Parsing World's Skylines using Shape-Constrained MRFs

Rashmi Tonge
CVIT, IIIT Hyderabad

Subhransu Maji
Toyota Technological Institute at Chicago

C. V. Jawahar
CVIT, IIIT Hyderabad

Abstract

We propose an approach for segmenting the individual buildings in typical skyline images. Our approach is based on a Markov Random Field (MRF) formulation that exploits the fact that such images contain overlapping objects of similar shapes exhibiting a "tiered" structure. Our contributions are the following: (1) A dataset of 120 high-resolution skyline images from twelve different cities with over 4,000 individually labeled buildings that allows us to quantitatively evaluate the performance of various segmentation methods, (2) An analysis of low-level features that are useful for segmentation of buildings, and (3) A shape-constrained MRF formulation that enforces shape priors over the regions. For simple shapes such as rectangles, our formulation is significantly faster to optimize than a standard MRF approach, while also being more accurate. We experimentally evaluate various MRF formulations and demonstrate the effectiveness of our approach in segmenting skyline images.

1. Introduction

We are interested in extracting the detailed structure of buildings within photographs of skylines as shown in Fig. 1. The skylines of cities such as Chicago, New York, Hong Kong and Tokyo, among others, are a subject of great interest among professional and amateur photographers alike, hence one can find an immense number of these pictures on the web. Some of these cities are known for their exceptionally tall buildings, others for their unique designs, and these photographs provide a gist of their architectural styles.

Automatic segmentation of individual buildings from images can be used in a number of applications for designers and artists such as renderings of these from novel viewpoints, information overlays, creation of virtual cities, and other applications such as 'geo-location' by matching individual buildings to a dataset of known buildings.

The proposed task is quite challenging for a number of reasons. Skylines typically contain many tightly packed buildings that partially occlude one another leading to complex occlusion patterns. Furthermore, different facades of the same building can appear quite different from one another due to sunlight. However, these images are highly structured – buildings are typically convex objects, roughly rectangular, and all the buildings stand on the ground plane. These constraints can be incorporated as priors for automatic segmentation algorithms.

Figure 1: Photos of skylines of Chicago and Miami and their labeling of individual buildings using our method.

Current semantic segmentation algorithms typically do not consider such detailed labels. For example, datasets such as PASCAL VOC [7], or MSRC [16] consider labeling of pixels into one of the dozens of labels. In geometric labeling [11], the goal is to roughly label pixels into a number of coarse level orientations such as frontal, left/right-facing, or semantic categories such as ground, sky or porous. In order to systematically study this problem, we introduce a dataset of 120 images from twelve cities of the world with buildings that are individually segmented. Each image typically contains between 30 – 40 buildings, and the dataset contains over 4,000 individual buildings, which serves as a test bed for our experiments (Sect. 3).

We study the problem in an automatic as well as interactive setting. In the interactive setting, we assume that we are provided with an image, some 'seed' pixels for each building, and the upper and lower boundaries delineating the region containing all the buildings (as seen in Fig. 2). In the 'automatic' setting we are only provided with the image and the upper and lower boundaries. On our dataset we found that automatic methods [11] for obtaining such regions work reasonably well, hence we focus on the task of segmenting the individual buildings. Our evaluation metrics and tasks are described in Sect. 3.1.

We also experimentally evaluate color and texture models for representing the appearance of buildings, and find that texture based Gaussian mixture models can provide sig-

nificant improvement over color models (Sect. 5.1). These serve as local evidence (or 'unary' potentials) in a Markov Random Field (MRF) formation of our problem. Several leading approaches for semantic segmentation are based on MRFs – a probabilistic model of pixel labels that incorporates local evidence and smoothness of nearby pixels labels. These approaches, though general purpose, do not easily allow the incorporation of higher-order priors such as the overall shape and size of the regions. To this end we propose a *shape-constrained MRF* that allows explicit control over the shape, and utilizes the fact the 'tired' structure exhibited by occluding buildings implies that only the upper boundary of an object is 'owned' by each object.

We propose several greedy approaches to optimize the proposed MRF formulation (Sect. 4). Similar to approaches like α-expansion [6], we pick one label at a time and update the pixels with respect to that label. However, unlike expansion moves where only background pixels can change to foreground, we allow refinement moves where foreground labels can change to *any* background as well. The 'tiered' structure of the labels allows us to infer the background label *underneath* each foreground pixel. Furthermore, one can order the buildings from front to back based on the y-coordinate of the 'seeds', which serves as a natural order in which we consider region refinement.

One such approach called *rectangle MRF* does this via an explicit search over all potential rectangles for each building. This search can be done quickly even on relatively high resolution images using 'integral images'. Another approach called *tiered MRF* does this via a dynamic programming, approximating the upper boundary of a building as a 1D monotonic curve, i.e., the x-coordinates along the curve are monotonic. The former approach allows us control the shape of each region but does a poor job at approximating its upper boundary. Hence we propose a hybrid approach called *refined MRF*, that starts with the solution of rectangle MRF and refines the upper boundary within the horizontal bounds of the rectangle using dynamic programming. This achieves the best results while being an order of magnitude faster than α-expansion using graph-cuts (Sect. 5.2).

The automatic setting is suitable for low-level image segmentation methods such as SLIC [1], graph-based segmentation [8], and gPb regions [2]. However, none of these methods explicitly consider shape priors. We show that starting from a set of regions automatically selected from any such segmentation method, one can improve the results using shape priors (Sect. 5.3).

2. Related work

There has been significant interest in the recent past to understand the natural outdoor by looking at the buildings, mountains and surroundings [3, 11]. Semantic understanding of the outdoor with additional geometric cues can help in 3D layouts and better visualization. Our work is related to this, except we aim to extract the fine-grained detailed structure of the regions within the image.

Figure 2: **The skyline-12 dataset.** Sample images from the dataset are shown in top 2 rows. Each image (middle) is annotated with individual buildings (bottom). In the interactive setting for segmentation the methods are also provided the top (red) and bottom (green) boundaries, as well as 'seeds' for each building shown as blue strokes.

Our work can be considered in the framework of semantic pixel labeling. Optimization for labeling pixels is a widely studied area of research. Most of the successful methods for semantic segmentation [12, 18] cast it as an energy minimization problem consisting of local and pairwise potentials in Markov Random Fields. Methods like [4, 15] popularized this framework for *binary* interactive segmentation of natural images in an energy minimization framework. Graph cut with α-expansion [6] has emerged as a popular approach to solve multi-label segmentation. The optimization reduces to a sequence of binary labeling problems each of which can be computed using graph-cuts. Although, extremely general, the process can be expensive for large images both in terms of computational complexity and memory. We introduce methods that are an order of magnitude faster and more accurate for labeling skyline images that exploits the spatial structure of the objects.

For tiered scenes, Felzenzwalb and Veksler [9] introduced a dynamic programming based solution to obtain a globally optimal solution. However the complexity scales exponentially with the number of labels, hence is impractical for our setting. Zheng *et al.* [19] propose a faster approximation to [9] by decomposing multi-label tiered la-

beling to a series of binary labeling problems exploiting the topological priors. Our approach takes a similar route, but we incorporate higher order priors such as the overall shape and aspect ratio of each region that cannot be easily expressed as topological priors. Another approach for incorporating topological priors such as inclusion or exclusion is [17], but is also computationally expensive. Freedman and Zhang [10] propose an approach for incorporating shape priors in a MRF formulation, but it assumes that the location of the shape is known making it unsuitable for our case. In our setting both topological and shape priors play a key role, and we show that the combination can improve results without sacrificing speed (Sect. 5.2).

Automatic segmentation methods exploit the local similarity in defining segments and boundaries [1, 2, 8]. While all these methods are quite accurate for generic segmentation, skylines prove to be much harder due to intra-region color and texture variations. We show that our automatic approach can be initialized from any of these unsupervised segmentation techniques and provides a significant boost over them by exploiting shape priors (Sect. 5.3).

3. The skyline-12 dataset

We introduce a new dataset **skyline-12** consisting 10 skyline images each of the following twelve cities — *Chicago, Dallas, Frankfurt, Hong Kong, Miami, New York, Philadelphia, Seattle, Shanghai, Singapore, Tokyo* and *Toronto*. The photographs taken during daytime with variety of dense and complex skylines. All the images are obtained from *Flickr* and are of an average resolution of 1500×2500 pixels, with largest image is of 4092×10476 pixels and smallest one is of 384×576 pixels.

All images in the dataset are manually annotated with the individual buildings at pixel level, as well as the upper and lower boundaries delineating the regions containing all the buildings. Moreover, to study the problem in the interactive setting we also provide 'seed' pixels for each building. Such seeds may be provided by the user in an interactive application, but in order to systematically evaluate various methods, we use the same seeds as input to various methods. Fig. 2 shows a sample image from our dataset with the annotations and seed pixels.

3.1. Tasks and evaluation

Interactive setting. In this setting the input is an image $\mathcal{I}$, the upper and lower boundaries delineating the region containing the buildings, as well as seed pixels $\{S_i\}$ for each building $b_i, i \in \{1, \ldots, N\}$. Output of the methods is a labeling of all the pixels in the building region into one of N labels or background. Performance is measured as the average overlap of the segmentations of each building b_i as explained below. Let $G_{\mathcal{I}}$ and $P_{\mathcal{I}}$ denote the ground-truth and the predicted labeling, and let $G^i_{\mathcal{I}}$ and $P^i_{\mathcal{I}}$ denote the set of pixels labelled as i in each. The overlap is computed as the intersection over union of these sets. The $\mathsf{AverageOverlap}(G_{\mathcal{I}}, P_{\mathcal{I}})$ is defined as:

$$\mathsf{AverageOverlap}(G_{\mathcal{I}}, P_{\mathcal{I}}) = \frac{1}{N} \sum_{i=1}^{N} \frac{G^i_{\mathcal{I}} \cap P^i_{\mathcal{I}}}{G^i_{\mathcal{I}} \cup P^i_{\mathcal{I}}}$$

We average this across all the images in the test set and report a single *Mean Average Overlap* (MAO) score for a method. This measure has been used in past for evaluation of segmentation in [2, 7, 13].

Automatic setting. In this setting we are only given an image $\mathcal{I}$ and the upper and lower boundaries as described earlier. The output of a segmentation algorithm is a labeling of each pixel in the image into M regions. We compute similar average overlap scores as before, but first compute a bipartite matching between the ground-truth regions and segmented regions. For all N ground truth regions, we compute the bipartite matching $m : N \to M$ of highest score where the score of matching is given by the intersection over union of the pixels. The average overlap in this setting is defined as:

$$\mathsf{AverageOverlap}(G_{\mathcal{I}}, P_{\mathcal{I}}) = \max_{m \in \mathcal{M}} \frac{1}{N} \sum_{i=1}^{N} \frac{G^i_{\mathcal{I}} \cap P^{m(i)}_{\mathcal{I}}}{G^i_{\mathcal{I}} \cup P^{m(i)}_{\mathcal{I}}}$$

Here unassigned ground truth regions get a score of zero. This measure is similar to the *Best Segment Score* (BSS) criteria used in [13] with the key difference that each segmented region can contribute to only one building. For a given automatic method, we report MAO scores after performing the matching of labels within each image.

4. Approach

We formulate the overall labeling as an energy minimization problem. For set of pixels P and set of possible labels L, the energy of a labeling $F : P \to L$, is defined as

$$E(F) = \sum_{p \in P} D_p(F_p) + \sum_{p,q \in N} V_{pq}(F_p, F_q) \qquad (1)$$

Where $V_{pq}(a, b) = \lambda \exp\left(-\gamma \left(I_p - I_q\right)^2\right) \cdot \mathbf{1}(a \neq b)$ and I_p denotes the image intensity at pixel p. The optimal labeling can be obtained by $F^* = \operatorname{argmin}_f E(f)$.

The unary term D_p measures the color and texture similarity of the pixel compared to the color and texture models estimated from a set of seed pixels (Sect. 5.1). In the interactive setting these seeds are provided as input, as described earlier. In the automatic setting, we initialize these seeds from unsupervised low-level segmentation algorithms.

A standard approach for solving multi-label MRF as described above is the α-expansion [6]. In each iteration a

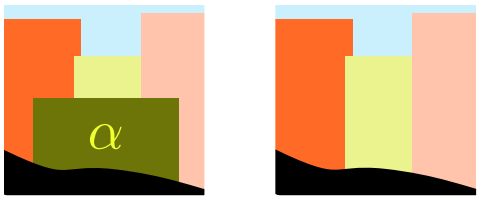


Figure 3: Given a label α (left) one can infer the background labels underneath α by copying the labels from the top to bottom because of the tiered structure (right).

label α is picked, and binary segmentation problem is formulated by replacing all the other labels to a single background label as follows:

$$E(F) = \sum_{p \in P} D'_p(F_p) + \sum_{p,q \in N} V_{pq}(F_p, F_q) \qquad (2)$$

where, $F : P \rightarrow \{0,1\}^P$, $D'_p(1) = D_p(\alpha)$ and $D'_p(0) = D_p(l_p^{bg})$ where l_p^{bg} is the current background label at pixel p. In typical labeling problems the background pixel label is unknown at pixels which are labelled α, hence only expansion moves are considered by setting the background costs of such pixels high. However due to the tiered nature of the labels we can induce the background labels for pixels labelled α by copying the background labels from the top to bottom as illustrated in Fig. 3. This allows us to simultaneously expand or contract the regions with label α. This is important as it allows us to only adjust the upper boundary of each building at a time leading to faster algorithms.

The optimal solution to the binary problem can be obtained using graph cuts. Although this is an effective and general purpose approach, running graph cuts can be quite expensive on large images such as ours, requiring several minutes to find the optimal labeling. Our key idea is to replace the search over binary segmentations by a search over a parametric shape family. For buildings we can explicitly search over the space of feasible rectangles much faster than possible segmentations. Furthermore, the 'tiered' structure of the buildings provides a natural ordering of the buildings according to their depth order. In practice we order the buildings according to the lowest seed pixel, i.e., the building with the lowest seed is considered first.

Our algorithm is as follows – we initialize the 'frontier' f to the the lower boundary l of the building region. At each iteration we pick the next building α in the ordered list. We formulate a binary segmentation problem using Eqn. 2 and estimate its upper boundary Ω_α. Then, we update the frontier by taking the column-wise maximum of the frontier, the upper boundary Ω_α. The corresponding labeling F is updated as well. This process is repeated a few times over all buildings. The algorithm is shown in Algo. 1 and few iterations of the process are shown in Fig. 4. Below we describe two efficient ways of searching over the upper boundary.

Algorithm 1 Greedy skyline segmentation

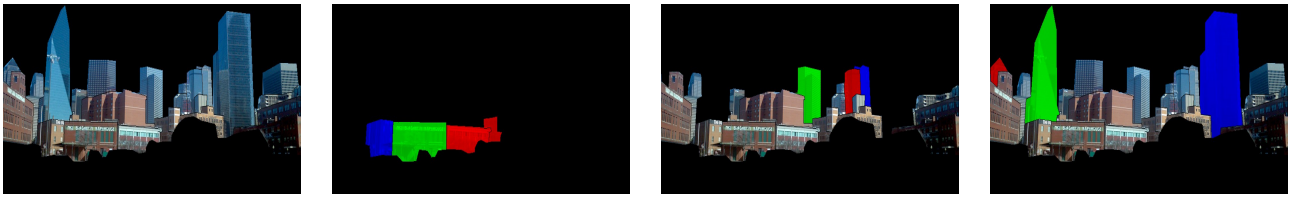
Require: data D, pairwise V, boundary (l, u)
1: Initialize, initial labeling F from unary labels
2: **for** iter := 1 **to** K **do**
3: Initialize, frontier $f \leftarrow l$
4: **for** α := 1 **to** N **do**
5: $\Omega_\alpha \leftarrow$ upperBoundary(α, F, D, V, f, u)
6: $f \leftarrow \max(f, \Omega_\alpha)$
7: F $\leftarrow$ updateLabels(F, Ω_α)
8: **end for**
9: **end for**

Rectangle MRF. In this formulation we constrain the upper boundary of the building to be exactly rectangular, i.e., for each building we only need to estimate the three values (L, T, R), the left, top and right of the building within the feasible set, i.e., within the current upper and lower boundaries and enclosing the seed pixels of the building. Moreover, we can also constrain the aspect ratios to a desired range, as well as enforce width and height constraints learned on the training data. For a given value of (L, T, R) the energy can be computed in $O(1)$ time using integral images of the unary and pairwise terms. For a region of size $m \times n$, i.e., m rows and n columns, there are $O(mn^2)$ rectangles to consider, hence the complexity of each iteration of *rectangle MRF* is $O(mn^2)$. Compare this to the worst case complexity of graph-cut which is $O(m^3n^3)$.

Tiered MRF. Constraining the upper boundaries as rectangles can be a poor approximation to many buildings. Here we refine the shape of the upper boundary. However, instead of a general 2D curve we restrict the upper boundary Ω to be 'x-monotonic', i.e., it intersects each column exactly once. This is a good approximation to buildings seen in typical skylines that are convex. The key advantage of the x-monotonic structure is that the optimal solution can be found using a simple extension of the dynamic programming algorithm proposed in [9, 19]. At each column j we maintain the optimal cost of a path ending in each row i. Let, l_j, u_j denote the lower and upper bounds at column j. Setting, $C_{i,-1} = 0, \forall i$ and $l_{-1} = u_{-1} = u_1$, we have the following recurrence relation for $C_{i,j}$ for $i \in [l_j, u_j]$:

$$C_{i,j} = \min_{k \in [l_{j-1}, u_{j-1}]} C_{k,j-1} + U_{i,j} + |X_{k,j} - X_{i,j}| + Y_{i,j} + \tau|k - i|$$

Where, $U_{i,j} = \sum_{t=l_j}^{i} D'_{(t,j)}(1) - D'_{(t,j)}(0)$, $X_{i,j} = \sum_{t=l_j}^{i} V_{(t,j-1),(t,j)}$, and $Y_{i,j} = V_{(i+1,j),(i,j)}$. Here $V_{p,q}$ is the cost of an edge between pixels p and q (Eqn. 1). The last term τ forces the path to be smoother. The terms U and X can be precomputed allowing evaluation of the expression on the right in $O(m)$ time. Thus, the complexity of computing the optimal path is $O(m^2n)$. The optimal path within l, u can be obtained by maintaining back-pointers.



Original Image | Iterations 1-3 | Iterations 16-18 | Iteration 21-23

Figure 4: Progression of *refined MRF* algorithm for a sample image. In the first 3 iterations, three 3 buildings are selected in the order, 1^{st} in *red*, 2^{nd} in *green* and 3^{rd} in *blue* color. Similarly, intermediate 3 and last 3 iterations are shown on the right.

Refined MRF. The *tiered MRF* approach does not respect the overall shape, hence we propose a hybrid approach where we *refine* the upper boundaries of a building using the dynamic programming approach proposed earlier *only* within the left and right edges of a building found by the *rectangle MRF*. This maintains the overall shape while allowing better fits to the upper boundary. For a building of width d this can be computed in $O(m^2 d)$.

Description	MAO
Color + Texture + Spatial	53.4%
w/o Color	50.3%
w/o Texture	37.2%
w/o Spatial	33.1%

Table 1: **Quality of the unary potentials.** MAO scores on the *validation* set using unary potentials only.

5. Experiments

Images within each city in the dataset are split into *training, validation* and *test* sets of 3, 3 and 4 images each respectively. This results in a *training/validation* set of 36 images and a *test* set of 48 images. We begin by seeking the best representation of appearance of buildings by evaluating the appearance models in isolation on the *validation* set. We then report segmentation results for two different scenarios. In the interactive setting seeds are provided as input, whereas in the automatic setting they are not. In both the settings we report MAO numbers on the *test* set. All the parameter optimization is performed on the *training/validation* set. In addition to accuracy, we also present a comparison of the running times of various methods.

One might be concerned about the potential overlap of images from the same city in the training and test set. However most of our modeling is image specific, with the exception of few parameters such as α and β (described in the next section) that trade off color and texture weights, the 'texton' dictionary used to estimate texture histograms, as well as the MRF parameters such as λ and τ. These parameters are kept fixed across all images. In an experiment where we randomly split the cities into two halves, and using all the images from cities in one half for estimating optimal parameters, while predicting the results on the later half, showed a difference in MAO of about 0.1% compared to using the entire set for training. Hence, we believe that the overlap is not a concern for overfitting in our approach.

5.1. Region representation

We start with a SLIC superpixel segmentation [1]. Superpixels that contain seed pixels are assigned to the majority label. To assign the affinity of a pixel to a region (unary potential), we use color and texture features. Color is modelled with GMM same as [15], with $C_p(b,k)$ representing the contribution towards the unary potential at pixel p in k^{th} cluster for b^{th} building (label). The texture model is built over a pre-trained textons as in [14]. We assign each pixel to a texton, and compute the histogram of all (we use 32) textons in its local neighbourhood of radius 10 pixels. For this purpose, we cluster the histograms of the foreground pixels using k-means (we choose, $k = 3$). The contribution of the texture $T_p(b,k)$ is defined as the χ^2 distance of the local histogram, $h_p(i)$ computed at pixel p for i^{th} texton, from the mean of the k^{th} cluster, H_{ilk}. i.e.,

$$T_p(b,k) = \sum_{i=1}^{32} \frac{(H_{ibk} - h_p(i))^2}{(H_{ibk} + h_p(i))} \tag{3}$$

Finally, unary potential $D_p(b)$ for pixel p and building (label) b is computed as,

$$\alpha(\beta \min_k C_p(b,k) + (1-\beta) \min_k T_p(b,k)) + (1-\alpha) S_p(b)$$

where $S_p(b)$ is horizontal distance of the p^{th} pixel from mean seed for the building. Parameters α and β are chosen by cross-validation. Fig. 5 shows color, texture and spatial models for a sample building in an image, along with the final unary potential.

Tab. 1 presents the quality of the unary potentials. Labels are obtained by taking the pixel-wise minimum of the costs of each label. The tables shows that all the three components (color, shape and texture) contribute to the final success. Color alone is not sufficient, possibly due to wide appearance variations of facades of a building caused by sunlight. Adding texture significantly improves the performance. The performance of the combination is not sensitive

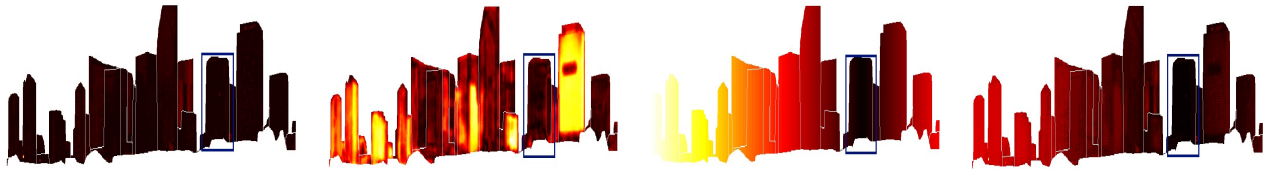


Figure 5: Figure shows *color, texture, spatial* models and the combined unary costs for a sample building (within the box) with $\alpha = 0.35$ and $\beta = 0.4$. The regions with low costs are darker (black) than the ones with high cost (yellow). The corresponding image is the left skyline in Figure 8. (Note: images are costs, rescaled and resized for visibility.)

over a wide range of α and β. For instance, MAO for validation set does not vary significantly over values of α between 0.2-0.4 and β between 0.2-0.6. Optimum values of α and β obtained on the *validation* set are 0.35 and 0.20. While calculating unary potentials for various experiments, all three models are normalized to unit variance.

5.2. Interactive segmentation

In the interactive setting we compare *tiered MRF*, *rectangle MRF* and *refined MRF* with a *standard MRF* formulation where α-expansion is used to solve the binary labeling problem. We use publicly available code for max-flow/min-cut for optimizing the problem [5]. For a fair comparison we run all the algorithms for $K = 2$ outer iterations (Algo. 1). In our experiments we found that no significant change in labeling after 2 iterations. For speed we also resize all the images to a maximum dimension of 2000 pixels, and the results rescaled to the original size for evaluation.

Tab. 2 presents results in the interactive setting. All the MRF formulations significantly improve over the unary potentials. *Our proposed approaches are about an order of magnitude faster than the α-expansion*. The *rectangle MRF* achieves results almost as good as the *standard MRF* while taking only 5.5s on average per image on commodity desktop with an Intel CPU @ 3.20GHz. Refinement on top improves performance for a small additional time of 3.7s (for a total of 9.2s). Tiered labeling is fast but not competitive showing the value of enforcing shape priors.

In a typical skyline image, many buildings have two visible facades, each with different color and texture due to sunlight, because of which the unary potentials are unreliable. Here shape priors can provide additional cues to guide segmentation. Fig. 6 shows the significance of shape priors in segmenting buildings. The *refined MRF* outperforms both *standard MRF*and *tiered MRF*, while preserving contiguity and shape of the segments. While it correctly segments buildings in most of the cases, there are images where rectangular shape prior is grossly incorrect. Two such examples are show in Fig. 7. In the first case, *refined MRF* fails due to irregular shapes of crowded and similar buildings. In the later case, the rectangular shape prior is incorrect due to concave shape of the buildings.

Method	MAO	Complexity/bldg.	Speed/img.
Unary only	54.5%	n/a	n/a
Standard MRF	62.3%	$O(m^3n^3)$	69.5s
Tiered MRF	59.4%	$O(m^2n)$	7.5 s
Rectangle MRF	62.0%	$O(mn^2)$	5.5 s
Refined MRF	**63.4%**	$O(mn^2 + m^2d)$	9.2 s

Table 2: **Speed and accuracy tradeoff in the interactive setting.** For various methods MAO scores, *worst case* computational complexities per building, and speed per image (in seconds) averaged over the *test* set are shown. All the methods are run for $K = 2$ outer iterations (Algo. 1). Images are resized to a maximum dimension of 2000 pixels for speed. The typical image is of size $m \times n = 1255 \times 2000$ pixels and has 34 buildings.

The automatic segmentations and ground-truth labels for some example images from the dataset of various interactive approaches are shown in Fig. 8. The *rectangle MRF* obtains a rough approximation of building structure quickly, which is then refined by *refined MRF* leading to more accurate boundaries.

5.3. Automatic segmentation

In the automatic setting, we start with a baseline segmentation, and refine it using our method. The initial segmentation method is used to estimate the seeds which are then used as input for the interactive segmentation methods described in the earlier section.

For the initial segmentation we use either SLIC [1], graph-based segmentation [8], or gPb regions [2]. The way we estimate seed regions is as follows: a skyline is partitioned into N vertical divisions and largest K segments are selected from each such division of the baseline. Buildings in a skyline are layered due to varying depth of buildings from camera. The uniform selection of segments is effective in selecting buildings in all layers. Generally, a skyline has $2 - 3$ such layers. In all experiments we set $N = 20$ and $K = 2$. Thus, we select $N \times K = 40$ uniformly distributed largest segments from the output of a segmentation algorithm and label these as different buildings. This serves as a *baseline*. A number of pixels within the segments are used as seeds for the interactive methods.

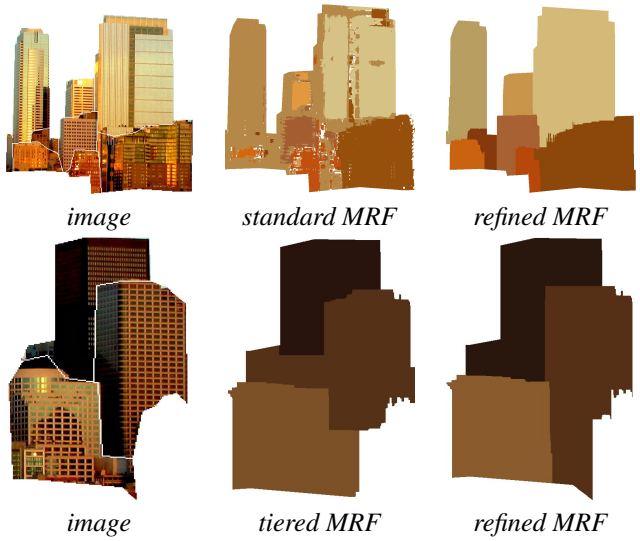


Figure 6: Two examples where *refined MRF* improves over the *standard MRF* and *tiered MRF*. On the top row, *standard MRF* over-segments the building. In the bottom row, lack of an explicit shape model in the *tiered MRF* causes it to incorrectly extend the rightmost building. In both these cases explicit shape priors enforced by the refined MRF enables it to correctly segment the buildings.

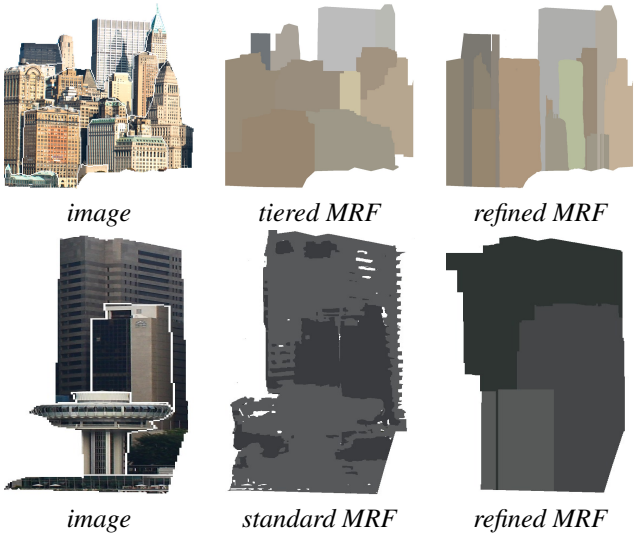


Figure 7: Some failures of the shape-constrained MRFs. On the image in the top row, the buildings are not rectangular and the *refined MRF* makes many mistakes, and the tiered structure alone is more appropriate. On the bottom row, the *refined MRF* incorrectly segments the concave buildings in the bottom.

Method	SLIC [1]	Graph based [8]	gPb [2]
Initial	24.56%	20.17%	26.35%
Tiered MRF	27.22%	25.86%	31.51%
Rectangle MRF	**27.33%**	**27.87%**	32.79%
Refined MRF	27.30%	27.42%	**33.13%**

Table 3: **Performance in the automatic setting.** Starting from various baseline segmentation algorithms such as SLIC, graph-based segmentation, and gPb regions, we perform an automatic labeling. The table shows the MAO scores for various the methods. Seeds obtained from gPb regions offer the best performance.

Tab. 3 compares various methods in the automatic setting. The *refined MRF* and *rectangle MRF* give significant performance boost over all these methods, an average 40% improvement over graph-based segmentation and 25% improvement over gPb, in few images showing as much as 60% improvement over the baseline. The running time of these methods are similar to those described in Tab. 2.

Among various low-level methods for segmentation, SLIC and graph-based use only color, while gPb uses both texture and color, hence the improved baseline. Nonetheless, our method improves over all of these methods mainly due the utilization of shape priors. In an interactive setting a user may use this as an input to guide effort in correction. The results for some images using the automatic approaches are shown in the last row of Fig. 8. Our method may be made fully automatic using methods such as [11] that can estimate the upper and lower boundaries. In our experiments we found that although these methods are fairly good, they still make mistakes. Hence to avoid confounding factors for mistakes in our analysis, we choose to include the boundary as part of the input for the automatic segmentation methods.

6. Summary

We presented a user-guided approach for extracting the structure of buildings within a skyline image. Our *shape-constrained MRF* approach lets us exploit the shape priors of the buildings and the tiered structure, allowing more accurate parsing. Compared to standard approaches for optimizing MRFs such as α-expansion, our *rectangle MRF* method is significantly faster, taking a few seconds to label a 3 mega-pixel image. Further refinement within the constraints of the rectangle improves accuracy. This coarse-to-fine approach for parsing may be used in other settings where an explicit search over shapes is faster than graph-cuts. Our preliminary results on improving automatic segmentation methods using shape priors are also promising. Finally, the **skyline-12** dataset consisting of 120 high resolution images with detailed annotations, and code for reproducing the results presented, will be available for download at the author's website.

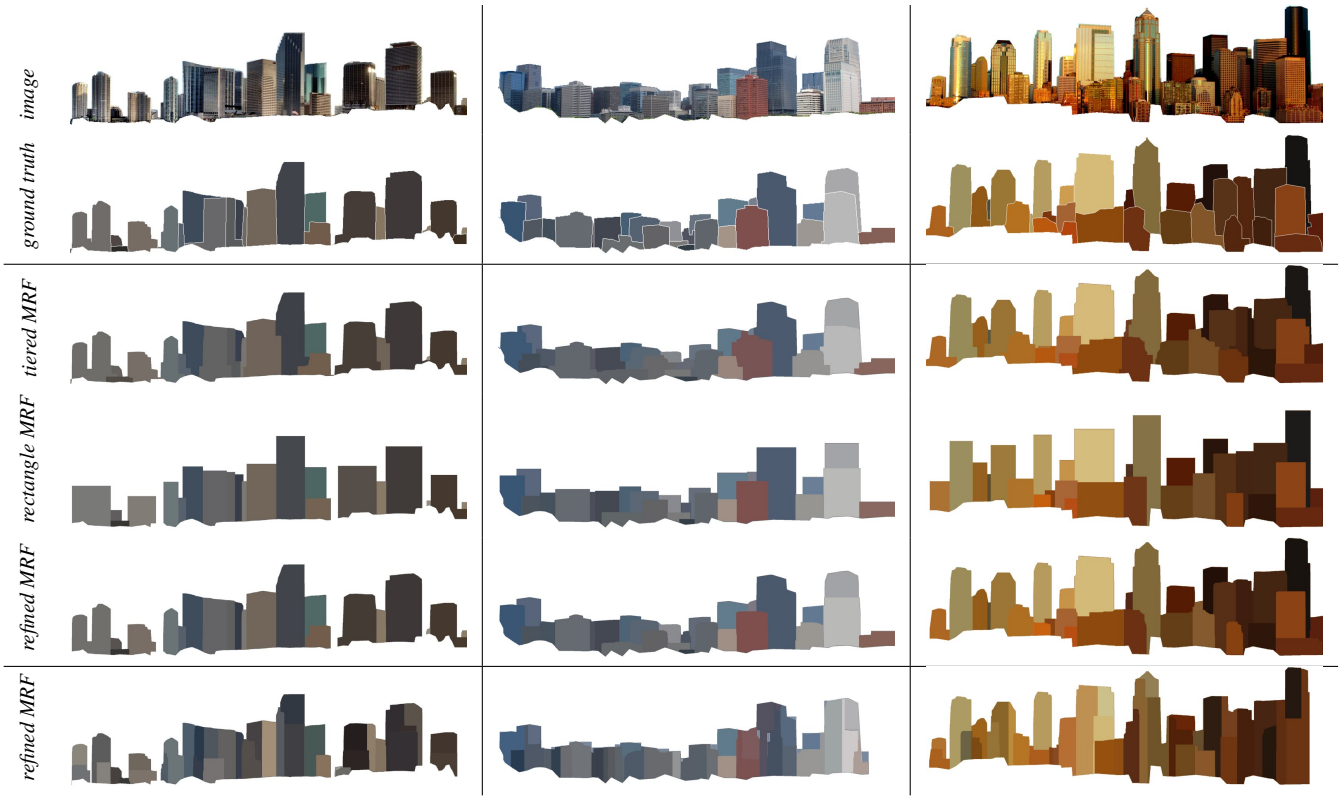


Figure 8: **Skyline segmentation results for three images**. In first and second row, original skylines within the upper and lower boundary and the corresponding ground truth segmentation are shown. In third, fourth, fifth and sixth row, outputs of the interactive *tiered MRF*, *rectangle MRF*, *refined MRF* and automatic *refined MRF* are shown respectively.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC Superpixels compared to state-of-the-art superpixel methods. *IEEE PAMI*, 2012. 2, 3, 5, 6, 7

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE PAMI*, 2011. 2, 3, 6, 7

[3] G. Baatz, O. Saurer, K. Kser, and M. Pollefeys. Large scale visual geo-localization of images in mountainous terrain. In *ECCV*, 2012. 2

[4] A. Blake, C. Rother, M. Brown, P. Pérez, and P. H. S. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, 2004. 2

[5] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *IEEE PAMI*, 2004.

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE PAMI*, 2001. 2, 3

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1, 3

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004. 2, 3, 6, 7

[9] P. F. Felzenszwalb and O. Veksler. Tiered scene labeling with dynamic programming. In *CVPR*, 2010. 2, 4

[10] D. Freedman and T. Zhang. Interactive graph cut based segmentation with shape priors. In *CVPR*, 2005.

[11] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 2007. 1, 2, 7

[12] P. Kohli, L. Ladický, and P. H. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 2009. 2

[13] T. Malisiewicz and A. A. Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*, 2007. 3

[14] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE PAMI*, 2004. 5

[15] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, 2004. 2, 5

[16] 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*, 2007.

[17] J. Xu, M. D. Collins, and V. Singh. Incorporating topological constraints within interactive segmentation and contour completion via discrete calculus. In *CVPR*, 2013. 3

[18] C. Zhang, L. Wang, and R. Yang. Semantic segmentation of urban scenes using dense depth maps. In *ECCV*, 2010. 2

[19] Y. Zheng, S. Gu, and C. Tomasi. Fast tiered labeling with topological priors. In *ECCV*, 2012. 2, 4