Efficiently Selecting Regions for Scene Understanding[*]

M. Pawan Kumar
Computer Science Department
Stanford University
pawan@cs.stanford.edu

Daphne Koller
Computer Science Department
Stanford University
koller@cs.stanford.edu

Abstract

Recent advances in scene understanding and related tasks have highlighted the importance of using regions *to reason about high-level scene structure. Typically, the regions are selected beforehand and then an energy function is defined over them. This two step process suffers from the following deficiencies: (i) the regions may not match the boundaries of the scene entities, thereby introducing errors; and (ii) as the regions are obtained without any knowledge of the energy function, they may not be suitable for the task at hand. We address these problems by designing an efficient approach for obtaining the best set of regions in terms of the energy function itself. Each iteration of our algorithm selects regions from a large dictionary by solving an accurate linear programming relaxation via dual decomposition. The dictionary of regions is constructed by merging and intersecting segments obtained from multiple bottom-up over-segmentations. To demonstrate the usefulness of our algorithm, we consider the task of scene segmentation and show significant improvements over state of the art methods.*

1. Introduction

As low-level vision reaches maturity, more and more researchers have started considering high-level scene understanding tasks such as scene segmentation, object detection and single view 3D reconstruction. Given the success of pixel-based approaches for low-level vision (such as image denoising, where each pixel is assigned an intensity value; or stereo reconstruction, where each pixel is assigned a disparity value), the initial expectation was that such models will also be suitable for scene understanding. However, this expectation has turned out to be overly optimistic as features extracted from patches surrounding pixels are prone to noise from background clutter. To address this issue, many researchers have advocated the use of *regions*, sets of connected pixels that share common characteristics such as color or texture, to extract reliable discriminative features.

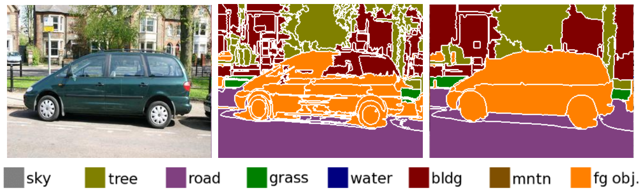


Figure 1. *The first image is the input to a region-based scene segmentation algorithm. The second image shows the result of using segments as regions. The third image is the result obtained using the task-specific regions selected from a large dictionary. Note that a single over-segmentation is not faithful to the scene boundaries. For example, it merges tire pixels with the road. Furthermore, the small size of the regions results in the algorithm confusing foreground regions with building. In contrast, our approach provides a clean segmentation of the foreground object.*

Typically, the regions are used to define an energy function whose value depends on the labels assigned to the regions. The labels and the energy function are task dependent. For instance, the labels may represent semantic classes for scene segmentation or depth values for 3D reconstruction. The energy function captures our knowledge about the particular task (such as the smoothness of labeling or contextual cues). However, the regions themselves are often selected independent of the task.

The choice of regions is an important one: while smaller regions allow us to capture the boundary of scene entities accurately, larger regions provide more reliable features. To balance this trade-off, most works so far have used segments obtained from bottom-up approaches as regions. However, since bottom-up methods have no knowledge of the energy function, the regions obtained in this manner may not be suitable for the task at hand. We argue that a successful scene understanding approach would use regions that are specifically chosen for the particular task. To this end, we design an efficient algorithm for selecting regions that approximately minimize the energy of a region-based model.

Our model is fairly generic and is capable of formulating many high-level problems. It consists of two layers: (i) each pixel is assigned to a unique region; and (ii) each region is assigned a unique label. Minimizing the energy associated with this model is extremely challenging due to the large number of possible pixel-to-region assignments. Furthermore, the regions are required to be connected, a difficult constraint to impose [24, 36]. In order to overcome these difficulties, we make clever use of bottom-up over-segmentation algorithms. We obtain multiple over-segmentations by changing the parameters of a bottom-up approach. The different parameter settings provide segments of different quantizations, from small segments that respect the scene boundaries to large segments that provide good features. We define our regions as sets of pixels obtained by merging and intersecting the segments with each other. While merging segments together provides large regions, their intersection with small segments ensures that they align well with the boundary (see Fig. 1).

While the use of bottom-up segmentations prunes down

[*] Supported by NSF under grant IIS 0917151, MURI contract N000140710747, and the Boeing company.

the pixel-to-region assignments and provides connected putative regions, we are still faced with the difficult task of selecting the best set of regions (in terms of the energy of the model) from a large dictionary. We address this problem by formulating it as an integer program and designing an accurate linear programming (LP) relaxation for it. Furthermore, we show how the LP relaxation can be solved efficiently by suitably modifying the dual decomposition framework [3]. To demonstrate the usefulness of our approach, we consider the task of scene segmentation and show significant improvements over state of the art methods.

2. Related Work

Earlier works on the various aspects of scene understanding built models whose variables corresponded to the pixels of an image. Each pixel was assigned a label using features extracted from a regularly shaped patch around it [12, 19] or at an offset from it [32]. However, the features extracted from such patches are not reliable in the presence of background clutter. For example, in scene segmentation, a patch around a boundary pixel of a tree may contain sky or building pixels. This may inhibit an algorithm from using the fact that trees are mostly green.

To avoid the problem of pixel-based methods, some works use an over-segmentation of the image obtained from one of the many available bottom-up approaches [1, 7]. Each segment is treated as a region and an energy function is defined over them [22, 27, 31]. However, these regions may not capture the boundaries between the scene entities accurately, which would imply that: (i) even the best possible labeling of the regions would have errors since each region can only be assigned one label; and (ii) the features obtained from the regions would not be robust to background clutter (for instance, in our scene segmentation example we may still end up with a tree region that contains sky or building pixels). To address this issue, many researchers have suggested heuristics for selecting a good over-segmentation. These heuristics use low-level cues such as the *stability* of a segment [27] or high-level cues such as bounding boxes of objects [22]. However, unlike our approach, these methods do not select the regions based on a global energy function for the specific high-level task under consideration.

In order to obtain regions that capture the boundaries accurately, researchers have now started using multiple over-segmentations of an image. Pontafaru *et al.* [26] observed that by taking the intersection of the over-segmentations, we can obtain regions that are true to the scene boundaries. However, such regions are often very small and do not allow us to capture high-level scene structure. For example, small regions do not provide useful shape-based cues, which may be essential in distinguishing man-made objects (such as buildings and roads that often contain straight lines) from natural ones (such as mountains that would provide a high residue for a line fitting algorithm).

The work of Kohli *et al.* [15] and its recent extension [21], provides a graph-cuts based algorithm for combining multiple over-segmentations. However, there are important differences between their approach and ours, namely: (i) While our approach selects regions for a given energy function, their method modifies the energy function itself to accommodate the multiple segmentations. Specifically, they restrict the region potentials to be of a form that penalizes pixels that do not take the dominant label of that region. This restriction limits the capability of the energy function to fully capture our knowledge about the high-level task; and (ii) Our regions do not overlap with each other, thereby providing a coherent explanation of the image. In contrast, the methods proposed in [15, 21] consider overlapping regions that can be assigned different labels. This effectively implies that the pixels within the overlap are used to support two contradicting hypotheses (that is, data is overcounted).

The method that is most related to ours is the recent work of Gould *et al.* [8, 9]. Like our approach, they construct a dictionary of putative regions and select the best set of non-overlapping regions based on the energy function of their model. However, unlike our LP relaxation, their method makes a small move at each iteration, which (as our results show) is prone to getting stuck in a bad local minima.

3. The Region-based Model

Our region-based model has two layers. The first layer consists of random variables $\mathcal{V}^P$ that correspond to the pixels of a given image $\mathbf{X}$. Each random variable can take one label from the set $\mathcal{L}^P = \{1, 2, \cdots, R\}$ that represents the region to which it belongs. Here R is the total number of regions in a given image (which has to be inferred automatically). A labeling of the first layer is a vector $\mathbf{Y}^P$ that divides the pixels into connected, non-overlapping regions. The second layer consists of random variables $\mathcal{V}^R(\mathbf{Y}^P)$ that correspond to the regions defined by $\mathbf{Y}^P$. Each random variable can take one label from the set $\mathcal{L}^R = \{1, 2, \cdots, L\}$ where L is the total number of region labels that we consider. The desired output of the high-level task is provided by a labeling $\mathbf{Y}^R$ of the second layer.

A labeling of the entire model is denoted by $\mathbf{Y} = (\mathbf{Y}^P, \mathbf{Y}^R)$. Its energy consists of two types of potentials:
(i) For each variable $v_r \in V^R(\mathbf{Y}^P)$ (corresponding to a region r) we define a unary potential $\theta_r(\mathbf{Y}^R_r; \mathbf{X})$ for assigning it the label $\mathbf{Y}^R_r$. The unary potential can capture information, such as, that green regions are likely to be grass or tree, while blue regions are likely to be sky.
(ii) For each pair of neighboring variables, whose corresponding regions share at least one boundary pixel, we define a pairwise potential $\theta_{rr'}(\mathbf{Y}^R_r, \mathbf{Y}^R_{r'}; \mathbf{X})$ for assigning labels $\mathbf{Y}^R_r$ and $\mathbf{Y}^R_{r'}$ to v_r and $v_{r'}$ respectively. The pairwise potentials can be used to encourage spatial contiguity and capture contextual information, such as, that boats are usually on water while cars are usually on roads.

Optionally, one may also include a regularization (for example, an ℓ_1 penalty) for the number of regions, which can be handled easily using our approach. However, for this work, we consider the total energy function to be of the form

$$\begin{aligned} E(\mathbf{Y};\mathbf{X}) = & \sum_{v_r \in \mathcal{V}^R(\mathbf{Y}^P)} \theta_r(\mathbf{Y}_r^R) + \\ & \sum_{(v_r, v_{r'}) \in \mathcal{E}^R(\mathbf{Y}^P)} \theta_{rr'}(\mathbf{Y}_r^R, \mathbf{Y}_{r'}^R), \quad (1) \end{aligned}$$

where $\mathcal{E}^R(\mathbf{Y}^P)$ is the set of all neighboring regions defined by $\mathbf{Y}^P$. Note that we have dropped the term $\mathbf{X}$ from the individual potentials to make the notation less cluttered. For the model described above, we consider the problem of obtaining the labeling $\mathbf{Y}^*$ that minimizes the energy function. This will provide us with the best set of regions for the task ($\mathbf{Y}^P$) as well as their labels ($\mathbf{Y}^R$).

4. Energy Minimization

As noted earlier, the main difficulty in energy minimization arises due to the fact that there are many possible labelings $\mathbf{Y}^P$ that group pixels into regions. Specifically, for a given $H \times W$ image there can be as many as HW regions (that is, each pixel is a region). Hence the total number of possible labelings $\mathbf{Y}^P$ is $(HW)^{(HW)}$. Furthermore, we also need to ensure that the inferred labeling $\mathbf{Y}^P$ provides connected regions, which is well-known to be a difficult constraint to impose [24, 36].

In order to overcome these problems, we make use of bottom-up over-segmentation approaches. Specifically, we minimize the energy using the following two steps: (i) construct a large dictionary of connected putative regions using multiple over-segmentations; and (ii) select the set of regions that minimize the energy (that is, infer $\mathbf{Y}^P$ and $\mathbf{Y}^R$). We begin by describing our algorithm for selecting regions from a dictionary. We then provide details of the dictionary that we found to be effective in our experiments.

4.1. Region Selection as Optimization

Given a dictionary of regions, we wish to select a subset of regions such that: (i) the entire image is explained by the selected regions; (ii) no two selected regions overlap with each other; and (iii) the energy $E(\mathbf{Y};\mathbf{X})$ is minimized. Note that the dictionary itself may contain overlapping regions of any shape or size. We do not place any restrictions on it other than the assumption that it contains at least one set of disjoint regions that explains the entire image. We formulate the above task as an integer program and provide an accurate linear programming (LP) relaxation for it.

Integer Programming Formulation. Before describing the integer program, we need to set up some notation. We denote the dictionary of regions by $\mathcal{D}$. The intersection of all the regions in $\mathcal{D}$ defines a set of super-pixels $\mathcal{S}$. The set of all regions that contain a super-pixel $s \in \mathcal{S}$ is denoted by $\mathcal{C}(s) \subseteq \mathcal{D}$. Finally, the set of neighboring regions is denoted by $\mathcal{E}$, where two regions r and r' are considered neighbors of each other (that is, $(r, r') \in \mathcal{E}$) if they do not overlap and share at least one boundary pixel.

To formulate our problem as an integer program we define binary variables $y_r(i)$ for each region $r \in \mathcal{D}$ and $i \in \mathcal{L}^I = \mathcal{L}^R \cup \{0\}$. These variables indicate whether a particular region is selected and if so, which label it takes. Specifically, if $y_r(0) = 1$ then the region r is not selected, else if $y_r(i) = 1$ where $i \in \mathcal{L}^R$ then the region r is assigned the label i. Similarly, we define binary variables $y_{rr'}(i,j)$ for all neighboring regions $(r, r') \in \mathcal{E}$ such that $y_{rr'}(i,j) = y_r(i)y_{r'}(j)$. Furthermore, we define unary and pairwise potentials corresponding to the augmented label set $\mathcal{L}^I$ as

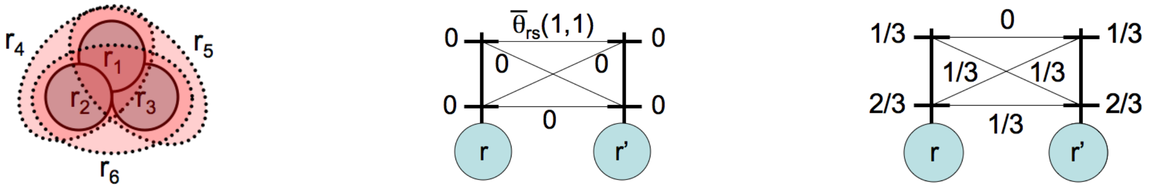
$$\begin{aligned} \overline{\theta}_r(i) &= \begin{cases} \theta_r(i) & \text{if } i \in \mathcal{L}^R, \\ 0 & \text{otherwise,} \end{cases} \\ \overline{\theta}_{rr'}(i,j) &= \begin{cases} \theta_{rr'}(i,j) & \text{if } i,j \in \mathcal{L}^R, \\ 0 & \text{otherwise.} \end{cases} \quad (2) \end{aligned}$$

The problem of obtaining the desired subset of regions can then be formulated as the following integer program:

$$\begin{aligned} \min_{\mathbf{y}} \quad & \sum_{r \in \mathcal{D}, i \in \mathcal{L}^I} \overline{\theta}_r(i) y_r(i) + \sum_{(r,r') \in \mathcal{E}, i,j \in \mathcal{L}^I} \overline{\theta}_{rr'}(i,j) y_{rr'}(i,j) \\ \text{s.t.} \quad & y_r(i), y_{rr'}(i,j) \in \{0,1\}, \forall r, r' \in \mathcal{D}, i,j \in \mathcal{L}^I, \\ & \sum_{i \in \mathcal{L}^I} y_r(i) = 1, \forall r \in \mathcal{D}, \\ & \sum_{j \in \mathcal{L}^I} y_{rr'}(i,j) = y_r(i), \forall (r,r') \in \mathcal{E}, i \in \mathcal{L}^I, \\ & \sum_{i \in \mathcal{L}^I} y_{rr'}(i,j) = y_{r'}(j), \forall (r,r') \in \mathcal{E}, j \in \mathcal{L}^I, \\ & \sum_{r \in \mathcal{C}(s)} \sum_{i \in \mathcal{L}^R} y_r(i) = 1, \forall s \in \mathcal{S}. \quad (3) \end{aligned}$$

The first set of constraints ensure that the variables $\mathbf{y}$ are binary. The second constraint implies that each region r should be assigned one label from the set $\mathcal{L}^I$. The third constraint enforces $y_{rr'}(i,j) = y_r(i)y_{r'}(j)$. The final constraint, which we call covering constraint, restricts each super-pixel to be covered by exactly one selected region.

Linear Programming Relaxation. Problem (3) is NP-hard since $\mathbf{y}$ is constrained to be binary (which specifies a non-convex feasible region). However, we can obtain an approximate solution to the above problem by relaxing the constraint on $\mathbf{y}$ such that $y_r(i)$ and $y_{rr'}(i,j)$ take (possibly fractional) values in the interval $[0, 1]$. The resulting LP relaxation is similar to the standard relaxation for energy minimization in pairwise random fields [6, 37], with the exception of the additional covering constraints. However, this relaxation is very weak when the pairwise potentials are not *submodular* (roughly speaking, when they encourage neighboring regions to take different labels) [11]. For example, consider the case where each region is either selected or not



(a) (b) (c)

Figure 2. **(a)** *Neighboring regions r_1, r_2 and r_3, shown using solid lines, consist of a single super-pixel. The regions shown using dashed lines are formed by two super-pixels. Specifically, $r_4 = r_1 \cup r_2$, $r_5 = r_1 \cup r_3$ and $r_6 = r_2 \cup r_3$.* **(b)** *The potentials corresponding to the clique of size 6 formed by the regions. The branches (horizontal lines) along the trellis (the vertical line on top of a region) represent the different labels that each region may take. We consider a two label case here. The unary potential $\overline{\theta}_r(i)$ is shown next to the i^{th} branch of the trellis on top of region r. The pairwise potential $\overline{\theta}_{rr'}(i,j)$ is shown next to the connection between the i^{th} branch of r and the j^{th} branch of r'. The only non-zero potential $\overline{\theta}_{rr'}(1,1) > 0$ corresponds to selecting both the regions r and r'. The optimal labeling of the clique must have an energy greater than 0 since at least two neighboring regions must be selected.* **(c)** *The optimal solution of the* LP *relaxation. The value of $y_r(i)$ is shown next to the i^{th} branch of r and the value of $y_{rr'}(i,j)$ is shown next to the connection between the i^{th} and j^{th} branches of r and r' respectively. Note that the solution satisfies all cycles inequalities, that is, $\sum_{(r,r')\in\mathcal{E}_C} y_{rr'}(0,0) + y_{rr'}(1,1) \geq 1$, where $\mathcal{E}_C$ is a cycle. Hence the solution lies within the feasible region of the relaxation. However, it can be easily verified that its objective function value is 0, thereby proving that the relaxation is not tight.*

(that is, $|\mathcal{L}^I| = 2$). For two neighboring regions r and r', the pairwise potential $\overline{\theta}_{rr'}(\cdot,\cdot)$ is 0 if one or both regions are not selected and $\overline{\theta}_{rr'}(1,1)$ otherwise (as defined by equation (2)). If $\overline{\theta}_{rr'}(1,1) > 0$ then the neighboring regions are encouraged to take different labels; that is, the pairwise potentials are non-submodular. This results in *frustrated cycles* for which the standard LP relaxation provides a weak approximation [29] (we tested this empirically for our problem, but do not include the results due to space limitations).

There are two common ways to handle non-submodular problems in the literature: (i) applying the roof duality relaxation [4, 29]; and (ii) using message passing algorithms [16, 20, 33] based on cycle inequalities [2]. Unfortunately, both these methods are not directly applicable in our case. Specifically, roof duality does not allow us to incorporate the covering constraints. Adding cycle inequalities still results in a weak approximation. For example, see Fig. 2 that shows a clique formed by three neighboring regions (r_1, r_2 and r_3) along with all the regions that overlap with at least one of them. Consider the case where $|\mathcal{L}^I| = 2$ and $\overline{\theta}_{rr'}(1,1) > 0$, thereby resulting in non-submodular pairwise potentials (shown in Fig. 2(b)). Since each super-pixel needs to be covered by exactly one selected region, it follows that the energy of the optimal assignment for this clique will be strictly greater than 0 (as at least two neighboring regions have to be selected). However, the LP relaxation, with all the cycle inequalities added in, still provides a fractional solution whose objective function value is 0 (shown in Fig. 2(c)).

Our example shows that cycle inequalities are not sufficient to make the relaxation tight. Instead, we require all the constraints that define the marginal polytope [37] (convex hull of the valid integral labelings) of the entire clique. We note here that the recent work of Sontag *et al.* [33] also advocates the use of such constraints. However, in their experiments they found cliques of size three (which are also a cycle of size three) to be sufficient. In contrast, we use cliques that are formed by three neighboring regions along with all the regions that overlap with at least one of the three regions. To the best of our knowledge, ours is one of the first examples in vision where constraints on cliques are essential for obtaining a good solution. Although a large number of constraints are required to specify the marginal polytope of a clique (their exact form is not important for this work), we show how the overall relaxation can be solved efficiently.

4.2. Solving the Relaxation

We use the dual decomposition framework that is well-known in the optimization community [3] and has recently been introduced in computer vision [18]. We begin by describing the general framework and then specify how it can be modified to efficiently solve our relaxation.

Dual Decomposition. Consider the following convex optimization problem: $\min_{\mathbf{z}\in\mathcal{F}} \sum_{k=1}^{M} g_k(\mathbf{z})$, where $\mathcal{F}$ represents the convex feasible region of the problem. The above problem is equivalent to the following: $\min_{\mathbf{z}_k\in\mathcal{F},\mathbf{z}} \sum_k g_k(\mathbf{z}_k)$, s.t. $\mathbf{z}_k = \mathbf{z}$. Introducing the additional variables $\mathbf{z}_k$ allows us to obtain the dual problem as

$$\max_{\boldsymbol{\lambda}_k} \min_{\mathbf{z}_k\in\mathcal{F},\mathbf{z}} \sum_k g_k(\mathbf{z}_k) + \sum_k \boldsymbol{\lambda}_k(\mathbf{z}_k - \mathbf{z}), \tag{4}$$

where $\boldsymbol{\lambda}_k$ are the Lagrange multipliers. Differentiating the dual function with respect to $\mathbf{z}$ we obtain the constraint that $\sum_k \boldsymbol{\lambda}_k = \mathbf{0}$, which implies that we can discard $\mathbf{z}$ from the above problem. The simplified form of the dual suggests the following strategy for solving it. We start by initializing $\boldsymbol{\lambda}_k$ such that $\sum_k \boldsymbol{\lambda}_k = \mathbf{0}$. Keeping the values of $\boldsymbol{\lambda}_k$ fixed, we solve the following slave problems: $\min_{\mathbf{z}_k\in\mathcal{F}} (g_k(\mathbf{z}_k) + \boldsymbol{\lambda}_k\mathbf{z}_k)$. Upon obtaining the optimal solutions $\mathbf{z}_k^*$ of the slave problems, we update the values of $\boldsymbol{\lambda}_k$ by projected subgradient descent where the subgradient with respect to $\boldsymbol{\lambda}_k$ is $\mathbf{z}_k^*$. In other words, we update $\boldsymbol{\lambda}_k \leftarrow \boldsymbol{\lambda}_k + \eta_t\mathbf{z}_k^*$ where η_t is the learning rate at iteration t. In order to satisfy the constraint $\sum_k \boldsymbol{\lambda}_k = \mathbf{0}$ we project the value of $\boldsymbol{\lambda}_k$ to $\boldsymbol{\lambda}_k \leftarrow \boldsymbol{\lambda}_k - (\sum_k \boldsymbol{\lambda}_k)/M$. Under fairly general conditions, this iterative strategy known as dual decomposition converges to the globally optimal solution of the original problem. We refer the reader to [3] for details.

Dual Decomposition for Selecting Regions. When using dual decomposition, it is crucial to select slave problems whose optimal solutions can be computed quickly. With this in mind, we choose three types of slave problems. Below we describe each of these slave problems and justify their use by providing an efficient method to optimize them.

The first type of slave problems is similar to the one used for energy minimization in pairwise random fields. Specifically, each slave problem is defined using a subset of regions $\mathcal{D}_T \subseteq \mathcal{D}$ and edges $\mathcal{E}_T \subseteq \mathcal{E}$ that form a tree. For each such graph $(\mathcal{D}_T, \mathcal{E}_T)$ we define the following problem:

$$\begin{aligned}\min_{\mathbf{y}\geq 0} \quad & \sum_{r\in\mathcal{D}_T,i} \left(\frac{\overline{\theta}_r(i)}{n_r} + \lambda^1_r(i)\right) y_r(i) + \\ & \sum_{(r,r')\in\mathcal{E}_T,i,j} \left(\frac{\overline{\theta}_{rr'}(i,j)}{n_{rr'}} + \lambda^1_{rr'}(i,j)\right) y_{rr'}(i,j), \\ \text{s.t.} \quad & \mathbf{y} \geq 0, \sum_{i\in\mathcal{L}^I} y_r(i) = 1, \\ & \sum_{j\in\mathcal{L}^I} y_{rr'}(i,j) = y_r(i), \sum_{i\in\mathcal{L}^I} y_{rr'}(i,j) = y_{r'}(j),\end{aligned} \tag{5}$$

where n_r and $n_{rr'}$ are the total number of slave problems involving $r \in \mathcal{D}$ and $(r,r') \in \mathcal{E}$ respectively. As the LP relaxation is tight for trees (that is, it has integral solutions) [6, 37], the above problem can be solved efficiently using belief propagation [25].

The second type of slave problems correspond to the covering constraints. Specifically, for each $s \in \mathcal{S}$ we define:

$$\min_{\mathbf{y}\geq 0} \quad \sum_{r\in\mathcal{C}(s),i} \left(\frac{\overline{\theta}_r(i)}{n_r} + \lambda^2_r(i)\right) y_r(i) \tag{6}$$

$$\text{s.t.} \quad \sum_{r\in\mathcal{C}(s)} \sum_{i\in\mathcal{L}^R} y_r(i) = 1, \sum_{i\in\mathcal{L}^I} y_r(i) = 1. \tag{7}$$

It can be verified that the constraint matrix for the above problem is totally unimodular. In other words, the optimal solution is integral and in fact, can be found efficiently in $O(L|\mathcal{C}(s)|)$ time (where L is the number of region labels and $|\mathcal{C}(s)|$ is the number of regions in $\mathcal{D}$ that cover s).

The third type of slave problems correspond to the clique constraints defined in §4.1. Specifically, for every clique defined by $\mathcal{D}_Q \subseteq \mathcal{D}$ and $\mathcal{E}_Q \subseteq \mathcal{E}$, we specify:

$$\begin{aligned}\min_{\mathbf{y}\in\mathcal{M}_Q} \quad & \sum_{r\in\mathcal{D}_Q,i} \left(\frac{\overline{\theta}_r(i)}{n_r} + \lambda^3_r(i)\right) y_r(i) + \\ & \sum_{(r,r')\in\mathcal{E}_Q,i,j} \left(\frac{\overline{\theta}_{rr'}(i,j)}{n_{rr'}} + \lambda^3_{rr'}(i,j)\right) y_{rr'}(i,j),\end{aligned} \tag{8}$$

where $\mathcal{M}_Q$ is the marginal polytope of the clique. Note that since $\mathcal{M}_Q$ is the convex hull of all valid integral labelings and the objective function is linear, it follows that the above problem has an integral optimal solution. Furthermore, it corresponds to the minimization of a sparse higher order function [17, 28]. In other words, although there are $O((L+1)^{|\mathcal{D}_Q|})$ possible labelings for the clique, only a small fraction of them are valid. For example, consider the clique shown in Fig. 2. If region r_4 is selected, then regions r_1 and r_2 cannot be selected since they overlap with r_4. Furthermore, we also know that exactly one of r_1 and r_4 must be selected. Using similar observations, the number of valid labelings of a general clique used in our relaxation (that is, a clique formed by three neighboring regions along with all the regions that overlap with at least one of them) can be shown to be $O((L|\mathcal{C}(s_Q)|)^3)$. Here, s_Q is the super-pixel s that is covered by at least one region in $\mathcal{D}_Q$ and has the largest corresponding set $\mathcal{C}(s)$. Note that we can also use cliques defined by m neighboring regions that can be optimized in $O((L|\mathcal{C}(s_Q)|)^m)$ time (we omit details on the derivation of the time complexity due to lack of space). However, we found $m = 3$ to be sufficient to obtain a tight relaxation.

We iteratively solve the above slave problems and update $\boldsymbol{\lambda}$. Upon convergence, we obtain the primal solution (that is, the subset of regions and their labels) in a similar manner to [16, 17, 18]. Briefly, this involves sequentially considering each super-pixel (in some arbitrary order) and picking the best region for it (according to the values of the dual variable $\boldsymbol{\lambda}$). The primal-dual gap provides us with an estimate of how good our solution is. Typically, we obtain a very small gap by using our approach. However, this accuracy comes at the cost of adding the (relatively) expensive clique constraints, which can make our approach intractable for very large dictionaries. Below we describe an effective strategy for dealing with this case.

Maintaining an Active Set. Although the dictionary may consist of a large number of regions, only a small fraction of them will be selected (since overlapping regions are not allowed). This is analogous to the problem of training a support vector machine, where only a few input vectors define the separating hyperplane. To take advantage of this sparsity, we use a *shrinking* strategy that was shown to be highly effective in the well-known SVM^{light} algorithm [14]. Specifically, we maintain an active set of regions $\mathcal{D}_A \subseteq \mathcal{D}$ and solve our relaxation for this (potentially much smaller) subset of the dictionary. The active set consists of those regions that were selected while optimizing at least one slave problem during the last T iterations. In other words, $r \in \mathcal{D}_A$ if at least one of the slave problems had an optimal integral labeling with $y_r(0) \neq 1$ in at least one of the last T iterations. Intuitively, if a region does not belong to the active set, then it is most probably not part of the best subset of regions.

Once the relaxation has been solved for the active set, we check whether the solution obtained is the optimal solution for the entire dictionary. If the primal-dual gap for the problem corresponding to the active set is 0, then this check simply involves running one iteration of the dual decomposition algorithm for the entire dictionary. If the value of the dual does not increase, then we are assured that we have found the best subset of regions (as this implies that the dual variables $\boldsymbol{\lambda}$ will not change in subsequent iterations). In practice, we run the dual decomposition algorithm for a fixed, small number of iterations for the entire dictionary and stop if there is no increase in the dual. Otherwise, we make all the regions active and re-run the algorithm. In our experiments, using active sets in this manner cuts down the run-time of our algorithm by more than a factor of 2.

Another advantage of maintaining an active set is that it provides us with an efficient heuristic for iteratively adding clique constraints. Specifically, we start with the basic LP relaxation (without any clique constraints). If we are unable to increase the value of the dual for T' iterations, then we add the slave problem corresponding to the clique with the maximum fraction of active regions. The intuition behind this is that a large fraction of active regions indicates that the algorithm is not able to decide which regions to pick. Adding that clique to the dual decomposition will make this decision easier by providing more support for the regions that minimize the clique slave problem. Note that other heuristics have been proposed in the literature for iteratively adding more constraints, most notably the work of Sontag *et al.* [33]. However, their approach is extremely expensive as it involves adding each clique separately and selecting the one that results in the maximum increase in the dual at that iteration. In practice, our strategy (that is much more efficient) worked as well as that of [33] for our problem and cut down the run-time by an order of magnitude compared to naive dual decomposition where all clique slaves are added at the beginning (using $T = 30$ and $T' = 5$).

4.3. Generating the Dictionaries

Ideally, we would like to form a dictionary that consists of all the regions obtained by merging and intersecting the segments provided by a bottom-up approach. However, this will clearly result in a very large dictionary that cannot be handled even using our efficient algorithm. Instead, we iteratively search over the regions using the following strategy. We initialize our dictionary with the regions obtained from one (very coarse) over-segmentation. In the subsequent iterations, we consider two different types of dictionaries (similar to [8]). The first dictionary consists of the current set of regions $\mathcal{D}_{cur}$ (those that have provided the best explanation of the image until now, in terms of the energy) as well as all the regions obtained by merging two neighboring regions in $\mathcal{D}_{cur}$. The second type of dictionary uses multiple over-segmentations to define the regions. Specifically, in addition to $\mathcal{D}_{cur}$, it also consists of all the regions obtained by merging every segment from the over-segmentations with all its overlapping and neighboring regions in $\mathcal{D}_{cur}$. Similarly, it also contains all the regions obtained by intersecting every segment with all its overlapping regions in $\mathcal{D}_{cur}$. While using the first dictionary results in larger regions (by merging neighboring regions together), the second dictionary allows us to correct any mistakes (merging two incompatible regions) by considering intersections of segments and regions.

It is worth noting that, unlike most of the previous works [10, 13, 15, 23, 31], our dictionaries define regions that are not just segments obtained by a bottom-up approach. This additional degree of freedom is important for obtaining regions that provide discriminative features while respecting scene boundaries. The dictionaries defined above may still be too large to be handled by our algorithm, especially if we wish to use coarse over-segmentations. We address this problem using the powerful move-making framework [5].

Move-making Strategy. If the dictionary under consideration is too large, then we divide it into smaller sub-dictionaries and obtain a strong local minimum solution over them. In more detail, for a given dictionary $\mathcal{D}$, consider a set of sub-dictionaries $\mathcal{D}_k$ such that $\bigcup_k \mathcal{D}_k = \mathcal{D}$. Given the current regions $\mathcal{D}_{cur}$ and a sub-dictionary $\mathcal{D}_k$, we define an auxiliary sub-dictionary $\mathcal{D}'_k$. If, for a segment a and a region $r \in \mathcal{D}_{cur}$, $r \cup a \in \mathcal{D}_k$ then for all $r' \in \mathcal{D}_{cur}$ that are neighbors of r, $r' \backslash a \in \mathcal{D}'_k$ (where $r' \backslash a$ is the set of pixels in r' but not in a). The auxiliary sub-dictionary $\mathcal{D}'_k$ ensures that if $r \cup a \in \mathcal{D}_k$ is selected then the regions required to complete the explanation of the image (that is, $r' \backslash a$) are present in $\mathcal{D}'_k$. To obtain a local minimum over a given set of sub-dictionaries, we iterate over the following steps until we can no longer decrease the value of the energy:

- Choose a sub-dictionary $\mathcal{D}_k$.
- Construct a new dictionary $\mathcal{D}_{new} = \mathcal{D}_{cur} \cup \mathcal{D}_k \cup \mathcal{D}'_k$.
- Update $\mathcal{D}_{cur}$ by selecting the best regions from $\mathcal{D}_{new}$.

The last step involves running our efficient dual decomposition algorithm on the dictionary $\mathcal{D}_{new}$.

Obtaining the Sub-dictionaries. The method of [8] can be thought of as a special case of the above move-making strategy where every sub-dictionary is of size 1. Such small moves often result in the algorithm getting stuck in a bad local minima. In contrast, our LP relaxation based approach allows us to use large sub-dictionaries at each iteration. We use the following simple strategy to divide a given dictionary into sub-dictionaries. Starting with sub-dictionaries of size 1, we merge the two most compatible sub-dictionaries together, where compatibility is measured by the average energy of the neighboring regions in the two sub-dictionaries. We only merge two sub-dictionaries together if the value of $|\mathcal{C}(s)|$ for the resulting sub-dictionary (that determines the efficiency of each dual decomposition iteration) does not exceed a certain threshold K (we use $K = 5$) for any $s \in \mathcal{S}$. A reader familiar with the energy minimization literature would notice that the method of [8] is analogous to iterated conditional modes while our approach is analogous to the more accurate α-expansion move [5].

5. Experiments

As an example application of our framework, we consider the task of scene segmentation. However, we would like to emphasize that our method for selecting regions based on the energy function of the high-level vision task is fairly generic and is applicable to several problems. In order to facilitate future research on the use of regions, we have made our efficient dual decomposition code available online (linked from the first author's homepage). Below we describe our experimental setup in detail.

Model. We use a similar model to the ones present in [8, 9]. Specifically, we compute the unary potential $\theta_r(i)$ of a region r being assigned to a semantic class i using multiclass logistic regression over a set of features extracted from the pixels belonging to the region. These features not only include color, texture and shape information, but also the percentage of pixels that lie above and below the horizon. The use of horizon helps capture important contextual cues like grass and road lie below sky. Similar to [8], we update the position of the horizon after each iteration by fixing the value of all other variables in the model (that is, the regions and their labels). The pairwise potentials $\theta_{rr'}(i,j)$ are of the form described in [9], that is $\theta_{rr'}(i,j) \propto \exp(-\Delta(r,r'))$, where $\Delta(r,r')$ measures the contrast between regions r and r'. Note that the pairwise potentials are independent of the labels. This has two advantages: (i) it encourages larger regions so that the boundary length is minimized; and (ii) it effectively implies that in our LP relaxation the label set $\mathcal{L}^I$ is of cardinality 2. This follows from the fact that if a region r is selected, then the energy function is minimized by assigning it a label $\mathbf{Y}_r^R = \arg\min_{i \in \mathcal{L}^R} \theta_r(i)$. We refer the reader to [8, 9] for more details.

Dataset. We use the publicly available Stanford background dataset [8]. It consists of 715 images (collected from standard datasets such as PASCAL, MSRC and geometric context) whose pixels have been labeled as belonging to one of seven background classes or a generic foreground class. For each image we use three over-segmentations obtained by employing different kernels for the standard mean-shift algorithm [7]. Similar to [8], we split the dataset into 572 images for training and 143 images for testing. We report results on four different splits.

Results. In order to evaluate the effectiveness of our approach for selecting a good set of regions, we compare it with other types of regions commonly used in computer vision. Specifically, we compare our approach to the following: (i) regions obtained by the intersection of the three over-segmentations of an image; (ii) regions defined by the segments of the best single over-segmentation (in terms of the energy of the model); and (iii) regions selected from a dictionary similar to ours by the method of [8] (using the code provided by the authors[1]).

For each type of regions, we evaluate the accuracy of the labeling obtained by measuring the percentage of pixels whose labels matched the ground-truth. Here, the label of a pixel is the label assigned to the region to which it belongs. Table 1 shows the average energy and accuracy for the different approaches. Note that all region-based methods outperform the pixel-based approach described in [8] (that provides comparable results to [32]). However, the choice of the regions greatly affects the value of the energy and hence the accuracy of the segmentation. By using large dictionaries and an accurate LP relaxation, our approach provides a statistically significant improvement (using paired t-test with $p = 0.05$) over other methods, both in terms of energy and accuracy. Our algorithm takes less than 10 minutes per image on average on a 2.4 GHz processor.

	Energy	Accuracy
Pixel	-	76.65 ± 1.20
Intersection	16150 ± 2005	76.84 ± 1.34
Segmentation	6796 ± 833	77.85 ± 1.50
[8]	4815 ± 592	78.52 ± 1.40
Our Method	**1630** ± **306**	**79.42** ± **1.41**

Table 1. *Results for the scene segmentation experiment. The mean and standard deviation of the energy and the pixel-wise accuracy over four folds is shown. The first row corresponds to a pixel-based model [8]. The second row uses regions obtained by intersecting the three over-segmentations. The third row shows the results obtained by using the best single over-segmentation (in terms of the energy function). The fourth row corresponds to the method described in [8]. The fifth row shows our method's results. Using multiple over-segmentations to define an accurate dictionary and selecting the regions by our* LP *relaxation based method results in a lower energy labeling that provides better accuracy.*

Fig. 3 shows some example segmentations obtained by the various approaches. Note that the regions corresponding to the intersection of over-segmentations respect the boundaries of the scene entities. However, they are too small to provide reliable features. Even using the best single over-segmentation results in regions that are not large enough. The method of [8] overcomes this problem to some extent by using an accurate dictionary of regions. However, as it is prone to getting stuck in a bad local minima, it sometimes selects regions that result in a high energy labeling. Our approach addresses this deficiency by allowing us to make large moves at each iteration.

6. Discussion

Although several works have shown that regions can greatly help in high-level vision, there is so far no consensus on the definition of regions. While some researchers use segments of a single over-segmentations as regions, others use the intersection of multiple over-segmentations or even the overlapping segments themselves. With this work we propose an intuitive definition of regions, namely regions are sets of connected pixels that minimize the energy for the task under consideration. Furthermore, regions of a given image do not overlap with each other, thereby providing a coherent explanation of the image. Such regions are concisely captured in our model, which is fairly generic and can be applied to several vision problems. To aid the use of our model,

[1] We thank Stephen Gould for helping us understand his code and for many useful discussions about our approach.

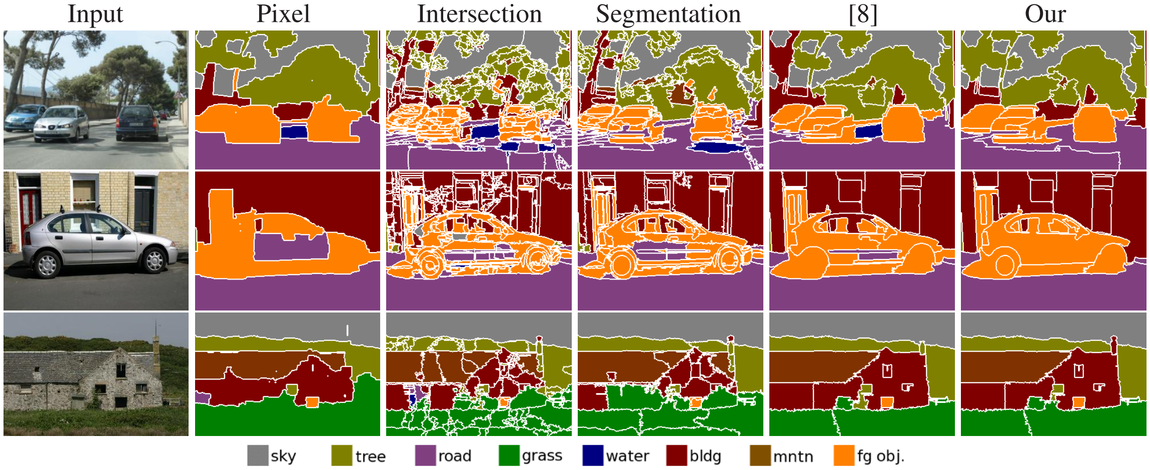


Figure 3. *Examples of scene segmentation obtained using different types of regions. Our method provides large regions that align with the scene boundaries.*

we proposed an energy minimization algorithm that avoids the problem of getting stuck in a bad local minima. This results in a significant decrease in the energy compared to the current methods, which in turn improves the accuracy. We believe that our algorithm will also benefit parameter learning methods that heavily rely on energy minimization (for example, the max-margin training regime [34, 35]). Better parameters may lead to further improvements in accuracy. Finally, we hope that our framework would not only assist existing region-based methods on various aspects of scene understanding (such as geometric reconstruction [13], object discovery [30] and object detection/segmentation [10, 23]) but also provide a unified formulation for them.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *CVPR*, 2009.

[2] F. Barahona and A. Mahjoub. On the cut polytope. *Mathematical Programming*, 1986.

[3] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[4] E. Boros and P. Hammer. Pseudo-Boolean optimization. *Discrete Applied Mathematics*, 2002.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 2001.

[6] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. A linear programming formulation and approximation algorithms for the metric labelling problem. *SIAM Journal on Disc. Math.*, 2005.

[7] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *ICCV*, 1997.

[8] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009.

[9] S. Gould, T. Gao, and D. Koller. Region-based segmentation and object detection. In *NIPS*, 2009.

[10] C. Gu, J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. In *CVPR*, 2009.

[11] P. Hammer. Some network flow problems solved with pseudo-Boolean programming. *Operations Research*, 1965.

[12] X. He, R. Zemel, and M. Carriera-Perpinan. Multiscale conditional random fields for image labeling. In *CVPR*, 2004.

[13] D. Hoiem, A. Efros, and M. Herbert. Geometric context from a single image. In *ICCV*, 2005.

[14] T. Joachims. Making large-scale SVM learning practical. In *Advances in Kernel Methods*. MIT Press, 1999.

[15] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.

[16] N. Komodakis and N. Paragios. Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles. In *ECCV*, 2008.

[17] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optmization for higher-order MRFs. In *CVPR*, 2009.

[18] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*, 2007.

[19] S. Konishi and A. Yuille. Statistical cues for domain specific image segmentation with performance analysis. In *CVPR*, 2000.

[20] M. P. Kumar and P. Torr. Efficiently solving convex relaxations for MAP estimation. In *ICML*, 2008.

[21] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Associative hierarchical CRFs for object class image segmentation. In *ICCV*, 2009.

[22] D. Larlus and F. Jurie. Combining appearance models and Markov random fields for category level object segmentations. In *CVPR*, 2008.

[23] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *CVPR*, 2009.

[24] S. Nowozin and C. Lampert. Global connectivity potentials for random field models. In *CVPR*, 2009.

[25] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kauffman, 1998.

[26] C. Pontafaru, C. Schmid, and M. Hebert. Object recognition by integrating multiple segmentations. In *ECCV*, 2008.

[27] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007.

[28] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order functions of discrete variables. In *CVPR*, 2009.

[29] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary MRFs via extended roof duality. In *CVPR*, 2007.

[30] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006.

[31] A. Saxena, M. Sun, and A. Ng. Make3D: Learning 3D scene structure from a single still image. *PAMI*, 2008.

[32] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[33] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening LP relaxations for MAP using message passing. In *UAI*, 2008.

[34] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *NIPS*, 2003.

[35] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector learning for interdependent and structured output spaces. In *ICML*, 2004.

[36] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008.

[37] M. Wainwright, T. Jaakkola, and A. Willsky. MAP estimation via agreement on trees: Message passing and linear programming. *IEEE Transacations on Information Theory*, 2005.