will be for α-expansion. Our label scoring function is derived from complementary slackness condition (7a).

[1] FastPD is a faster version of PD3.

Algorithm 1 LPDG-Sweep

```
for $t = 0, 1, \ldots$ do
    Compute LPDG-based label scores $\omega(\cdot) \in \{\omega_1(\cdot), \omega_2(\cdot), \omega_3(\cdot)\}$.
    Sort label scores: $\{\alpha_1, \ldots, \alpha_k\} = \text{sort}(\omega(\cdot))$
    for $i = 1, \ldots, k$ do
        $\alpha_i$-expansion: Update variables: $\{h_i, h_i(\alpha), y_{ij}(\alpha), y_{ji}(\alpha)\}$
    end for
end for
```

4.1. Local Primal-Dual Gap

We first define a quantity we call *Local Primal-Dual Gap* (LPDG) for every label and variable. Given a pair of primal-dual solutions $(\mathbf{x}^p, \{h_i, h_i(\cdot), y_{ij}(\cdot)\})$, LPDG is formally defined as:

$$lpdg(\alpha, i) = h_i(x_i^p) - h_i(\alpha). \tag{10}$$

Using the ball analogy of dual variables from Fig. 3 again, LPDG can be seen as the height difference between the balls at node i corresponding to the primal labelling x_i^p and the label α. Positive values of LPDG indicate violations of complementary slackness, and we can say that dual variable $h_i(\alpha)$ is in *deficit* by an amount of $lpdg(\alpha, i)$. Negative values of LPDG indicate that complementary slackness is satisfied and that dual variable $h_i(\alpha)$ is in *surplus* by amount $lpdg(\alpha, i)$. A dual variable in deficit $h_i(\alpha)$ needs to be "fixed" by the algorithm and a neighbouring dual variable in surplus $h_j(\alpha), j \in \mathcal{N}(i)$ can help this process by via the message-variables $y_{ij}(\alpha)$, $y_{ji}(\alpha)$. Thus, $lpdg(\alpha, i)$ quantifies the amount of violation in complementary slackness conditions at node i, label α, and $lpdg(\alpha, j)$ quantifies how much node j can help correct this violation. More specifically, we can state the following property of LPDG:

Proposition 1 *Slackness: If LPDG for all nodes is non-positive,* i.e. $lpdg(\alpha, i) \leq 0$, $\forall i \in \mathcal{V}, \alpha \in \mathcal{L}$*, then complementary slackness conditions hold and we have achieved an* f_{app}*-approximate solution.*

4.2. LPDG-based Label Scoring Functions

LPDG gives us an appropriate language to describe our goals for label scoring. In order to get the highest drop in primal-dual gap, we need to pick a label α that has the most deficit. However, we also need to make sure there is enough surplus so that we may actually make an improvement. Keeping this in mind, we propose the following three label scoring functions:

- LPDG-crisp:
  $\omega_1(\alpha) = \sum_{i \in \mathcal{V}} \mathbb{I}_{[0\ \infty)}\left(lpdg(\alpha, i)\right)$,

$$\text{where} \quad \mathbb{I}_S(y) = \begin{cases} 1 & y \in S \\ 0 & \text{else} \end{cases} \tag{11}$$

Algorithm 2 LPDG-Partial-Sweep

```
for $t = 0, 1, \ldots$ do
    Compute LPDG-based label scores $\omega(\cdot) \in \{\omega_1(\cdot), \omega_2(\cdot), \omega_3(\cdot)\}$.
    Find highest label score: $\alpha = \text{argmax}_{\hat{\alpha}}\, \omega(\hat{\alpha})$
    $\alpha$-expansion: Update variables: $\{h_i, h_i(\alpha), y_{ij}(\alpha), y_{ji}(\alpha)\}$
end for
```

- LPDG-deficit:
  $\omega_2(\alpha) = \sum_{i \in \mathcal{V}} lpdg(\alpha, i) \cdot \mathbb{I}_{[0\ \infty)}\left(lpdg(\alpha, i)\right)$
- LPDG-tradeoff:
  $\omega_3(\alpha) = \sum_{i \in \mathcal{V}} \left|lpdg(\alpha, i)\right|$

LPDG-crisp ignores the actual LPDG values and simply counts the number of nodes in deficit. LPDG-deficit on the other hand also incorporates the how much these nodes are in deficit. Finally, LPDG-tradeoff incorporates both the amount of deficit and the surplus over all nodes.

Finally, we propose the following two adaptive primal-dual Expansion algorithms: 1) LPDG-Sweep, shown in Alg 1 that uses an LPDG-scoring-function-based permutation of labels to perform α-expansion in each sweep, and 2) LPDG-Partial-Sweep, show in Alg. 2, that performs partial sweeps where LPDG-based reordering of labels is performed after each alpha-expansion.

The standard Expansion algorithm goes through all labels in one iteration, and thus has a linear runtime complexity in the number of labels in the problem. This makes it inefficient on problems with very large label sets. Our partial-sweep scheme based on LPDG scores is not bound to the linear complexity and only expands labels which are relevant to the problem instance.

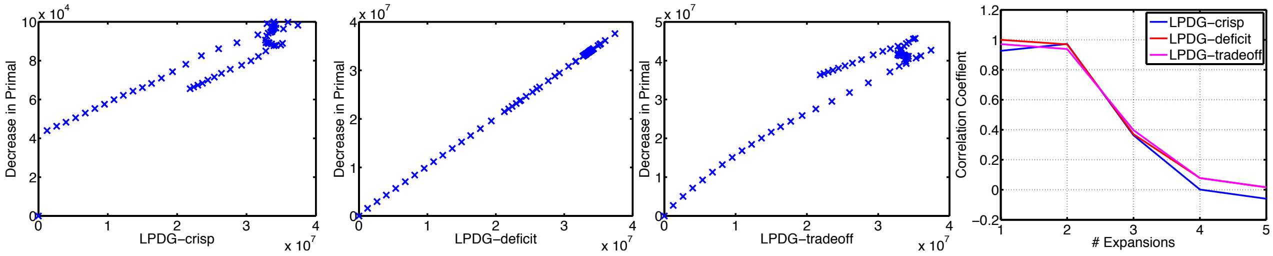
5. Experiments

We evaluated our method on the problems of object segmentation and stereo matching.

Stereo: We use image pairs from the Middlebury Stereo Dataset.[2] We used the energy function made available by Alahari *et al.* [1]. The number of disparity labels in different problem instances were: Tsukuba (16), Venus (20), Cones (60) and Teddy (60).

Object Segmentation: For this application, we used the pairwise energy functions constructed by Ladicky *et al.* [17] that are based on the method of Shotton *et al.* [26]. We tested our algorithm on some images from the MSRC [26] (21 object labels) and PASCAL Visual Object Category dataset [5] (20 foreground and 1 background labels) datasets.

[2] http://vision.middlebury.edu/stereo/data/.



(a) LPDG-crisp. (b) LPDG-deficit. (c) LPDG-tradeoff. (d) Correlation Coefficients.

Figure 4: **Correlation between LPDG-based scores and decrease in energy.** Graph (a) shows the relationship between the true decrease in energy and the LPDG-crisp score. Each point on the graph corresponds to a proposed label-expansion at the start of the energy minimization procedure. The x-axes shows what was the LPDG-crisp score for the particular label expansion, while the y-axes shows what was the decrease in the energy of the solution when this label-expansion was performed. Graphs (b) and (c) show the corresponding plots for the LPDG-deficit and LPDG-trade-off scores respectively. It can be seen that all scores are highly correlated at the start. Graph (d) shows that the correlation between the LPDG scores and the decrease in the energy falls as we perform more expansion moves.

Correlation between LPDG-ranking and True-ranking To quantify the performance of our LPDG-based metrics in predicting usefulness of labels, we measure the correlation between the proposed label LPDG scores and the true decrease in the primal energy. To do this, we performed the following experiment: we first let k sweeps of alpha-expansion occur in the usual fashion (*i.e.*, in some static pre-fixed ordering over the variables). After the completion of these k standard sweeps, we loop over all labels and for each label l we record all three of our label scores and also record the drop in energy if this label was expanded on *immediately* after finishing the k standard sweeps. Fig 4 shows the correlation plots for the $k = 0$ case, *i.e.* without any sweeps. We can see a very strong correlation for all three of our label scores. Fig 4 also shows the correlation coefficient as a function of k. We can see that the correlation is very high initially and then slows decays away as the problem moves towards convergence. However, it is important to remember that with methods like FastPD it is the *first few* iterations that are responsible for *most* of the time taken by the algorithm and this is precisely where LPDG is most helpful.

Comparison with FAST-PD. We compare the performance of our method with the FAST-PD algorithm proposed by Komodakis *et al.* [16], which is an efficient version of the standard Expansion algorithm. To ensure our experiments assess the relative performance of our proposed LPDG scores and are not influenced by the particular details of the primal-dual alpha-expansion implementation, we use the use the implementation provided by Komodakis *et al.* for all our experiments.[3] All methods reported in this paper differ only in the order of expansions and thus the relative performance differences can be directly attributed to our improved score function.

Fig. 5 shows the energy-vs-iteration and energy-vs-time plots for stereo problems, while Fig. 2 and 6 shows the same for object segmentation problems. We can see that in all cases LPDG-based dynamic label schedules perform significantly better than the standard static schedules. For most problem instances, our guided approach can reduce the running time by more than 50%.

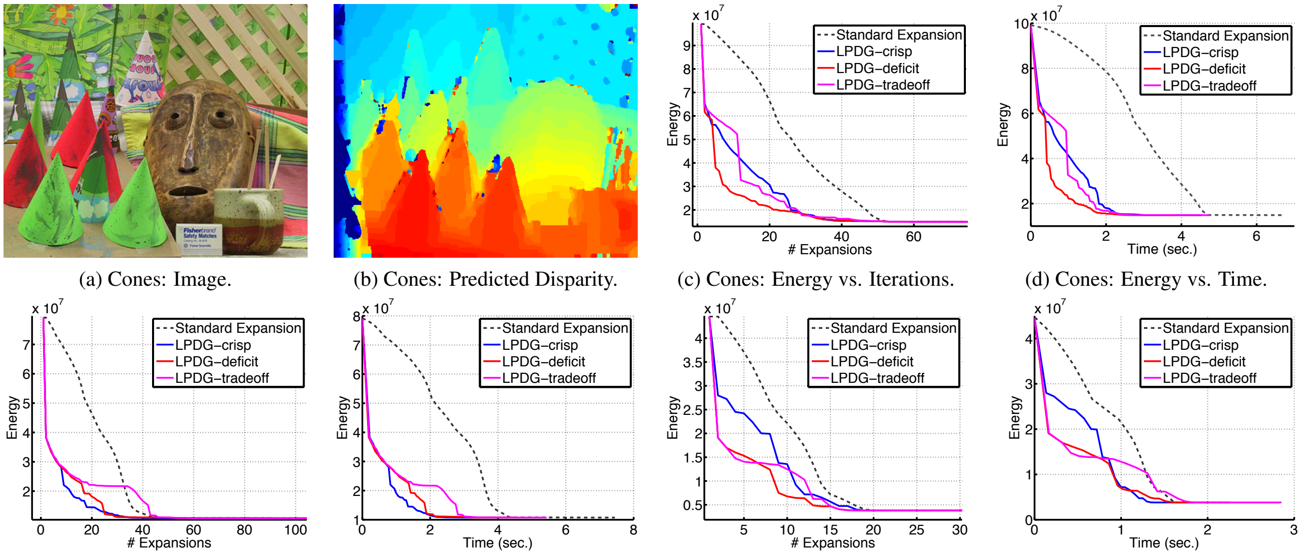
Complete vs Partial Label Sweep. We experimented with partial sweeps but on Pascal it performed very similar to the full sweep algorithm, which also converges very quickly. We believe the reason for this is that the energies are fairly confident, and most labels are only expanded on once. On Stereo, we were able to observe improvement of partial sweeps over full sweeps, typically by 10-20%. We believe problems with extremely large label spaces where individual instances contain only a small subset of these labels would most benefit from our partial sweep algorithm. Fig. 7 shows performance of all scoring function for full and partial sweep on the Venus stereo problem.

Initialization vs. Dynamic Schedules. For all methods presented in this paper, we initialized the primal labelling uniformly with ones, *i.e.* $x_i^p = 1, \forall i \in [n]$. Of course, a better initialization scheme based on unary potentials might help, *e.g.*, we can initialize with the labels that minimize node energies independently, *i.e.* $x_i = \operatorname{argmin}_{s \in \mathcal{L}} \theta_i(s)$. However, we found in our experiments that even with a *bad* uniform initialization, LPDG-based methods outperformed standard Expansion with a good initialization based on unary potentials. This demonstrates the power of LPDG based dynamic scheduling. Of course, ideally we should use both a good initialization and a good dynamic scheduling scheme. Fig. 8 shows the results.

6. Conclusions

We presented a novel method for proposing good expansion moves that significantly speeds up the Expansion and FAST-PD energy minimization algorithms. The results

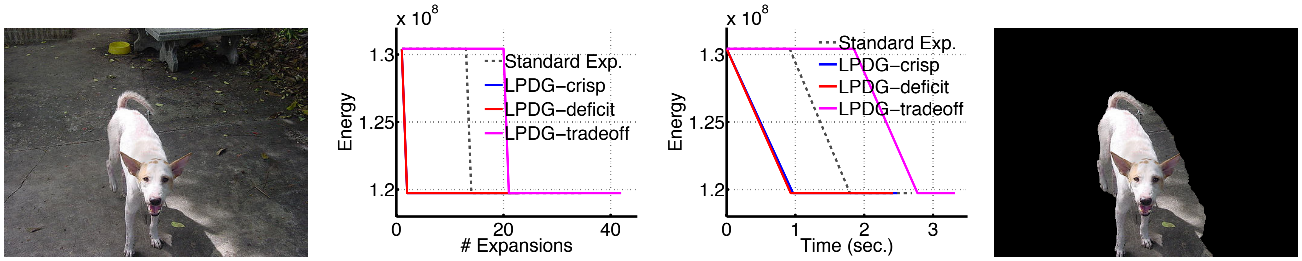
[3] http://www.csd.uoc.gr/~komod/FastPD/index.html.



(a) Cones: Image. (b) Cones: Predicted Disparity. (c) Cones: Energy vs. Iterations. (d) Cones: Energy vs. Time.

(e) Teddy: Energy vs. Iterations. (f) Teddy: Energy vs. Time. (g) Venus: Energy vs. Iterations. (h) Venus: Energy vs. Time.

Figure 5: Results on energy functions used for Stereo-Matching. (a) Cones image from the Middlebury Stereo Dataset. (b) The computed disparity map by minimizing the energy used in [1]. (c) Graph showing how the energy of the solution changes with number of expansions of the different variants of guided Expansions and standard FAST-PD. (d) Graph showing how the energy of the solution obtained by different variants of guided Expansion and FAST-PD changes with running time. (e) and (f) Results for Teddy image pair. (g) and (h) Results for Venus image pair.



(a) Image. (b) Energy vs. Iterations. (c) Energy vs. Time. (d) Segmentation.

Figure 6: Results on energy functions used for Object Segmentation. (a) Image from the PASCAL VOC 2009 dataset. (b) Solution obtained by minimizing the pairwise energy function used in [17]. (c) Graph showing how the energy of the solution changes with number of expansions of the different variants of guided Expansions and standard FAST-PD. (d) Graph showing how the energy of the solution obtained by different variants of guided Expansion and FAST-PD changes with running time.

of our experiments have demonstrated that primal and dual solutions can be used to make good predictions on which label-expansion will lead to lower energy solutions. We believe that the theory developed in this paper, and our method will help significantly reduce the time taken for energy minimization in problems which are defined over large label sets (*e.g*. denoising).

This paper focused on developing the LPDG-based ranking scores for Expansion moves. Extending these ideas to general moves like Range [29] and Fusion [18, 32] is an interesting direction for future work.

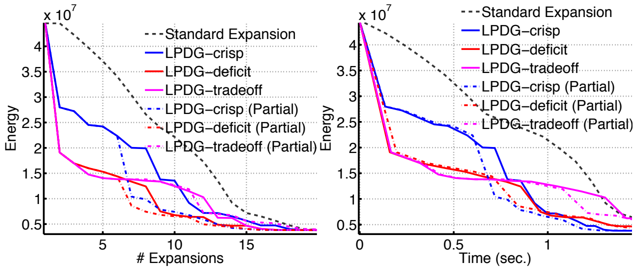
References

[1] K. Alahari, P. Kohli, and P. H. S. Torr. Dynamic hybrid algorithms for map inference in discrete mrfs. *PAMI*, pages 1846–1857, 2010. `http://cms.brookes.ac.uk/staff/Karteek/data.tgz`. 1867, 1869, 1871

[2] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010. 1865

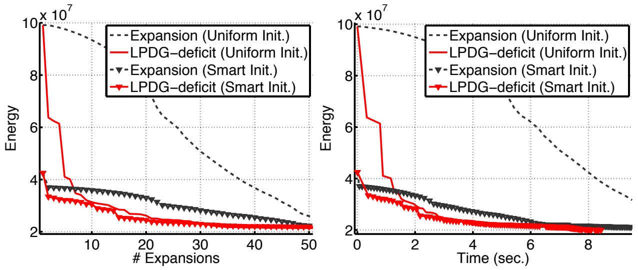
[3] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *ICCV*, 2001. 1865

[4] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *PAMI*, 20(12):1222–1239, 2001. 1865



(a) Venus: Energy vs. Iteration. (b) Venus: Energy vs. Time.

Figure 7: Comparative evaluation of the LPDG-partial sweep algorithm (Alg 2) on the Venus stereo problem instance.



(a) Cones: Energy vs. Iteration. (b) Venus: Energy vs. Time.

Figure 8: Initialization vs. Dynamic Schedules.

[5] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2009 Results. http://www.pascal-network.org /challenges/VOC/voc2009/workshop/index.html. 1866, 1869

[6] B. Glocker, H. Heibel, N. Navab, P. Kohli, and C. Rother. Triangleflow: Optical flow with triangulation-based higher-order likelihoods. In *11th European Conference on Computer Vision (ECCV)*, Crete, Greece, September 2010. 1865

[7] S. Gould, F. Amat, and D. Koller. Alphabet soup: A framework for approximate energy minimization. In *CVPR*, pages 903–910, 2009. 1866

[8] X. He, R. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *ECCV*, 2006. 1865

[9] I. Hiroshi. Higher-order clique reduction in binary graph cut. In *CVPR*, 2009. 1865

[10] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 75(1), 2007. 1865

[11] H. Ishikawa. Higher-order gradient descent by fusion-move graph cut. In *ICCV*, 2009. 1867

[12] P. Kohli, M. Kumar, and P. Torr. P3 and beyond: Solving energies with higher order cliques. In *CVPR*, 2007.

[13] P. Kohli and P. H. S. Torr. Effciently solving dynamic markov random fields using graph cuts. In *ICCV*, pages 922–929, 2005. 1867

[14] V. Kolmogorov. A note on the primal-dual method for the semi-metric labeling problem. University College London Tech. Report., 2007. 1868

[15] N. Komodakis and G. Tziritas. Approximate labeling via graph cuts based on linear programming. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(8):1436–1453, 2007. 1866, 1868

[16] N. Komodakis, G. Tziritas, and N. Paragios. Performance vs computational efficiency for optimizing single and dynamic mrfs: Setting the state of the art with primal-dual strategies. *Computer Vision and Image Understanding*, 112(1):14–29, 2008. 1866, 1867, 1868, 1870

[17] L. Ladickỳ, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical crfs for object class image segmentation. *ICCV*, 2009. 1866, 1869, 1871

[18] V. S. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for markov random field optimization. *PAMI*, 2010. 1866, 1867, 1871

[19] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *SIGGRAPH*, 2004. 1865

[20] S. Ramalingam, P. Kohli, K. Alahari, and P. H. S. Torr. Exact inference in multi-label crfs with higher order cliques. *CVPR*, 2008. 1865

[21] C. Rother, V. Kolmogorov, and A. Blake. "Grabcut": interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, 2004. 1865

[22] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002. 1865

[23] M. I. Schlesinger. Syntactic analysis of two-dimensional visual signals in noisy conditions (in Russian). *Kibernetika*, 4:113–130, 1976. 1868

[24] G. Sebastiani and F. Godtliebsen. On the use of gibbs priors for bayesian image restoration. *Signal Processing*, 56(1):111 – 118, 1997. 1865

[25] S. E. Shimony. Finding maps for belief networks is np-hard. *Artificial Intelligence*, 68(2):399–410, August 1994. 1868

[26] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(1), 2009. 1865, 1866, 1869

[27] D. Sontag and T. Jaakkola. New outer bounds on the marginal polytope. In *NIPS*, 2007. 1867

[28] 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 with smoothness-based priors. *PAMI*, 30(6):1068–1080, 2008. 1865

[29] O. Veksler. Graph cut based optimization for mrfs with truncated convex priors. In *CVPR*, 2007. 1866, 1871

[30] T. Werner. A linear programming approach to max-sum problem: A review. *PAMI*, 29(7):1165–1179, 2007. 1868

[31] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *CVPR*, 2008. 1865

[32] O. J. Woodford, P. H. S. Torr, I. D. Reid, and A. W. Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *PAMI*, 2009. 1866, 1867, 1871