MONOCULAR EXTRACTION OF 2.1D SKETCH

Mohamed Amer, Raviv Raich, and Sinisa Todorovic

School of EECS, Oregon State University, Corvallis, OR 97331

ABSTRACT

The 2.1D sketch is a layered representation of occluding and occluded surfaces of the scene. Extracting the 2.1D sketch from a single image is a difficult and important problem arising in many applications. We present a fast and robust algorithm that uses boundaries of image regions and T-junctions, as important visual cues about the scene structure, to estimate the scene layers. The estimation is a quadratic optimization with hinge-loss based constraints, so the 2.1D sketch is smooth in all image areas except on image contours, and image regions forming "stems" of the T-junctions correspond to occluded surfaces in the scene. Quantitative and qualitative results on challenging, real-world images—namely, Stanford depth-map and Berkeley segmentation dataset—demonstrate high accuracy, efficiency, and robustness of our approach.

***Index Terms*—** Layered 2.1D sketch, T-Junctions, segmentation, quadratic optimization with hinge-loss penalty

1. INTRODUCTION

This paper is about extracting, from a single image, a layered 3D-scene representation, called the 2.1D sketch [1–4]. In this representation, surfaces of the scene are assumed to form planes (layers) whose normals lie along the camera viewing direction. Occluding surfaces of the scene comprise a layer closer to the camera than the layer formed by occluded surfaces. Actual 3D distances are mapped onto integer distances between the corresponding layers.

Monocular extraction of the 2.1D sketch is required by many applications, including range analysis [3], object recognition [5], and image-based walkthrough [6]. However, due to the loss of 3D information in the imaging process, this problem is very difficult. For example, the depth-ordering between objects in the scene may not be well-defined, because of self-occlusions and entanglements.

Prior work seeks to model interactions between image features, such as regions, contours and junctions, and thus infer the 2.1D sketch. Examples of these models include the "dead leaves" model of occlusion [1]; minimum description length of image support maps [3]; layered Markov Random Fields and hierarchical graphical models [4]; and classifiers discriminating image regions into vertical, horizontal, or support surfaces in the scene [5, 6]. These approaches typically make heuristic assumptions about the number of scene layers. Although occlusions may occur over large spatial extents in the image, these methods usually analyze only pairwise (local) interactions between image features. Our approach addresses these limitations by conducting a global analysis of higher-order interactions among image features, to infer the data-driven number of scene layers and their globally consistent layout.

Our work is most related to Gestalt-based methods that analyze T-junctions as power visual cues of occlusion [1, 7, 8]. The "cap" of the T indicates the occlusion boundary between two surfaces, and the "stem" of the T is formed by an occluded surface and the background. However, some T-junctions do not arise from occlusion, but brightness discontinuities along a surface. This ambiguity is resolved in [7, 8] by estimating a globally consistent 2.1D sketch, based on *gradients* of the depth map. The gradients are inferred from stems and caps of the T-junctions, and used to estimate either a diffusion map [7], or a directed acyclic graph of T-junctions in which the graph edges point to occluded image parts [8]. However, this gradient based formulation has limitations. T-junctions could provide only the orientation of depth-map gradients, whereas the gradient magnitudes (e.g., edge weights in the DAG [8]) have to be heuristically selected.

Our key contribution is a new, more natural way of exploiting T-junctions within a hinge-loss regularized quadratic optimization for extracting the 2.1D sketch. Instead of using the depth-map gradients, we specify an optimization problem which directly constraints that image regions forming the T's "cap" should be closer to the camera than regions forming the "stem". Our optimization relies on an iterative framework where every iteration involves the solution of large scale set of linear equations. To efficiently solve for the linear equation at each iteration, we use a two step method. Two step methods offer significant speedups (e.g., Krylov space methods [9]), and thus gain rising popularity in many fields, including sparse reconstruction and compressed sensing. We consider a similar framework for the reconstruction of layered representation.

Fig. 1 shows an illustration of our approach. As input to the optimization, we use image segments and detected T-junctions to enforce smoothness and global-consistency constraints on the resulting 2.1D sketch. Since extracting T-junctions and image regions are two difficult problems in and of themselves, this paper only focuses on the optimization, for brevity. In the following, we will thus assume that region boundaries and "stems" and "caps" of T-junctions are given, while the image segmenter and T-junction detector we used in our experiments are briefly described in the results section.

2. PROBLEM FORMULATION

This section formulates a new optimization problem for estimating the 2.1D sketch. We proceed with necessary notation.

Let $D(x, y)$ denote the 2.1D sketch, obtained from the image support S, $D(\cdot, \cdot) : (x, y) \in S \mapsto D(x, y)$. Let $B \subset S$ be the input set of points that belong to region boundaries. We will account for B in our quadratic optimization via the boundary mask $M(\cdot, \cdot) : (x, y) \in S \mapsto M(x, y)$, where $M(x, y) = 1$ if $(x, y) \notin B$ and zero otherwise. We estimate $D(x, y)$ using the following two principles.

Smoothness – Image regions, by definition, represent 3D scene surfaces with homogeneous photometric properties. Therefore, it is reasonable to expect that an image region cannot split over several distinct layers in the 2.1D sketch. This means that $D(x, y)$ should be smooth within every image region. This can be enforced through

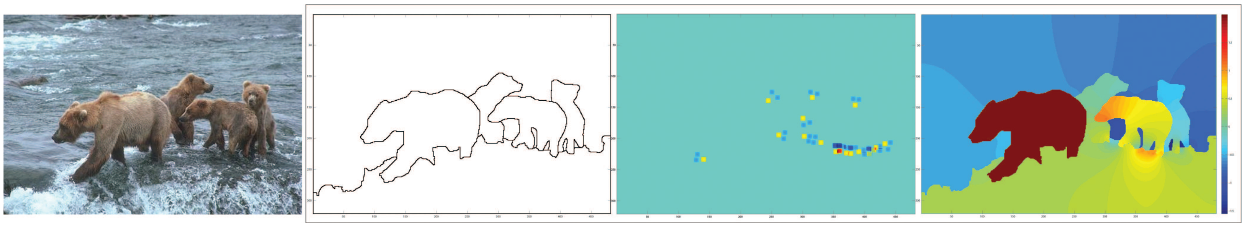

Fig. 1. Our approach: From the input image (left) we extract segments (middle left) and T-junctions (middle right). The segments and foreground-ground relations between pairs of segments obtained from the detected T-junctions are used to enforce smoothness and global-consistency constraints in a quadratic optimization which estimates the 2.1D sketch. The scene-depth layers are color-coded so that layers closer to the camera have "warmer" colors.

the minimization of restricted Laplacian of $D(x,y)$ as

$$\min_{D(\cdot,\cdot)} \iint M(x,y)\left(\left(\frac{\partial D(x,y)}{\partial x}\right)^2+\left(\frac{\partial D(x,y)}{\partial y}\right)^2\right)dxdy. \tag{1}$$

While the resulting D would be constant over every image region, as required, these constants may take arbitrary values, since (1) can be solved separately for each region. Therefore, (1) needs to be augmented with constraints that would enforce globally consistent dependencies between the values of D, as explained below.

T-junction consistency – Instead of constraining the gradients of D, as done in prior work [7, 8], we use a more natural way to incorporate T-junctions in estimation of the 2.1D sketch—namely, the value of $D(x,y)$ at the "cap" of a T-junction should be strictly larger than the value at the two sides of the "stem". For the ith T-junction, we consider three points: above the "cap" (x_{i1},y_{i1}), to the right of the "stem" (x_{i2},y_{i2}), and to the left of the "stem" (x_{i3},y_{i3}), and specify the following constraints for $i=1,2,\dots,T$

$$D(x_{i1},y_{i1})\geq D(x_{i2},y_{i2})+1,\ D(x_{i1},y_{i1})\geq D(x_{i3},y_{i3})+1. \tag{2}$$

By combining (1) and (2), the resulting D will be constant within each segment and satisfy the required constraints, and thus represent the desired 2.1D sketch. Note that T-junctions, being local cues, may introduce globally inconsistent constraints, e.g., when two objects are intertwined. The above formulation is capable of gracefully solving this problem. The global inconsistencies will result in D with smoothly varying values in the corresponding regions. This will reflect that some objects may indeed extend across two distinct scene layers.

To formulate our algorithm, we simplify notation by representing $D(x,y)$ and $M(x,y)$ with column vectors $\boldsymbol{d}$ and $\boldsymbol{m}$, where all pixels are stacked in the vector. Operators can be expressed as matrices, and are denoted by boldface capital letters. Specifically, the restricted Laplacian from (1) is given by $\mathbf{d}^{\mathrm{T}}\mathbf{L}\mathbf{d}$ where $\mathbf{L}=\mathbf{D}_x^{\mathrm{T}}\mathrm{diag}(\mathbf{m})\mathbf{D}_x+\mathbf{D}_y^{\mathrm{T}}\mathrm{diag}(\mathbf{m})\mathbf{D}_y$, and $\mathbf{D}_x$ and $\mathbf{D}_y$ are the differentiation operators along the x-axis and the y-axis, respectively. Note that diag$(\mathbf{m})$ corresponds to a diagonal matrix with $\mathbf{m}$ on its diagonal. Also, the multiplication with diag$(\mathbf{m})$ corresponds to an elementwise multiplication with $\mathbf{m}$. To represent the T-junction constraints of (2), we define $\mathbf{a}_{i1}$ and $\mathbf{a}_{i2}$ vectors of the same size as $\boldsymbol{d}$, such that $\mathbf{a}_{i1}^{\mathrm{T}}\mathbf{d}{=}D(x_{i1},y_{i1}){-}D(x_{i2},y_{i2})$ and $\mathbf{a}_{i2}^{\mathrm{T}}\mathbf{d}{=}D(x_{i1},y_{i1}){-}D(x_{i3},y_{i3})$. Note that $\mathbf{a}_{i1}$ allows us to measure the "cap-to-right-of-stem" difference, and $\mathbf{a}_{i2}$ allows us to measure the "cap-to-left-of-stem" difference. Using the above vector notation in (1) and (2), we formalize our algorithm as

$$\min_{\mathbf{d}}\ \mathbf{d}^{\mathrm{T}}\mathbf{L}\mathbf{d}$$
$$\text{subject to}\quad \mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}\geq 1,\ \ i=1,\dots,T,\ \ j=1,2. \tag{3}$$

The constraints $\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}\geq 1$ can be replaced with a penalty term $f_1(\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d})$ in the minimization (3), where $f_\gamma(\cdot)$ is the standard hinge loss function, given by $f_\gamma(\delta){=}(\gamma{-}\delta)\mathbf{1}(\delta{<}\gamma)$, and where $\mathbf{1}(\cdot)$ is the indicator function. As a result, we obtain the equivalent optimization problem

$$\min_{\mathbf{d}}\ J(\mathbf{d}),\ \ J(\mathbf{d})=\mathbf{d}^{\mathrm{T}}\mathbf{L}\mathbf{d}+2\lambda\textstyle\sum_{i=1}^{\mathrm{T}}\sum_{j=1}^{2} f_1(\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}), \tag{4}$$

where λ controls the trade-off smoothness vs T-junction constraints. Minimizing $J(\mathbf{d})$ is not trivial, since the hinge loss function $f_1(\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d})$ is not differentiable at $\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}{=}1$. We proceed with an optimization transfer approach for the solution of (4).

3. ALGORITHMIC SOLUTION

This section presents optimization transfer [10] for solving the minimization in (4). The key idea of our approach is to replace the minimization that cannot be solved in closed-form, with an iterative method with strong theoretical guarantees of convergence.

Any general objective function of the form $J(\mathbf{d})$, as in (4), can be minimized with respect to $\boldsymbol{d}$ by iteratively minimizing a surrogate function, $H(\mathbf{d},\mathbf{d}')$, as $\mathbf{d}^{(k+1)}=\arg\min_{\mathbf{d}} H(\mathbf{d},\mathbf{d}^{(k)})$. For this, $H(\mathbf{d},\mathbf{d}')$ must satisfy: $J(\mathbf{d}){\leq}H(\mathbf{d},\mathbf{d}')$ and $J(\mathbf{d}){=}H(\mathbf{d},\mathbf{d})$. These two criteria guarantee that $J(\mathbf{d}^{(k)})$ is non-increasing with iterations k. Also, under some mild conditions, the optimization transfer guarantees convergence to local minima [10].

We use the optimization transfer to minimize $J(\mathbf{d})$ in (4). To identifying surrogate $H(\mathbf{d},\mathbf{d}')$ for $J(\mathbf{d})$, consider the following bound on the hinge loss function:

$$f_\gamma(\delta)\leq\frac{\delta^2}{4|\gamma|}+(\gamma-\delta)\mathbf{1}(\delta'<\gamma). \tag{5}$$

Since $f_1(\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d})=f_{1-\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}'}(\mathbf{a}_{ij}^{\mathrm{T}}(\mathbf{d}-\mathbf{d}'))$, from (5), we have

$$f_1(\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d})\leq\frac{(\mathbf{a}_{ij}^{\mathrm{T}}(\mathbf{d}-\mathbf{d}'))^2}{4|1-\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}'|}+(1-\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d})\mathbf{1}(\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}'<1). \tag{6}$$

From (6) and (4), we derive the upper bound of $J(\mathbf{d})$ as

$$\begin{aligned}H(\mathbf{d},\mathbf{d}')=\mathbf{d}^{\mathrm{T}}\mathbf{L}\mathbf{d}+\lambda\Big(\sum_{i=1}^{\mathrm{T}}\sum_{j=1}^{2}&\frac{(\mathbf{a}_{ij}^{\mathrm{T}}(\mathbf{d}-\mathbf{d}'))^2}{2|1-\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}'|}\\&+2(1-\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d})\mathbf{1}(\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}'<1)\Big).\end{aligned} \tag{7}$$

From a quick glance at (7), $H(\mathbf{d},\mathbf{d}')$ has the standard quadratic form

$$H(\mathbf{d},\mathbf{d}')=\mathbf{d}^{\mathrm{T}}\mathbf{A}(\mathbf{d}')\mathbf{d}-2\mathbf{b}(\mathbf{d}')^{\mathrm{T}}\mathbf{d}+c(\mathbf{d}'), \tag{8}$$

where

$$\mathbf{A}(\mathbf{d}') = \mathbf{L} + \lambda \sum_{i=1}^{\mathrm{T}} \sum_{j=1}^{2} \frac{\mathbf{a}_{ij}\mathbf{a}_{ij}^{\mathrm{T}}}{2|1 - \mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}'|}, \tag{9}$$

$$\mathbf{b}(\mathbf{d}') = \lambda \big(\sum_{i=1}^{\mathrm{T}} \sum_{j=1}^{2} \big[\frac{\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}'}{2|1 - \mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}'|} + \mathbf{1}(\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}' < 1) \big] \mathbf{a}_{ij} \big), \tag{10}$$

$$c(\mathbf{d}') = \lambda \big(\sum_{i=1}^{\mathrm{T}} \sum_{j=1}^{2} \frac{(\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}')^2}{2|1 - \mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}'|} + 2\,\mathbf{1}(\mathbf{a}_{ij}^{\mathrm{T}}\mathbf{d}' < 1) \big). \tag{11}$$

Next, we apply the optimization transfer on (4), i.e., minimize $H(\mathbf{d}, \mathbf{d}')$, given by (8), which yields

$$\mathbf{A}(\mathbf{d}^{(k)})\mathbf{d}^{(k+1)} = \mathbf{b}(\mathbf{d}^{(k)}). \tag{12}$$

In the following, we explain how to solve the system of linear equations in (12). Note that the inversion of $\mathbf{A}(\mathbf{d}^{(k)})$ is not trivial. Recall that $\mathbf{A}(\mathbf{d}^{(k)})$ contains the restricted Laplacian operator, defined as a matrix of size $N \times N$, where N is the total number of pixels in the image. In our experiments, this component of $\mathbf{A}(\mathbf{d}^{(k)})$ is often impossible to store, let alone invert, due to its large size. Instead, we consider an accelerated iterative solution to the minimization of (8)

$$\mathbf{f}^{(l+1)} = \mathbf{f}^{(l)} + \alpha_l(\mathbf{f}^{(l-1)} - \mathbf{f}^{(l)}) + \beta_l(\mathbf{b}(\mathbf{d}^{(k)}) - \mathbf{A}(\mathbf{d}^{(k)})\mathbf{f}^{(l)}), \tag{13}$$

where α_l and β_l are coefficients that can be found directly by substituting the right-hand-side of (13) into (8), and minimizing with respect to α_l and β_l. Our main motivation to use (13) is that it provides faster convergence than the standard gradient descent, which can be obtained by setting $\alpha_l = 0$ in (13). Typically, the convergence of (13) is of the order of square root of the convergence time of gradient descent. In theory, the solution can be obtained after the iterations have converged. In practice, a finite number of iterations suffices, i.e., $\mathbf{d}^{(k+1)} = \mathbf{f}^{(L)}$, where L is sufficiently large to allow a desired decrease in the objective function.

Per iteration, complexity is dominated by application of $\mathbf{A}(\mathbf{d}^{(k)})$ to $\mathbf{f}^{(l)}$. Complexity of the first term in $\mathbf{A}(\mathbf{d}^{(k)})$ (i.e., the restricted Laplacian) is $O(N)$, where N is the number of image pixels, since differentiation along the x-axis, y-axis, and multiplication with a mask, are all $O(N)$. The contribution of the second term in $\mathbf{A}(\mathbf{d}^{(k)})$ associated with the T-junctions is $O(T)$ where T is the number of T junctions. Since T is by definition smaller than N, the overall complexity per iteration is $O(N)$.

4. RESULTS

This section presents qualitative and quantitative evaluation of our approach. We use challenging, real-world images of Berkeley segmentation dataset [11], shown in Figs. 1, 2, and 3, and Stanford Make3D dataset [12], shown in Fig. 4. The Berkeley dataset does not provide ground-truth annotations of the 2.1D sketch; therefore we use these images for the qualitative evaluation. The Stanford dataset consists of eight images, each accompanied by a real-valued depth map. The ground-truth depths were collected using a laser scanner. To provide image segmentation as input to our algorithm, we used the Berkeley's latest code available online, though any other off-the-shelf segmenter could have been used. T-junctions are detected by finding the intersection of any three boundaries, which is done by overlaying segments on top of each other and finding points that had three boundaries sharing it. After detecting T-junctions, their "stems" and "caps" are simply identified by examining the variances of pixel values across the edges forming each T-junction. Evaluation of the image segmenter and T-junction detector is not our focus.

Qualitative evaluation – Figures 1–4 show examples of our results. Layers closer to the camera are color-coded with "warmer" colors. As can be seen, our approach successfully estimates the scene layers and their relative distances from the camera.

Quantitative evaluation – Most prior work uses only qualitative evaluation (e.g., [7, 8]). Accuracy of our layer ordering is estimated for the Stanford images with respect to their ground-truth depth maps. Since the depth maps have real values, we adapt them to be suitable for comparison with our resulting 2.1D sketches. To this end, we first segment the Stanford images using the Berkeley's code, and average the associated depth map values over each region in the segmentation. Then, we form a region adjacency matrix, W_{depth}, whose entries are defined as $\text{sign}(d_i - d_j)$, where d_i and d_j are average depths of regions i and j. We compute a similar matrix, W_{sketch}, based on estimated values of the 2.1D sketch, $\boldsymbol{d}$, over image regions. Finally, the error is computed as the Hamming distance between $W_{\text{depth-map}}$ and W_{sketch}, averaged over all pairs of regions. As can be seen in Fig. 5, this error over eight Stanford images is about 37%. Fig. 5 also shows that our approach is relatively insensitive to lower recall rates of T-junction detection in the image, up to the recall of 80%. The relatively simple T-junction detector that we use in this paper has recall that is on average larger than 85%.

Only qualitative comparison is possible with [7, 8] and other competing approaches, as they do not report quantitative results. On their much simpler examples (e.g., synthetic images) we achieve qualitatively the same performance.

5. CONCLUSION

We have presented a new approach to monocular extraction of the 2.1D sketch from the image segmentation and T-junctions. The problem is formulated as an optimization, involving the minimization of a smoothness term, which incorporates information about the segmentation, subject to a set of constraints that enforce Gestalt cues of the T-junctions. The T-junction constraints are incorporated into the objective function through a hinge-loss regularization term. We have specified an optimization-transfer based, iterative method to solve the problem. The algorithm is evaluated both qualitatively and quantitatively. The qualitative tests demonstrate that, given image segmentation and T-junctions, our algorithm is capable of resolving a layered scene representation which is consistent with human interpretation of challenging natural images. Quantitative results that compare a ground truth depth map with our performance demonstrate that our method can correctly recover the 2.1D sketch under the decreasing recall rate of T-junction detection. Occlusion may lead to an incorrect 2.1D sketch. This limitation is a consequence of using only low-level segmentation and T-junctions as visual cues. Our future work will deal with extensions that will include high-level image interpretation in the optimization constraints.

6. REFERENCES

[1] M. Nitzberg and D. Mumford, "The 2.1-D sketch," in *ICCV*, 1990, pp. 138–144.

[2] Edward H. Adelson, "Layered representation for vision and video," in *IEEE W. Representation of Visual Scenes*, 1995.

[3] T. Darrell and A.P. Pentland, "Cooperative robust estimation using layers of support," *IEEE TPAMI*, vol. 17, no. 5, pp. 474–487, 1995.

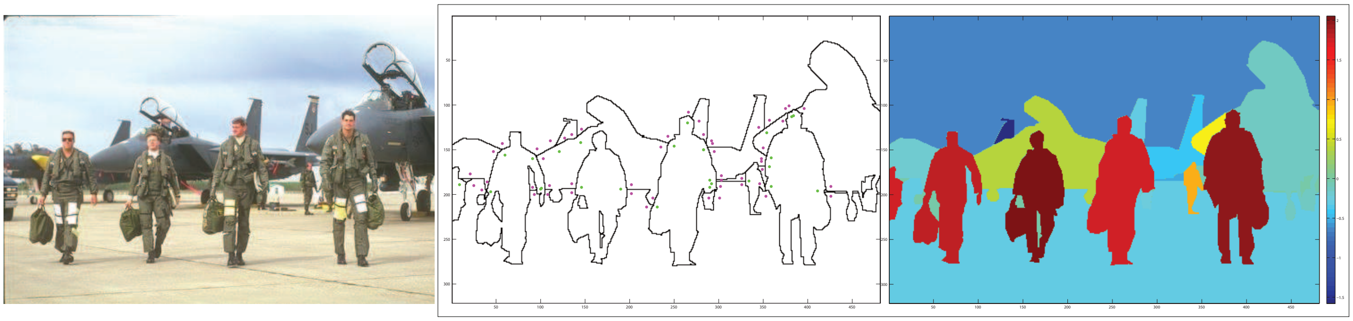

Fig. 2. An example from the Berkeley segmentation dataset [11]. Original image (left), input segmentation and T-junctions (middle), the output 2.1D sketch. The two airplanes are split into distinct layers due to occlusion, but the ordering of the scene layers is correctly estimated.

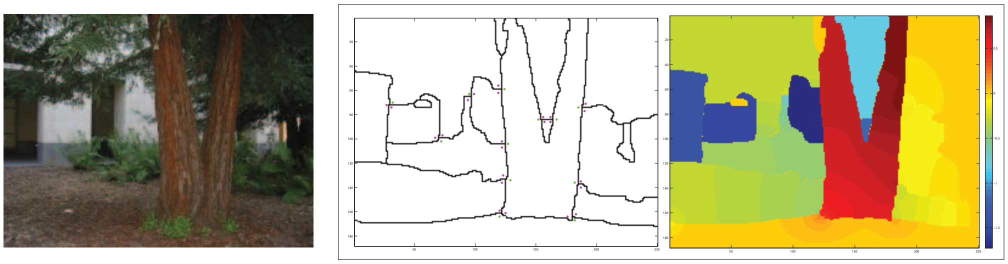

Fig. 4. An example from Stanford dataset [12] (left), and the associated ground-truth depth map (right). The input segmentation and T-junctions (middle left), and the output layered scene representation (middle right).

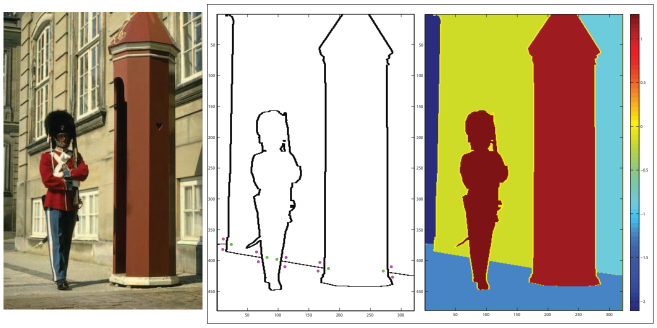

Fig. 3. An example from Berkeley dataset [11]: due to occlusion of the building, the correct ordering of layers cannot be resolved based on segmentation and T-junctions alone. Higher-level reasoning (e.g., about texture similarity) is needed to fix this problem.

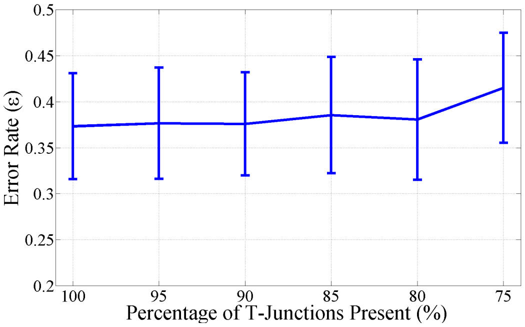

Fig. 5. Our approach is relatively insensitive to a drop in recall of T-junction detections up to 80%.

[4] R. Gao, T. Wu, S. Zhu, and N. Sang, "Bayesian Inference for Layer Representation with Mixed Markov Random Field," in *EMMCVPR*, 2007.

[5] D. Hoiem, A. Stein, A.A. Efros, and M. Hebert, "Recovering occlusion boundaries from a single image," in *ICCV*, 2007.

[6] Derek Hoiem, Alexei A. Efros, and Martial Hebert, "Automatic photo pop-up," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 577–584, 2005.

[7] J.M. Morel and P. Salembier, "Monocular Depth by Nonlinear Diffusion," in *ICVGIP*, 2008, pp. 95–102.

[8] M. Dimiccoli and P. Salembier, "Hierarchical region-based representation for segmentation and filtering with depth in single images," in *ICIP*, 2009.

[9] M. Hanke, "Accelerated Landweber iterations for the solution of ill-posed equations," *Numerische mathematik*, vol. 60, no. 1, pp. 341–373, 1991.

[10] D.R. Hunter and K. Lange, "A Tutorial on MM Algorithms.," *The American Statistician*, vol. 58, no. 1, pp. 30–38, 2004.

[11] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*, 2001.

[12] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3D scene structure from a single still image," *IEEE TPAMI*, vol. 31, no. 5, pp. 824–840, 2009.