Noname manuscript No.
(will be inserted by the editor)

Monocular Extraction of 2.1D Sketch Using Constrained Convex Optimization

Mohamed R. Amer · Siavash Yousefi · Raviv Raich · Sinisa Todorovic

Received: date / Accepted: date

Abstract This paper presents an approach to estimating the 2.1D sketch from monocular, low-level visual cues. We use a low-level segmenter to partition the image into regions, and, then, estimate their 2.1D sketch, subject to figure-ground and similarity constraints between neighboring regions. The 2.1D sketch assigns a depth ordering to image regions which are expected to correspond to objects and surfaces in the scene. This is cast as a constrained convex optimization problem, and solved within the optimization transfer framework. The optimization objective takes into account the curvature and convexity of parts of region boundaries, appearance, and spatial layout properties of regions. Our new optimization transfer algorithm admits a closed-form expression of the duality gap, and thus allows explicit computation of the achieved accuracy. The algorithm is efficient with quadratic complexity in the number of constraints between image regions. Quantitative and qualitative results on challenging, real-world images of Berkeley segmentation, Geometric Context, and Stanford Make3D datasets demonstrate our high accuracy, efficiency, and robustness.

Keywords 2.1D sketch · figure-ground assignment · image segmentation · convex quadratic optimization

M. Amer
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, 97331. Tel.: +1541 908 4765. E-mail: amerm@onid.orst.edu.

S. Yousefi
Department of Electrical Engineering, University of Washington, Seattle, WA E-mail: siavash@uw.edu.

R. Raich
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, 97331. E-mail: raich@eecs.oregonstate.edu.

S. Todorovic
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, 97331. E-mail: sinisa@eecs.oregonstate.edu.

1 Introduction

Our goal is to estimate a layered depth map – called 2.1D sketch [1, 12, 19, 40] – of the scene from a single image. The term 2.1D sketch was first introduced by [40] as an extension to the primal sketch representation in [22, 36]. The 2.1D sketch is defined as a piece-wise planar representation of surfaces and their relative depth ordering in the scene, where the plane normals lie along the camera viewing direction.

Monocular estimation of 2.1D sketch is a long-standing problem, encountered in many applications, including range analysis [12,35,44,46], object recognition [20,24,25,34,49], and image-based visualizations [26]. This problem is difficult, because there are infinitely many 3D scenes that could explain the 2D image. In addition, the depth ordering of objects may be ill-defined, e.g., in the cases of self-occlusions and entanglements.

A viable approach accounts for constraints about scene layouts, and thus rules out implausible solutions. We here focus on low-level constraints. Our approach is aimed as an initial step of diverse higher-level vision systems, and thus is *not* informed about any specific objects and surfaces occurring in the scene, their numbers, scales, and layouts. Low-level constraints may be inferred from image features, and their perceptual organization. For example, a T-junction unequivocally indicates the presence of a partial occlusion, and thus variation in the scene depth. However, detection of the low-level constraints is typically noisy. For instance, T-junctions can be easily confused with T-like patterns in an image texture.

We use regions as basic image features, since: i) region boundaries often coincide with object boundaries, and thus facilitate extracting local figure-ground constraints; and ii) spatial locality of regions allows a piecewise planar approx-

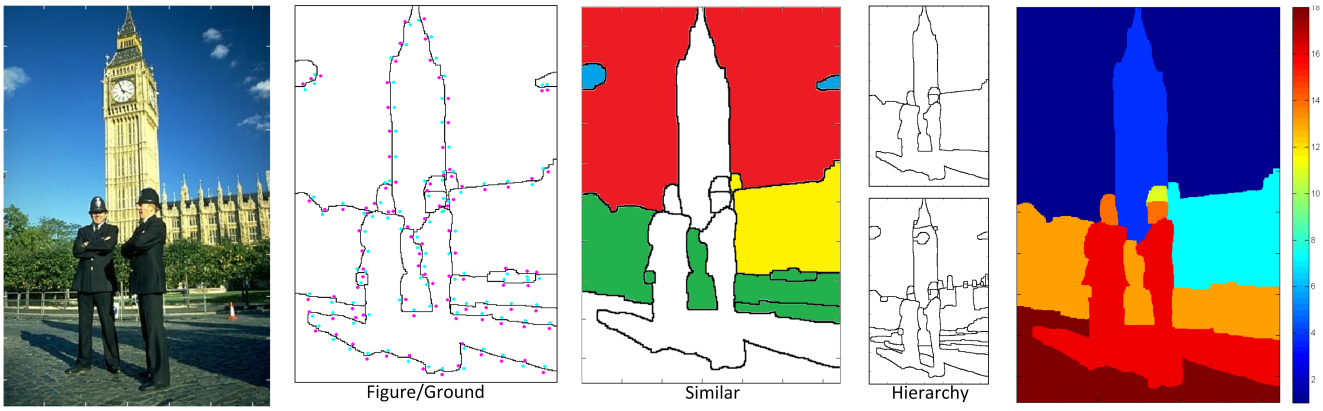


Fig. 1 Our approach: An input image is first segmented. We then estimate local Figure-Ground (marked as colored points) and Adjacent Layers relationships (similar regions and hierarchically related regions) along shared boundaries between regions. Hierarchy is established based on region nesting given by the segmentation tree which defines similarity relationship. A convex optimization uses these region relationships to estimate the 2.1D sketch. The optimization favors placing regions with similar appearance, and region-subregion pairs at near-by depth layers. The surfaces estimated as closer to the camera are depicted with warmer colors.

imation of the 3D scene, where the regions correspond to the frontally viewed planes.

As the right segmentation scale is unknown, we use a hierarchical segmentation of the image in order to have a high recall of true object boundaries, while maintaining the minimum region size sufficiently large so as to enable robust estimation of the 2.1D sketch. Thus, our goal is to assign a depth ordering to all multiscale image regions. The resulting depth map may merge a number of oversegmented regions to a single-depth layer.

An overview of our approach is illustrated in Figure 1. Given an image segmentation, we compute the depth map as a 2D, piecewise constant function with discontinuities at region boundaries. We first estimate two types of region relationships, which are then used as soft constraints of convex optimization for computing the 2.1D sketch. The first relationship is the *local* figure-ground (FG) depth ordering between neighboring regions that share a boundary. It is estimated based on the local curvature and convexity of boundary parts shared by the regions. Since a shared region boundary can be split into a number of parts, two neighboring regions may have a number of (noisy) FG relations, which may not necessarily all be the same. The estimated FG polarity is used in the convex optimization to favor placing the regions closer (F) and farther away (G) from the camera in the resulting depth map. The convex optimization uses the FG relations to separate regions into different layers. The second relationship is called Adjacent Layers (AL). Neighboring regions that are similar in appearance (i.e., color and texture) are estimated to have the AL relationship. In addition, all region-subregion pairs in the hierarchical segmentation are specified to have the Hierarchical Relationship (HR). This relationship serves to indicate image parts which have been oversegmented, or occupied by objects and their parts, i.e., image parts that correspond to surfaces of the scene which indeed belong to adjacent depth layers. The convex optimization uses the AL relationships to favor placing corresponding regions at near-by depth layers (or even at a single depth layer). This would amount to correcting oversegmentation, or constraining the depth differences of objects and their parts.[1]

The estimates of region relations are noisy, because they are obtained from local, low-level cues, largely affected by variations in illumination, scale, and viewpoint. The convex optimization jointly considers all region relations, and corrects the ones whose satisfaction would violate that the resulting depths of regions represent a *preorder* relationship. For example, a local F-G estimate of a pair of regions may be switched in the convex optimization to G-F, and, similarly, an AL estimate may be corrected to place the two regions at relatively distant ordinal depths. We say that the 2.1D sketch is consistent over a set of regions, if their estimated ordinal depths are reflexive and transitive, i.e., represent preorder. Our approach takes an input local cues estimates, and fuse the information from all cue to get a the most consistent depth representation.

Our formulation represents a tournament problem [4, 7, 9, 31, 48], where the goal is to rank n objects (i.e., image regions) according to some transitive characteristic (distance from the camera), by means of successive pairwise comparisons (i.e., FG, AL, and HR relations). The 2.1D sketch can thus be defined as tournament, i.e., an acyclic, directed, weighted subgraph of the graph of image regions, where edge weights represent differences in the regions' ordinal depths. Therefore, finding the 2.1D sketch amounts to iden-

[1] A region-subregion pair may not correspond to an object and its part (e.g., sky seen through tree branches). In this case, even human vision could not correctly identify their depths, without resorting to higher-level semantic cues, which are beyond our scope.

tifying a subset of directed edges with a minimum total weight, whose removal leaves a maximum-weight acyclic subgraph.

The tournament problem is known to be NP-hard and APX-hard [4,7,9,31,48], which means that there is no polynomial-time approximation scheme to solve the problem within every fixed percentage greater than zero. Therefore, estimating the 2.1D sketch is not only NP-hard, but also hard to approximate. The literature presents a large number of approximations aimed at special graphs (e.g., planar graphs). For general graphs, as is our case, recently reported approximate algorithms claim to have the best efficiency-accuracy trade off [38], with expected $(1+\epsilon)$-approximation guarantee (e.g., $\epsilon = 2$), and runtime which is singly exponential in $1/\epsilon$.

As one of the key contributions, this paper formulates depth-map estimation as a constrained convex quadratic optimization, aimed at efficiently approximating the NP-hard tournament problem. The convex problem is solved using a new algorithm, specified within the optimization transfer framework. Our algorithm is efficient with the quadratic complexity in the number of constraints between image regions. We empirically observe that its convergence rate is close to exponential, i.e., the values of our non-negative objective function follow a decreasing exponential function over the number of iterations. The algorithm provides explicit accuracy guaranties of the solution.

The new algorithm is suited for our (and other) problem(s) with a large number of variables. This is empirically validated via a comparison with the common gradient descent with backtracking line search. In our experiments, we typically observe relatively large values of the condition number of the Hessian, which negatively affects the convergence rate of the gradient descent. In turn, this makes identifying the right stopping criterion very hard. This is a fundamental problem, since the gradient descent does not provide any guarantees of closeness of its result to the global optimum. By contrast, the analytic tractability of our optimization transfer can save many iterations of the backtracking step size selection, and speed up convergence. This is because our algorithm admits a closed-form expression of the duality gap, and thus allows for explicit computation of the accuracy achieved at convergence. Our experiments demonstrate that our algorithm consistently yields better solutions than the gradient descent with backtracking line search.

This paper is organized as follows. Section 2 explains our contributions relative to prior work. Section 3 describes image segmentation, and estimation of FG and AL relations between regions. Sections 4–6 formulate extracting the 2.1D sketch as convex optimization. Sections 7–9 specify our optimization algorithm, and discuss its certificate of accuracy, convergence, and complexity. Section 10 presents our experimental evaluation. Derivation of certain details of our formulation and optimization algorithm are presented in the supplement material.

2 Prior Work and Our Contributions

The literature on monocular extraction of the 2.1D sketch can be broadly divided into two groups—namely, approaches that exploit domain knowledge about constraints between known objects and surfaces in the scene, and methods that are agnostic about the image content. The former group is typically aimed at interpreting specific scenes that have previously been seen in training, which allows them to learn spatial context among objects in the scene, and thus estimate the depth map [20, 23–25, 30, 34, 46, 49]. A thorough review and comparison of our approach with this group is beyond our scope. The latter group is related to our approach. They typically specify a depth-map model aimed at capturing the perceptual organization of image features of a wide range of previously unseen scenes. Representative approaches of this group include: (i)Contour-based 3D shape recovery [25, 34, 35] (ii)"Dead leaves" model of occlusion [15, 40]; (iii) Minimum description length of image support maps [12]; and (iv) Markov Random Fields (MRFs) [39], Layered MRFs [19, 45], and hierarchical graphical models [14]. In general, these models have intractable inference. This is usually addressed by making heuristic assumptions about the number of depth layers present, and limiting the range of spatial relationships among image features considered for perceptual grouping. These approaches typically use only qualitative evaluation, often presented only on a small set of images.

Below, we review popular low-level approaches that use T-junctions and image defocus as features. T-junctions are used in [14, 39, 40] to identify *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 [39], or a directed acyclic graph of T-junctions in which graph edges point to partially occluded scene parts [14]. However, T-junctions can provide only the orientation of depth-map gradients, whereas the gradient magnitudes have to be heuristically estimated [14]. Explicit detection of T-junctions may be avoided by directly minimizing the Mumford-Shah segmentation functional to reconstruct occluded object boundaries, and thus estimate the 2.1D sketch [15]. Defocus of certain image parts can also be used as image feature [13, 16, 29, 32, 41, 42]. Estimating the surface of exact focus in real images, however, is challenging, especially when the image is corrupted by other types of noise (e.g., motion blur).

A special case of the 2.1D sketch is the figure-ground perceptual organization of the scene. Note that our use of the term figure-ground relationship between regions concerns a distance ordering of the corresponding objects from the

camera. In the literature, the notion of figure-ground is often used in a more general sense, where objects of interest appear as foreground. It has been shown that figure-ground perception is influenced by many low-level factors [17, 18, 43, 51], including: local shape characteristics of the region boundary (e.g., convexity), its global shape characteristics (e.g., symmetry, orientation), appearance of the region itself (e.g., size, texturedness), and spatial layout of its surrounding regions. Typically, figure/ground assignment is made for each pixel along a region boundary [17], or boundary segments between two junctions [43], based on the above factors. Also, image regions can be assigned figure-ground labels, based on estimating their ordinal depths [33]. This method uses the conditional random field (CRF) spanned over image regions to estimate occluding and occluded scene parts, based on low-level properties of region shapes and T-junctions. They heuristically limit to a small number the set of image regions input to their algorithm to handle complexity.

Similar to the second group of approaches, we use low-level image features – namely, regions and their figure-ground, appearance similarity, layout, and hierarchical relationships. We address the computational issues of existing CRF/MRF based methods by formulating depth-map estimation as a convex optimization problem, and providing accuracy guaranties of the obtained solution. Our algorithm is efficient, and may easily handle a large number of multiscale image regions. A comparison with the common backtracking projection gradient descent algorithm, presented here, demonstrates many advantages of our algorithm. We do not limit the number of input regions, and show that accounting for their hierarchical nesting improves performance. We are not aware of any previous work that uses nesting of regions as a depth cue. Prior work proposed evaluation metrics which evaluate FG relationships across segmentation boundaries [17, 43]. We use image segments, generated by the gPb-OWT-UCM algorithm [6, 37], as input features to our approach. Since gPb-OWT-UCM segmentation first extracts and then closes salient image contours to form segments, our approach can be viewed as related to contour-based methods for 3D shape estimation [35]. To the best of our knowledge, this paper presents the first quantitative results of *low-level*, monocular, 2.1D sketch estimation on outdoors natural scenes datasets, including the Berkeley segmentation datasets BSD300 [37] and BSD500 [6], Stanford Make3D dataset [46], and Geometric Context Dataset [27]. Other work focus is on indoor scenes such as [30, 47] which are out of our scope. For evaluation, we extend FG annotations of the BSD300 and BSD500 datasets. Specifically, BSD300 provides FG annotations of pairs of regions in only 200 images. We extend this annotation to additional 100 test images of BSD300. In addition, we provide FG annotations for 200 new images of BSD500. Our annotations will be made public. This paper also specifies new quantitative evaluation metrics that account for differences in human judgment of the true depth orderings of surfaces in the scene.

Our preliminary approach has been published in [5], and extended here by: (i) Formulating a quadratic optimization over image regions, instead of pixels; (ii) Specifying a new closed-form dual formulation, and thus providing arbitrarily good certificates of our solutions; (iii) Deriving a new optimization-transfer algorithm; and (iv) Presenting first-ever quantitative results of monocular depth-map estimation.

3 Feature Extraction

This section presents our initial step, wherein we extract image regions and estimate their FG and AL relations.

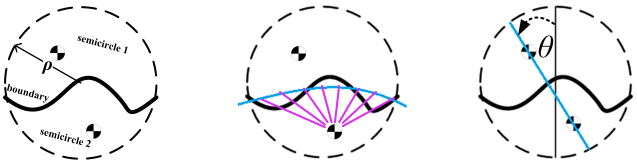
Access to image regions is provided by the state-of-the-art, multiscale segmentation algorithm gPb-OWT-UCM, presented in [6, 37]. gPb-OWT-UCM first detects image contours using a globalized probability of boundary (gPb) detector, then, computes the Oriented Watershed Transform (OWT) to close the contours, and thus produce all regions, and, finally, constructs the Ultrametric Contour Map (UCM) from region boundaries to represent the image by a strict hierarchy of regions. Region boundaries in the UCM are characterized by likelihoods of being true object boundaries. For a given threshold (scale), the UCM yields a set of closed contours with likelihoods above the threshold that partition the image. As the threshold is decreased, new closed contours are introduced in the image segmentation, partitioning the previously obtained regions into finer-scale subregions.

To relax any assumptions about object scale, we use all regions from a range of UCM scales, starting from an empirically estimated optimal minimum scale (see Sec. 10). The regions are then organized in a segmentation tree [3,21]. The root represents the entire image, nodes closer to the root correspond to large regions, and their descendant nodes represent smaller subregions. The segmentation tree encodes hierarchical (parent-child) relationships (HR) between the regions.

In the following, we explain how to estimate FG, AL, and HR relationships of regions.

3.1 Local FG Relations between Regions

For every pair of neighboring regions, we estimate their local FG relations at points regularly sampled along the shared region boundary. We closely follow the approach of [17], where the authors empirically evaluated the optimal values of input parameters, which control the estimation of FG relationships along region boundaries, on the BSD300 dataset. Specifically, we regularly sample boundary points with the step 10 boundary pixels apart. Circles with radius $\rho = 22\%$ of the boundary length are placed at each sample point. The



(a) (b) (c)

Fig. 2 Estimating the FG relationship of two neighboring regions, as in [17]: (a) A circle (dashed) with radius ρ is placed at each sample point along the shared boundary (bold black); the boundary splits the circle into two semicircles. (b) *Convexity* of a semicircle is defined as the sum of lengths of the straight lines connecting the center of mass and the parabola fitted through the boundary points. (c) *Lower region* is the angle between the vertical image axis and a line that connects the centers of mass of the two semicircles.

boundary partitions each circle into two semicircles 1 and 2, as illustrated in Fig. 2. Each sample point p is characterized by a descriptor, $\boldsymbol{x}(p)$, with the following three elements: (i) Area(p) is defined as a log ratio of the areas of the two semicircles, Area$(p) = \log \frac{\text{area}(1)}{\text{area}(2)}$; (ii) Convexity$(p)$ is defined as a log ratio of the two semicircles' convexities, Convexity$(p) = \log \frac{\text{conv}(1)}{\text{conv}(2)}$, where conv$(1)$ is a the sum of lengths of the straight lines connecting the center of mass and the parabola fitted through the boundary points; and (iii) Lower Region(p) is defined as the cosine of angle θ between the vertical image axis, and a straight line that connects the centers of mass of the two semicircles (Fig. 2c). θ is measured counter clockwise, from the vertical image axis. The same descriptor $\boldsymbol{x}(p)$ was used in [17]. As in [17, 43], we classify $\boldsymbol{x}(p)$ to identify whether the semicircle 1 is figure or ground. The logistic regression based criterion, $P(\text{figure}|\boldsymbol{x}(p)) = 1/[1 + \exp(-\boldsymbol{\varpi}^T \boldsymbol{x}(p))] \geq 0.5$, is used to declare that the semicircle 1 is figure and the semicircle 2 is ground. The model parameters $\boldsymbol{\varpi}$ are learned using iteratively reweighed least squares to maximize the joint likelihood of training data, as in [17].

3.2 Adjacent-layer (AL) and Hierarchical Relationships (HR) between Regions

Similar neighboring regions, and parent-child pairs of the segmentation tree are estimated to have the AL relationship. Region similarity is estimated as a χ^2 distance between the 300-bin histograms of codewords present in two regions. The dictionary of codewords is extracted per each image.

To this end, similar to the features used in [6]. We follow the standard approach that is evaluated in [50]. We characterize every pixel in the image with a descriptor consisting of the following 11 features: (i) Lab color values; (ii) 4 responses of the rotationally invariant, nonlinear MR8 filter bank; and (iii) 4 responses of the Laplacian of Gaussian filters. The pixel descriptors are clustered using the K-means (with $K = 300$). Pixels grouped within one cluster are labeled with a unique codeword ID of that cluster. The histograms of codeword occurrences within each region are used to estimate their similarity with neighboring regions.

Given FG, AL, and HR relations, we estimate the 2.1D sketch, such that the resulting depth ordering of regions is consistent, as explained in the following section.

Features	Explanation
FG	Figure-Ground relationships (cues).
AL	Adjacent Layers relationships (cues).
HR	Hierarchical relationships (cues).
ρ	The radius of the circle used to extract features.
p_n	A point on an edge, where features are to be extracted at point n.
$\mathbf{x}(p)$	A vector of features extracted from point p.
$\boldsymbol{\varpi}$	A vector of logistic regression model parameters.
Primal	Explanation
K	Number of regions in an image.
d_k	A column vector of depth values assigned to region k
$a_{j,n}$	An element in matrix $\mathbf{A}$ defining FG relationships.
$b_{j,m}$	An element in matrix $\mathbf{B}$ defining AL and HR relationships.
α	Exponential function input parameter for FG relations.
β	Exponential function input parameter for AL and HR relations.
γ	Regularization parameter.
ϵ_n	An lower bound on nth depth difference.
δ_n	An lower bound on mth depth difference.
$h_\epsilon(\cdot)$	Function of ϵ_n, parametrized by α.
$h_\delta(\cdot)$	Function of δ_n, parametrized by β.
$f(\cdot)$	Primal objective function.
$\tilde{f}(\cdot)$	Surrogate function for the primal objective function.
Dual	Explanation
ν, λ, ξ	Lagrangian multipliers.
$\mathbf{w}$	A vector of all the Lagrangian multipliers.
$g(\cdot)$	Dual objective function.
$\tilde{g}(\cdot)$	Surrogate function for the dual objective function.

Table 1 Notation table of concepts used in Sec. 4-9.

4 Problem Formulation

This section formulates estimating 2.1D sketch as a constrained convex optimization problem. We gradually introduce each term of the objective function and constraints, in three stages. Below, we first define necessary notation. Our notation is also summarized in Table 1.

Matrices are denoted with bold-faced, capital letters, and column vectors are denoted with bold-faced, small letters. $\mathscr{S} = \{1, 2, \ldots, K\}$ denotes an image segmentation with K regions. $\mathbf{d} = [d_1, \ldots, d_k, \ldots, d_K]^{\mathrm{T}}$ denotes a column vector of depth values assigned to all regions $k \in \mathscr{S}$. Points $n = 1, \ldots, N$ sampled along region boundaries are characterized by local FG estimates (Sec. 3.1), which are represented by a sparse matrix, $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n, \ldots, \mathbf{a}_N]_{K \times N}$. Each column vector $\mathbf{a}_n$ is defined as follows. Let $k(n)$ and $k'(n)$ denote the indices of two neighboring regions whose shared boundary contains point n, where $k(n)$ is classified as figure, and $k'(n)$ is classified as ground. Then, the nth FG estimate can be represented by $\mathbf{a}_n \in \mathbb{R}^K$ as

$$a_{jn} = \begin{cases} 0 & , \ j \neq k(n), \ j \neq k'(n), \\ -1 & , \ \text{if } j = k(n) \ , \\ 1 & , \ \text{if } j = k'(n) \ . \end{cases} \quad (1)$$

From (1), the depth difference at nth point of a region boundary can be computed as $d_{k'(n)} - d_{k(n)} = \mathbf{a}_n^{\mathrm{T}}\mathbf{d}$. In case there exists a depth map, $\mathbf{d}^*$, consistent with all FG estimates, then $\mathbf{a}_n^{\mathrm{T}}\mathbf{d}^* > 0$ for $n = 1, 2, \ldots, N$. Thus, finding $\mathbf{d}^*$, could be formulated as a linear feasibility problem:

$$-\mathbf{A}^{\mathrm{T}}\mathbf{d} \prec \mathbf{0}, \tag{2}$$

where $\prec$ denotes the element-wise $<$ relation.

Finding the 2.1D sketch as in (2), however, suffers from two drawbacks. First, (2) does not address potential errors in FG estimation. Second, (2) does not have a unique solution. If $\mathbf{d}'$ is a solution of (2), so is $\mathbf{d}'' = c \cdot \mathbf{d}'$, for all $c > 0$. Without proper regularization, a solution of the form $c \cdot \mathbf{d}$, where $c \to 0$, may introduce numerical issues. To overcome these issues, we relax the constrains in (2) as

$$\mathbf{a}_n^{\mathrm{T}}\mathbf{d} \geq \epsilon_n,\ n = 1, 2, \ldots, N, \tag{3}$$

where $\epsilon_n \in \mathbb{R}$ is an (unknown) lower bound on nth depth difference. Importantly, the relaxation in (3) allows for negative values of ϵ_n, and thus addresses incorrectly estimated FG relations. Since we expect that FG estimation error is relatively low, we seek $\mathbf{d}^*$ and $\boldsymbol{\epsilon}^* = [\epsilon_1^*, \ldots, \epsilon_n^*, \ldots \epsilon_N^*]^{\mathrm{T}}$, so as to encourage most ϵ_n^* values to be positive. This can be achieved by minimizing objective $\sum_{n=1}^{N} h_{\boldsymbol{\epsilon}}(\epsilon_n; \alpha)$, which is monotonically decreasing in ϵ_n, for $n = 1, 2, \ldots, N$. We define $h_{\boldsymbol{\epsilon}}(\epsilon_n; \alpha)$ as a convex function of ϵ_n, parameterized by an input parameter, $\alpha > 0$. Specifically, in our experiments, we use $h_{\boldsymbol{\epsilon}}(\epsilon_n; \alpha) = \exp(-\alpha\epsilon_n)$. From (3), the depth estimation can be formalized as

$$\begin{aligned} \min_{\mathbf{d}, \boldsymbol{\epsilon}}\ & \frac{\gamma}{2}\|\mathbf{d}\|_2^2 + \sum_{n=1}^{N} h_{\boldsymbol{\epsilon}}(\epsilon_n; \alpha) \\ \text{s.t.}\ & -\mathbf{A}^{\mathrm{T}}\mathbf{d} + \boldsymbol{\epsilon} \preceq \mathbf{0}, \end{aligned} \tag{4}$$

where $\preceq$ denotes the element-wise $\leq$ relation. (4) maximizes depth differences of every pair of neighboring regions, such that each figure region is closer to the camera relative to its respective ground region pair, and the resulting depth ordering of all regions is consistent. The quadratic regularization of $\mathbf{d}$ in (4), with regularization parameter γ, prevents solutions $\epsilon_n \to \infty$, $n = 1, ..., N$.

Finding the 2.1D sketch as in (4) does not explicitly address errors in image segmentation (e.g., oversegmentation), and does not take into account that object and their parts should be at near-by depths. Therefore, we extend (4) so as to account for the AL relationships between regions. Let $m = 1, 2, \ldots, M$ index all pairs of regions in $\mathscr{S}$, estimated as having the AL relationship. Similar to $\mathbf{A}$, we define matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_m, \ldots, \mathbf{b}_M]_{K \times M}$, where each $\mathbf{b}_m$ is a column vector that encodes the AL and HR relationship of the mth pair of regions, $k(m)$ and $k'(m)$, as

$$b_{jm} = \begin{cases} 0 & , \ j \neq k(m),\ j \neq k'(m), \\ -1 & , \ \text{if } j = k(m), \\ 1 & , \ \text{if } j = k'(m). \end{cases} \tag{5}$$

Small depth differences can be enforced by adding the following constraint to our optimization in (4):

$$|\mathbf{b}_m^{\mathrm{T}}\mathbf{d}| \leq \delta_m,\ \delta_m \geq 0,\ m = 1, 2, \ldots, M, \tag{6}$$

We enforce depth differences $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_m, \ldots, \delta_M]^{\mathrm{T}}$ to be close to zero. This can be achieved by minimizing objective $\sum_{m=1}^{M} h_{\boldsymbol{\delta}}(\delta_m; \beta)$, which is monotonically increasing in δ_m, for $m = 1, 2, \ldots, M$. We define $h_{\boldsymbol{\delta}}(\delta_m; \beta)$ is a convex function of δ_m, parameterized by an input parameter, $\beta > 0$. In our experiments, we specify $h_{\boldsymbol{\delta}}(\delta_m; \beta) = \exp(\beta\delta_m)$.

Adding constraint (6) to (4), we obtain the primal problem:

$$\begin{aligned} \min_{\mathbf{d}, \boldsymbol{\epsilon}, \boldsymbol{\delta}}\ & f(\mathbf{d}, \boldsymbol{\epsilon}, \boldsymbol{\delta}) = \frac{\gamma}{2}\|\mathbf{d}\|_2^2 + \sum_{n=1}^{N} h_{\boldsymbol{\epsilon}}(\epsilon_n; \alpha) + \sum_{m=1}^{M} h_{\boldsymbol{\delta}}(\delta_m; \beta) \\ \text{s.t.}\ & -\mathbf{A}^{\mathrm{T}}\mathbf{d} + \boldsymbol{\epsilon} \preceq \mathbf{0}, \quad \mathbf{B}^{\mathrm{T}}\mathbf{d} - \boldsymbol{\delta} \preceq \mathbf{0}, \quad -\mathbf{B}^{\mathrm{T}}\mathbf{d} - \boldsymbol{\delta} \preceq \mathbf{0}. \end{aligned} \tag{7}$$

Note that there is no need to explicitly constrain $\mathbf{0} \preceq \boldsymbol{\delta}$ in (7). From (7), we seek to: (i) maximize depth differences of neighboring regions so the figure is closer to the camera relative to the ground, (ii) minimize depth differences between region pairs believed to represent the same scene layer, and (iii) produce a consistent depth ordering.

The following section presents a solution of (7) which is common practice. That method will serve as a baseline for our new approach, presented in Sec 6.

5 The Unconstrained Formulation

A common practice is to rewrite (7) as an unconstrained, convex minimization problem, and, then, solve it using a gradient descent method, as follows.

First, note that the two right-hand side inequalities in (7) could be replaced with the inequality $|B^T\mathbf{d}| \preceq \boldsymbol{\delta}$. It follows that a solution of (7) must satisfy both constraints $-\mathbf{A}^{\mathrm{T}}\mathbf{d} + \boldsymbol{\epsilon} \preceq \mathbf{0}$ and $|B^T\mathbf{d}| - \boldsymbol{\delta} \preceq \mathbf{0}$ with equality. This can be proved by contradiction. Assume that a specific set of argument values $\mathbf{d}, \boldsymbol{\epsilon}, \boldsymbol{\delta}$ is a solution of (7) that does not lie on the boundary of an inequality involving ϵ_n. Then, ϵ_n can be enlarged, which will lower $h_\epsilon(\epsilon_n; \alpha)$ and thus the objective function f as well. This contradicts the assumption that the aforementioned set of argument values is a solution of (7). The proof for an inequality involving δ_m is similar.

Consequently, (7) could be rephrased as follows:

$$\min_{\mathbf{d}} \frac{\gamma}{2}\|\mathbf{d}\|_2^2 + \textstyle\sum_{n=1}^{N} h_\epsilon(\mathbf{a}_n^T\mathbf{d}; \alpha) + \sum_{m=1}^{M} h_\delta(|\mathbf{b}_m^T\mathbf{d}|; \beta). \tag{8}$$

This is an unconstrained, convex minimization problem, which could be solved using a gradient descent method.[2]

[2] Note that Newton's and other Hessian based methods would not be better alternatives to the gradient descent in our case, due to a large number of unknown variables in $\boldsymbol{d}$ (on the order of 10^2).

Specifically, to solve (8), we here use the popular gradient descent with backtracking line search [8, p. 464], whose convergence is known to be at least linear and better than that of other gradient descent methods. We apply the following gradient descent rule:

$$\mathbf{d}^{(t)} = \mathbf{d}^{(t-1)} - \sigma \cdot \frac{\partial u(\mathbf{d})}{\partial \mathbf{d}}, \;\; t = 1, 2, ... \tag{9}$$

where $\frac{\partial u(\mathbf{d})}{\partial \mathbf{d}}$ in (11) is the gradient of the unconstrained objective in (8) and step σ is optimized in every iteration t using the backtracking line search.

$$\begin{aligned}\frac{\partial u(\mathbf{d})}{\partial \mathbf{d}} &= \gamma \mathbf{d} - \alpha \sum_{n=1}^{N} \mathbf{a}_n^T \exp(-\alpha \mathbf{a}_n^T \mathbf{d}) \\ &+ \beta \sum_{m=1}^{M} \text{sign}(\mathbf{b}_m^T \mathbf{d}) \mathbf{b}_m^T \exp(\beta |\mathbf{b}_n^t \mathbf{d}|)\end{aligned} \tag{10}$$

The main advantage of the gradient descent to solving (7) is simplicity. However, as our experiments demonstrate, a number of disadvantages makes it unsuitable for solving our vision problem. One issue is that its convergence rate depends critically on the condition number of the Hessian, i.e., the ratio of the smallest eigenvalue to the largest one [8, p. 475]. In our experiments, we typically observe relatively large values of the condition number of the Hessian. It is well-known that this negatively affects the convergence rate of the gradient descent. In some cases, the gradient descent is so slow that it is very hard to identify the right stopping criterion, resulting in poor solutions. This is a fundamental problem, since the gradient descent does not provide any guarantees of closeness of its result to the global optimum.

Poor performance of the gradient descent on monocular estimation of 2.1D sketch, both in terms of accuracy and convergence rate, has motivated us to seek an alternative solution to (7). In the following section, we specify the dual formulation of (7), which allows us to derive a new optimization algorithm. Our algorithm admits a closed-form expression of the duality gap, and thus allows for explicit computation of the accuracy achieved at convergence. Our experiments demonstrate that the new algorithm consistently yields better solutions than the aforementioned gradient descent with backtracking line search.

6 The Dual Formulation

This section presents the dual formulation of (7) which has a number advantages over the unconstrained formulation. First, the dual formulation presents simpler constraints for our problem – specifically, separable non-negativity constraints. Second, in general, the dual formulation cannot always be derived in closed-form. Our approach allows a closed-form derivation of the dual, and, consequently, an explicit computation of the duality gap. This, in turn, allows us to provide arbitrarily good certificates of our solution [8].

The dual formulation can be derived from the following Lagrangian of (7):

$$\begin{aligned}\mathscr{L} = &\frac{\gamma}{2}\|\mathbf{d}\|_2^2 + \sum_{n=1}^{N} h_{\boldsymbol{\epsilon}}(\epsilon_n; \alpha) + \sum_{m=1}^{M} h_{\boldsymbol{\delta}}(\delta_m; \beta) \\ &+ \boldsymbol{\nu}^{\mathrm{T}}(-\mathbf{A}^{\mathrm{T}}\mathbf{d} + \boldsymbol{\epsilon}) + \boldsymbol{\lambda}^{\mathrm{T}}(-\mathbf{B}^{\mathrm{T}}\mathbf{d} - \boldsymbol{\delta}) + \boldsymbol{\xi}^{\mathrm{T}}(\mathbf{B}^{\mathrm{T}}\mathbf{d} - \boldsymbol{\delta}),\end{aligned} \tag{11}$$

where $\boldsymbol{\nu}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\xi}$ are vectors of non-negative Lagrange multipliers, corresponding to the inequality constrains in (7). Given (11), we reformulate our 2.1D sketch estimation as

$$\max_{\boldsymbol{\nu} \succeq \mathbf{0}, \boldsymbol{\lambda} \succeq \mathbf{0}, \boldsymbol{\xi} \succeq \mathbf{0}} \; \min_{\mathbf{d}, \boldsymbol{\epsilon}, \boldsymbol{\delta}} \; \mathscr{L}(\mathbf{d}, \boldsymbol{\epsilon}, \boldsymbol{\delta}; \boldsymbol{\nu}, \boldsymbol{\lambda}, \boldsymbol{\xi}). \tag{12}$$

Since $\mathscr{L}$ is smooth with respect to $\mathbf{d}$, $\boldsymbol{\epsilon}$, and $\boldsymbol{\delta}$, the optimal parameters, $\mathbf{d}^*$, $\boldsymbol{\epsilon}^*$, and $\boldsymbol{\delta}^*$, can be readily found from $\partial\mathscr{L}/\partial\mathbf{d} = 0$, $\partial\mathscr{L}/\partial\boldsymbol{\epsilon} = 0$, $\partial\mathscr{L}/\partial\boldsymbol{\delta} = 0$. From the definitions of the regularizing functions, $h_{\boldsymbol{\epsilon}}(\epsilon_n; \alpha) = \exp(-\alpha\epsilon_n)$, $n = 1, \ldots, N$, and $h_{\boldsymbol{\delta}}(\delta_m; \beta) = \exp(\beta\delta_m)$, $m = 1, \ldots, M$, appearing in (11), we derive:

$$\mathbf{d}^* = \frac{1}{\gamma}(\mathbf{A}\boldsymbol{\nu} + \mathbf{B}(\boldsymbol{\lambda} - \boldsymbol{\xi})), \tag{13}$$

$$\epsilon_n^* = \frac{1}{\alpha} \log \frac{\alpha}{\nu_n}, \; n = 1, 2, \ldots, N, \tag{14}$$

$$\delta_m^* = \frac{1}{\beta} \log \frac{\xi_m + \lambda_m}{\beta}, \; m = 1, 2, \ldots, M. \tag{15}$$

By substituting $\mathbf{d}^*$, $\boldsymbol{\epsilon}^*$, and $\boldsymbol{\delta}^*$ in (11), we obtain the following dual objective in terms of the Lagrange multipliers:

$$\begin{aligned}\min_{\boldsymbol{\nu}, \boldsymbol{\lambda}, \boldsymbol{\xi}} \; g(\boldsymbol{\nu}, \boldsymbol{\lambda}, \boldsymbol{\xi}) = \Big[&\frac{1}{2\gamma}\|\mathbf{A}\boldsymbol{\nu} + \mathbf{B}(\boldsymbol{\lambda} - \boldsymbol{\xi})\|_2^2 \\ &+ \left(\sum_{n=1}^{N} \frac{\nu_n}{\alpha} \log \frac{\nu_n}{\alpha} - \sum_{n=1}^{N} \frac{\nu_n}{\alpha} \right) \\ &+ \left(\sum_{m=1}^{M} \frac{\lambda_m + \xi_m}{\beta} \log \frac{\lambda_m + \xi_m}{\beta} - \sum_{m=1}^{M} \frac{\lambda_m + \xi_m}{\beta} \right) \Big] \\ \text{s.t.} \quad &\boldsymbol{\nu} \succeq \mathbf{0}, \boldsymbol{\lambda} \succeq \mathbf{0}, \boldsymbol{\xi} \succeq \mathbf{0}\end{aligned} \tag{16}$$

The dual formulation in (16) is presented as minimization, rather than the standard maximization, for maintaining the convex formulation. The dual formulation, given by (16), is solved by applying optimization transfer (OT) to the objective $g(\boldsymbol{\nu}, \boldsymbol{\lambda}, \boldsymbol{\xi})$. We first compute the surrogate functions $\tilde{g}(\boldsymbol{\nu}, \boldsymbol{\nu}^{(t)})$, $\tilde{g}(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(t)})$, and $\tilde{g}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(t)})$, then they are minimized with respect to $\boldsymbol{\nu}$, $\boldsymbol{\lambda}$, $\boldsymbol{\xi}$ as defined in the next section.

7 Optimization Transfer

There is a variety of solvers for convex optimization that we initially thought could be applied to our dual optimiza-

tion problem (16). However, in our implementation of available MATLAB solvers and cvx [11], we encountered memory and convergence issues due to the large number of constraints. This is most likely because these solvers are not able to efficiently exploit the structure of (16). Specifically, it is well-known that cvx approximates entropy terms in a given objective function. Thus, cvx approximates our two terms $\sum_n \nu_n \log(\nu_n)$ and $\sum_m (\lambda_m + \xi_m)\log(\lambda_m + \xi_m)$ in (16), resulting in convergence issues. Therefore, in this paper, we present a new optimization transfer algorithm that efficiently addresses the above issues.

Optimization transfer (OT) has been successfully used in image processing [2,52], and problems with a large number of variables [28]. The analytic tractability of OT enables convergence speed-ups, and thus large computational savings, relative to competing methods. For example, in comparison with the standard backtracking projected gradient descent method, OT avoids the computationally intensive step of selecting the backtracking step size. We proceed with a general description of OT, and its application to our problem.

In OT, instead of directly minimizing a given function $f(\mathbf{x})$ with respect to $\mathbf{x}$, one considers minimizing a surrogate function $\tilde{f}(\mathbf{x}, \mathbf{x}')$, where (i) $f(\mathbf{x}) \leq \tilde{f}(\mathbf{x}, \mathbf{x}')$ for all $\mathbf{x}$ and $\mathbf{x}'$; and (ii) $f(\mathbf{x}') = \tilde{f}(\mathbf{x}', \mathbf{x}')$ for all $\mathbf{x}'$. In this way, the generally intractable minimization of $f(\mathbf{x})$ is transferred to the following iterative minimization of the surrogate:$\mathbf{x}^{(t+1)} = \arg\min_{\mathbf{x}} \tilde{f}(\mathbf{x}; \mathbf{x}^{(t)})$. It is straightforward to show that solutions $\mathbf{x}^{(t)}$ are guaranteed to decrease the original objective $f(\mathbf{x})$ using the following sequence of relations: $f(\mathbf{x}^{(t+1)}) \leq \tilde{f}(\mathbf{x}^{(t+1)}, \mathbf{x}^{(t)}) \leq \tilde{f}(\mathbf{x}^{(t)}, \mathbf{x}^{(t)}) = \tilde{f}(\mathbf{x}^{(t)})$. The first inequality is due to property (i), the second inequality is due to minimizing the surrogate, and the last equality follows from property (ii). The main challenge in OT is to find a suitable $\tilde{f}(\mathbf{x}, \mathbf{x}^{(t)})$ that satisfies the two key requirements: (1) $\tilde{f}(\mathbf{x}, \mathbf{x}')$ can be minimized with respect to $\mathbf{x}$ in closed-form, yielding closed-form iterations; and (2) $\tilde{f}(\mathbf{x}, \mathbf{x}^{(t)})$ is a tight upper bound of $f(\mathbf{x})$ for reduced computational complexity.

We apply OT to the objective of (16) with respect to each parameter $\boldsymbol{\nu}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\xi}$, separately, while fixing the other two. This coordinate descent yields efficient update rules. In the sequel, we explain how to minimize the three surrogate functions to (16), one for each $\boldsymbol{\nu}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\xi}$. To this end, we will need the following auxiliary column vector and matrix:

$$\mathbf{w} \triangleq [\boldsymbol{\nu}^{\mathrm{T}}, \boldsymbol{\lambda}^{\mathrm{T}}, \boldsymbol{\xi}^{\mathrm{T}}]^{\mathrm{T}}, \qquad \mathbf{C} \triangleq [\mathbf{A}, \mathbf{B}, -\mathbf{B}]. \tag{17}$$

7.1 Minimization with respect to $\boldsymbol{\nu}$

We first identify the $\boldsymbol{\nu}$-dependent terms in the objective of (16), and, then, derive their surrogate function. $\tilde{g}(\boldsymbol{\nu}, \boldsymbol{\nu}^{(t)})$ denotes the surrogate of $g(\boldsymbol{\nu}, \boldsymbol{\lambda}, \boldsymbol{\xi})$ consisting of the terms that depend on $\boldsymbol{\nu}$. Since $\tilde{g}(\boldsymbol{\nu}, \boldsymbol{\nu}^{(t)})$ upper bounds $g(\boldsymbol{\nu}, \boldsymbol{\lambda}, \boldsymbol{\xi})$, we consider bounding the first and second terms in (16) to derive $\tilde{g}(\boldsymbol{\nu}, \boldsymbol{\nu}^{(t)})$.

The full derivation of $\tilde{g}(\boldsymbol{\nu}, \boldsymbol{\nu}^{(t)})$ is presented in the supplement material. As detailed in the supplement material, $\tilde{g}(\boldsymbol{\nu}, \boldsymbol{\nu}^{(t)})$ is separable over individual elements of $\boldsymbol{\nu}$, $\tilde{g}(\boldsymbol{\nu}, \boldsymbol{\nu}^{(t)}) = \sum_{n=1}^{N} \tilde{g}(\nu_n, \nu_n^{(t)})$. Consequently, the minimization of $\tilde{g}(\boldsymbol{\nu}, \boldsymbol{\nu}^{(t)})$ with respect to $\boldsymbol{\nu}$ can be solved separately using the following N convex problems: $\min_{\nu_n \geq 0}\ \tilde{g}(\nu_n, \nu_n^{(t)}), n = 1, 2, \ldots, N$. Solving each of these problems analytically gives the following update rules for ν_n, $n = 1, 2, \ldots, N$:

$$\nu_n^{(t+1)} = \nu_n^{(t)} \max\left[0, 1 - \frac{\alpha[\mathbf{w}^{(t)}]^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}\mathbf{A}\mathbf{e}_n + \gamma \log(\nu_n^{(t)}/\alpha)}{\alpha\kappa\nu_n^{(t)} + 2\gamma}\right]. \tag{18}$$

7.2 Minimization with respect to $\boldsymbol{\lambda}$

As in Sec. 7.1, we first identify the $\boldsymbol{\lambda}$-dependent terms in the objective of (16), and, then, derive their surrogate function. $\tilde{g}(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(t)})$ denotes the surrogate of $g(\boldsymbol{\nu}, \boldsymbol{\lambda}, \boldsymbol{\xi})$ consisting of the terms that depend on $\boldsymbol{\lambda}$. To derive $\tilde{g}(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(t)})$, we consider bounding the first and third terms in (16). In the supplement material we present the full derivation of $\tilde{g}(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(t)})$, and shows that it can be conveniently written as $\tilde{g}(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(t)}) = \sum_{m=1}^{M} \tilde{g}(\lambda_m, \lambda_m^{(t)})$. This separability allows us to minimize $\tilde{g}(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(t)})$ with respect to $\boldsymbol{\lambda}$ via solving the following M convex problems separately: $\min_{\lambda_m \geq 0}\ \tilde{g}(\lambda_m, \lambda_m^{(t)})$, $m = 1, 2, \ldots, M$. The analytical solution of these problems gives the following update rules for $\lambda_m^{(t)}$, $m = 1, 2, \ldots, M$:

$$\lambda_m^{(t+1)} = \max\left[0, \lambda_m^{(t)} - \frac{\beta[\mathbf{w}^{(t)}]^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}\mathbf{B}\mathbf{e}_m + \gamma \log[(\xi_m^{(t)} + \lambda_m^{(t)})/\beta]}{\beta\kappa + 4\gamma/(\xi_m^{(t)} + \lambda_m^{(t)})}\right]. \tag{19}$$

7.3 Minimization with respect to $\boldsymbol{\xi}$

Note that the $\boldsymbol{\xi}$-dependent terms in the objective of (16) differ from those depending on $\boldsymbol{\lambda}$ only by one term, which uses $-\mathbf{B}$, instead of $\mathbf{B}$. Hence, from (19), we can immediately write the update rules for elements of $\boldsymbol{\xi}$ as

$$\xi_m^{(t+1)} = \max\left[0, \xi_m^{(t)} + \frac{\beta[\mathbf{w}^{(t)}]^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}\mathbf{B}\mathbf{e}_m - \gamma \log[(\xi_m^{(t)} + \lambda_m^{(t)})/\beta]}{\beta\kappa + 4\gamma/(\xi_m^{(t)} + \lambda_m^{(t)})}\right]. \tag{20}$$

8 The Solution and Dual Convergence

Since our dual in (16) is convex, convergence of the update rules (18)–(20) to the globally optimal Lagrange multipliers $\boldsymbol{\nu}^*$, $\boldsymbol{\lambda}^*$, and $\boldsymbol{\xi}^*$ is guaranteed. Given $\boldsymbol{\nu}^*$, $\boldsymbol{\lambda}^*$, and $\boldsymbol{\xi}^*$, our

solution of the primal parameters, $\mathbf{d}^*$, $\boldsymbol{\epsilon}^*$, and $\boldsymbol{\delta}^*$, can be computed from expressions (13)–(15). This solution is optimal, because our primal problem in (7) is convex and satisfies the Slater's feasibility conditions [8, p. 226-227]. By the Slater's theorem, our duality gap converges to zero, i.e., the primal and dual objectives at convergence are equivalent, $f(\mathbf{d}^*, \boldsymbol{\epsilon}^*, \boldsymbol{\delta}^*) = g(\boldsymbol{\nu}^*, \boldsymbol{\lambda}^*, \boldsymbol{\xi}^*)$.

This is illustrated in Fig. 3, which shows our results for an image from the BSD dataset [37]. Fig. 3 plots the duality gap values, i.e., the differences $[f(\mathbf{d}^{(t)}, \boldsymbol{\epsilon}^{(t)}, \boldsymbol{\delta}^{(t)}) - [-g(\boldsymbol{\nu}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\xi}^{(t)})]]$, after iterations $t = 1{:}12000$. The figure also illustrates the corresponding depth map estimates, $\mathbf{d}^{(t)}$. Fig. 3 shows that the duality gap converges to zero, and thus the dual solution immediately provides the optimal solution for the primal. Importantly, the log-linear plots of our convergence rates in Fig. 3 are linear functions after a small number of initial iterations. This suggests that the duality gap has an exponential decay in the number of iterations. For example, in Fig. 3, after point C, there is very little difference in the resulting depth maps, as the number of iterations increases. In our experiments, we generally observe that for reducing the duality gap by an order of magnitude, we need to additionally run our algorithm only a fixed number of iterations. In addition, the plots suggest that our convergence rate is relatively insensitive to large changes of the input number of regions, and tends to become faster for fewer regions, as expected.

Note that in addition to the depth map estimate, $\mathbf{d}^*$, the other two components of our solution can be useful for higher-level algorithms (e.g., object recognition). Specifically, $\boldsymbol{\epsilon}^*$ and $\boldsymbol{\delta}^*$ can be used as cues about: (i) Confidence in the obtained solution, and (ii) Errors in the low-level segmentation, and local estimation of FG, AL, and HR relations. For example, negative ϵ_n^* (or large δ_m^*) indicates that the corresponding low-level estimate of the FG (or AL or HR) relation is not globally consistent. Consequently, the relevance of regions with such relationships for higher-level algorithms could be appropriately down-weighted. Conversely, regions with large positive values of ϵ^* could be assigned relatively high confidence, and relevance, depending on particular objectives of high-level algorithms.

9 Complexity

From (18), (19), and (20), our per-iteration complexity is dominated by two terms: $[\mathbf{w}^{(t)}]^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}\mathbf{A}\mathbf{e}_n$ and $[\mathbf{w}^{(t)}]^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}\mathbf{B}\mathbf{e}_m$, where $\mathbf{w}^{(t)}$ and $\mathbf{C}$ are given by (17). We first explain complexity of $\mathbf{C}^{\mathrm{T}}\mathbf{A}$ and $\mathbf{C}^{\mathrm{T}}\mathbf{B}$, and then state the overall complexity.

The terms $\mathbf{C}^{\mathrm{T}}\mathbf{A}$ and $\mathbf{C}^{\mathrm{T}}\mathbf{B}$ have computational complexity $O((N+M)^2)$. This is because the columns of either the $K \times N$ matrix $\mathbf{A}$, or the $K \times M$ matrix $\mathbf{B}$ contain only 2 nonzero elements per column. Note that $\mathbf{C}^{\mathrm{T}}\mathbf{A}$ and $\mathbf{C}^{\mathrm{T}}\mathbf{B}$ can be computed prior to running the OT iterations. Thus, computing $\mathbf{C}^{\mathrm{T}}\mathbf{A}$ and $\mathbf{C}^{\mathrm{T}}\mathbf{B}$ adds only a fixed complexity of $O((N+M)^2)$ to the overall complexity of the algorithm.

As $\mathbf{w}^{(t)}$ changes in every iteration, the computation of $[\mathbf{w}^{(t)}]^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}\mathbf{A}$ and $[\mathbf{w}^{(t)}]^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}\mathbf{B}$ can be done in $O((N+M)^2)$. Intuitively, when updating one of the $O(N+M)$ Lagrange multipliers, we compute values of all other $O(N+M)$ Lagrange multiplies associated with all pairwise constraints between image regions. This amounts to a computational complexity of $O((N+M)^2)$ per iteration.

From our experiments, the OT algorithm presented in Sec. 7 yields the duality gap with an exponential decay in the number of iterations. Consequently, in practice, it suffices to choose a finite number of T iterations, because they would improve the accuracy of our algorithm by T orders of magnitude. Thus, our overall computational complexity is $O(T(N+M)^2)$.

10 Results

We test different aspects of our approach through a set of variants. Each variant differs in certain steps depending on the cues used to infer the layering. We refer to Figure/Ground cues as (FG), Adjacent Layers as (AL), and Hierarchical Relations as (HR). In each variant we evaluate the effect of adding extra cues, compared to the default approach, V(FG-AL-HR), which allows us to evaluate their relative impact on performance, as described below.

Default setup – V(FG-AL-HR): In this variant we use all available cues. First, images are segmented by gPb-OWT-UCM [6], at various Pb values. We start from $\mathrm{Pb}_0 = 40\%$, and vary Pb to 100% of the maximum value 255, in increments by 10%, $\mathrm{Pb} = \mathrm{Pb}_0{:}10{:}100$ in [%]. Then, the resulting regions are organized in the segmentation tree by their size and nesting. For every pair of neighboring regions, we regularly sample points along the shared boundary at $\rho = 22\%$ of the boundary length, and estimate the polarity of FG relations associated with the sample points (Sec. 3.1). The FG classifier is learned on the training images of BSD300 dataset [37]. Then the top 50% of most similar neighboring regions are said to have the AL relations. Similarly, the top 50% of most similar parent-child regions in the segmentation tree are said to have HR relations (Sec. 3.2). Finally, the FG, AL, and HR relations are input to our depth map estimator. We quantitatively evaluate and illustrate our results on the leaf regions of the segmentation tree.

Variants of our approach: V(FG-AL) does not account for the hierarchical relations of regions. V(FG) only account for Figure ground relationships. Variants V(FG-AL) and V(FG) are obtained only for a single scale of gPb-OWT-UCM, for $\mathrm{Pb}_0 = 40\%$ of the maximum Pb value.

Baselines: We compare V(FG-AL-HR), V(FG-AL), and V(FG) with the following three baselines. The first method,

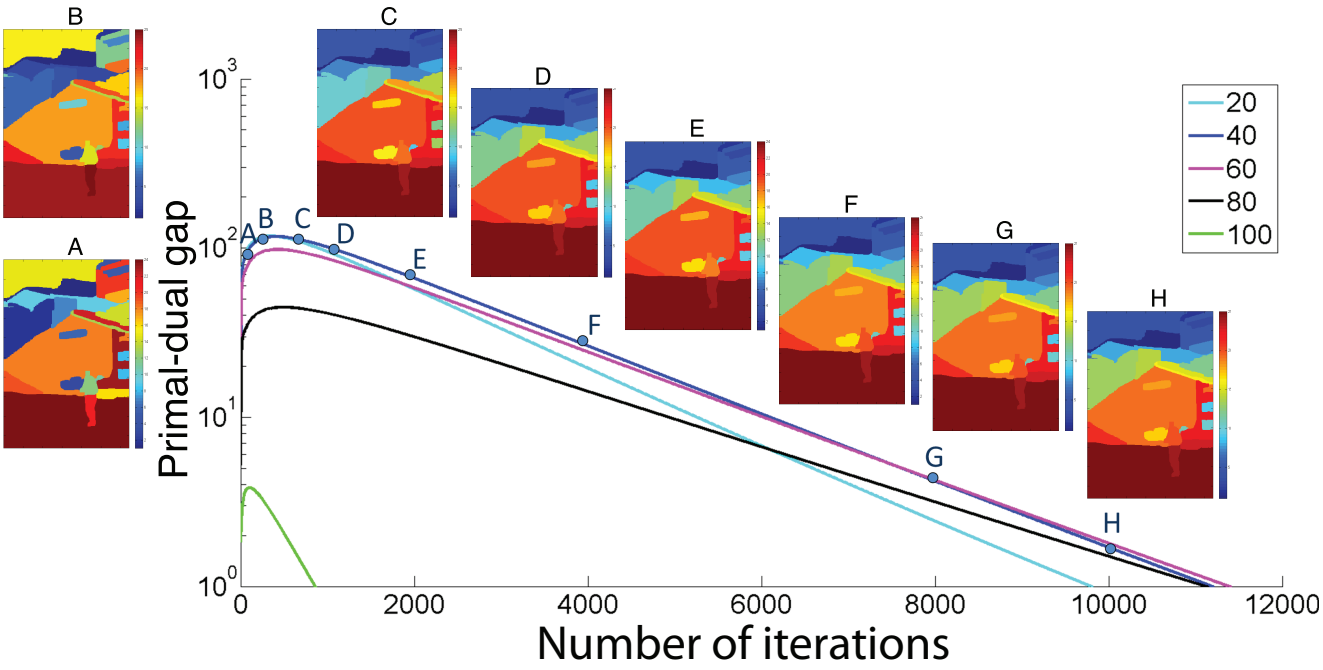


Fig. 3 The linear-log plots of convergence rates of the duality gap after $t = 1{:}12000$ iterations, for an image from the BSD dataset [37]. The letters A–H denote our color-coded depth map estimation (warmer colors are closer to the camera) after a certain number of iterations. As can be seen, there is little difference in the results from C to H, for significantly different numbers of iterations. We observe a linear relation on the log-linear plots, i.e., an exponential decay of the duality gap in the number of iterations. The different plots indicate different numbers of regions in the input image segmentation, obtained starting from different initial gPb threshold values: 20, 40,..., 100 (see Sec. 3). As can be seen, our convergence rate is relatively insensitive to large changes in the input number of regions, and becomes faster for fewer regions.

called *Random*, assigns random ordinal depths to input regions. The second method, called *Vertical*, assigns depth distances to image regions that are inversely proportional to their vertical distances from the image top. That is, Vertical assumes that the top of the image is farther away from the camera (e.g., sky) than the bottom of the image (e.g., road). The third method, called Gradient(FG-AL-HR), is the gradient descent, described in Sec. 5.

Ideal setting: We define two ideal settings, GT(FG), and GT(FG-AL)[3], where we provide ground truth information to our algorithm to estimate our upper performance bound.

Datasets: Our focus is on outdoors natural scenes. We use the following datasets for evaluation: (i) Stanford (Make3D) [46]; (ii) Geometric context dataset (GCD) [27]; and (iii) Berkeley segmentation datasets BSD300 and BSD500 [6, 37]. Other datasets such as [30, 47] are constructed using indoor scenes which are out of our scope.

Make3D consists of 400 training and 135 test images of natural and urban outdoor and indoor scenes. Continuous, absolute 3D coordinates of points in the scene are provided for every image. To generate ground truth for our evaluation, we flatten the 3D coordinates as follows. We first segment the images, and, then, average 3D coordinates of points within every segment. This immediately gives ordinal depths of the segments, as illustrated in Fig. 4. Note that our process of generating ground truth for the Make3D dataset may result in errors. The ground-truth depth layering of the scene is obtained by first segmenting the images into regions, followed by using the average laser-estimated depth of every region. Thus, error in ground truth may come from undersegmented parts of the scene.

Fig. 4 Generating ground truth for an example image from Make3D: (left to right) Continuous 3D coordinates are flattened within each region obtained from gPb-OWT-UCM to compute ordinal depths of the regions. The color codes of depths in the leftmost and rightmost images are different, since the former shows absolute depths and the latter shows estimated ordinal depths; however, FG relations should be the same.

GCD consists of 50 test and 250 train images of outdoor, natural and urban scenes. GCD provides annotations of occlusion boundaries, which we directly map to FG relations of our regions.

[3] Since we do not have ground truth annotations for the AL relations, we use the computed ones for GT(FG-AL).

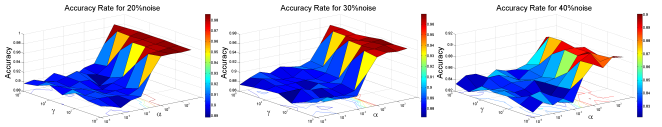


Fig. 5 FG accuracy of GT(FG-AL) on BSD300 when introducing error in the input ground-truth FG relations by randomly picking 20%, 30% and 40% of points sampled along region boundaries and reversing their ground-truth polarity. We set $\beta = 10$, and vary input parameters α and γ.

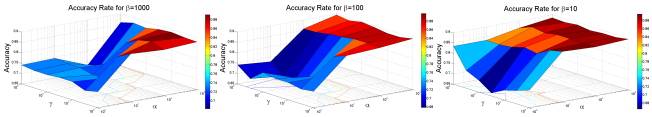


Fig. 6 FG accuracy of GT(FG-AL) on BSD300 for different values of input parameters α, β, and γ. The plots have a plateau, where our performance is relatively insensitive to variations in α, β, and γ values.

BSD300 and *BSD500* consist of natural scenes with a wide range of objects in various spatial layouts, and at a wide range of depths and scales. BSD300 has 200 training, and 100 validation images. BSD500 adds 200 new testing images to BSDS300. Ground-truth segmentations are provided for each image. However, BSD300 and BSD500 do not have annotations of ordinal depths of the ground-truth segments. Instead, BSD300 provides FG annotations of 200 images [17]. We extend this original FG annotation to another 100 images in BSD300, and to yet another 200 images in BSD500. In ambiguous cases: (i) When a FG relation gradually changes polarity along the shared boundary (e.g., the road and policeman in Fig. 1), we annotated a unique FG interpretation; and (ii) When high-level semantic cues are not sufficient to identify a FG relation, we randomly annotate one region of the two as figure. Our FG annotations of BSD300 and BSD500 will be made public.

Evaluation metrics: We use the following metrics: (i) Precision and recall of AL estimates; (ii) Hamming distance between the ground-truth and estimated depth maps; and (iii) Accuracy of FG assignments.

AL accuracy is only computed for BSD, because Make3D do not provide ground-truth segmentations. For BSD images, we say that two regions of gPb-OWT-UCM are correctly assigned the AL relationship by our depth-map estimation, if they both fall within the same ground-truth segment.

The Hamming distance serves to evaluate the consistency of our results. It is evaluated only for Make3D, because only this datasets provide the ground-truth 2.1D sketch. We compute the Hamming distance by comparing corresponding elements of two quadratic matrices, $W_{2.1D}$ and W_G. Elements of $W_{2.1D}$ are defined as $\text{sgn}(d_i - d_j)$, where d_i and d_j are the estimated ordinal depths of regions i and j. Similarly, elements of W_G are computed as $\text{sgn}(d_i^{'} - d_j^{'})$, where $d_i^{'}$ and $d_j^{'}$ are the ground-truth (flattened) depths of regions i and j. Such a region-based Hamming distance is referred to as region-based error (RBE).

The FG accuracy is computed as a percentage of correct FG estimates, associated with all points sampled along shared region boundaries as done in [17]. FG error is defined as (100 - FG accuracy)/100. For Make3D, FG accuracy is directly computed on the segmentation obtained from gPb-OWT-UCM, where the ground truth is computed by flattening the true depths within each region, as explained in the paragraph above. For BSD300 and BSD500 images, there are two different types of evaluations, one is similar to [17] where the error is computed on a subset of the points sampled along the contour. The other type as in [6, 43], where it is a computed on all points on the edges. FG accuracy is estimated so as to reduce the effect of gPb-OWT-UCM errors in segmentation. In particular, we first dilate region boundaries of the ground-truth segmentation. Then, we compute accuracy of FG estimates along only those region boundaries produced by gPb-OWT-UCM which are within a 5-pixel vicinity of the dilated ground-truth boundaries. All the results reported are for the per pixel FG error, with exception of Tab. 2 which is reported on sampled points along the contours.

10.1 Quantitative evaluation

To our knowledge, this paper presents the first quantitative evaluation of monocular 2.1D sketch estimation. Prior work uses only qualitative evaluation (e.g., [1, 12, 14, 19, 39, 40]), or evaluates only FG relationships of regions [17, 33, 43].

Empirical Parameter Estimation: GT(FG-AL) is evaluated on BSD300, to estimate our upper performance bound when the input consists of human segmentation, and ground-truth FG relations. In Tab. 5, we show that GT(FG-AL) outperforms GT(FG).

Figures 5–6 show GT(FG-AL)'s sensitivity to the specific choices of parameters α, β, and γ. For Fig. 5, we set $\beta = 10$, vary α and γ, and introduce noise in the ground-truth FG relations by reversing their polarity at randomly picked 20%, 30% and 40% of all pairs of neighboring regions. As can be seen, GT(FG-AL) achieves nearly perfect 99.1% FG accuracy, when the level of noise up to 20%. A visual inspection of these error cases shows that they are typically ill-defined, such that even human annotators make a random guess. In such cases, GT(FG-AL)'s interpretation is typically acceptable, but different from the human annotation, and thus counted as error. The choice of $\beta = 10$ is justified on Fig. 6. The figure shows GT(FG-AL)'s performance for 40% of FG noise, and various α, β, and γ values. We use 40% of FG noise, since our local FG estimation us-

Approach	Lower-Region	Area	Convexity	All 3 Features
Our local (BSD300)	65.1	66.4	64.3	72.7
classifier (BSD500)	(67.6)	(69.1)	(65.2)	(76.7)
[17] (BSD300)	64.4	67.8	60.1	74.2

Table 2 Local FG estimation, of pixels sampled along the contours of the image with distance ρ, in [%] on BSD300–200 images, and BSD500 (given in the parentheses), computed as in [17]. Due to implementation differences, our results on BSD300 slightly differ from those in [17].

Approach	Lower-Region	Area	Convexity	All 3 Features
Our local (BSD300)	62.4	63.1	60.8	70.3
classifier (BSD500)	(64.1)	(66.3)	(62.7)	(73.6)
[43] (BSD300)	–	55.6	–	–
[33] (BSD300)	61.9	–	68.4	–

Table 3 Local FG estimation, per pixel, in [%] on BSD300–200 images, and BSD500 (given in the parentheses), computed as in [33, 43]. Due to implementation differences, our results on BSD300 slightly differ from those in [33, 43].

Variants	BSD	Make3D	
	FG accur.	FG accur.	RBE
V(FG-AL-HR)–gPb	74.9 (72.1)	92.6	10.3
V(FG-AL)–gPb	73.4 (72.2)	90.2	10.1
V(FG)–gPb	71.8 (69.3)	83.9	15.7
Vertical–gPb	63.8 (59.1)	79.3	15.5
Gradient–gPb	69.2 (66.3)	84.2	12.1
V(FG-AL-HR)–Pb	71.2 (70.4)	89.3	12.8
V(FG-AL)–Pb	70.8 (71.0)	86.7	12.2
V(FG)–Pb	70.1 (68.6)	81.8	16.3
[43]–Pb	68.9	NA	NA
[33]–Pb	69.1	NA	NA
V(FG)–M	68.2 (63.6)	86.4	16.5

Table 4 FG accuracy and RBE in [%] on BSD300-200 images, BSD500 (given in the parentheses), and Make3D. For comparison with [25, 33, 43], we report their results when using Pb as input. V(FG) uses Meanshift segmentation (–M), Pb-UCM (–Pb), or gPb-OWT-UCM (–gPb) as input. Vertical and Gradient use Pb-UCM (–Pb), or gPb-OWT-UCM (–gPb).

ing a classifier at boundary points typically has an error rate of 40%. As can be seen, the plots have a plateau, where our performance is the best and relatively insensitive to parameter variations around $\alpha = 10$, $\beta = 10$, and $\gamma = 10$, used in all our experiments.

Classifiers and Features: Tab. 3 shows average accuracy of our local FG estimation on BSD300 and BSD500, and demonstrates how much our depth-map estimation improves the initial processing stage. As in [17], we extract image features: "lower-region", "convexity", and "area", from circles centered at points which are regularly sampled along region boundaries (Sec. 3.1), and test performance when using either each individual feature, or all three features jointly. In the former case, the FG classifier is a deterministic rule that declares the semicircle with a larger feature value (e.g., larger "convexity") as figure. In the latter case, we apply the logistic regression to the descriptor that consists of all three features, where the classifier is learned on training images of BSD300. Due to small implementation differences, our results on BSD300 slightly differ from those in [17].

For comparison with [33, 43], our local FG results are evaluated on the 200 images of BSD300 (100 training, and 100 test images). As can be seen in Tab. 3, the best performance is achieved when combining all three features with the logistic regression, which we use in our variants V(FG-AL-HR),V(FG-AL), and V(FG).

Tab. 4 shows our FG accuracy after the depth-map estimation by V(FG-AL-HR) and V(FG-AL) using Pb and gPb. Tab. 5 shows our FG accuracy after the depth-map estimation using human segmentation. From our local FG results in Tab. 3 and FG accuracy in Tab. 5, V(FG)-H improves performance of the logistic regression by 14.3% on BSD300, and 10.7% on BSD500. Also, from Tab. 5, replacing ground-truth FG relations, used in GT(FG), with logistic regression responses, used in V(FG)-H, downgrades V(FG)-H's performance by 10.6% on BSD300 and 10.5% on BSD500 relative to that of GT(FG).

Finest segmentation scale: Fig. 7 shows the FG error and RBE of V(FG-AL-HR) and the baselines Random and Vertical. As can be seen, the FG error and RBE change as a function of the number of regions in the input segmentation produced by gPb-OWT-UCM. The Pb threshold on region boundaries is varied, starting from Pb_0 to 100% of the maximum Pb value, in increments by 10%. As the initial threshold Pb_0 is increased, the total number of regions in the multiscale segmentation becomes smaller, and, consequently, our depth-map estimation runs faster. However, Fig. 7 shows that the fewer regions in the input segmentation, the larger the FG error and RBE of V(FG-AL-HR). Our FG accuracy decreases when Pb0 falls below 20%. We do not find that regime of segmentation interesting, and thus we do not include these results in Fig. 7. This is because the resulting oversegmentation for Pb0 ¡ 20% consists of numerous tiny superpixels (of size 3-4 pixels), whose boundaries cannot provide robust cues for our approach. Thus, we choose $Pb_0 = 40\%$ as a satisfactory trade-off between complexity and accuracy. Our Pb0=40% is higher than the PB thresholds used in [33, 43]. From Fig. 7, our FG error becomes larger as the Pb0 value increases. Therefore, our comparison with [33,43] in Tab. 4 is unfair to us. Yet, we achieve better FG accuracy. On BSD500, for certain Pb_0 values, the baseline method Random produces FG error which differs from the expected 0.5. This is because of the aforementioned "alignment" of the computed and ground-truth boundaries to reduce the effects of segmentation error, which leads to evaluating only the "aligned" subset of FG relations. V(FG-AL-HR) outperforms Vertical by 11.1% and 13% on BSD300 and BSD500 respectively.

Impact of Input Segmentations: Tab. 4 shows V(FG)'s RBE and FG accuracy for distinct input segmentations. V(FG)–

	BSD – FG accur
Vertical–H	67.6 (61.9)
Gradient–H	90.7 (93.1)
GT(FG)–H	95.2 (94.8)
GT(FG-AL)–H	97.8 (98.5)
V(FG)–H	84.6 (84.3)
[33]–H	82.8
[43]–H	78.3

Table 5 FG accuracy in [%] on BSD300-200 images, BSD500 (given in the parentheses). For comparison with [25, 33, 43], we report their results when using ideal human segmentation (–H) as input. Vertical and Gradient use human segmentation.

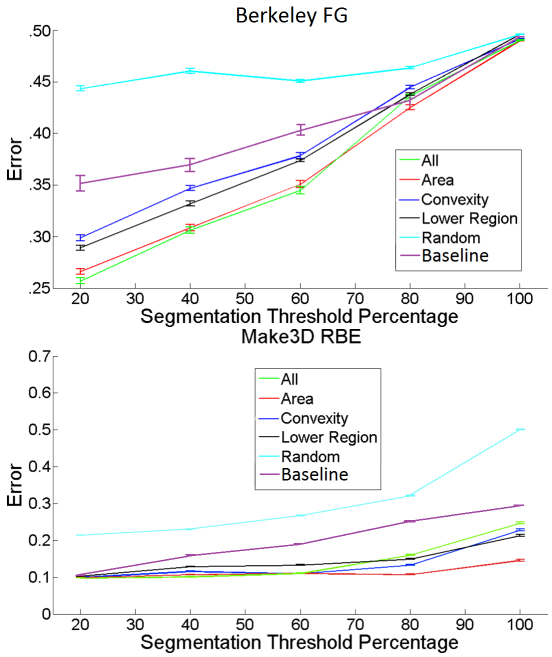


Fig. 7 The FG error and RBE of V(FG-AL-HR), and the two baselines Random and Vertical, using each individual feature "area", "convexity", and "lower-region", and "all" three features on BSD500, and Make3D. The segmentation threshold is Pb0, i.e., the finest scale of the multiscale segmentation used. As Pb_0 increases, we get fewer regions from gPb-OWT-UCM [6], and consequently worse performance (best viewed in color).

gPb uses the gPb-OWT-UCM segmentation with $Pb_0 = 40\%$ [6]. V(FG)–M uses the Meanshift segmentation [10] as shown in Fig. 14. For fair comparison of V(FG)–gPb and V(FG)–M, we empirically searched for the optimal combination of Meanshift's three input parameters: feature bandwidth b_f, spatial bandwidth b_s, and minimum region area $S_{\min}$. Specifically, we varied these parameters as $b_f = 5.5{:}0.5{:}8.5$, $b_s = 4{:}2{:}10$, and $S_{\min} = 100{:}200{:}900$, and reported in Tab. 4 the best results on a given dataset. Tab. 4 shows that V(FG)–gPb outperforms V(FG)–M on all datasets. This is typically because Meanshift generally produces oversegmentation. As a result, V(FG)–M usually infers a larger number of depth layers than the ground truth. Therefore, we use gPb-OWT-UCM for V(FG-AL-HR).

Impact of Hierarchy: Tab. 4 shows that accounting for hierarchical region relationships in V(FG-AL-HR) improves performance relative to that of V(FG) and V(FG-AL) on all the datasets. As can be seen in Tab. 4, V(FG-AL-HR) outperforms the baseline Vertical–H which uses the human segmentation as input.

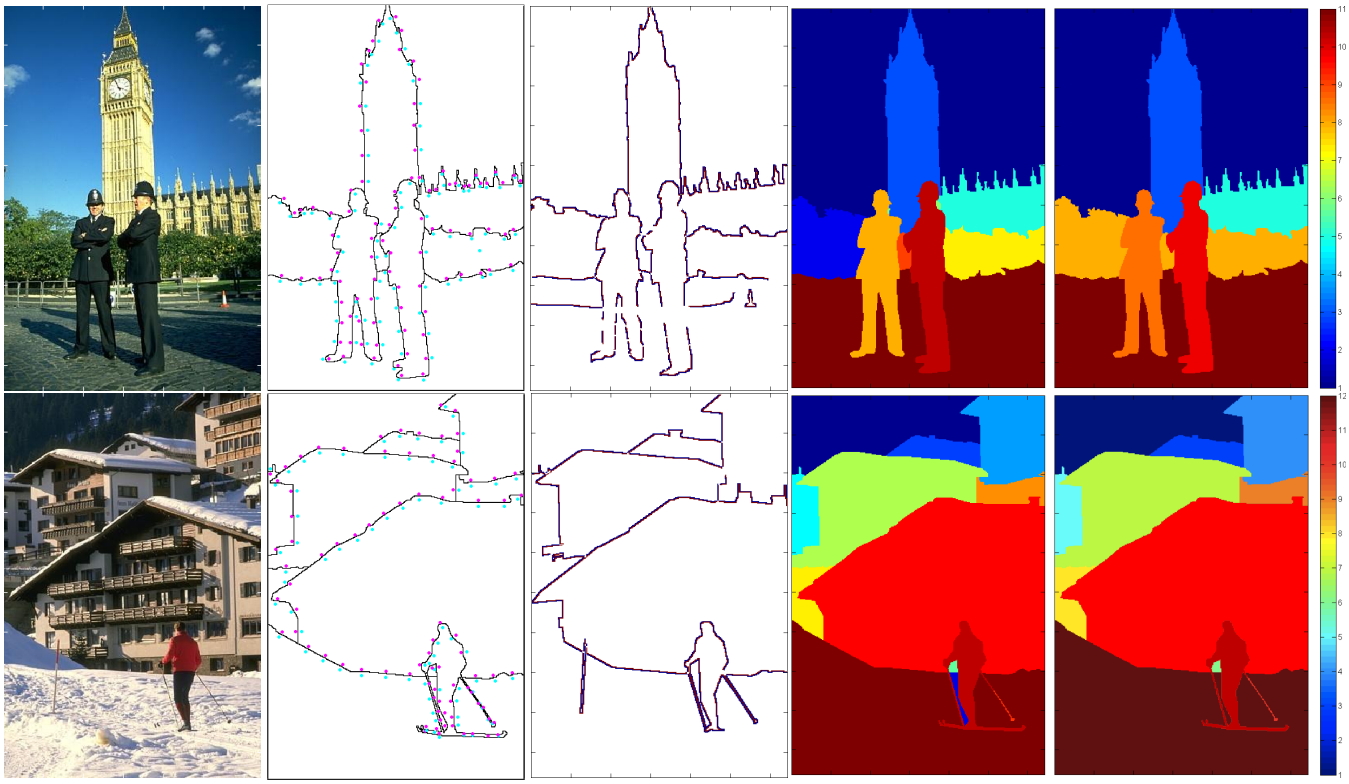
Comparison with Gradient Descent: Tab. 4 shows V(FG-AL-HR) outperforms Gradient, for the same input regions and their relationships, on all three datasets. Convergence of Gradient is declared when the objective function of (8) has not changed for 100 consecutive iterations, usually considered appropriate in practice. However, on our datasets, Gradient often fails to find a good (local) optimum when it meets the stopping criterion, resulting in worse depth-map estimates than V(FG-AL-HR). Gradient also incurs a larger running time than V(FG-AL-HR), due to the costly backtracking line search, as illustrated in Fig. 13, whereas our algorithm does not "waste" computational resources on parameter fine-tuning.

Comparison: On the GCD dataset, V(FG-AL-HR) yields FG accuracy of 71.4%. We compare V(FG-AL-HR) with the state-of-the-art higher-level approach of [25], where occlusion boundaries are estimated in a single image. The main difference from V(FG-AL-HR) is that [25] uses training examples of different types of surfaces to learn *typical surface layouts* (e.g., the sky and ground are always on the top and bottom of the image), and a number of additional image features, including 3D geometric context cues. The additional domain knowledge that they incorporate in their conditional random field model (CRF) for labeling watershed contours in the image gives FG accuracy of 79.9% with retraining the CRF on GCD [25]. Their advantage of 8.5% comes at the price of larger running times measured in minutes per image [25], and inability to address atypical scene layouts (e.g., like in BSD500) which have not been previously seen in training. By contrast, V(FG-AL-HR) is a low-level, generic approach capable of addressing atypical, previously unseen surface layouts, and runs in less than 5 seconds on GCD images.

Running Time: For BSD images with 480×320 pixels, gPb-OWT-UCM extracts multiscale regions, whose number is on the order of 10^2. In our MATLAB implementation, V(FG-AL-HR) takes about 5 seconds per image on a 2.66GHz, 3.49GB RAM PC (excluding the segmentation time).

10.2 Qualitative evaluation

Figures 8–12 show examples of our results. As can be seen, V(FG-AL-HR) successfully estimates the ordinal depths of image regions. The captions of the figures explain in more detail some key properties of our approach. For example,



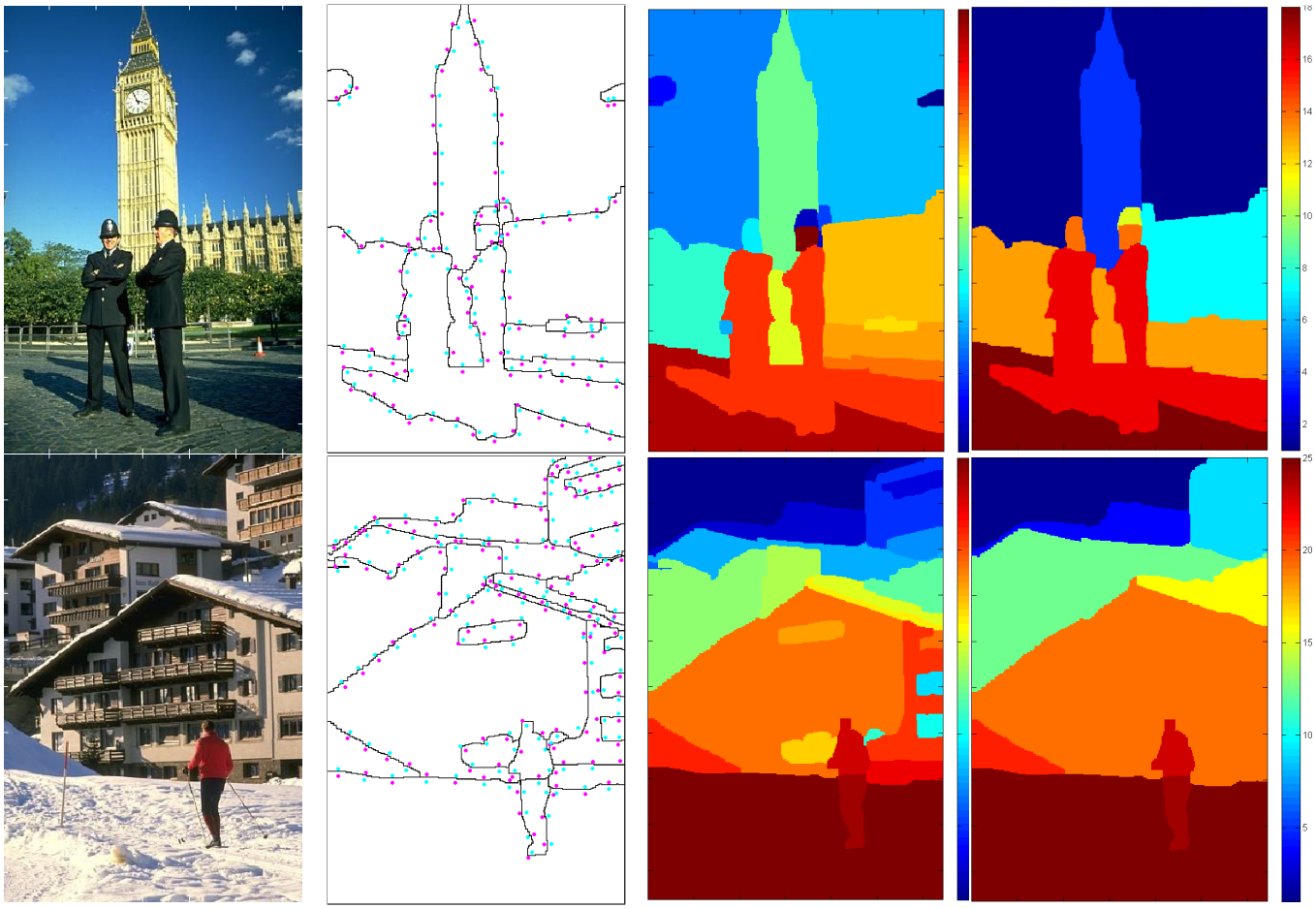
(a) Input images (b) Human segmentation (c) Ground truth FG (d) V(FG)-H (e) V(FG-AL)-H

Fig. 8 Example images from BSD300 [37], and their depth map estimates (depth layers closer to the camera are color-coded with "warmer" colors; the colors are normalized and distributed evenly over the estimated depth range for each image): (b) Local FG estimates of the logistic regression at points along region boundaries in the human segmentation (cyan marks foreground, and magenta indicates background); (c) Ground truth labeling of edges with FG labels. (d) V(FG)-H's depth map estimate; (e) V(FG)-H's estimation significantly improves by accounting for the AL relationships (see the merged trees behind the cops into a single layer; also the elongated snow region between the skier and the stick is placed at the correct depth layer with the rest of the snow area in the image);

in Fig. 8, we show cases when accounting for hierarchical relationships between region-subregion pairs in V(FG-AL-HR) improves performance relative to that of V(FG-AL). Fig. 10 shows that the baseline method Vertical fails in images where objects violate the assumption that their ordinal depths are inversely proportional to their distances from the image top, whereas V(FG-AL-HR) successfully handles these cases. Also, Fig. 11 shows examples when accounting for the AL relationships between regions in V(FG-AL-HR) improves V(FG)'s performance. As can be seen, V(FG-AL-HR) handles oversegmentation well by merging regions which are estimated to have the AL relationship. However, in the case when important boundaries are not detected in the image by a low-level segmenter, our optimization framework does not have necessary constraints to infer depth. Fig. 11 also qualitatively compares the result of V(FG-AL-HR) with that of the higher-level approach of [25]. V(FG-AL-HR) is not aware of any particular objects and typical scene layouts, and yields a similar depth map to that of [25]. Fig. 13 illustrates a comparison of V(FG-AL-HR) with Gradient. As can be seen, Gradient takes a longer running time to stop at a worse local optimum than V(FG-AL-HR). Finally, Fig. 12 illustrates our results on an example image from Make3D dataset.

11 Conclusion

We have presented a new approach to monocular extraction of the 2.1D sketch that does not have access to domain knowledge about typical object occurrences and scene layouts. It takes as input a multiscale image segmentation, and relationships between image regions – namely, local estimates of figure-ground along region boundaries, appearance similarity between neighboring regions, and parent-child relationships. Given this input, ordinal depths of the regions are found by optimization transfer of a convex optimization problem. A new optimization transfer algorithm has been derived. The algorithm provides explicit guarantees of solution accuracy. Its complexity is quadratic in the number of constraints, which makes it generally well-suited for convex problems with a large number of variables. We have empirically observed that the convergence rate of our algorithm is close to exponential.



(a) Input images (b) gPb-OWT-UCM (c) V(FG-AL)-gPb (d) V(FG-AL-HR)-gPb

Fig. 9 Example images from BSD300 [37], and their depth map estimates (depth layers closer to the camera are color-coded with "warmer" colors; the colors are normalized and distributed evenly over the estimated depth range for each image): (b) The finest segmentation scale of gPb-OWT-UCM [6] for the initial threshold of $\mathrm{Pb}_0 = 40\%$, and local FG estimates of the logistic regression; (c) V(FG-AL)'s depth map estimate; (d) V(FG-AL-HR)'s depth map estimate improves over that of V(FG-AL), as V(FG-AL-HR) additionally accounts for hierarchical relations between region-subregion pairs. The depth maps of V(FG-AL-HR) and V(FG)-H may differ in certain ordinal depths (i.e., color codes), since V(FG-AL-HR) and V(FG)-H use different segmentations. But both V(FG-AL-HR) and V(FG)-H identify consistent depth orderings of their input regions.

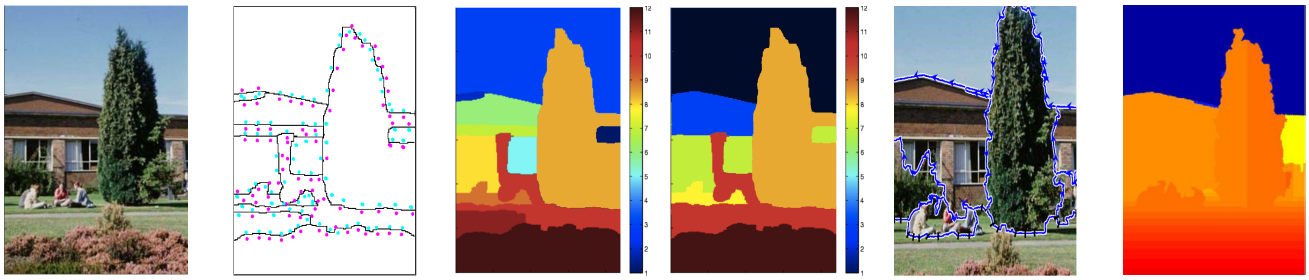
To our knowledge, we have presented the first quantitative evaluation of monocular 2.1D sketch estimation. For evaluation, we have mapped the ground-truth continuous, 3D coordinates of scenes in Stanford Make 3D dataset (Make3D) [46] to discrete ordinal depth maps. This discretized ground truth has allowed for computing a new metric of our performance — region-based Hamming distance. We have also evaluated our figure-ground assignments to pairs of neighboring regions on Geometric context dataset (GCD) [27], and Berkeley segmentation datasets (BSD300 and BSD500) [6, 37]. The results demonstrate that our algorithm successfully estimates the ordinal depths of input image regions corresponding to unique depth layers in the scene. Also, the results show that our algorithm is relatively insensitive to a specific choice of input parameters.

We have empirically observed that accounting for hierarchical relationships between regions, and appearance similarity between neighboring regions generally improves performance. This improvement is typically in terms of placing oversegmented image regions onto the same ordinal depth in the 2.1D sketch. Failures occur typically due to errors in the input low-level segmentation. Specifically, when important boundaries are not detected in the image by a low-level segmenter, our optimization framework does not have necessary constraints to infer depth changes in the scene.

We have also considered a simpler approach to solving the original convex optimization problem using the gradient descent with backtracking line search. In our experiments, the gradient descent consistently takes a longer running time to stop at a worse local optimum, yielding worse depth-map estimates, than our approach.

(a) Input image (b) gPb-OWT-UCM (c) V(FG-AL)-gPb (d) V(FG-AL-HR)-gPb (e) Vertical-gPb

Fig. 10 An example image from BSD 300 [37]: see the caption for Fig. 8. (b) The finest-scale segmentation of gPb-OWT-UCM [6] for $\text{Pb}_0 = 40\%$, and local FG estimates of the logistic regression (foreground and ground are marked cyan and magenta); (c) V(FG-AL)'s depth map estimate; (d) V(FG-AL-HR) outperforms V(FG-AL) by merging oversegmented regions; (e) Vertical on gPb-OWT-UCM fails, due to the presence of objects that violate the assumption that their ordinal depth is inversely proportional to their distance from the image top; our approach successfully handles these cases.



(a) Input image (b) gPb-OWT-UCM (c) V(FG) (d) V(FG-AL-HR) (e) Hoiem et al. [25] (f) Depth map of [25]

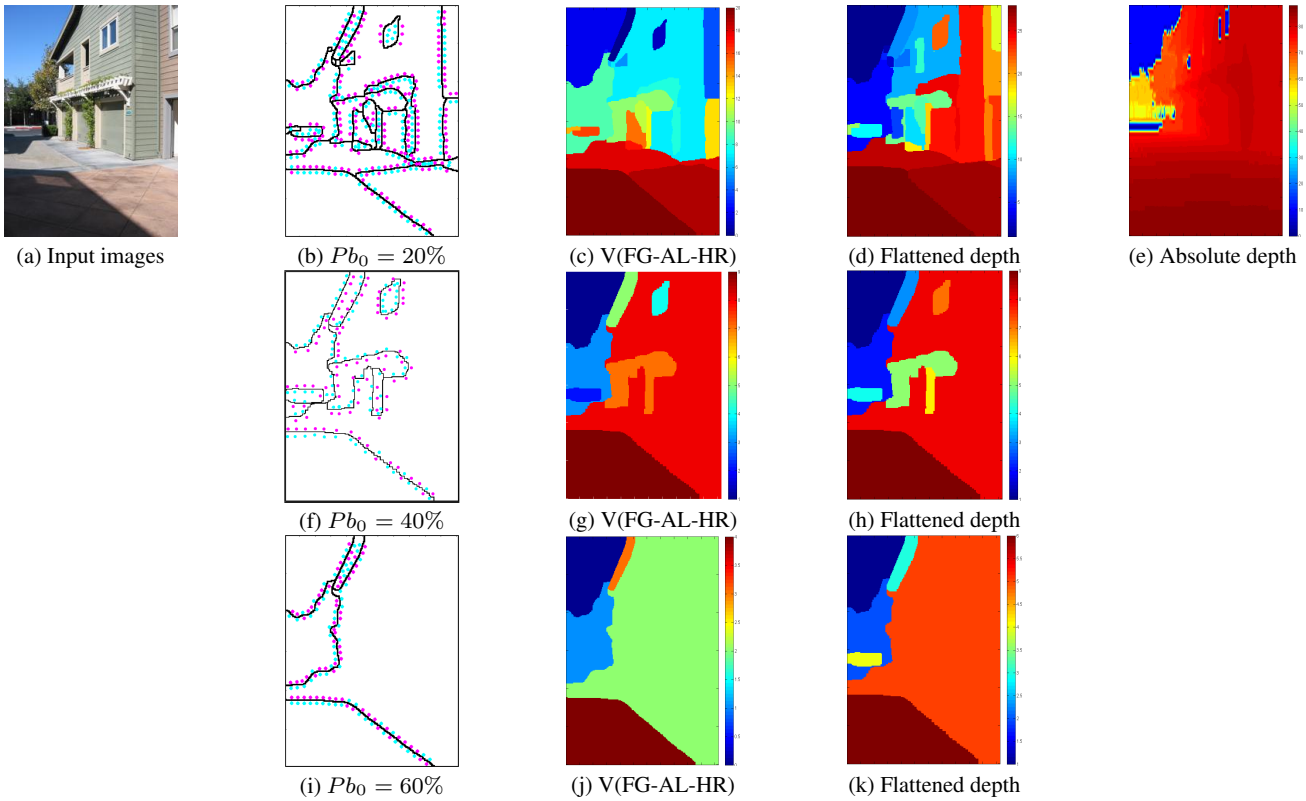
Fig. 11 An example image from GCD: (b)–(d) see the caption of Fig. 8. V(FG-AL-HR) improves upon V(FG) by accounting for the AL relationships (e.g., V(FG-AL-HR) places the windows of the house at the same depth layer). V(FG-AL-HR) is limited by the segmentation errors. (e)–(f) The results of the high-level CRF-based approach, presented in [25]. While V(FG-AL-HR) is not aware of any particular objects and typical scene layouts, it accurately yields a consistent depth map, which is nearly the same as that in the results of the high-level approach of [25].

Acknowledgment

This work was supported in part by grants NSF RI 1302700 and DARPA MSEE FA 8650-11-1-7149.

References

1. Adelson, E.H.: Layered representation for vision and video. In: IEEE W. Representation of Visual Scenes (1995)
2. Afonso, M., Bioucas-Dias, J., Figueiredo, M.: Fast image recovery using variable splitting and constrained optimization. Image Processing, IEEE Transactions on **19**(9), 2345–2356 (2010)
3. Ahuja, N., Todorovic, S.: Connected segmentation tree – a joint representation of region layout and hierarchy. In: CVPR (2008)
4. Alon, N.: Ranking tournaments. SIAM J. Discrete Math. **20**, 137–142 (2006)
5. Amer, M., Raich, R., Todorovic, S.: Monocular extraction of 2.1D sketch. In: ICIP (2010)
6. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE TPAMI **33**, 898–916 (2011)
7. Bar-Noy, A., Naor, J.: Sorting, minimal feedback sets, and hamilton paths in tournaments. SIAM J. Discrete Math. **3**(1), 7–20 (1990)
8. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge Univ. Press (2004)
9. Charbit, P., Thomasse, S., Yeo, A.: The minimum feedback arc set problem is np-hard for tournaments. Combinatorics, Probability & Computing **16**, 1–4 (2007)
10. Comaniciu, D., Meer, P.: Meanshift: a robust approach toward feature space analysis. IEEE TPAMI **24**(5), 603–619 (2002)
11. `http://cvxr.com/cvx/`
12. Darrell, T., Pentland, A.: Cooperative robust estimation using layers of support. IEEE TPAMI **17**(5), 474–487 (1995)
13. Darrell, T., Wohn, K.: Pyramid based depth from focus. In: CVPR, pp. 504 –509 (1988)
14. Dimiccoli, M., Salembier, P.: Hierarchical region-based representation for segmentation and filtering with depth in single images. In: ICIP (2009)
15. Esedoglu, S., March, R.: Segmentation with depth but without detecting junctions. Journal of Mathematical Imaging and Vision **18**, 7–15 (2003)
16. Favaro, P., Soatto, S., Burger, M., Osher, S.: Shape from defocus via diffusion. IEEE TPAMI **30**(3), 518 –531 (2008)
17. Fowlkes, C.C., Martin, D.R., Malik, J.: Local figure-ground cues are valid for natural images. J Vision **7**(8), 2 (2007)
18. Fragkiadaki, K., Shi, J.: Figure-ground image segmentation helps weakly-supervised learning of objects. In: ECCV (2010)
19. Gao, R., Wu, T., Zhu, S., Sang, N.: Bayesian Inference for Layer Representation with Mixed Markov Random Field. In: EMMCVPR (2007)
20. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV, pp. 1–8 (2009)
21. Gu, C., Lim, J.J., Arbelaez, P., Malik, J.: Recognition using regions. In: CVPR (2009)
22. Guo, C., Zhu, S., Wu, Y.: Primal sketch: Integrating texture and structure. Computer Vision and Image Understanding (2007)
23. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. In: CVPR (2009)
24. Hoiem, D., Efros, A., Hebert, M.: Closing the loop in scene interpretation. In: CVPR (2008)
25. Hoiem, D., Efros, A., Hebert, M.: Recovering occlusion boundaries from an image. IJCV **91**(3), 328–346 (2010)
26. Hoiem, D., Efros, A.A., Hebert, M.: Automatic photo pop-up. ACM Trans. Graph. **24**(3), 577–584 (2005)



(a) Input images (b) $Pb_0 = 20\%$ (c) V(FG-AL-HR) (d) Flattened depth (e) Absolute depth

(f) $Pb_0 = 40\%$ (g) V(FG-AL-HR) (h) Flattened depth

(i) $Pb_0 = 60\%$ (j) V(FG-AL-HR) (k) Flattened depth

Fig. 12 Make3D: see the caption of Fig. 8. The segmentations are obtained using gPb-OWT-UCM [6], for different initial thresholds on region boundaries $Pb_0 = 20, 40, 60\%$. V(FG-AL-HR) uses the input segmentation shown to its left. V(FG-AL-HR) successfully places two regions that correspond to the two separate sky parts (bottom row) onto the same depth layer, and thus handles partial occlusion.

27. Hoiem, D., Efros, A.A., Hebert, M.: Geometric context from a single image. In: ICCV, pp. 654–661 (2005)
28. Hunter, D., Lange, K.: A Tutorial on MM Algorithms. The American Statistician **58**(1), 30–38 (2004)
29. Hwang, T., Clark, J., Yuille, A.: A depth recovery algorithm using defocus information. In: CVPR, pp. 476–482 (1989)
30. Jia, Z., Gallagher, A., Chang, Y.J., Chen, T.: A learning based framework for depth ordering" ieee conference on computer vision and pattern recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)
31. Kenyon-Mathieu, C., Schudy, W.: How to rank with few errors. In: ACM symposium on Theory of Computing (STOC), pp. 95–103 (2007)
32. Krishnan, A., Ahuja, N.: Range estimation from focus using a non-frontal imaging camera. IJCV **20**(3), 169–186 (1996)
33. Leichter, I., Lindenbaum, M.: Boundary ownership by lifting to 2.1d. In: ICCV, pp. 9–16 (2009)
34. Liu, B., Gould, S., Koller, D.: Single image depth estimation from predicted semantic labels. In: CVPR (2010)
35. Malik, J., Maydan, D.: Recovering three dimensional shape from a single image of curved objects. TPAMI **11**(6), 555–566 (1989)
36. Marr, D.: Visual information processing: the structure and creation of visual representations. IJCAI (1979)
37. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV (2001)
38. Mathieu, C., Schudy, W.: How to rank with fewer errors: A PTAS for feedback arc set in tournaments. Journal of the ACM pp. 95–103 (2011)
39. Morel, J., Salembier, P.: Monocular depth by nonlinear diffusion. In: ICVGIP, pp. 95–102 (2008)
40. Nitzberg, M., Mumford, D.: The 2.1-D sketch. In: ICCV, pp. 138–144 (1990)
41. Pentland, A.P.: A new sense for depth of field. IEEE TPAMI **9**, 523–531 (1987)
42. Rajagopalan, A., Chaudhuri, S.: A variational approach to recovering depth from defocused images. IEEE TPAMI **19**(10), 1158–1164 (1997)
43. Ren, X., Fowlkes, C.C., Malik, J.: Figure/ground assignment in natural images. In: ECCV, pp. 614–627 (2006)
44. Roy-Chowdhury, A.K., Chellappa, R.: Statistical bias in 3d reconstruction from a monocular video. TIP pp. 1057–1062 (2005)
45. Saund, E.: Perceptual organization of occluding contours generated by opaque surfaces. In: CVPR, pp. II: 624–630 (1999)
46. Saxena, A., Sun, M., Ng, A.Y.: Make3D: Learning 3D scene structure from a single still image. IEEE TPAMI **31**(5), 824–840 (2009)
47. Scharstein, D., Pal, C.: Learning conditional random fields for stereo. In: CVPR (2007)
48. Stamm, H.: On feedback problems in planar digraphs. Springer Berlin / Heidelberg (1991)
49. Sun, M., Bradski, G.R., Xu, B.X., Savarese, S.: Depth-encoded Hough voting for joint object detection and shape recovery. In: ECCV, pp. 658–671 (2010)
50. Varma, M., Garg, R.: Locally invariant fractal features for statistical texture classification. In: ICCV (2007)
51. Vecera, S.P., Vogel, E.K., Woodman, G.F.: Lower region: A new cue for figure-ground assignment. Journal of Experimental Psychology: General **131**, 194–205 (2002)
52. Wright, S., Nowak, R., Figueiredo, M.: Sparse reconstruction by separable approximation. Signal Processing, IEEE Transactions on **57**(7), 2479–2493 (2009)

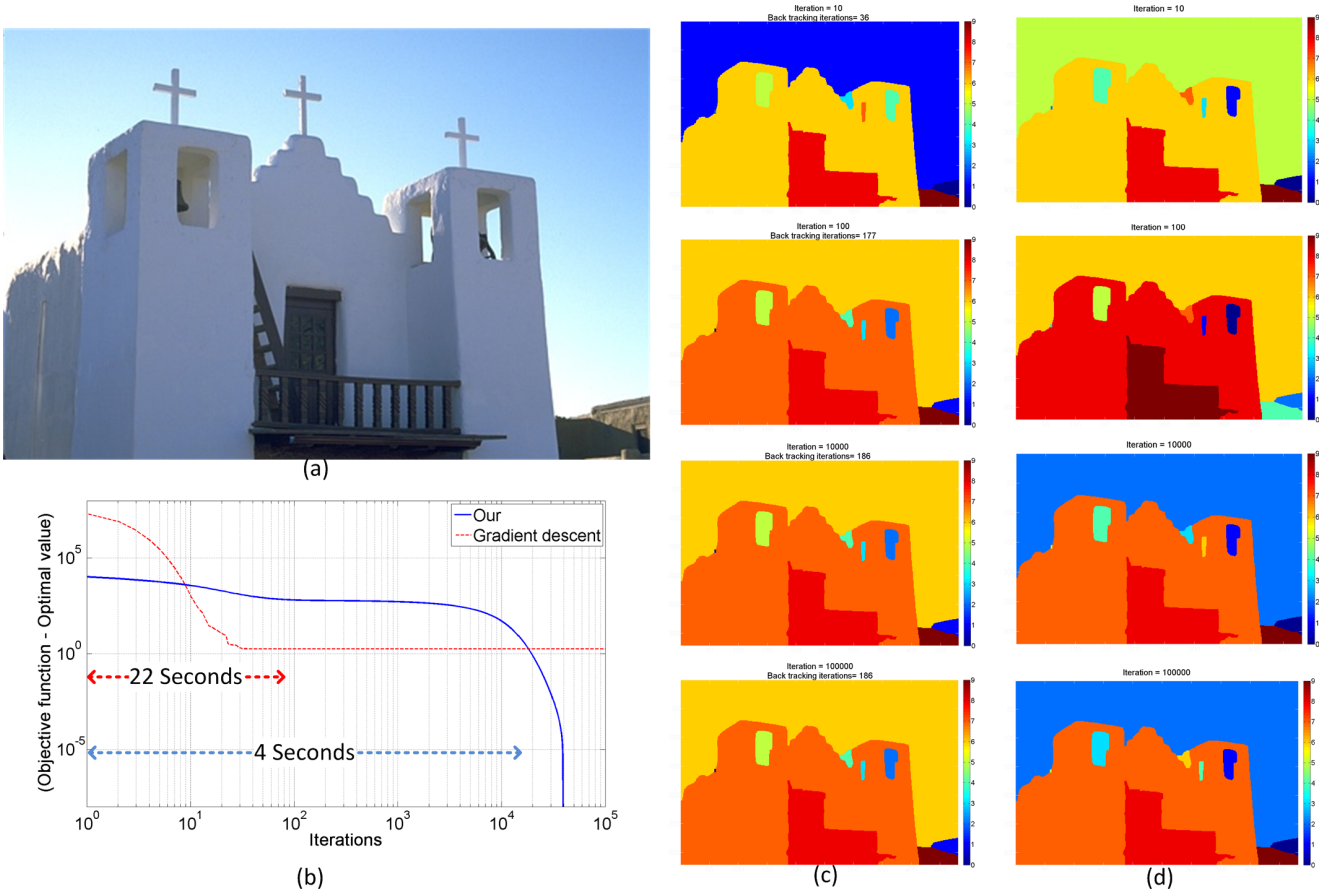


Fig. 13 (a) Input image. (b) Iterative minimization of the objectives in (7) by V(FG-AL-HR), and (8) by Gradient. Gradient meets the stopping criterion early, and yields a worse local optimum than V(FG-AL-HR). Also, due to backtracking line search, Gradient takes longer to reach the stopping criterion. (c) and (d) Depth-map estimates after a specified number of iterations by Gradient and V(FG-AL-HR).

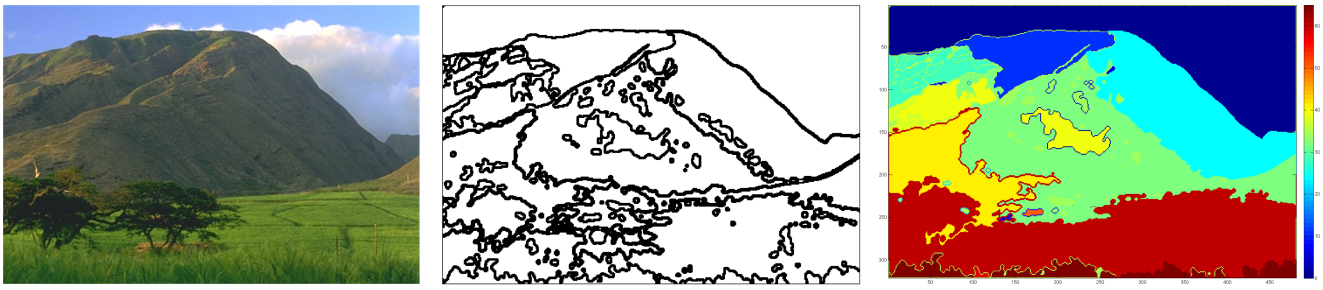


Fig. 14 (a) Input image. (b) Segmentation using the meanshift algorithm with $S_{min} = 300$ (c) The result of V(FG-AL). Some small regions got merged with bigger regions in the depth-map estimate by V(FG-AL) due to accounting for similarity between regions.