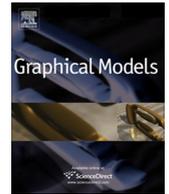




ELSEVIER

Contents lists available at ScienceDirect

Graphical Models

journal homepage: www.elsevier.com/locate/gmod

A global energy optimization framework for 2.1D sketch extraction from monocular images

Cheng-Chi Yu ^a, Yong-Jin Liu ^{a,*}, Matt Tianfu Wu ^b, Kai-Yun Li ^c, Xiaolan Fu ^c^aTNList, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China^bCenter for Vision, Cognition, Learning and Art, Department of Statistics, University of California, Los Angeles (UCLA), USA^cState Key Lab of Brain and Cognitive Science, Institute of Psychology, Chinese Academy of Sciences, Beijing 100101, China

ARTICLE INFO

Article history:

Received 28 February 2014

Received in revised form 26 March 2014

Accepted 28 March 2014

Available online 13 April 2014

Keywords:

2.1D sketch

Global optimization

Local features

Hybrid differential evolution

ABSTRACT

The 2.1D sketch is a layered image representation, which assigns a partial depth ordering of over-segmented regions in a monocular image. This paper presents a global optimization framework for inferring the 2.1D sketch from a monocular image. Our method only uses over-segmented image regions (i.e., superpixels) as input, without any information of objects in the image, since (1) segmenting objects in images is a difficult problem on its own and (2) the objective of our proposed method is to be generic as an initial module useful for downstream high-level vision tasks. This paper formulates the inference of the 2.1D sketch using a global energy optimization framework. The proposed energy function consists of two components: (1) one is defined based on the local partial ordering relations (i.e., figure-ground) between two adjacent over-segmented regions, which captures the marginal information of the global partial depth ordering and (2) the other is defined based on the same depth layer relations among all the over-segmented regions, which groups regions of the same object to account for the over-segmentation issues. A hybrid evolution algorithm is utilized to minimize the global energy function efficiently. In experiments, we evaluated our method on a test data set containing 100 diverse real images from Berkeley segmentation data set (BSDS500) with the annotated ground truth. Experimental results show that our method can infer the 2.1D sketch with high accuracy.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

The 2.1D sketch is a decomposition of an image domain into overlapping regions ordered by occlusion. An illustrative example is shown in Fig. 1, in which the 2.1D sketch of a domain D is a set of four overlapped regions R_1, R_2, R_3, R_4 , satisfying $R_1 \cup R_2 \cup R_3 \cup R_4 = D$, and there is a partial depth ordering \succ on these regions, $R_1 \succ R_2 \succ R_3 \succ R_4$, where $R_i \succ R_j$ means R_i is in front of R_j . The 2.1D sketch was first proposed by Nitzberg et al. [1,2], and was also called layered image representation by Adelson [3–5] for image

coding and motion analysis. The goal of the 2.1D sketch is, on one hand, to recover the partial layer ordering of regions in an input image, and on the other hand, to keep multiple reasonable and distinct solutions accounting for the intrinsic ambiguity caused by occlusion. Solving the 2.1D sketch is a critical step for scene understanding and other high-level vision tasks in both still image and video, such as foreground/background separation, depth estimation from a single image (also called 2.5D sketch), image and video coding, motion analysis, etc. It remains a very challenging problem in the vision literature.

This paper presents a global energy optimization method for inferring the 2.1D sketch from a monocular real image. Fig. 2 shows a real example. The 2D image regions

* Corresponding author.

E-mail address: liuyongjin@tsinghua.edu.cn (Y.-J. Liu).

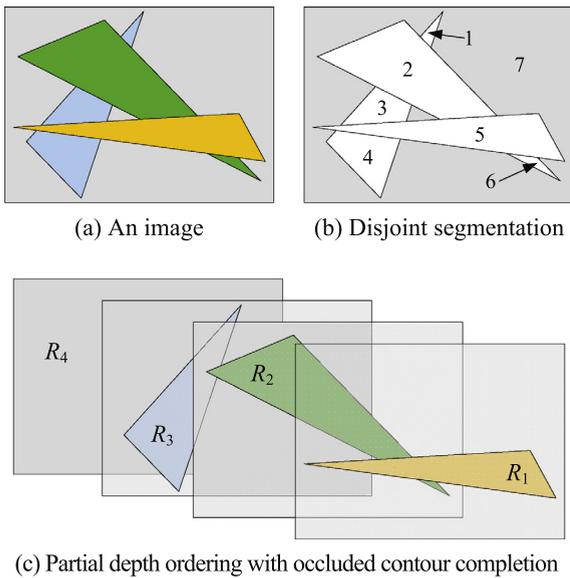


Fig. 1. 2.1D sketch of an image domain.

in Fig. 2(b) are computed by some off-the-shelf multi-scale low-level segmentation algorithms (e.g., the Berkeley multi-scale segmentation algorithm, called gPb-owt-ucm in [6], is used in our experiments). The reason that we adopt the over-segmented 2D image regions as input is to

recover the underlying latent objects in the 2.1D sketch. In this paper, we focus on inferring the partial depth ordering of objects (Figs. 2(c) and (d)) using the low-level segmented image regions as input (i.e., without resorting to the object information). We do not address the occluded contour completion problem. Meanwhile, we formulate the inference problem under an global energy optimization framework and therefore do not account for the multiple plausible solutions of the 2.1D sketch.

Inferring the partial depth ordering of a set of image regions belongs to the well-known tournament problem [7] which is known to be a NP-hard and APX-hard problem in general. In the tournament problem, the objective is to rank n elements (i.e., the image regions in this paper) using some transitive properties (i.e., the depth of image regions from the observer) and some successive pairwise comparisons (i.e., the marginal information of the globally consistent partial depth ordering). To address this issue, in this paper we propose a global energy optimization framework to model the 2.1D sketch, which leads to a nonlinear optimization in inference, and we utilize a hybrid evolution algorithm to solve it.

2. Related work

In the vision literature, generally speaking, the 2.1D sketch problem or layered representation has been discussed from a variety of perspectives, such as line drawing

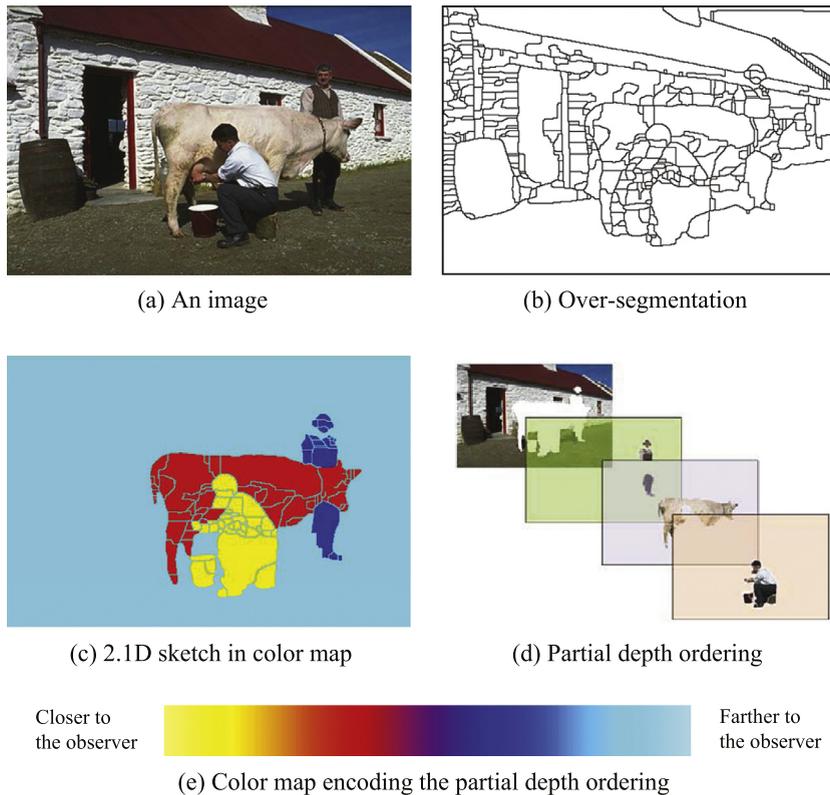


Fig. 2. A real example of 2.1D sketch. The partial depth order in (c) is generated by the proposed method in an interactive mode. The result is visualized by a color map (e) used in [8]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

interpretations [9,10], depth segmentation [11,12], occlusion recovery [13,14], contour illusory and completion [15,16] and foreground/background separation [17,18].

Nitzberg and Mumford [2] formulated the 2.1D sketch problem as an energy minimization problem defined similarly to the Mumford–Shah energy function. Their algorithm proceeded in a three-step procedure: (1) finding edges and T-junctions, (2) hypothesizing continuations, and (3) minimizing the defined energy function combinatorially with the constraints related to the edges and continuations. Only the third step was implemented in [2]. For the second step, a set of continuous curves by smoothly connecting disjoint contour segments are required for contour integrals in the energy function. This corresponds to the occluded contour completion problem which is itself a difficult problem. So the method in [2] can only handle simple synthesized images. Adelson [3] proposed the concept of layered representation for image coding and motion analysis. Since then, there has been limited work on the 2.1D sketch problem with limited progress being achieved, especially on the problem of inferring a 2.1D sketch from real images. A quadratic objective function was designed in [19] to incorporate ordering hints from T-junctions and the discontinuities at image contours for 2.1D sketch extraction. However, it was still unclear how to incorporate other local features such as saliency, color and textures used in this paper. For a special type of image texture that is formed by spatial repetition of a large number of texels, an unsupervised extraction method for 2.1D textures was proposed in [20]. In [21] a set of local cues, including semantics, positions, compactness, shared boundaries and junctions, were used in an Adaboost prediction of corresponding occlusions. In this paper, we deal with 2.1D sketches extracted from more general natural images and propose a unified global framework to solve the potential conflict of local cues.

2.1D sketch can find a wide range of applications including figure/ground segregation, image segmentation, image editing and objects' partial retrieval. By following the energy function for 2.1D sketch extraction defined in [2], Eshedoglu and March [11] proposed to segment an image with depth but without detecting junctions, using a variational formulation technique. Ren et al. [17] presented a method for figure/ground assignment according to local image evidence learned in a Markov random field (MRF) model. Yu et al. [18] proposed a model for figure/ground segregation based on a hierarchical MRF, using clique potentials in the MRF to encode local logical decision rules and demonstrating a system that automatically integrates sparse local relative depth cues arising from T-junctions over long distances into a global ordering of relative depths. Wang et al. [16] used some shape priors (e.g., templates for objects) to do partitioning based on the Swendsen-Wang cuts algorithm for integration of region-based and curve-based segmentation. Recently Hoiem et al. [15] proposed a method for recovering occlusion boundaries by learning a conditional random field (CRF) model. Gao et al. [22] proposed a mixed random field formulation with a hierarchical graphical model for the 2.1D sketch. The graphical model [23] extracted from 2.1D sketches can also be used in a conceptual design process [24]. An image editing method was

proposed in [25] by detecting and rearranging repeated scene elements, which utilized a partial depth ordering of overlapping repeated elements. By projecting a 3D object into a set of planes determined by optimal viewpoints [26], the 2.1D sketches can also be used for partial retrieval of both 2D shapes [27,28] and 3D objects [29].

In this paper we make the following contributions for inferring the 2.1D sketch from a monocular image:

- We present a generic energy function for modeling and computing the 2.1D sketch which can leverage different features (local and global) extracted from 2D image regions.
- Our method can serve as a universal framework for computing the 2.1D sketch of both simple artificial images and complex natural images. As a comparison, previous work [22,2] can only handle simple or synthetic images.

3. Overview of the proposed method

Given an input image, we first compute the over-segmented 2D regions using the gPb-owt-ucm algorithm in [6], and extract the local features (Section 4) in these regions for inferring (1) the local figure-ground relations and (2) the same depth layer relations. For example, in Fig. 1(b), the over-segmented regions 2 and 3 are in the figure-ground relation and the over-segmented regions 1, 3 and 4 are in the same depth layer relation. We pose the inference problem under the energy optimization framework (Section 5). The energy function consists of two parts: (1) one is defined based on the local partial ordering relations (i.e., figure/ground) between two adjacent over-segmented regions, which capture the marginal information of the global partial depth ordering, and (2) the other is defined based on the same depth layer relations among all the over-segmented regions, which groups regions belonging to the same object to account for the over-segmentation issues. Once the energy function is minimized by an efficient hybrid evolution algorithm, the optimal 2.1D sketch is obtained.

4. Local feature extraction

Given an over-segmented image computed by the algorithm in [6], the local features introduced in this section only consider the local relations among over-segmented regions.

We use the following two local relations abbreviated as F–G and SL. The F–G (figure–ground) relation is referred to as that two adjacent over-segmented regions do not belong to the same layer in the 2.1D sketch, and thus we need to determine which is foreground and which is background. The SL (same layer) relation is referred to as that two over-segmented relations have the same depth, and thus they need to be combined into one layer. To compute the 2.1D sketch, we first need to combine over-segmented regions with SL relations, and then we need to compute the partial depth order of the remaining regions with F–G relations.

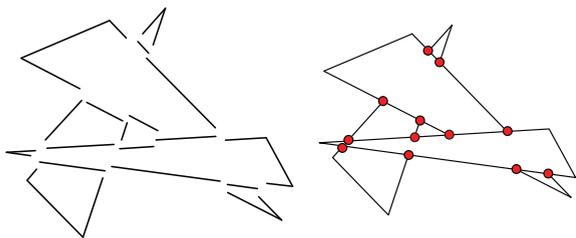


Fig. 3. Graph-based T-junction detection. Left: boundaries of over-segmented regions as shown in Fig. 1. Right: the nodes (shown in red) in the graph are the T-junctions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1. F–G relations

In this study we use three F–G relations, i.e., T-junction, boundary and saliency features.

4.1.1. T-junction features

T-junctions are significant local visual cues to analyze the depth relation between partially overlapped regions, which were used in [19,2] without implementation details. In this paper we use the following simple strategy. As illustrated in Fig. 3, we extract all boundary segments of over-segmented regions and build a graph $G = (V, E)$, where the node $v_r \in V$ is the intersection point of three distinct boundary segments, i.e., $v_r = \partial S_i \cap \partial S_j \cap \partial S_k, i \neq j \neq k$, and ∂S_i is the boundary of segment S_i . The edges $e_s \in E$ are region boundaries separated by the nodes, i.e., $\bigcup_{e_s \in E} e_s = (\bigcup_i S_i) \setminus (\bigcup_{v_r \in V} v_r)$. Each non-degenerate nodes in G corresponds to a T-junction feature. The depth relation locally around the T-junction is that, the cap of the T-junction is closer to the observer than the regions indicated by the stem of the T-junction.

We use a matrix T to represent all the T-junction features:

$$T_{m \times tnum} = [t_1, \dots, t_i, \dots, t_{tnum}] \quad (1)$$

where m is the number of over-segmented regions, $tnum$ is the total number of T-junctions and each t_i is a $m \times 1$ vector which represents the information of i th T-junctions, i.e., the j th element in t_i satisfies:

$$t_i(j) = \begin{cases} -2 & \text{if } j = \text{cap}(i) \\ 1 & \text{if } j = \text{stem}(i) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $\text{cap}(i)$ returns the region ID of the cap of the i th T-junction and $\text{stem}(i)$ returns the regions IDs (there are two regions belongs to the stem) of the stem of the i th T-junction.

4.1.2. Boundary features

For each edge $e_i \in E$ in the graph G , we compute a boundary feature b_i as proposed in [30]. We also use a matrix B to represent all the boundary features:

$$B_{m \times bnum} = [b_1, \dots, b_i, \dots, b_{bnum}] \quad (3)$$

where m is the number of over-segmented regions, $bnum$ is the total number of boundary edges in G and each b_i is a $m \times 1$ vector whose j th element satisfies:

$$b_i(j) = \begin{cases} -1 & \text{if } j = \text{bfigure}(i) \\ 1 & \text{if } j = \text{bground}(i) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $\text{bfigure}(i)$ and $\text{bground}(i)$ returns respectively the region IDs of the figure (closer to observer) and ground (farther to observer) judged by the i th boundary feature using the method proposed in [30].

4.1.3. Saliency features

In most cases, salient regions have high possibility to be the figure regions, especially in the images where a single salient object exists. We compute the saliency for each over-segmented regions using the method proposed in [31] with an improvement of using soft image abstraction [32]. We use a matrix S to represent all the saliency features:

$$S_{m \times snum} = [s_1, \dots, s_i, \dots, s_{snum}] \quad (5)$$

where m is the number of over-segmented regions, $snum$ is the number of all F–G relations judged by saliency features and each s_i is a $m \times 1$ vector whose j th element satisfies:

$$s_i(j) = \begin{cases} -1 & \text{if } j = \text{sfigure}(i) \\ 1 & \text{if } j = \text{sground}(i) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where $\text{sfigure}(i)$ and $\text{sground}(i)$ returns respectively the region IDs of the figure and ground judged by the i th saliency feature.

4.2. SL relations

We use the following three SL relations, i.e., color, texture and hierarchical features.

4.2.1. Color features

Since the image domain is over-segmented, if two adjacent regions have similar colors, we regard that they locally have a SL relation. We use the HSL color space; i.e., if the mean color vectors $HSL(p) = (h_p, s_p, l_p)$ and $HSL(q) = (h_q, s_q, l_q)$ of two adjacent regions p and q satisfy

$$\|HSL(p) - HSL(q)\|_2 < \text{disc} \quad (7)$$

where $\|\cdot\|$ is Euclidean 2-norm and disc is a small threshold, an SL relation is set to these two regions. We use a matrix C to represent all the color features:

$$C_{m \times cnum} = [c_1, \dots, c_i, \dots, c_{cnum}] \quad (8)$$

where m is the number of over-segmented regions, $cnum$ is the number of all the SL relations judged by color features and each c_i is a $m \times 1$ vector whose j th element satisfies:

$$c_i(j) = \begin{cases} -1 & \text{if } j = \text{CSL}_1(i) \\ 1 & \text{if } j = \text{CSL}_2(i) \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where $\text{CSL}_1(i)$ and $\text{CSL}_2(i)$ return the IDs of two regions which are in a SL relation judged by the i th color feature. The order of $\text{CSL}_1(i)$ and $\text{CSL}_2(i)$ does not affect the result.

4.2.2. Texture features

Since the color features have been considered in Eq. (18), we construct the texture features using the gray level of the image $I(x, y)$. Let $(x, y) \in R_i$ be all the pixels belonging to the region R_i , and w_i and h_i be the width and height of R_i . We use the following scale image features [33–35] for defining the texture feature of R_i :

- Angular second-moment feature \mathbf{a}_i :

$$\mathbf{a}_i = \sum_{(x,y) \in R_i} I^2(x, y) \quad (10)$$

- Contrast feature \mathbf{c}_i :

$$\mathbf{c}_i = \sum_{k=1}^{\max\{w_i, h_i\}} k^2 \sum_{|x-y|=k, (x,y) \in R_i} I(x, y) \quad (11)$$

- Inverse different moment feature \mathbf{i}_i :

$$\mathbf{i}_i = \sum_{(x,y) \in R_i} \frac{I(x, y)}{1 + (x - y)^2} \quad (12)$$

- Entropy feature \mathbf{e}_i :

$$\mathbf{e}_i = - \sum_{(x,y) \in R_i} I(x, y) \log I(x, y) \quad (13)$$

- Correlation feature \mathbf{cor}_i :

$$\mathbf{cor}_i = - \sum_{(x,y) \in R_i} \frac{xyI(x, y) - u_x u_y}{s_x s_y} \quad (14)$$

where

$$\begin{aligned} u_x &= \sum_{(x,y) \in R_i} xI(x, y), \quad u_y = \sum_{(x,y) \in R_i} yI(x, y), \\ s_x^2 &= \sum_{(x,y) \in R_i} I(x, y)(x - u_x)^2, \\ s_y^2 &= \sum_{(x,y) \in R_i} I(x, y)(y - u_y)^2 \end{aligned} \quad (15)$$

We take four scanning directions (up–down, left–right and two diagonals) and assemble all the scale image features into a normalized texture vector:

$$tv_i = (\mathbf{a}_i^1, \mathbf{c}_i^1, \mathbf{i}_i^1, \mathbf{e}_i^1, \mathbf{cor}_i^1, \dots, \mathbf{a}_i^4, \mathbf{c}_i^4, \mathbf{i}_i^4, \mathbf{e}_i^4, \mathbf{cor}_i^4) \quad (16)$$

We infer locally that two regions p and q have the SL relations if they satisfy:

$$\|tv_p - tv_q\|_2 < dist \quad (17)$$

where $\|\cdot\|$ is Euclidean 2-norm and $dist$ is a small threshold. We use a matrix X to represent all the texture features:

$$X_{m \times texnum} = [x_1, \dots, x_i, \dots, x_{texnum}] \quad (18)$$

where m is the number of over-segmented regions, $texnum$ is the number of all SL relations judged by texture features and each x_i is a $m \times 1$ vector whose j th element satisfies:

$$x_i(j) = \begin{cases} -1 & \text{if } j = TSL_1(i) \\ 1 & \text{if } j = TSL_2(i) \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

where $TSL_1(i)$ and $TSL_2(i)$ return the IDs of two regions which are in a SL relation judged by the i th texture feature. The order of $TSL_1(i)$ and $TSL_2(i)$ does not affect the result.

4.2.3. Hierarchical information

State-of-the-art segmentation algorithms such as the Berkeley multi-scale segmentation algorithm [6] can provide hierarchical segmentation information. A hierarchical segmentation example is shown in Fig. 4. If such hierarchical information is available, our method can make use of it for helping judge the SL relation as follows.

Let p and q be two over-segmented regions in a hierarchical segmentation tree. We compute a segmentation distance dis_{seg} between p and q as:

$$dis_{seg}(p, q) = D_{leaf} - D_{ancestor}(p, q) \quad (20)$$

where D_{leaf} is the depth of the leaf nodes representing the over-segmented regions and $D_{ancestor}(p, q)$ is the depth of the most recent common ancestor nodes of leaf nodes p and q . Our inference is based on the heuristic that if the segmentation distance is small for p and q , the oriented watershed transform value of the boundary is also small and p and q are more likely to be SL relation.

To account for the hierarchical information, we modify the judgement formula of color features and texture features by:

$$\|HSL(p) - HSL(q)\|_2 < disc - \lambda_1 \cdot dis_{seg}(p, q) \quad (21)$$

$$\|tv_p - tv_q\|_2 < dist - \lambda_2 \cdot dis_{seg}(p, q) \quad (22)$$

where λ_1 and λ_2 are two weighting coefficients.

There are a few parameters $disc, dist, \lambda_1, \lambda_2$ in local feature definitions and we will specify them in Section 6.

5. A global energy optimization framework

Each of the local features introduced in Section 4 infers a partial depth ordering on some over-segmented regions locally. Only using local information cannot ensure a globally consistent partial depth order and there will be some misjudgement and conflicts among the partial depth orders inferred locally. We propose below a global energy optimization framework to obtain a globally consistent partial depth order. Note that our framework is general and any local features not defined in Section 4 can be naturally incorporated into this framework.

5.1. A global energy

Let $S = \{1, 2, \dots, m\}$ be a finite index set, where m is the number of over-segmented regions. The solution to specify a 2.1D sketch is an index set $D = \{d_1, d_2, \dots, d_m\}$, where $1 \leq d_i \leq m$. If there are no SL relations in m regions, D is a permutation π of the set S , where π is a bijective map and $D = \{\pi(1), \pi(2), \dots, \pi(m)\}$.

The global energy $E_{2.1d}$ we define below represents the conflicts of the partial depth order inferred locally. When the value of $E_{2.1d}$ is minimized, the global conflict is also minimized, which leads to an optimal solution of the 2.1D sketch.

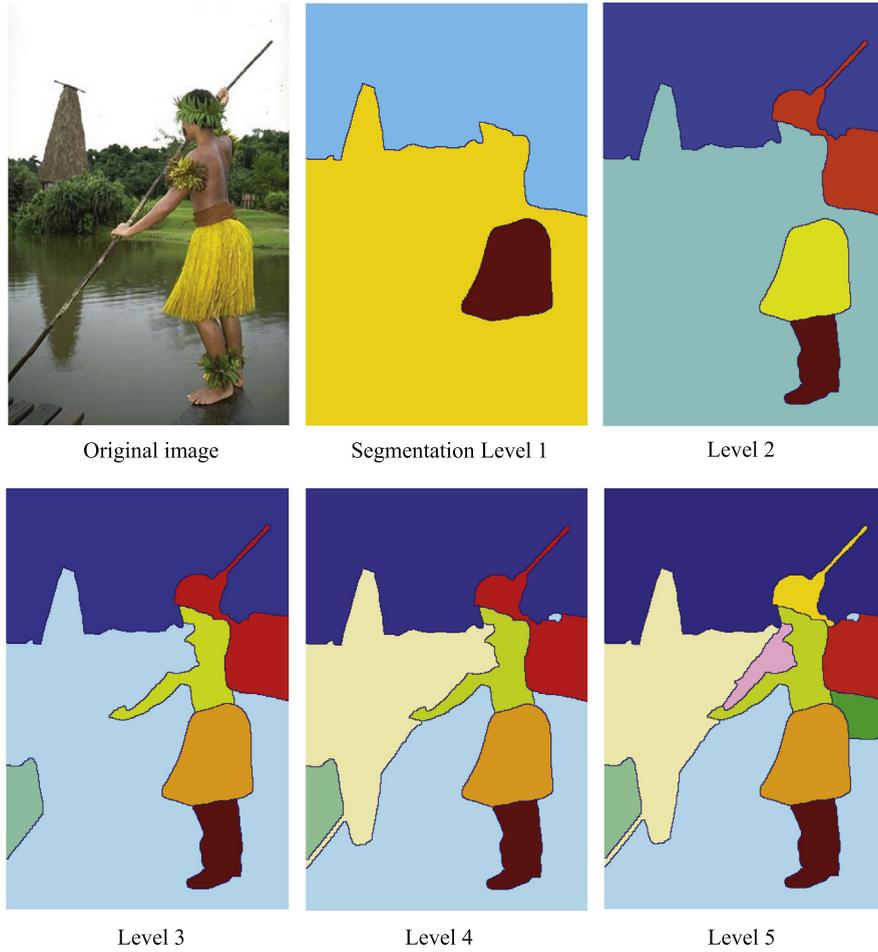


Fig. 4. A hierarchical segmentation of an image in Berkeley data set BSDS500 [6]. The different regions in the segmentation are shown in different colors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The global energy $E_{2.1d}$ consists of two terms, representing the F–G and SL relations respectively:

$$E_{2.1d} = E_{FG} + E_{SL} \quad (23)$$

We denote a parameterized exponential function as:

$$E(\lambda, x) = e^{-\lambda x} \quad (24)$$

Then E_{FG} is defined as:

$$E_{FG} = \sum_{i=1}^{tnum} E(\lambda_t, \alpha_i) + \sum_{j=1}^{bnum} E(\lambda_b, \beta_j) + \sum_{k=1}^{snum} E(\lambda_s, \gamma_k) \quad (25)$$

where $\lambda_t > 0$, $\lambda_b > 0$, $\lambda_s > 0$ are three weighting coefficients whose values are specified in the next section, α_i is the i th element in the m -vector $\alpha = T^T D$, β_j is the j th element in the m -vector $\beta = B^T D$, γ_k is the k th element in the m -vector $\gamma = S^T D$ and T, B, S are matrices defined in Eqs. (1), (3), (5) respectively.

We note that if all the local judgements for F–G relations are correct, the values α_i, β_j and $\gamma_k, 1 \leq i \leq tnum, 1 \leq j \leq bnum, 1 \leq k \leq snum$, are all positive and E_{FG} reaches its minimum. To see this property, we explain it using the T-junction features for an example and the other two

local (boundary and saliency) features can be explained in the same way. Assume in an image there are three regions with the partial depth ordering $R_1 > R_2 > R_3$, i.e., $D = (1 \ 2 \ 3)^T$. If they are judged correctly by a T-junction feature $t_1 = (-2 \ 1 \ 1)^T$, then $t_1^T \cdot D = 3$ achieves the maximum positive value. For any incorrect judgment, e.g., $t'_1 = (1 \ -2 \ 1)^T$, $t'_1{}^T \cdot D = 0 < t_1^T \cdot D$. Given any $\lambda_t > 0$, $E(\lambda_t, t_1^T \cdot D)$ achieves the minimum value.

Since local judgments by different local features may have conflicts, the function $E(\lambda, x)$ plays a role of penalties for the wrong judgment of F–G relations and the minimum value of E_{FG} corresponds a global optimal solution to the F–G relations.

Denote an absolute version of a m -vector r by $|r| = (|r_1| \ |r_2| \ \cdots \ |r_m|)$. The E_{SL} in Eq. (23) is defined as:

$$E_{SL} = \sum_{u=1}^{cnum} E(-\lambda_c, \zeta_u) + \sum_{v=1}^{texnum} E(-\lambda_{tex}, \eta_v) \quad (26)$$

where $\lambda_c > 0$, $\lambda_{tex} > 0$ are two weighting coefficients whose values are specified in the next section, ζ_u is the u th element in the m -vector $\zeta = |C^T D|$, η_v is the v th element in

the m -vector $\eta = |X^T D|$, C and X are matrices defined in Eqs. (8), (18) respectively.

If all the local judgements for SL relations are correct, all the values ζ_u and η_v , $1 \leq u \leq cnum$, $1 \leq v \leq texnum$, are zero, which corresponds to a minimum value of E_{SL} . If there are some wrong or conflict SL relations judged by local features, some pairs of two regions have different layer indices and thus some absolute values ζ_u or η_v are positive, leading to an increase of E_{SL} .

5.2. Energy minimization and 2.1D sketch extraction

The minimization of the global energy $E_{2.1d}$ corresponds to a global optimal solution D_{opt} of the partial depth ordering. To solve the nonlinear optimization problem $\arg \min_D E_{2.1d}$, we use a variant of hybrid differential evolution algorithm. Compared to classical multidimensional numerical solutions such as downhill simplex method or direction-set method which may be easily trapped into a local optimization, differential evolution [36] has been demonstrated to be a powerful tool for finding the global optimization. How to generate new solutions from existing one is important in differential evolution. Below we present a variant of hybrid differential evolution with simulated annealing, which is used to escape from possible local optimum attractions.

We represent the solution space for 2.1D sketch by a $m \times n$ matrix DS :

$$DS = [D_1, D_2, \dots, D_n] \quad (27)$$

where D_i , $0 < i \leq n$, is a m -vector, m is the number of over-segmented regions and we use $n = 100$ in our experiment. We use the following steps for solving the 2.1D sketch problem:

1. Initialization. We randomly generate n permutations of the index set $\{1, 2, \dots, m\}$ to initialize the matrix DS .
2. Hybrid mutation. Denote the iteration number by s , and let u_i^{s+1} and v_i^{s+1} be intermediate m -vectors used later for updating D_i^s into D_i^{s+1} . Set

$$v_i^{s+1} = \alpha D_{r_1}^s + \beta D_{best}^s + \varepsilon_1 (D_{r_2}^s - D_{r_3}^s) + \varepsilon_2 (D_{r_4}^s - D_{r_5}^s)$$

where $r_1^s, r_2^s, r_3^s, r_4^s, r_5^s \in [1, n]$ ($n = 100$ in our experiment) are random and mutually different integers, x_{best}^s is the best solution in the s th iteration in DS , and we set the parameters $\alpha = 0.8$, $\beta = 0.2$, $\varepsilon_1 = 0.6$, $\varepsilon_2 = 0.06$.

3. Crossover. Set

$$u_i^{s+1} = \begin{cases} v_i^{s+1} & U(0, 1) \leq CR \\ D_i^s & \text{otherwise} \end{cases}$$

where $U(0, 1)$ is a uniform random number between 0 and 1, and we set the crossover rate $CR = 0.9$.

4. Selection. We compute an updating probability by

$$p = \begin{cases} 1, & \text{if } E_{2.1d}(u_i^{s+1}) \leq E_{2.1d}(D_i^s) \\ e^{-\frac{E_{2.1d}(u_i^{s+1}) - E_{2.1d}(D_i^s)}{Z^s}}, & \text{otherwise} \end{cases}$$

where Z^s is the temperature at iteration s that is decreased with the increasing generation:

$$Z^{s+1} = \mu Z^s$$

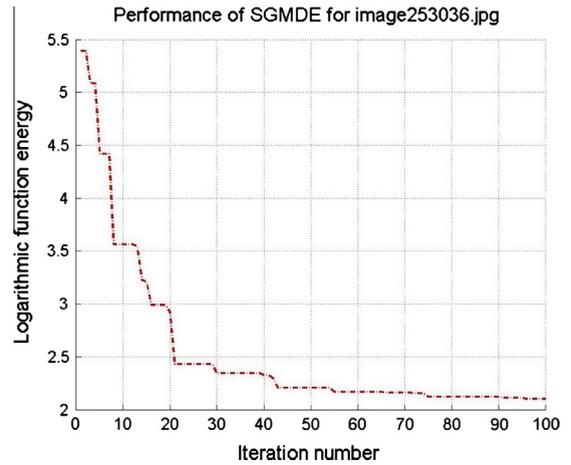


Fig. 5. An iteration process working on the image (ID 253036) in Berkeley data set (BSDS500) [6], using the hybrid differential evolution.

We set the initial temperature $Z^0 = 100$ and $\mu = 0.1$. Given the probability p , we update the solution space in the next generation as:

$$D_i^{s+1} = \begin{cases} u_i^{s+1}, & \text{if } U(0, 1) < p \\ D_i^s, & \text{otherwise} \end{cases}$$

5. Clone selection and Gaussian mutation. The best 10 candidate solutions in DS are selected for the cloning operation, in which Gaussian variations are added:

$$x_i^{s+1}(j) = x_i^{s+1}(j) + \lambda^{s+1} \cdot rand_j(G(0, 1))$$

$$1 \leq i \leq 20, 1 \leq j \leq m$$

where $rand_j(G(0, 1))$ is a random number in $(-\infty, \infty)$ generated by standard Gaussian distribution $G(0, 1)$ with mean $\mu = 0$ and standard deviation $\sigma = 1$, and $\lambda^{s+1} = 0.1 \|D_{best}^{s+1}\|_2$.

6. Stopping criterion. The total number of iteration reaches a predefined threshold $n_s = 100$.

An iteration process of the numerical solution working on the image (ID 253036) in Berkeley data set (BSDS500) [6] is illustrated in Fig. 5. The MATLAB code is available to be downloaded at¹ and readers can repeat all experiments presented in this paper.

The detailed complexity analysis and the performance of above specified hybrid differential evolution with comparison with classical numerical methods are presented in Section 6.2.

One advantage of the hybrid differential evolution is that it allows continuous optimization, i.e., all the elements in D_{best} are not limited to integers. Given the solution $D_{best} = (d_1 \ d_2 \ \dots \ d_m)^T$, we obtain the partial depth ordering of m regions by the matrix R :

$$R_{ij} = \begin{cases} 0 & |d_i - d_j| \leq 0.5 \\ 1 & d_i - d_j > 0.5 \\ -1 & d_j - d_i > 0.5 \end{cases}, \quad 1 \leq i < j \leq m \quad (28)$$

¹ <http://cg.cs.tsinghua.edu.cn/people/~Yongjin/2.1D-source.zip>.

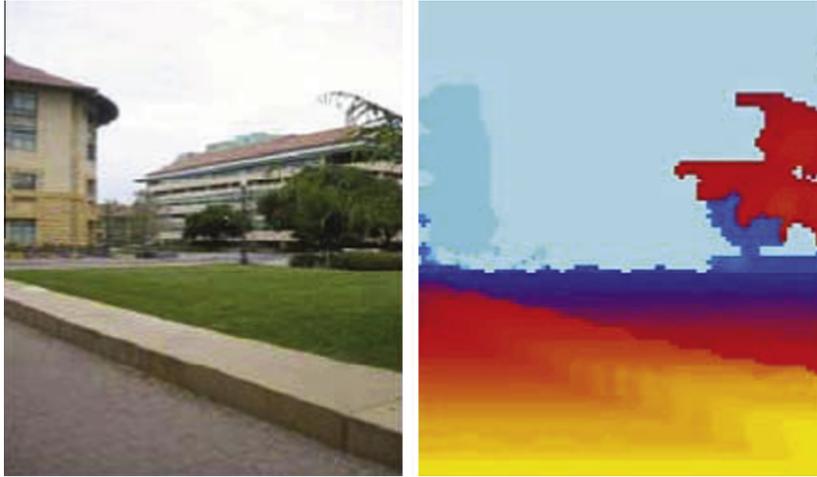


Fig. 6. An example image (left) and its depth ground truth (right) in the Make3D data set.

If $R_{ij} = 0$, then regions i and j are in the same layer. If $R_{ij} = 1$, then region i is foreground and region j is background. If $R_{ij} = -1$, then region j is foreground and region i is background.

6. Algorithm evaluation and comparisons

There are several public available data sets that we can use for testing our proposed global energy optimization framework, including Cornell Make3D data set [8], Berkeley 3-D object data set (B3DO) [37] and Berkeley data set (BSDS500) [6]. However, we found that the Make3D and B3DO data sets were not suitable for our application. We explain the reason with the following examples. For the image in the Make3D data set shown in Fig. 6 left, due to the limited laser scanning resolution in the large scale of outdoor scene, the buildings and the sky have the same depth ground truth as shown in Fig. 6 right. However, in our application, the buildings and sky have a clear occlusion relation and should be judged as the different layers. For the image in the B3DO data set shown in Fig. 7 left, due to the limited resolution of the Kinect hardware, the clothes, the quilt and the bed all have the same depth ground truth as shown in Fig. 7 right. However, in our

application, the clothes, the quilt and the bed are clearly in different layers with different partial depth ordering.

To evaluate our proposed method, we select 100 images from the Berkeley data set (BSDS500) using the following principles: (1) The images are representative, including different scenarios and different target objects. (2) The images have good over-segmentation results. (3) The objects/regions in the images have clearly partial depth orders. The selected images with their IDs in BSDS500 are illustrated in Fig. 8. All these 100 selected images are over-segmented using the Berkeley segmentation algorithm [6]. Human subjects were invited to add partial depth orders to these over-segmented regions manually and the results were recoded in the matrix $T^{2.1d}$ as the ground truth. For $1 \leq i < j \leq m$, (1) if $T_{ij}^{2.1d} = 0$, then regions i and j are in the same layer; (2) if $T_{ij}^{2.1d} = 1$, then region i is foreground and region j is background; (3) if $T_{ij}^{2.1d} = -1$, then region j is foreground and region i is background. We use the ground truth $T^{2.1d}$ to measure the accuracy of different 2.1D sketch extraction methods.

For each test image, we define the accuracy η of the 2.1D sketch using the result matrix R (defined in Eq. (28)) and the ground truth $T^{2.1d}$ as the follows:

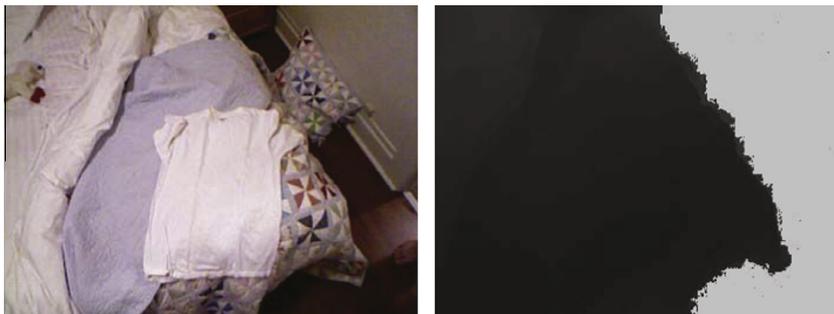


Fig. 7. An example image (left) and its depth ground truth (right) in the B3DO data set.

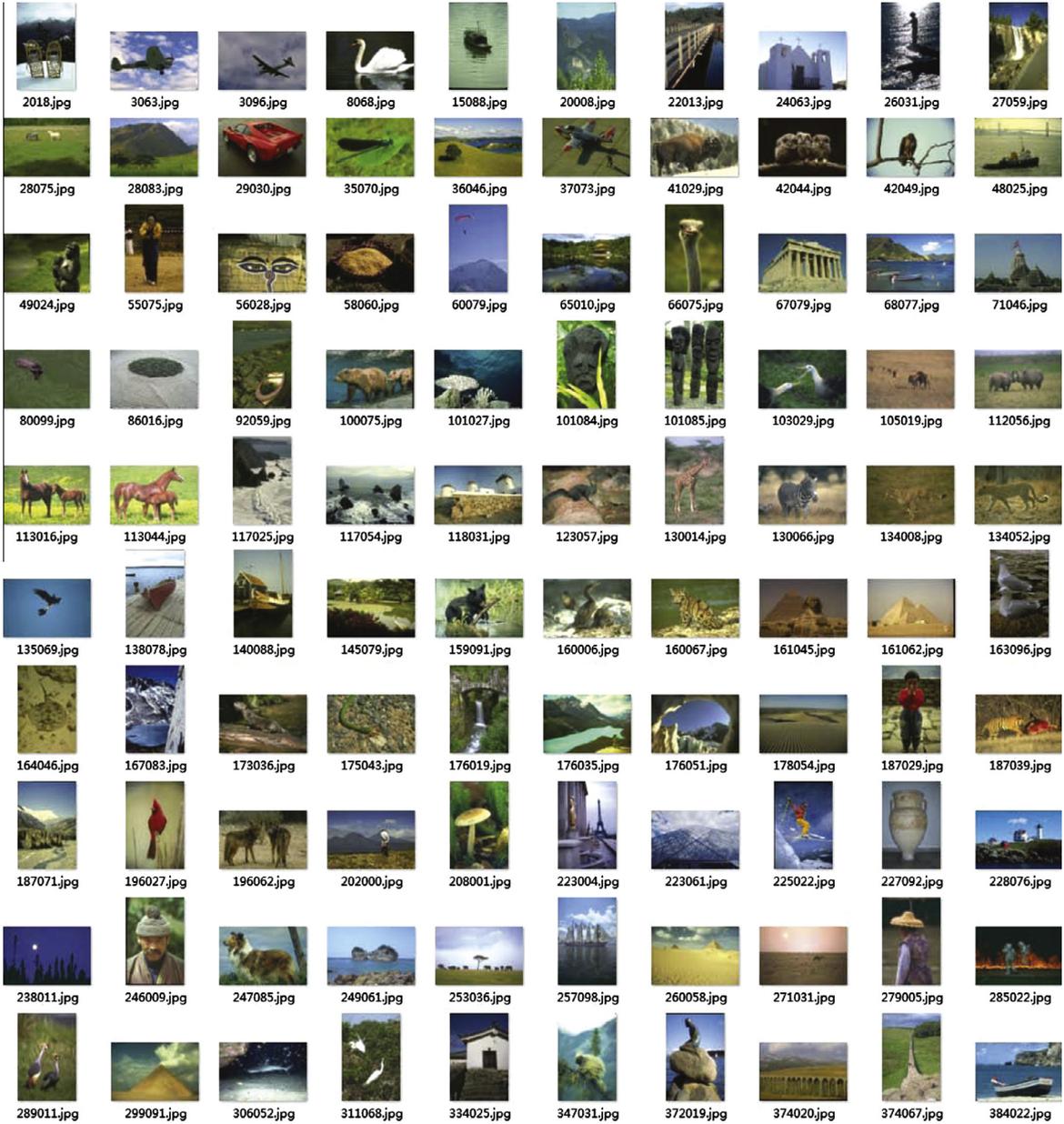


Fig. 8. The selected 100 images with their indices in Berkeley data set (BSDS500).

$$\eta = \frac{\sum_{i=1}^m \sum_{j=i+1}^m \text{Flag}_{ij}}{\binom{m}{2}} \quad (29)$$

where

$$\text{Flag}_{ij} = \begin{cases} 1 & R_{ij} = T_{ij}^{2,1d} \\ 0 & R_{ij} \neq T_{ij}^{2,1d} \end{cases} \quad (30)$$

All the experimental results and comparisons summarized below used the ground truth $T^{2,1d}$ with the images selected from the public available Berkeley data

set (BSDS500). Both image data and the MATLAB code are available and the readers can repeat the experimental results.

6.1. Parameter settings

There are a few parameters in our method. One set of parameters are the thresholds *disc* and *dist* used in Eqs. (7) and (17) respectively, and their scale weights λ_1 and λ_2 relative to the segmentation distance defined in Eqs. (20)–(22). In all experiments, our method is not sensitive to these four parameters and we tune these parameters according to the following observations. In the HSL space,

Table 1

The average accuracy and running time of 2.1D sketch extraction using three different numerical schemes. The time is measured at the platform of MATLAB R2010 in a PC (Intel (R) Core (TM) i7C2600 CPU 3.4 GHz) running Windows 7.

	Our method	GA [38]	PSO [39]
Average accuracy (%)	83.4	79.6	79.2
Average time (sec)	18.0	16.8	17.2

the range of each dimension is $H \in [0, 360]$, $S \in [0, 100]$ and $L \in [0, 100]$. So we choose the threshold $disc$ to be about 4% of the diagonal length of the HSL range cube, i.e., $disc = 15$. The magnitude of the texture vector defined in Eq. (16) is normalized to be 1 and we choose the threshold $dist$ to

Table 2

The average accuracy of 2.1D sketch extraction with different objective functions and methods.

Objective function	E_{FG} (25)	E_{SL} (26)	$E_{2,1d}$ (23)	Method [19]	(23) Without hierarchical information (21, 22)
Average accuracy (%)	75.8	43.3	83.4	48.3	79.5

be 0.1. Our experiments show that in Eqs. (21) and (22), good results can be obtained if the relative scale between $disc$ (or $dist$) and dis_{seg} is chosen to be within [5%, 10%]. So we set $\lambda_1 = 1$ and $\lambda_2 = 0.01$.

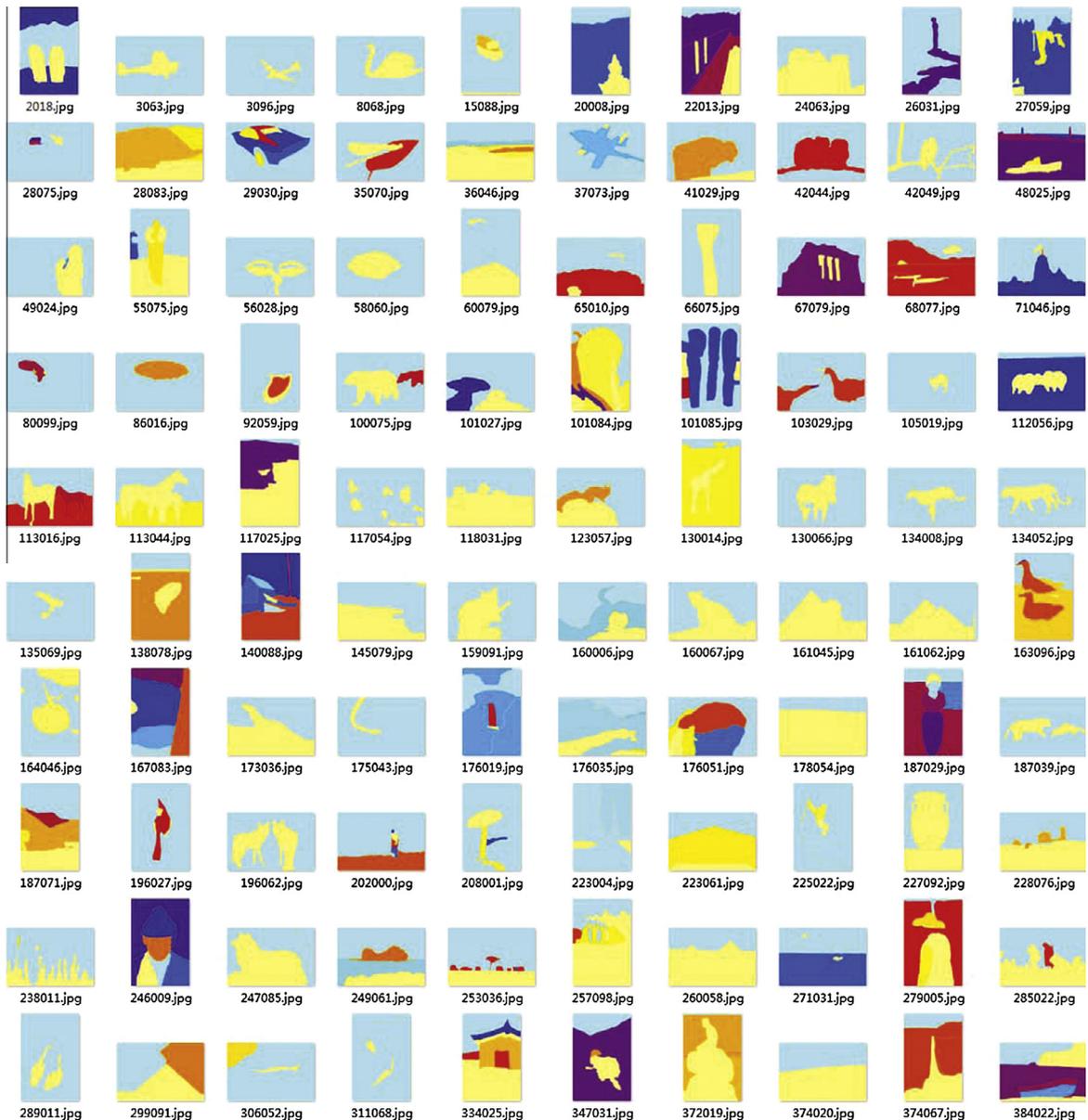


Fig. 9. The 2.1D sketches of 100 images in Fig. 8 by minimizing the global energy $E_{2,1d}$: their partial depth orders are encoded by the color map shown in Fig. 2(e). The original 2.1D sketches can be downloaded at <http://cg.cs.tsinghua.edu.cn/people/~Yongjin/2.1D-source.zip>.

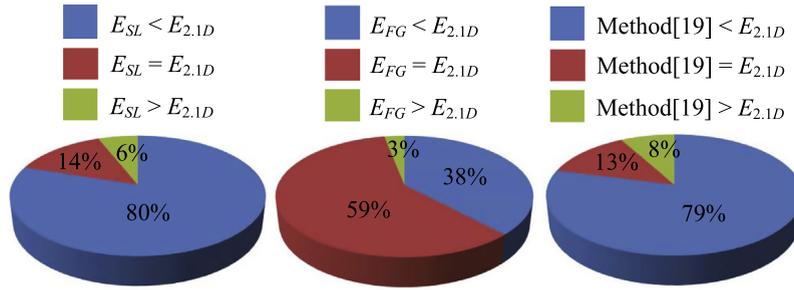


Fig. 10. The statistic data of accuracy comparison of the energy $E_{2.1d}$ to E_{FG} , E_{SL} and the method [19] on 100 test images shown in Fig. 8.

Table 3

The number of images for which 2, 3, 4, 5 and 6 layers existed in the extracted 2.1D sketches as represented in Fig. 9.

Layer numbers	2	3	4	5	6
Image numbers	28	25	23	16	8

The other set of parameters are weights $\lambda_t, \lambda_b, \lambda_s, \lambda_c$ and λ_{tex} used in Eqs. (25) and (26), which provide different combinations of local features. For different image types, the significance of each local feature is varied and the optimal set of these weighting parameters should be determined by the given image database. We use the following optimization method to determine these parameters based on BSDS500 database.

We define a functional Ξ in a five dimensional space $(\lambda_t, \lambda_b, \lambda_s, \lambda_c, \lambda_{tex})$. We choose the range of each dimension to be $[0, 10]$. Each point p in the hypercube $[0, 10]^5$ determine a deterministic global energy function defined in Eq. (23) and we denote it by $E_{2.1d}(p)$. With $E_{2.1d}(p)$, we compute an accuracy $\eta_i(p)$ for each image i in BSDS500. The functional Ξ is defined as

$$\Xi(p) = \sum_i \eta_i(p) \quad (31)$$

The optimal set of parameters is the point p' that maximizes the functional value Ξ in the hypercube $[0, 10]^5$. To find such an optimal p' , we estimate the analytic form of Ξ using RBF interpolant. We sample the hypercube $[0, 10]^5$ using an interval 2 in each dimension. For each sample point p_i , we compute the value $\Xi(p_i)$. The RBF interpolating function is then given by

$$\tilde{\Xi}(p) = \sum_i w_i \Phi(p - p_i) \quad (32)$$

where Φ is the radial basis function and w_i is a weight for each sample p_i . We choose the Gaussian radial basis function $\Phi(r) = e^{-r^2}$ due to its positive definite property. The weights w_i in Eq. (32) are obtained by solving the linear system from the interpolating constraints:

$$\tilde{\Xi}(p_j) = \Xi(p_j) = \sum_i w_i \Phi(p_j - p_i) \quad (33)$$

Given the analytic form of $\tilde{\Xi}$, we find its maximum value in $[0, 10]^5$ using the BFGS algorithm in multi-dimensions. The

corresponding optimal parameters are $\lambda_t = 0.2, \lambda_b = 0.2, \lambda_s = 0.2, \lambda_c = 4$ and $\lambda_{tex} = 1$.

6.2. Numerical scheme analysis and performance

Heuristic global optimization methods, including genetic algorithms (GAs), particle swarm optimization (PSO) algorithms, simulated annealing (SA) algorithms, differential evolution (DE) algorithms, utilize various stochastic search strategies to obtain the global optimums of nonlinear optimization problems. These optimization techniques do not use the derivatives of objective functions and are suitable for minimization of our objective function (23). The numerical scheme presented in Section 5.2 is an improvement based on a combination of SA and DE methods. We first briefly summarize the complexity of this numerical scheme and then compare it with a classic GA algorithm [38] and a PSO algorithm [39].

In our variant of hybrid differential evolution with simulated annealing, n is the population scale of solutions, m is the number of over-segmented regions and s is the maximum number of iterations. The complexities of the six steps in the numerical scheme are follows:

1. Initialization: $O(mn)$.
2. Hybrid mutation: $O(mns)$.
3. Crossover: $O(mns)$.
4. Selection: $O(ns)$.
5. Clone selection and Gaussian mutation: $O(mns)$.
6. Stopping criterion: $O(s)$.

Thus, the total complexity is $O(mns)$, which is the same as the GA algorithm [38] and the PSO algorithm [39]. The average accuracy and running time of our 2.1D sketch extraction method using these three numerical schemes over 100 images selected from BSDS500 are summarized in Table 1, which show that the numerical scheme presented in Section 5.2 has the highest average accuracy at the cost of a slightly more running time.

6.3. Evaluation of 2.1D sketch extraction

In the global energy $E_{2.1d}$ defined in Eq. (23), there are two terms E_{FG} and E_{SL} , which characterize the 2.1D sketch representation from different perspectives. We first evaluate the performance of the combinations of these two terms. Fig. 10 summarizes the statistic data for comparison

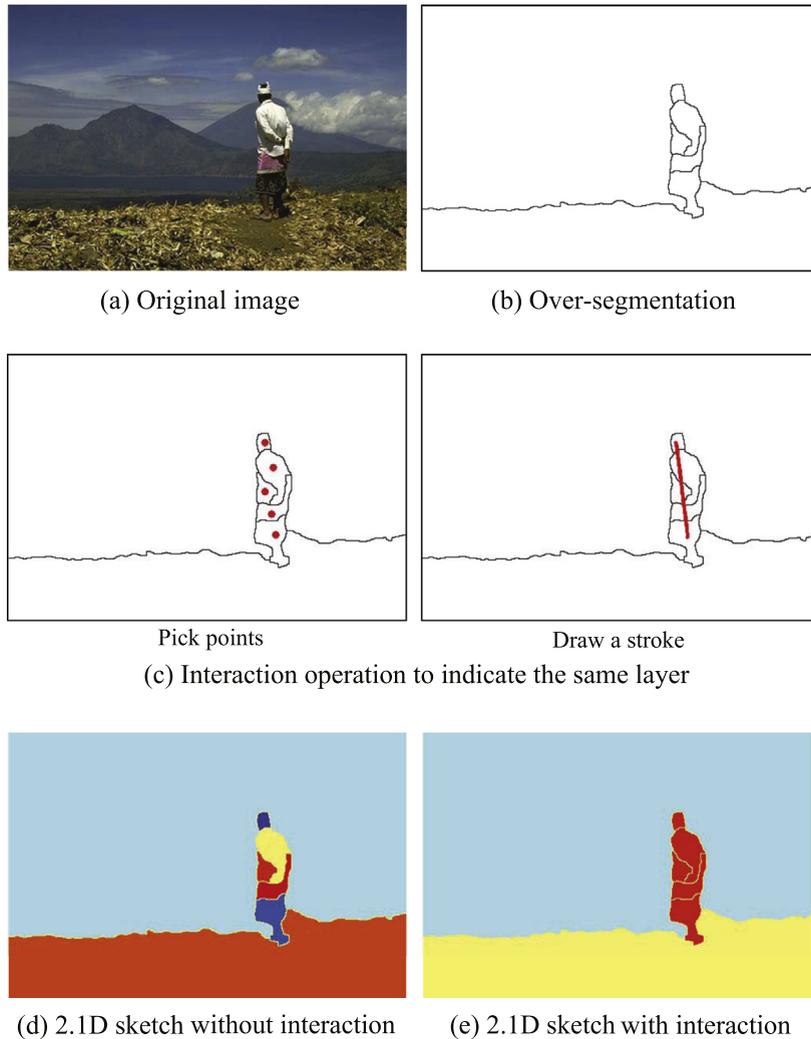


Fig. 11. The interactive operation for 2.1D sketch extraction. The partial depth orderings in the 2.1D sketches in (d) and (e) are encoded using the color map shown in Fig. 2(e). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of the accuracy of extracted 2.1D sketches by minimizing the energies E_{FG} (Eq. (25)), E_{SL} (Eq. (26)) and $E_{2.1d}$ (Eq. (23)) on 100 test images shown in Fig. 8. The original full data is also presented in Table 4 in Appendix A. The average accuracy of different energies is also summarized in Table 2. In all the 100 test images, there are 97 images for which $E_{2.1d}$ is better than or equally well to E_{FG} ; there are 94 images for which $E_{2.1d}$ better than or equally well to E_{SL} . The average accuracy of $E_{2.1d}$ is 83.4%, which is also the best when compared to the average accuracy of E_{FG} (75.8%) and E_{SL} (43.3%). These results show that the combinatorial form of $E_{2.1d}$ achieves the best performance of 2.1D sketch extraction.

Our proposed global energy optimization method offers a general framework that can naturally incorporate any local features. In the presented study, we use a special hierarchical information inherent in the Berkeley segmentation dataset, as characterized in Eqs. (21) and (22). It is interesting to explore whether this hierarchical information helps

the 2.1D sketch extraction. Table 2 also summarizes the results for this comparison, which show that the energy function $E_{2.1d}$ incorporating hierarchical information in Eqs. (21) and (22) has the average accuracy (83.4%) better than the average accuracy (79.5%) using the energy function $E_{2.1d}$ without the hierarchical information.

Secondly we compare the proposed global energy optimization method with previous 2.1D sketch extraction methods. There are only a few 2.1D sketch extraction methods [19,22,2] in which due to the high computational complexity or some strong assumptions on the continuity of image contours, the methods [22,2] can only handle artificially synthesized simple images but cannot handle the real images as shown in Fig. 8. We thus compare our method with the method in [19]. The statistic data of comparison between our method ($E_{2.1d}$) and the method in [19] is shown in Fig. 10. The full results using the method [19] are also summarized in Table 4, and its average accuracy is summarized in Table 2. The results show that in all the

Table 4

The accuracies of 2.1d sketches extracted from 100 images shown in Fig. 8, using different energy functions and methods. The images are indexed using their original ID in the Berkeley data set (BSDS500) [6]. For each image, the best accuracy is shown in red color.

Image ID	E_{FC}	E_{SL}	$E_{2.1d}$	Method [19]	Image ID	E_{FC}	E_{SL}	$E_{2.1d}$	Method [19]
2018	0.893	0.321	0.964	0.536	135069	1.000	1.000	1.000	1.000
3063	1.000	1.000	1.000	0.333	138,078	0.800	0.400	0.800	0.467
3096	1.000	0.333	1.000	0.333	140,088	0.357	0.679	0.393	0.321
8068	1.000	0.000	1.000	1.000	145,079	1.000	0.000	1.000	0.333
15,088	0.619	0.476	0.714	0.286	159,091	0.667	0.500	1.000	0.333
20,008	1.000	0.667	1.000	0.000	160,006	1.000	0.333	1.000	0.667
22,013	0.524	0.381	0.571	0.286	160,067	1.000	1.000	1.000	1.000
24,063	1.000	0.333	1.000	0.833	161,045	1.000	1.000	1.000	0.333
26,031	0.333	0.500	0.833	0.167	161,062	1.000	1.000	1.000	1.000
27,059	0.733	0.133	0.667	0.533	163,096	0.467	0.533	0.511	0.267
28,075	0.533	0.267	0.667	0.333	164,046	0.733	0.267	1.000	0.267
28,083	0.800	0.333	0.800	0.533	167,083	0.400	0.200	0.400	0.533
29,030	0.278	0.306	0.444	0.194	173,036	1.000	0.333	1.000	0.000
35,070	1.000	0.500	1.000	1.000	175,043	1.000	0.000	1.000	1.000
36,046	0.810	0.333	0.905	0.571	176,019	0.533	0.400	0.733	0.200
37,073	0.467	0.467	0.800	0.467	176,035	0.857	0.190	0.857	0.619
41,029	1.000	0.667	1.000	0.667	176,051	0.476	0.095	0.524	0.381
42,044	0.400	0.400	0.800	0.333	178,054	1.000	0.667	1.000	0.333
42,049	0.524	0.143	1.000	0.476	187,029	0.571	0.464	0.607	0.250
48,025	0.644	0.422	0.867	0.622	187,039	1.000	0.333	1.000	1.000
49,024	0.833	0.333	0.833	0.167	187,071	1.000	0.500	1.000	0.600
55,075	0.800	0.400	0.800	0.933	196,027	0.600	0.700	0.700	0.300
56,028	0.733	0.067	1.000	0.533	196,062	1.000	0.667	1.000	0.333
58,060	1.000	0.000	1.000	1.000	202,000	0.571	0.238	0.571	0.333
60,079	0.667	0.000	1.000	0.000	208,001	0.667	0.467	0.800	0.733
65,010	0.667	0.600	0.667	0.333	223,004	0.400	0.400	0.400	0.600
66,075	1.000	1.000	1.000	1.000	223,061	1.000	1.000	1.000	0.667
67,079	0.700	0.400	0.700	0.400	225,022	0.833	0.167	1.000	0.833
68,077	0.810	0.333	0.571	0.857	227,092	0.333	0.333	1.000	0.167
71,046	0.571	0.381	0.476	0.143	228,076	0.800	0.400	1.000	0.400
80,099	0.500	0.600	0.500	0.300	238,011	0.700	0.300	0.700	0.300
86,016	0.500	0.800	0.600	0.200	246,009	0.143	0.429	0.143	0.333
92,059	0.667	0.167	0.667	0.333	247,085	0.667	0.167	0.667	0.667
100,075	1.000	0.000	1.000	0.667	249,061	0.833	0.500	0.833	0.333
101,027	0.500	0.200	0.800	0.600	253,036	0.591	0.227	0.985	0.818
101,084	0.476	0.476	0.524	0.667	257,098	0.800	0.600	0.800	0.700
101,085	0.214	0.107	0.321	0.571	260,058	1.000	1.000	1.000	0.000
103,029	0.400	0.600	0.533	0.267	271,031	0.667	0.500	0.833	0.333
105,019	1.000	1.000	1.000	0.000	279,005	0.714	0.429	0.714	0.238
112,056	1.000	0.000	1.000	1.000	285,022	0.800	0.733	0.800	0.533
113,016	0.700	0.400	0.800	0.700	289,011	1.000	0.000	1.000	0.333
113,044	0.800	0.400	0.800	0.700	299,091	1.000	0.667	1.000	0.667
117,025	1.000	0.333	1.000	0.000	306,052	1.000	0.667	1.000	0.333
117,054	0.806	0.306	1.000	0.222	311,068	0.833	0.500	1.000	0.667
118,031	1.000	0.167	1.000	0.500	334,025	0.528	0.528	0.528	0.556
123,057	1.000	0.333	1.000	0.167	347,031	0.833	0.167	0.833	0.333
130,014	0.700	0.400	1.000	0.600	372,019	0.571	0.571	0.929	0.429
130,066	1.000	0.000	1.000	1.000	374,020	1.000	1.000	1.000	0.000
134,008	1.000	0.000	1.000	1.000	374,067	0.833	0.667	1.000	0.667
134,052	1.000	1.000	1.000	0.000	384,022	0.667	0.619	0.667	0.429

100 test images, there are 79 images for which the $E_{2.1d}$ minimization method is better than the method [19] and there are 13 images for which the $E_{2.1d}$ minimization method works equally well to the method [19]. As presented in Table 2, the average accuracy of $E_{2.1d}$ minimization method is 83.4%, which is much better than the method [19] with only 48.3% average accuracy.

The 2.1D sketches of 100 test images shown in Fig. 8, recognized by minimizing the global energy $E_{2.1d}$, are shown with a color mapping in Fig. 9. The number of layers existed in 2.1D sketches extracted from the 100 test images is diverse from 2 to 6, and the statistic data of layer numbers is summarized in Table 3.

Limitation of the presented method. Our method currently mainly use the low level image features to determine the partial depth ordering of regions in the image. Due to the lack of high-level semantic information, in some cases our method may separate the regions of the same object into layers with different depth orders. One example is shown in Fig. 11, in which the over-segmented regions of the old man (Fig. 11(b)) were incorrectly judged as having different partial depth orders (Fig. 11(d)). Due to the lack of high-level semantic information, our method cannot merge them into a layer with SL relations, since these regions have significant difference in color and texture. To remedy this drawback, our method provides an optional

interactive operation. The user can draw some strokes or pick some points (Fig. 11(c)) to indicate that the underlying regions should be merged in a same layer with SL relations. One example of correct 2.1D sketch extracted after the interactive operation is shown in Fig. 11(e).

7. Conclusion

The 2.1D sketch is a layered image representation with a partial depth ordering, which can provide a good starting point for the downstream high-level vision tasks such as depth estimation (2.5D sketch), motion analysis, and image and video coding. Most previous 2.1D sketch extraction methods only relied on one or several local features, and can only handle simple or synthesized images. In this paper, we propose a global energy optimization framework which can incorporate different local features into a global unified solution. Two relations (F-G and SL) are modeled and local features are unified into a global form with a parameterized exponential function type. The experimental results on 100 selected images from the Berkeley data set (BSDS500) show that our proposed global solution has a good accuracy of extracting 2.1D sketches from real images.

Acknowledgments

The authors would like to thank the reviewers for their constructive comments that helped to improve this paper. This work was partially supported by the Natural Science Foundation of China(61322206, 61272228), the National Basic Research Program of China(2011CB302202) and National High-tech R&D Program of China(2012AA011801). The work of Y.J. Liu was supported in part by Tsinghua University Initiative Scientific Research Program (20131089252).

Appendix A. Appendix

In this appendix, we present in Table 4 the full original data of the accuracy of extracted 2.1D sketches by minimizing the energies E_{FG} (Eq. (25)), E_{SL} (Eq. (26)) and $E_{2.1d}$ (Eq. (23)) on 100 test images shown in Fig. 8, which is used to generate the statistical data summarized in Fig. 10.

References

- [1] M. Nitzberg, D. Mumford, T. Shiota, *Filtering, Segmentation, and Depth*, Springer Verlag Inc., New York, 1993.
- [2] M. Nitzberg, D.B. Mumford, The 2.1-d sketch, in: IEEE International Conference on Computer Vision (ICCV' 90), 1990, pp. 138–144.
- [3] E.-H. Adelson, Layered representations for vision and video, in: Proceedings IEEE Workshop on Representation of Visual Scenes, 1995, pp. 3–9.
- [4] J.-Y.-A. Wang, E.-H. Adelson, Layered representation for motion analysis, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR' 93), 1993, pp. 361–366.
- [5] J.-Y.-A. Wang, E.-H. Adelson, Representing moving images with layers, IEEE Trans. Image Process. 3 (5) (1994) 625–638.
- [6] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 898–916.
- [7] T. Coleman, A. Wirth, Ranking tournaments: local search and a new algorithm, ACM J. Exp. Algorithmics 14 (2010) 6:2.6–6:2.2.
- [8] A. Saxena, S.H. Chung, A.Y. Ng, 3-D depth reconstruction from a single still image, Int. J. Comput. Vis. 76 (1) (2008) 53–69.
- [9] J. Malik, Interpreting Line Drawings of Curved Objects, Ph.D. Thesis, Stanford University, CA, USA, uMI order no. GAX86-12762, 1986.
- [10] Q. Fu, Y.-J. Liu, W. Chen, X. Fu, The time course of natural scene categorization in human brain: simple line-drawings vs. color photographs, J. Vis. 13 (9) (2013), <http://dx.doi.org/10.1167/13.9.1060> (Article No. 1060).
- [11] S. Esedoglu, R. March, Segmentation with depth but without detecting junctions, J. Math. Imag. Vis. 18 (1) (2003) 7–15.
- [12] A. Torralba, A. Oliva, Depth estimation from image structure, IEEE Trans. Pattern Anal. Mach. Intell. 24 (9) (2002) 1226–1238.
- [13] L.-R. Williams, A.-R. Hanson, Perceptual completion of occluded surfaces, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR' 94), 1994, pp. 104–112.
- [14] E. Saund, Perceptual organization of occluding contours generated by opaque surfaces, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR' 99), 1999, pp. 1063–6919.
- [15] D. Hoiem, A.A. Efros, M. Hebert, Recovering occlusion boundaries from an image, Int. J. Comput. Vis. 91 (3) (2011) 328–346.
- [16] J. Wang, M. Betke, E. Gu, Mosaicshape: stochastic region grouping with shape prior, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR' 05), 2005, pp. 902–908.
- [17] X. Ren, C.C. Fowlkes, J. Malik, Figure/ground assignment in natural images, in: Lecture Notes in Computer Science (ECCV' 06), 2006, pp. 614–627.
- [18] S.X. Yu, T.S. Lee, T. Kanade, A hierarchical markov random field model for figure-ground segregation, in: Proc. 3rd Int. Conf. Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR'01), Springer-Verlag, 2001, pp. 118–133.
- [19] M.R. Amer, R. Raich, S. Todorovic, Monocular extraction of 2.1d sketch, in: Proc. Int. Conf. Image Processing (ICIP 2010), 2010, pp. 3437–3440.
- [20] N. Ahuja, S. Todorovic, Extracting texels in 2.1d natural textures, in: IEEE International Conference on Computer Vision (ICCV' 07), 2007, pp. 1–8.
- [21] X. Chen, Q. Li, D. Zhao, Q. Zhao, Occlusion cues for image scene layering, Comput. Vis. Image Understan. 117 (1) (2013) 42–55.
- [22] R.-X. Gao, T.-F. Wu, S.-C. Zhu, N. Sang, Bayesian inference for layer representation with mixed markov random field, in: Proc. 6th Int. Conf. Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR'07), 2007, pp. 213–224.
- [23] Y.-J. Liu, K. Tang, A. Joneja, Sketch-based free-form shape modelling with a fast and stable numerical engine, Comput. Graph. 29 (5) (2005) 778–793, <http://dx.doi.org/10.1016/j.cag.2005.08.007>.
- [24] C.-X. Ma, Y.-J. Liu, H.-Y. Yang, D.-X. Teng, H.-A. Wang, G.-Z. Dai, Knitsketch: a sketch pad for conceptual design of 2d garment patterns, IEEE Trans. Automat. Sci. Eng. 8 (2) (2011) 431–437.
- [25] M.-M. Cheng, F.-L. Zhang, N.J. Mitra, X. Huang, S.-M. Hu, Repfinder: Finding approximately repeated scene elements for image editing, in: ACM SIGGRAPH' 10, 2010, pp. 83:1–83:8.
- [26] Y.-J. Liu, Q.-F. Fu, Y. Liu, X. Fu, A distributed computational cognitive model for object recognition, Sci. China (Ser. F: Inf. Sci.) 56 (9) (2013), <http://dx.doi.org/10.1007/s11432-013-4994-3> (Article No. 092101(13)).
- [27] S. Joshi, A. Srivastava, W. Mio, X. Liu, Hierarchical organization of shapes for efficient retrieval, in: Lecture Notes in Computer Science (ECCV' 04), 2004, pp. 570–581.
- [28] C.-X. Ma, Y.-J. Liu, H.-A. Wang, D.-X. Teng, G.-Z. Dai, Sketch-based annotation and visualization in video authoring, IEEE Trans. Multimedia 14 (4) (2012) 1153–1165.
- [29] Y.-J. Liu, X. Luo, A. Joneja, C.-X. Ma, X.-L. Fu, D.-W. Song, User-adaptive sketch-based 3-d cad model retrieval, IEEE Trans. Automat. Sci. Eng. 10 (3) (2013) 783–795.
- [30] C.C. Fowlkes, D.R. Martin, J. Malik, Local figure-ground cues are valid for natural images, J. Vis. 7 (8) (2007) 2:1–2:9.
- [31] M.-M. Cheng, G.-X. Zhang, N.J. Mitra, X. Huang, S.-M. Hu, Global contrast based salient region detection, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR' 11), 2011, pp. 409–416.
- [32] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, N. Crook, Efficient salient region detection with soft image abstraction, in: IEEE International Conference on Computer Vision (ICCV' 13), 2013, pp. 1529–1536.
- [33] M. Crosier, L.D. Griffin, Using basic image features for texture classification, Int. J. Comput. Vis. 88 (3) (2010) 447–460.
- [34] R.M. Haralick, Statistical and structural approaches to texture, Proc. IEEE 67 (5) (1979) 786–804.
- [35] R.M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, IEEE Trans. Syst. Man Cybern. SMC-3 (6) (1973) 610–621.

- [36] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces, Tech. Rep. TR-95-012, Berkeley, CA (1995).
- [37] A. Janoch, S. Karayev, Y. Jia, J.-T. Barron, M. Fritz, K. Saenko, T. Darrell, A category-level 3-d object dataset: Putting the kinect to work, in: ICCV Workshop on Consumer Depth Cameras in Computer Vision 2011, 2011, pp. 1168–1174.
- [38] D.-E. Goldberg, [Genetic Algorithms in Search, Optimization, and Machine Learning](#), Addison-Wesley, Reading Menlo Park, 1989.
- [39] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.