Wang N, Ai HZ, Tang F. Who blocks who: Simultaneous segmentation of occluded objects. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 28(5): 890–906 Sept. 2013. DOI 10.1007/s11390-013-1385-6

Who Blocks Who: Simultaneous Segmentation of Occluded Objects

Nan Wang[1] (王 楠), *Student Member*, *IEEE*, Hai-Zhou Ai[1] (艾海舟), *Senior Member*, *IEEE* and Feng Tang[2] (汤 锋), *Member*, *IEEE*

[1] *Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*

[2] *Multimedia Interaction and Understanding Lab, HP Labs, Palo Alto, CA 94304-1126, U.S.A.*

E-mail: aaron.nan.wang@gmail.com; ahz@mail.tsinghua.edu.cn; feng.tang@hp.com

Received December 20, 2012; revised June 19, 2013.

Abstract In this paper, we present a simultaneous segmentation algorithm for multiple highly-occluded objects, which combines high-level knowledge and low-level information in a unified framework. The high-level knowledge provides sophisticated shape priors with the consideration of blocking relationship between nearby objects. Different from conventional layered model which attempts to solve the full ordering problem, we decompose the problem into a series of pairwise ones and this makes our algorithm scalable to a large number of objects. Objects are segmented in pixel level with higher-order soft constraints from superpixels, by a dual-level conditional random field. The model is optimized alternately by object layout and pixel-wise segmentation. We evaluate our system on different objects, i.e., clothing and pedestrian, and show impressive segmentation results and significant improvement over state-of-the-art segmentation algorithms.

Keywords object segmentation, occlusion reasoning, object graph, conditional random field, random forest

1 Introduction

Object segmentation is a fundamental task in computer vision. Different from conventional image segmentation problems, e.g., superpixelization[1-3] and interactive ones[4-5], object segmentation usually needs specific object knowledge to provide high-level information. Recently it receives the renewed interest partially due to the improvement of efficient object detection algorithms[6-9]. With the help of object detection algorithms, elaborate object models are proposed to facilitate the segmentation framework. Many of these models are designed for single object modeling, e.g., [10-13]. However, there are still numerous scenarios containing multiple objects, such as group images for social activities[14], crowd scenes in public and jammed traffic[15]. In these scenarios, multiple objects exist and might occlude each other densely. Therefore, a typical challenge for segmenting objects is the inter-object occlusion. And this challenge also implies that different from single object modeling algorithms, we should take object-wise information into consideration and model them jointly (as shown in Fig.1).

In this paper, we propose to estimate the shapes for multiple occluded objects with a global view of the scene. We treat each object as a node and model their interactions (blocking relationship) as edges. We use an object-based graph because the variation of object shapes in occlusion scenarios can be distinguished by their individual and context information. An object-based graph can give more direct understanding of

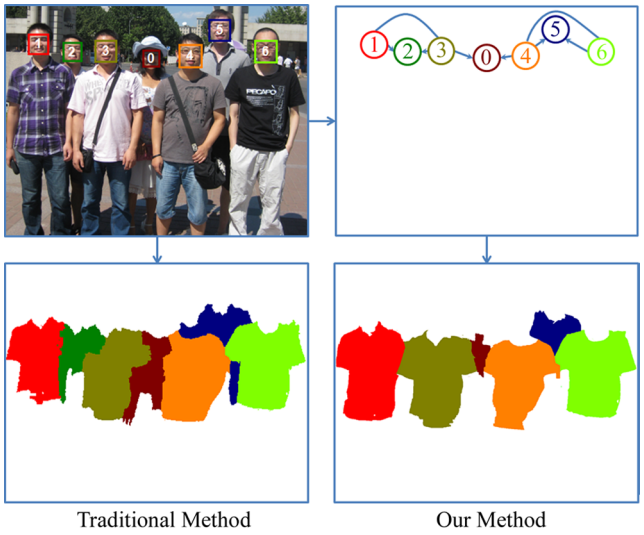


Fig.1. Taking clothing segmentation for example, for highly-occluded images, it benefits from the blocking information between neighbors.

Regular Paper

This work is supported in part by the National Natural Science Foundation of China under Grant No.61075026 and the National Basic Research 973 Program of China under Grant No.2011CB302203.

the whole scene. When compared with previous approaches, our contributions are twofold:

Blocking Model. As a key aspect of our method, a blocking model is proposed for occlusion reasoning. The conventional layered approach[16] attempts to solve the full layer ordering problem in a brute-force way. This method works well for images with a small number of objects, but is time-consuming for the ones with a large number of objects, since it involves a full permutation problem. However, there are many objects in a photo. We observe that it is unnecessary to estimate the layer order for each pair of objects. For example, the layer order for objects far from each other is not critical. So we cast the full layer order problem into a series of pairwise ones, which makes our algorithm scalable to a large number of objects. The blocking model is built on two kinds of heuristic features with the following intuitions: 1) relative positions of objects imply the blocking relationship; 2) unblocked objects contain more repetitive patterns. These two kinds of features complement each other in determining the blocking relationship. We base the second kind of features on appearance similarities. This makes the algorithm generalizable to many object types.

Joint Layout and Segmentation Model. We propose a joint model for object layout and segmentation. The layout model is defined on an object-based graph and depicts the scene with a global view of the image. It provides sophisticated shape priors for objects in the consideration of blocking relationship between them. These shape priors directly reduce the ambiguities in overlapping regions which would be difficult to solve for segmentation algorithms using only low-level features. The segmentation model is a variant of the conditional random field which integrates shape priors from high-level knowledge, higher-order constraints from superpixels and low-level appearance model from image features. The two models are optimized iteratively, which obtains more robust segmentation results.

We will briefly discuss related work in Section 2, describe the multi-object model in Section 3, and discuss how to learn the model in Section 4 and present the inference framework in Section 5. Experimental results of different object segmentation problems will be shown in Section 6, demonstrating significant improvements compared with conventional methods and the application to verify detection hypotheses with segmentations. Section 7 concludes the paper.

2 Related Work

There are innovative methods which combine top-down and bottom-up information for object segmentation. Borenstein and Ullman[10] proposed a patch-based algorithm to segment instances of a particular object category. Leibe and Schiele[17] used a probabilistic formulation and incorporated the knowledge about the recognized category. Levin and Weiss[12] simultaneously trained top-down and bottom-up cues and yielded high quality segmentation with a handful of patches. Winn and Jojic[18] presented an unsupervised algorithm which learns object class model for segmentation. However, these methods are designed for single-object segmentation and occlusions are not taken into consideration.

Winn and Shotton[19] proposed a "Layout Consistent Random Field" addressing partial occlusions by allowing invisible parts and the approach was extended to handle various viewpoints and scales in [20]. However, these methods do not exploit the global information provided by the image (e.g., object relative locations[21]). And it is not clear about the performance when using the grid-like parts in objects with high shape variability[10]. The papers of [22-23] simultaneously learn detector and segmentor using edge-based features, but their occlusion reasoning is based on the image coordinates, which makes their algorithms hard to handle cases where objects locate closely to each other[14].

Kumar *et al.*[11] proposed the OBJCUT method to get object specific prior for segmentation. However, their method can only deal with self-occlusion and cannot detect occluded objects. The method most similar to ours is the Layered Model proposed by Yang *et al.*[16], which represents ordered layers of object detections to estimate refined object appearance. They try to solve the full layer ordering problem and enumerate all possible orders, but this makes the algorithm intractable for a large number of objects.

Note that some other techniques can also deal with occlusions in a sense without explicit occlusion models, e.g., co-segmentation[24] and multi-phase labeling[25]. These techniques are out of the range of this paper and we will focus on the direct modeling of occlusions.

More recently, there are several attempts on combining detection and segmentation into the same framework. Ladicky *et al.*[26] described a probabilistic framework combining detection and segmentation based on solvable Robust P^n potentials[27]. Maire *et al.*[28] addressed the combination problem of image segmentation, figure/ground organization and object detection by solving a generalized eigen-problem. Partial occlusion can be addressed by these methods, but they do not supply sophisticated shape priors for objects, which is important for the cases with dense occlusions.

A preliminary version of this work appeared in [29] while significant improvements are made in this paper. We introduce a joint layout and segmentation model

and improve the original layout model by proposing more accurate object boundary approximation. Interleaved inference and learning algorithms are proposed to optimize and parameterize this joint model. We also provide additional experimental results on new datasets with different objects as well as additional analysis of our system components. And finally we describe how to use our segmentation results to verify the detection hypotheses.

3 Multi-Object Modeling

Inter-object occlusions in group images, crowded scene or jammed traffic are usually severe and dense. Our approach isolates object detection and segmentation algorithm because we believe that in such scenarios, more flexible approach should be used to locate objects with the consideration of the impact of occlusions, e.g., using a face detector to find occluded persons[14], or using a tracking algorithm to locate occluded objects.

3.1 Roadmap of the Proposed Approach

Now we describe our multi-object model for occlusion reasoning, object layout and segmentation through a high-level overview. Details of the approach will be discussed in the next subsections.

Given an image I with K pixels and O objects detected in it, our approach is aimed to infer object shape in pixel level for each of them considering the occlusion between neighbors. The whole approach is illustrated in Fig.2.

Object Level Notions. Let x_n denote the features extracted for each object, including object location and size, superpixel image[3], RGB features in inferred object region, where $n \in \{1, \ldots, O\}$. Suppose M candidate object shapes are generated for each object (as will be described in Section 4, these candidate shapes are generated by a modified random forest), and then $y_n \in \{1, \ldots, M\}$ called as *object layout*, represents the object shape selected for each object. The object pairs possibly occluded by each other are denoted as the edge set $\mathcal{E}_L$. For any pair of neighbor objects, binary blocking indicator b_{mn} denotes the blocking relationship. When object m blocks object n, $b_{mn} = 1$; otherwise $b_{mn} = 0$.

Pixel-Superpixel Level Notions. Let $\boldsymbol{I}_i$ be the feature vector associated with the i-th pixel where $i \in \{1, \ldots, K\}$ and $z_i \in \{0, 1, \ldots, O\}$ be the label assigned to pixel i. Similarly, let $\boldsymbol{I}_{K+r}$ be the feature vector associated with the r-th superpixel where $r \in \{1, \ldots, R\}$ and R is the superpixel number and $z_{K+r} \in \{0, 1, \ldots, O\}$ be the label assigned to super-

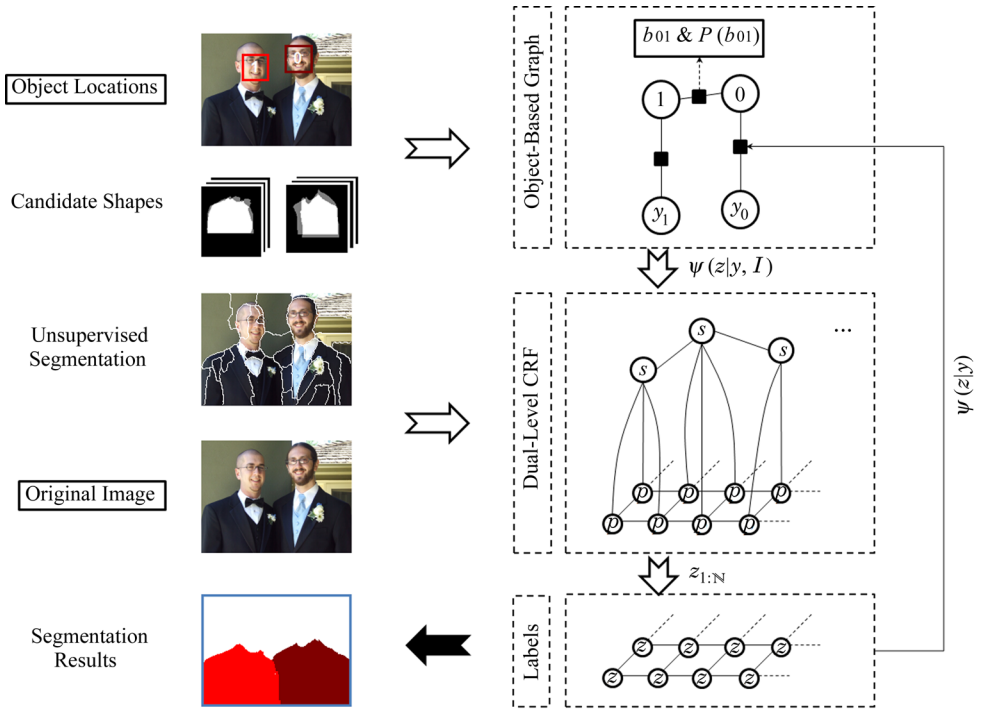


Fig.2. Illustration of our approach. The original image and object locations are input. The object-based graph and dual-level conditional random fields (CRF) are built and combined to pursue an appropriate explanation of both low-level image features and high-level object semantics. Final results are obtained from pixel-level labels.

pixel r. In the label set $\{0, 1, \ldots, O\}$, 0 represents the background and $\{1, \ldots, O\}$ are different objects.

Let $\boldsymbol{x} = (x_1, \ldots, x_O)$, $\boldsymbol{y} = (y_1, \ldots, y_O)$, and $\boldsymbol{z} = (z_1, \ldots, z_{K+R})$. Given an image I and an object layout $\boldsymbol{y}$, the problem of segmentation requires us to obtain the final labels $\boldsymbol{z}$ by maximizing the posterior probability $P(\boldsymbol{z}|I, \boldsymbol{y})$. In our algorithm, $P(\boldsymbol{z}|I, \boldsymbol{y})$ is assumed to be a conditional random field (CRF)[30] as in conventional settings. The posterior distribution $P(\boldsymbol{z}|I, \boldsymbol{y})$ is a Gibbs distribution and can be written as: $P(\boldsymbol{z}|I, \boldsymbol{y}) = \exp(-E(\boldsymbol{z}|I, \boldsymbol{y}))/Z_S$, where Z_S is the partition function, and $E(\boldsymbol{z}|I, \boldsymbol{y})$ is the energy function. Thus the final segmentation result can be obtained by:

$$\tilde{\boldsymbol{z}} = \arg\max_{\boldsymbol{z}} P(\boldsymbol{z}|I, \boldsymbol{y}) = \arg\min_{\boldsymbol{z}} E(\boldsymbol{z}|I, \boldsymbol{y}). \quad (1)$$

Note that Z_S is omitted in the last part of the above equation, because Z_S is the same for all possible labelings $\boldsymbol{z}$.

Currently, the object layout $\boldsymbol{y}$ is assumed to be known. Now we will show how to obtain $\boldsymbol{y}$. Intuitively, the object layout is affected by the true object shapes and the blocking relationship. Specifically, the shapes of all objects are expected to give better explanation to both cues. However, the true object shapes cannot be known. Here, we use the segmentation result $\boldsymbol{z}$ to approximate those shapes. So in our algorithm, given object-level features $\boldsymbol{x}$, blocking relationship $\boldsymbol{b}$ and the approximated object shapes $\boldsymbol{z}$, the optimal object layout $\boldsymbol{y}$ is obtained by maximizing the posterior probability $P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{b})$.

$$\tilde{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{b}).$$

To solve the above optimization problem, object shapes $\boldsymbol{z}$ and blocking relationship $\boldsymbol{b}$ are required. The former one can be obtained by optimizing (1). The remaining part is how to get the blocking relationship $\boldsymbol{b}$. In our algorithm, the blocking relationship $\boldsymbol{b}$ is predicted based on both self-similarities and the contextual information. Formally, it is obtained by maximizing the posterior probability $P(\boldsymbol{b}|\boldsymbol{x})$ as follows:

$$\tilde{\boldsymbol{b}} = \arg\max_{\boldsymbol{b}} P(\boldsymbol{b}|\boldsymbol{x}).$$

3.2 Object Segmentation

We use our previous algorithm of segmenting objects with high-order constraints from superpixel[31]. But different from the original one in [31] and the robust P^n model in [27], our algorithm here also incorporates superpixel-wise interactions which give more power to refine the segmentation results (a similar routine can be found in [32]). Formally, the segmentation energy function $E(\boldsymbol{z}|I, \boldsymbol{y})$ is defined as follows:

$$E(\boldsymbol{z}|I, \boldsymbol{y}) = \sum_{i=1:K+R} \psi_i(z_i|I, \boldsymbol{y}) + \sum_{(i,j)\in\mathcal{E}_S} \psi_{ij}(z_i, z_j|I), \quad (2)$$

where $\mathcal{E}_S$ is the edge set in the segmentation graph model including pair of four-connected pixels, pixels and their corresponding superpixels and pairs of superpixels sharing the same boundaries.

The unary potential ψ_i encodes the cost of assigning a label to the i-th pixel or superpixel. It is computed from the shape prior and appearance model both guided by object layout $\boldsymbol{y}$. Then the unary potential can be written as:

$$\psi_i(z_i|I, \boldsymbol{y}) = \theta_s \psi_i^s(z_i|I, \boldsymbol{y}) + \theta_a \psi_i^a(z_i|I, \boldsymbol{y}), \quad (3)$$

where θ_s and θ_a are parameters weighting the potentials obtained from object layout $\boldsymbol{y}$ (ψ_i^s) and color (ψ_i^a) respectively. These potentials take different forms for pixels and superpixels.

Shape Prior. For each pixel, the probability of assigning a label based on object layout $\boldsymbol{y}$ can be calculated as follows:

$$p_i(z_i) = \begin{cases} \prod_{n=1}^{O} p_i(0|y_n), & \text{if } z_i = 0, \\ \dfrac{p_i(1|y_m)}{\sum_k p_i(1|y_k)}(1 - p_i(0)), & \text{if } z_i \neq 0, \end{cases} \quad (4)$$

where $p_i(1|y_m)$ is the probability of the i-th pixel being foreground in the m-th object's shape mask. So intuitively, for pixel i, $z_i = 0$ if and only if it is background for all object shape masks. Thus the shape potential is computed as:

$$\psi_i^s(z_i) = \begin{cases} -\log p_i(z_i), & \text{if } i \leqslant K, \\ -\log \dfrac{\sum_{k\in R_{i-K}} p_k(z_i)}{|R_{i-K}|}, & \text{if } i > K, \end{cases} \quad (5)$$

where R_{i-K} is the pixel set in superpixel $i - K$ and $|R_{i-K}|$ is the number of pixels in this superpixel.

Appearance Prior. RGB histograms are extracted as the appearance model for both objects (background) and superpixels, which are denoted as $\mathcal{H}_m^{\mathrm{OBJ}}$ and $\mathcal{H}_r^{\mathrm{SPX}}$ respectively. The appearance potential is computed as:

$$\psi_i^a(z_i) = \begin{cases} -\log \mathcal{H}_{z_i}^{\mathrm{OBJ}}(\boldsymbol{I}_i), & \text{if } i \leqslant K, \\ -\log[\mathcal{H}_{z_i}^{\mathrm{OBJ}}, \mathcal{H}_{i-K}^{\mathrm{SPX}}]_{\mathrm{INT}}, & \text{if } i > K, \end{cases}$$

where $\mathcal{H}(\cdot)$ is the value of bin $\cdot$ in the histogram, and $[\cdot, \cdot]_{\mathrm{INT}}$ is the histogram intersection similarity between two histograms. We choose this histogram similarity criterion due to its compatibility to the histogram-based probability.

The pairwise potential ψ_{ij} encodes the interactions between pixels and superpixels. The basic idea is to encourage consistency between involved nodes in CRF. The form of ψ_{ij} is inspired by the data-dependent contrast model used in [33]:

$$\psi_{ij}(z_i, z_j) = \begin{cases} 0, & \text{if } z_i = z_j, \\ g_{ij}(z_i, z_j|I), & \text{if } z_i \neq z_j. \end{cases}$$

In our algorithm, $g_{ij}(z_i, z_j|I)$ is computed based on the type of nodes the edge involved in and defined as:

$$g_{ij}(z_i, z_j|I) = \begin{cases} \lambda_a \exp(-\beta_a \|\boldsymbol{I}_i - \boldsymbol{I}_j\|^2), & \text{if } i \leqslant K, j \leqslant K, \\ \lambda_b \beta_b \, |SP_{j-K}\,|, & \text{if } i \leqslant K, j > K, \\ \lambda_b \beta_b |SP_{i-K}\,|, & \text{if } i > K, j \leqslant K, \\ \lambda_c \exp(-\beta_c [\mathcal{H}^{\text{SPX}}_{i-K}, \mathcal{H}^{\text{SPX}}_{j-K}]_{\mathcal{X}}), & \text{if } i > K, j > K, \end{cases}$$

where $|SP_{\{i-K,j-K\}}|$ are the cardinalities of superpixels (superpixels are indexed after all pixels). $\beta_a, \beta_b, \beta_c$ are calibration parameters to normalize the core function value in each condition into a comparable range and $\lambda_a, \lambda_b, \lambda_c$ are parameters to weight pairwise potentials. Note that we isolate them in the middle two cases for consistency in formulation. g_{ij} is only reached when $z_i \neq z_j$. The other symbols are explained below.

In the first case, both nodes are pixels, so it is calculated as a normal contrast sensitive model. Its intuition is that two nearby pixels have higher probability to take the same label if they are similar to each other, so higher cost will be assigned if their labels are different. β_a is calculated as the inverse of the average of all possible $\|\boldsymbol{I}_i - \boldsymbol{I}_j\|^2$ which is the Euclidean distance between vector $\boldsymbol{I}_i$ and $\boldsymbol{I}_j$.

In the middle two cases, only one node is a pixel. The potential is computed based on the cardinality of the corresponding superpixel[27]. The calibration parameter β_b is calculated as the inverse of the average of all superpixel cardinalities.

In the final case, both nodes are superpixel. The potential is derived by generalizing the conventional contrast sensitive model. A similar intuition is that two superpixels have higher probability to be assigned the same label if they are similar to each other. The difference here is that we use Chi-Square histogram distance $[\cdot, \cdot]_{\mathcal{X}}$ to replace the Euclidean distance in pixel-wise ones. The calibration parameter β_c is calculated as the inverse of the average of all possible $[\mathcal{H}^{\text{SPX}}_{i-K}, \mathcal{H}^{\text{SPX}}_{j-K}]_{\mathcal{X}}$.

3.3 Layout Model

We assume that the object-graph is a Markov random field, so the object layout probability can be formulated as:

$$P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{b}) = \frac{1}{Z_L} \prod_n \varphi_n(y_n|x_n, z^n)^{\beta^u} \cdot \prod_{(m,n)\in\mathcal{E}_L} \varphi_{mn}(y_m, y_n|x_m, x_n, b_{mn})^{\beta^p_{mn}}, \tag{6}$$

where Z_L is the partition function to make sure that $P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{b})$ is a distribution, z^n is the segmentation results for the n-th object, and $\mathcal{E}_L$ is the edges in the object layout model which will be decided in runtime based on object locations and sizes.

The unary potentials φ_n measure the consistency between the object candidate shape and the actual boundaries. Suppose the y_n candidate binary shape (obtained by thresholding the object foreground-background mask) for object n is $s^n_{y_n}$, and then φ_n is defined as their Jaccard coefficient:

$$\varphi_n = \frac{|s^n_{y_n} \cap z^n|}{|s^n_{y_n} \cup z^n|}. \tag{7}$$

A uniformed unary parameter β^u is assumed for all unary potentials. Note that while obtaining the segmentation result z^n, superpixel information is incorporated. This is an essential part of our algorithm. If the segmentation is only performed in pixel level, the result obtained will be more similar to the supplied high-level shape prior $\boldsymbol{y}$, which means this unary potential in (3) cannot supply enough information to help the layout model jump out of local minima (e.g., caused by poor initialization). We will show more detailed analysis in Section 6.

The pairwise potentials penalize the conflict foreground part of the neighbor shapes and the blank space left in the overlapping region, and encourage the larger size of the unoccluded region. Let A be the overlapping region of two detected object boxes, and $s^n_{y_n}$ and $s^m_{y_m}$ be the candidate binary shape of the n-th and the m-th objects respectively. For clarity, we abbreviate $s^n_{y_n}$ and $s^m_{y_m}$ into s^n and s^m respectively. Then φ_{mn} is calculated as:

$$\varphi_{mn}(y_m, y_n|x_m, x_n, b_{mn}) = \exp(-\beta_a \varphi^a_{mn} - \beta_b \varphi^b_{mn} - \beta_c \varphi^c_{mn}),$$

where

$$\varphi^a_{mn} = \sum\nolimits_{k\in A} s^m_k s^n_k / |A|, \tag{8}$$

$$\varphi^b_{mn} = \sum\nolimits_{k\in A} (1 - s^n_k)(1 - s^m_k) / |A|, \tag{9}$$

$$\varphi^c_{mn} = \frac{\sum_k s^n_k b_{mn} + \sum_k s^m_k (1 - b_{mn})}{\sum_k s^n_k + \sum_k s^m_k}, \quad (10)$$

where s^n_k is the binary indicator of the pixel as the foreground in A, the same with s^m_k and $|A|$ is the size of the overlapping region. In (8), the numerator accumulates the number of pixels where occluded and unoccluded shapes are both foregrounds. This conflict brings the ambiguity for segmentation. In (9), we penalize the blank space left in the overlapping region by the neighbor shapes. This potential is required because it corrects the bias to smaller object shape from the other two pairwise potentials (φ_a and φ_c). These two functions are illustrated in Fig.3.

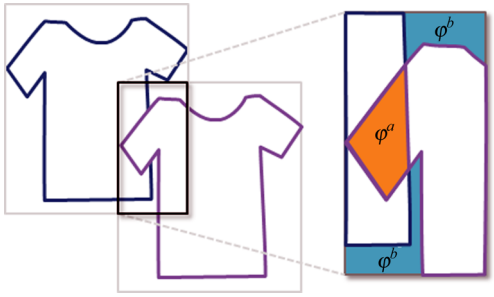


Fig.3. Illustration of pairwise potential functions of the layout model.

The former two potentials do not distinguish unoccluded and occluded objects. In (10), we encourage a bigger foreground size of the unblocked object in the overlapping region by calculating their foreground size proportions.

Since the blocking relationship might be different for different object pairs as shown in Subsection 5.2, β^p_{mn} is used to represent the blocking relationship confidence. This is different from unary parameter β^u. So we set $\beta^p_{mn} = P(b_{mn}|x_m, x_n)$, where $P(b_{mn}|x_m, x_n)$ is the blocking probability calculated as (12) in Subsection 4.2.

3.4 Blocking Model

Heuristically, the object at a lower location is more likely to block those at a higher location. However, estimating the occlusion relationship by relative object location is not always reliable. For example, in Fig.1, the 0th and 2nd face locations are lower than their neighbors, but they actually stand behind them.

We assume that the blocking relationships are independent for all neighboring pairs. So $P(\boldsymbol{b}|\boldsymbol{x})$ can be factored as:

$$P(\boldsymbol{b}|\boldsymbol{x}) = \prod\nolimits_{(m,n)\in\mathcal{E}_L} P(b_{mn}|x_m, x_n).$$

The blocking probability between pairs of objects $P(b_{mn}|x_m, x_n)$ is obtained by random forest. The details can be found in Subsection 4.2.

4 Model Learning

We learn the model using a supervised algorithm based on manually labeled dataset. Object shapes are labeled respectively, with foreground as 1 and background as 0. The blocking relationship is inferred from the segmentation labeling results. For objects nearby, we accumulate the foreground size of them in the overlapping region A and determine the ground-truth blocking relationship b^g_{mn} and confidence $P(b^g_{mn})$ based on the foreground proportion:

$$b^g_{mn} = \sum_{k\in A} \tilde{s}^m_k > \sum_{k\in A} \tilde{s}^n_k,$$

$$P(b^g_{mn}) = \frac{\max\left(\sum_{k\in A} \tilde{s}^m_k, \sum_{k\in A} \tilde{s}^n_k\right)}{\sum_{k\in A} \tilde{s}^m_k + \sum_{k\in A} \tilde{s}^n_k},$$

where $\tilde{s}^m_k$ is the binary label of the k-th pixel of the m-th object. The intuition of blocking relationship ground-truth determination is that the unblocked object is the one with larger foreground in the overlapping region. And the confidence is the proportion of the foreground pixels in this region.

4.1 Object Shape Modeling

As stated in Section 3, M object shapes are sampled for each individual object. These candidate template shapes are learned using random forest, since it is very efficient and straightforward to implement[34]. The weak feature is designed based on the self-similarity properties of object (Fig.4). Specifically, rectangle patches are densely sampled in the object region. The weak feature is defined as the histogram similarity (inverse $\mathcal{X}^2$ distance) between patches from possible foreground region. In each internal node the training data is split by comparing a weak feature with randomized thresholds.

Each decision tree is a binary tree (illustrated in Fig.4), where each leaf (Fig.4) records the object shape fallen into it. In contrary to conventional decision tree, the information gain is not available for a high-dimensional clothing shape. Actually, the object shape model is used to sample the object shape distribution with low-level features bridging the semantic gap between them. The model is expected to maintain compact shape distribution for the terminal nodes and hence generate specific shape for a testing image. So we propose to use the shape consistency to calculate

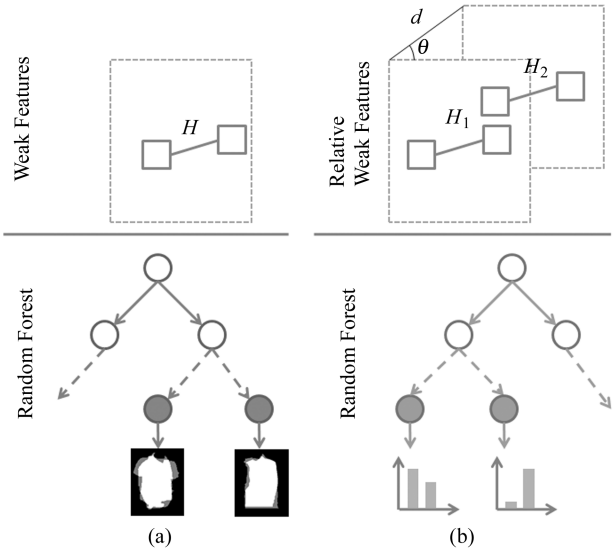


Fig.4. (a) Weak features and decision tree for shape sampling. (b) Weak features and decision tree for blocking prediction.

the decision tree split metrics. The consistency between any two binary shapes is defined as their Jaccard coefficient:

$$c(\tilde{s}^m, \tilde{s}^n) = \frac{\mathrm{FG}(\tilde{s}^m) \cap \mathrm{FG}(\tilde{s}^n)}{\mathrm{FG}(\tilde{s}^m) \cup \mathrm{FG}(\tilde{s}^n)}, \tag{11}$$

where $\mathrm{FG}(\cdot)$ is the set of foreground pixels of binary shape.

All object shapes are normalized with the same size based on the detection results. Let $\tilde{\boldsymbol{s}} = \{\tilde{s}^1, \ldots, \tilde{s}^S\}$ be a shape set, and then the consistency of $\tilde{\boldsymbol{s}}$ is defined as the minimum among all pairs of shapes:

$$c(\tilde{\boldsymbol{s}}) = \min_{m,n \in \{1,\ldots,S\}} c(\tilde{s}^m, \tilde{s}^n).$$

The feature in each internal node is selected as the one producing the most consistent partition, i.e.:

$$\tilde{H} = \arg\max_{H,\tau} c(\tilde{\boldsymbol{s}}^+_{H,\tau}, \tilde{\boldsymbol{s}}^-_{H,\tau}),$$

where $\tilde{\boldsymbol{s}}^+_{H,\tau}$ and $\tilde{\boldsymbol{s}}^-_{H,\tau}$ are the subsets produced by comparing the response of H with threshold of τ. In internal nodes, H and τ are both randomly selected[34]. The learning process is halted when the remaining object shapes are consistent enough or there are no weak features that can gain significant consistency improvement. The consistency threshold is manually set as 0.9 keeping high modeling performance and moderate tree size.

A set of decision trees is learned and each tree is trained on a different sampled dataset[34]. Since the features used in our model imply the self-similarity of objects, the output object shape model can address the occlusion moderately (referring to Subsection 6.2.2).

For an input image, the algorithm traverses each decision tree based on weak features and final comes to a leaf node where one or several shape samples are maintained. A probabilistic map is generated for each leaf node and used in following procedures.

4.2 Blocking Distribution Modeling

The blocking distribution models the pairwise blocking relationship between objects. This is essentially a binary classification problem for each object pair. Random forest[34] is used as the classifier.

For each decision tree, a blocking distribution between neighbors is learned based on *relative* features, as shown in Fig.4. Here we set the upper-right person as the origin. The first set of features is the relative object location, including relative angle θ and relative distance δ. The second set of features is related to the one used in Subsection 4.1. As shown in Fig.4, for neighboring objects, H_1 and H_2 involve one patch of the patch pair in the overlapping region and the other outside. The difference $\Delta(H_1, H_2) = H_1 - H_2$ is used as the weak feature. The intuition is that the appearance of the overlapping region should be more similar to the unblocked object. Still, in the internal nodes, these continuous features are used by comparing with the randomly-selected thresholds and the one with the maximum information gain is selected. The leaf nodes maintain the probability that object a blocks object b using a non-parametric model, i.e., we count the blocking and unblocking frequency from training samples fallen into them.

A set of decision trees are learned and each tree is trained on a different sampled datasets. The final blocking distribution is obtained by averaging outputs of all trees:

$$P(b_{mn}|x_m, x_n) = \frac{1}{T} \sum\nolimits_{t=1}^{T} P_t(b_{mn}|x_m, x_n), \tag{12}$$

where T is the number of blocking decision trees and $P_t(b_{mn}|x_m, x_n)$ is the blocking distribution obtained by a single decision tree.

5 Model Inference

Given an image and its object detection results, we are aiming at inferring the blocking relationship, object layout and segmentation results simultaneously.

5.1 Data Preparation

The possible object region is obtained by object detection or other assistant techniques, e.g., discriminative part detection or object tracking. The edges $\mathcal{E}_L$ for layout model are determined for the neighbor rectangles with large enough overlapping region. We use

a threshold 10% of the minimal rectangle which preserves most real blocking pairs while eliminates those which are not so affected by blocking relationship.

For each object, M candidate objects are obtained using the object shape model learned in Subsection 4.1. For each pair of neighbors, the blocking distribution is obtained using (12).

5.2 Blocking Relationship Determination

We assume that the blocking relationships between any pairs of objects are independent. Then the optimal blocking relationship $\tilde{\boldsymbol{b}}$ can be obtained by maximizing the individual local blocking distribution, i.e.,

$$\begin{aligned}\tilde{\boldsymbol{b}} &= \arg\max_{\boldsymbol{b}} P(\boldsymbol{b}|\boldsymbol{x}) \\ &= \prod\nolimits_{(m,n)\in\mathcal{E}_L} \arg\max_{b_{mn}} P(b_{mn}|x_m, x_n).\end{aligned}$$

Yang *et al.*[16] searched the optimal ordering from all possible permutations. In our method, we just use the Maximum a posteriori (MAPs) solution, as we observe that the uncertain blocking relationship usually occurs when neighbors locate closely but do not block each other (Fig.5). It means that "who blocks who" is not critical in these cases. We incorporate the confidence $P(b_{mn}|x_m, x_n)$ to the layout model in (6) as the pairwise weight, which decreases the effect of neighbor constraints when the relationship is not so confident. Thanks to the way of incorporating blocking relationship as soft constraints, our algorithm performs well even though we do not permute all possible blocking relationship.

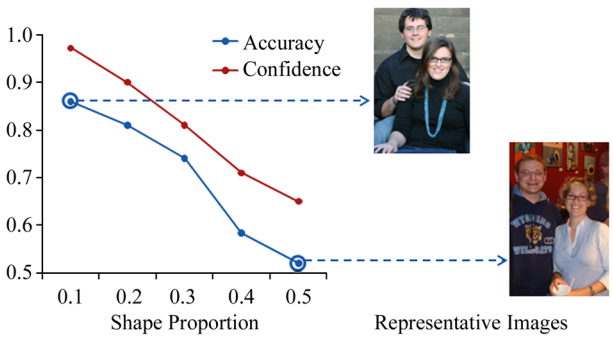


Fig.5. Detailed view of the relationship between blocking prediction accuracy, confidence and its importance for layout modeling.

5.3 Segmentation and Layout Sampling

The object segmentation results $\boldsymbol{z}$ and object layout $\boldsymbol{y}$ depend on each other. Here we use a coordinate ascent algorithm for inference. It iterates between two procedures: layout sampling and object segmentation.

5.3.1 Layout Sampling

Given blocking configuration $\boldsymbol{b}$, the MAP solution of (6) can be approximated by Loopy Belief Propagation[35]. However, in our method, we propose to sample the layout instead of a single MAP solution. The two reasons are as follows.

The first is as suggested by Kumar *et al.*[11], when using the top-down model as latent variable to guide the bottom-up image segmentation, the expectation of the log likelihood of an Markov random field (MRF) can be efficiently optimized by a single graph cut optimization (as shown in the following subsection, the dual-level CRF defined in (2) can be optimized by alpha-expansion). Importantly, they argued that multiple samples are necessary for some difficult cases in which RGB distribution of the background is similar to that of the object.

The other is from an insight of our model. In Subsection 4.1, an object shape random forest is learned for each person individually. Random forest increases its generalization accuracy nearly monotonically with respect to the number of decision tree[36]. The final voting procedure of outputs will make the model more robust to noise. However, the MAP inference selects only one result from all tree outputs, and this will decrease the generalization ability (e.g., data noise, incorrect graph structure or approximate solution of graph model). From this point of view, sampling from the layout model, where the interaction of the neighbors is taken into account, is a novel and effective voting method to improve the individual random forest performance.

For convenience of sampling, the sum-product algorithm is used for MAP inference[37] and then multi-solutions are sampled around the local (global for tree-graph) minimum by Gibbs sampling. Here, the posterior distribution of layout solution is approximated as[11]:

$$p(\boldsymbol{y}) = \frac{\prod_{(m,n)\in\mathcal{E}_L} B_{mn}(y_m, y_n)}{\prod_m B_m(y_m)^{q_m-1}},$$

where B_m and B_{mn} are the unary and pairwise beliefs calculated by the sum-product algorithm. q_m is the neighbor number of the m-th node.

5.3.2 Object Segmentation

Conditional random field with higher order constraints has been used in segmentation problem and gets significant achievement in recent years[27,32]. The pair-wise potentials in (2) take the form of Potts Model and thus are submodular[38], so we can optimize the dual-level CRF using the alpha-expansion algorithm[4,38-39] efficiently.

The procedure is performed as follows: first, we only use the shape prior in (3) and set $\theta_a = 0$, and get the segmentation results, which actually align to the image based on the pixel and superpixel information. Second, we learn the appearance model $\{\mathcal{H}_m^{\mathrm{OBJ}}\}_{m=0}^{O}$ and turn on the appearance prior, and then run alpha-expansion again to obtain the final labeling results by incorporating appearance information. Additional, we can improve the result by refining the appearance model iteratively as suggested in [4].

Superpixels are used as soft constraints in our algorithm, so their inaccuracy in approximating true object boundaries can be decreased. The final segmentation result is obtained from pixel-level labeling, i.e., only $\{z_i | i \leqslant K\}$ are used.

5.3.3 *Interleaved Optimization*

The optimization of $\boldsymbol{z}$ and $\boldsymbol{y}$ relies on each other. An interleaved algorithm is used for the optimization and detailed in Fig.6.

Input: image and object locations I and $\boldsymbol{x}$
Output: blocking relationship between objects $\boldsymbol{b}$, object layout $\boldsymbol{y}$ and segmentation results $\boldsymbol{z}$
1. Obtain blocking relationship $\boldsymbol{b}$ and their confidence
2. Generate multiple shape candidates for each object
3. Obtain intial shape prior by averaging all candidates
4. **for** each iteration t
   optimize (5) to get $\boldsymbol{z}$
   sample distribution (12) to get $\boldsymbol{y}$
5. **end for**

Fig.6. Interleaved optimization algorithm.

Note that initially both $\boldsymbol{z}$ and $\boldsymbol{y}$ are unknown. So we use the average of all candidates generated by the object shape model (Subsection 4.1) as the initial shape prior. The reason is that the object shape model is built based on self-similarity features which can reason the occlusion in part.

6 Experiments

The proposed method is a general framework which is applicable for segmenting group objects of many types. In this paper, it is applied to segmenting people clothing and pedestrians. We analyze the impact and performance of each component of our system in clothing segmentation of group images[14]. In these images heavily-occluded people can be located by a face detector and thus we can evaluate the blocking information excluding the influence of other factors. We also show results on segmenting pedestrians and compare our algorithm with state-of-the-art methods.

6.1 Learning Configurations

Random forests for object shape modeling and blocking distribution modeling are implemented similarly for every object class. Both of the random forests consist of 25 decision trees. For each node of them, $\sqrt{F}$ weak self-similarity features[40] are randomly selected (relative object location features in blocking distribution modeling are used for all nodes), where F is the weak feature number. The size of rectangles used in self-similarity features is 4×4, and they are densely sampled with step 2. Twenty thresholds are generated randomly for each feature and then the optimal one is selected. No pruning is applied for decision trees.

Weights of the model required to be specified are: shape weight θ_s, appearance weight θ_a, pairwise weights λ_a, λ_b and λ_c in segmentation energy (2), unary and pairwise weights β^u, β_a, β_b and β_c in layout energy (6). The other parameters are mostly used for calibration and their calculations have been specified. In general, the learning of these weights is not an easy problem. Since the inference algorithm is performed iteratively, we propose to learn the weights for each iteration respectively in Fig.7. Since the optimization of the whole model is slow, a heuristical approach (similar to [27]) is used to search for the weights. We learn weights one by one and fix the former weights as the optimal values when learning the latter ones. The order of parameters learning is $(\theta_a, \lambda_a, \lambda_b, \lambda_c)$ for segmentation model and $(\beta_a, \beta_b, \beta_c)$ for layout model. A coarse-to-fine searching procedure is performed for each weight. In the coarse level, weights are searched for in $\{2^{-5}, 2^{-4}, \ldots, 2^4, 2^5\}$, while in the fine level, weights are searched for around the optimal value from the coarse level, i.e., $\{2^{w-2}, 2^{w-1.75}, \ldots, 2^{w+1.75}, 2^{w+2}\}$, where w is the optimal value from the coarse level.

Input: training images and groundtruth, maximum number of iterations T
Output: model parameters: segmentation weights $\{(\theta_a^t, \lambda_a^t, \lambda_b^t, \lambda_c^t)\}_{t=1}^{T}$ and layout weights $\{(\beta_a^t, \beta_b^t, \beta_c^t)\}_{t=0}^{T}$
1. Learning layout weights for $t = 0$ using the initial prior from object shape models (Subsection 4.1) over the validation set
2. **for** iteration $t = 1{:}T$
   Learning $(\theta_a^t, \lambda_a^t, \lambda_b^t, \lambda_c^t)$ using layout weights $(\beta_a^{t-1}, \beta_b^{t-1}, \beta_c^{t-1})$ by searching over the validation set
   Learning $(\beta_a^t, \beta_b^t, \beta_c^t)$ using segmentation weights $(\theta_a^t, \lambda_a^t, \lambda_b^t, \lambda_c^t)$ by searching over the validation set
3. **end for**

Fig.7. Iterated parameters learning algorithm. In inference, only the ratios between model parameters matter. So we fix one parameter of layout weights and segmentation weights as 1 and learn the other ones, i.e., $\theta_s = 1$ and $\beta^u = 1$ for all iterations. We omit the two weights in the output.

6.2 Clothing Segmentation and Analysis

First of all, we evaluate our method on the public available dataset of group images[14]. All the images with high occlusions between neighbors are downloaded from Flickr by different key words: wedding, family and group images. We manually label 281 images with 1 051 persons. We partition the ground truth dataset into two halves randomly and train our model using one half and test it with the other. We detect face location and scale images using a face detector[41]. When modeling each person, we normalize the image to make the face with the size of 24×24. The skin is roughly filtered off using the skin color model extracted from the face.

6.2.1 Object Shape Modeling

While building the object shape model, we observe that the background occupies much larger proportion than the foreground since many people's clothes are occluded. So we define the consistency between binary shape pairs based on the foreground as (11). The out-of-bag accuracy of the model is illustrated in Fig.8 with respect to the number of trees. The curve shows that the combination of weak features can improve the accuracy significantly. The second added tree can give an out-of-bag accuracy improvement of 7.2%.

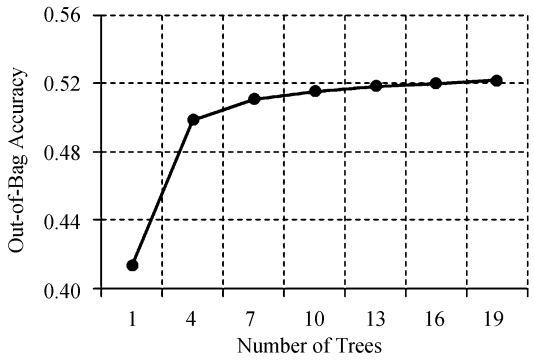


Fig.8. Out-of-bag accuracy of the object shape model with respect to different number of trees.

6.2.2 Blocking Modeling

The blocking performance is evaluated for the ones whose confidence $P(b_{ij}^g) > 0.7$, since the blocking relationship is not critical for the others, as shown in Fig.5.

Experiments are performed with different configurations: only face information (i.e., relative distance d and angle θ), only clothing information (i.e., differences of patch-wise similarities $\Delta(H_1, H_2)$) and both of them (our full blocking model). Based on the curves in Fig.9, we can observe that face information achieves baseline accuracy, since it portrays the basic structure of group people. However it cannot handle the occlusion among several people abreast, which limits its capability for blocking relationship prediction. Moreover, its performance will not increase with respect to the number of trees, since actually there are not so many features available for our random selection.

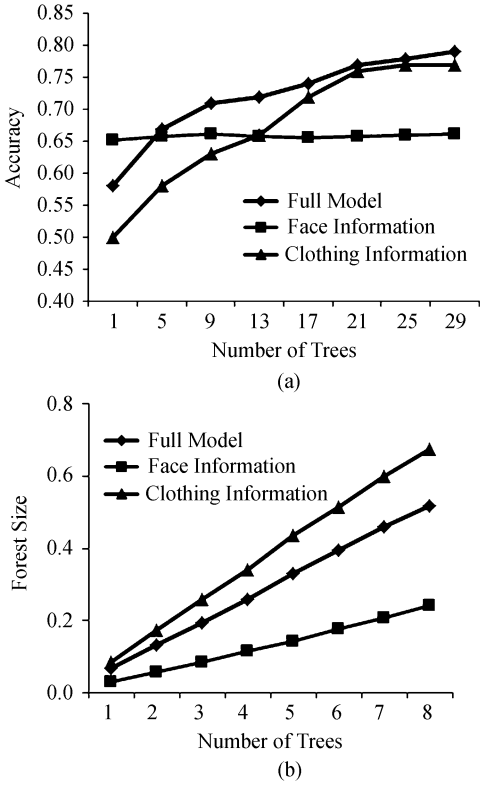


Fig.9. Detailed comparison of blocking models. (a) Blocking accuracy with respect to the number of trees. (b) Forest size with respect to the number of trees.

Both clothing information and our full model can achieve better accuracy with a large enough number of trees. However, the experiments show that to achieve the same performance, the full model needs fewer trees. In addition, as shown in Fig.9(b), the sizes of the random forest based on both kinds of information are much smaller than that based on only clothing information. So the conclusion is that the clothing information can capture the blocking relationship between persons, while face information can enhance it by reducing the required number of trees and nodes. As a comparison, we implement a method similar to that in [42] which uses relative size and location to infer the blocking relationship. However, its MAP result only achieves 56.2% accuracy. The reason may be that in our dataset, people stand closer, which reduces the power of object relative size and, moreover, no self-similarity features are explored in their method[42].

We also show how blocking accuracy affects the segmentation accuracy in Fig.10. As we show in Fig.5,

blocking accuracy changes with respect to its importance which can be defined by the portion of the smaller/larger shape occupied in the overlapping region. The curves in clothing segmentation shows that the blocking accuracy decreases when the shape portion increases as in such cases blocking relationship is not so obvious. However, the segmentation accuracy does not decrease monotonously. The main reason is that when blocking relationship is not obvious, people usually just stand close to each other and image-level cues are powerful enough to distinguish them. Such phenomenon cannot be found in pedestrian segmentation because occluded cases are much harder to locate and pose and appearance affects more.

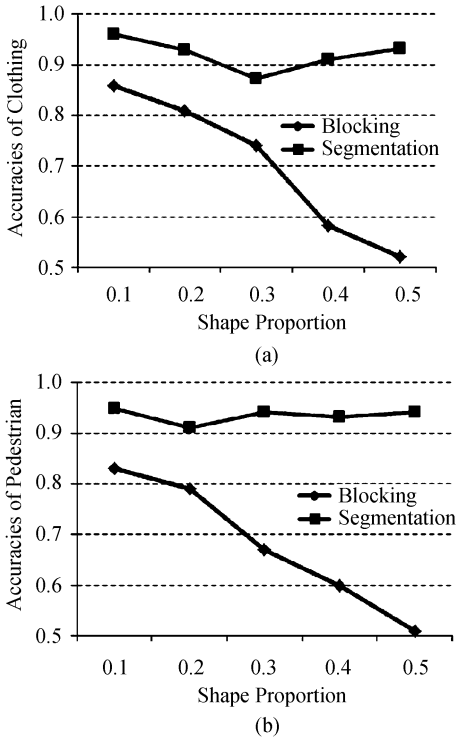


Fig.10. Relationship between blocking and segmentation accuracy.

We also conduct experiments to compare the performance of MAP result of object layout and its n-best samples in Table 1, where Forest means using only shape random forest, while the other two methods use the complete system with different layout optimization strategies. Using only the MAP result slightly improves the performance of random forest in clothes segmentation and moderately in pedestrian segmentation. The reason might be that pedestrian shapes vary largely due to pose variations and average shapes from random forest without any prior bias might ignore thin parts, e.g., arms and legs. However, in both cases, sampling with object shape priors (segmentation results from the dual-level CRF incorporated with sophisticated color model in pedestrian segmentation) boosts the accuracy effectively.

Table 1. Performance Comparison of Object Layout on Clothing and Pedestrian

	Forest (%)	MAP (%)	Sampling (%)
Clothing	90.7	91.1	93.8
Pedestrian	91.2	93.0	94.3

6.2.3 *Segmentation*

Segmentation accuracy is evaluated as the pixel-level consistency based on the manually-labeled data. Detailed experimental results are reported in Table 2 with different settings. The results show that all components contribute to the final results and we give detailed analysis as follows.

Shape Prior. In this setting category, we use different shape priors to guide the bottom-up segmentation and show our algorithm's capability in modeling the blocking relationship by comparing it with these methods. The first kind of shape prior, "Single Mask", is the probabilistic map of clothing shape counting on all training data. The second kind of shape prior, "Forest", is the probabilistic map of clothing shape counting on shapes output from the object shape model (Subsection 4.1). When implementing these two methods, no iterations between layout and segmentation inference are performed. "Single Mask" incorporates no blocking information while providing the high-level shape information. However, since self-similarity features are integrated in the object shape model, "Forest" includes implicit blocking information in its outputs and thus gives an observable improvement (2.5%) compared with the naïve "Single Mask" shape prior (an example can be found in Fig.11 marked with black rectangle).

Segmentation Potentials. In this setting category, we investigate the impacts of appearance and higher-order

Table 2. Segmentation Accuracy in Different Settings

Segmentation	Shape Prior (%)		Segmentation Potentials ($nIte = 3$) (%)		Iterated Inference (%)		
Accuracy	Single Mask	Forest	¬A&H	¬H&A	$nIte = 1$	$nIte = 2$	$nIte = 3$
Clothing	88.2	90.7	89.1	91.3	92.5	93.6	93.8
Pedestrian	89.3	91.2	90.0	93.8	92.0	93.9	94.3

Note: ¬A&H means appearance cues are turned off while higher-order cues on. ¬H&A means higher-order cues are turned off while appearance cues on. $nIte = 3$ means our full model.

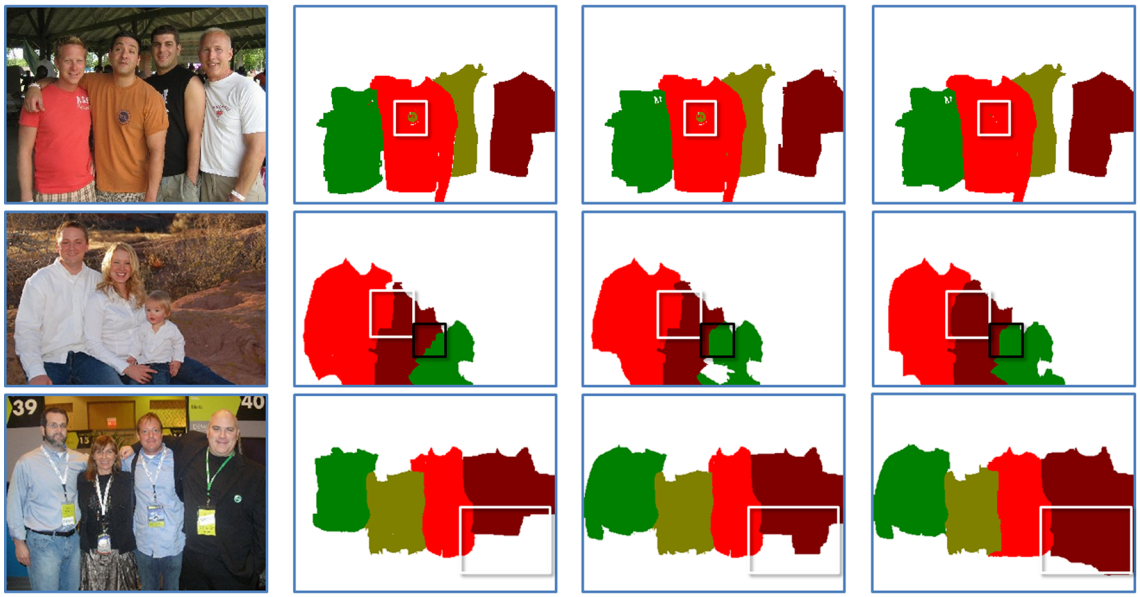


Fig.11. Clothing segmentation results for group images. The segmentation results in rectangles show the qualitative improvements with different algorithm settings. (a) Input image. (b) "Single Mask" prior. (c) "Forest" prior. (d) Full model.

potentials in the segmentation framework. We turn off either of them in (2) and perform three times of iterations between layout and segmentation inference as in our full model (shown in Fig.12). As shown by the data, appearance cues play an important role in the framework. Without appearance information, the segmentation accuracy drops from 93.8% to 89.1%. Its importance is expectable for two reasons. The first one is that people's clothes in our dataset are mostly distinguishable in appearance from each other and from the background. The second one is that the layout model depends on the inferred object shapes (recalling the unary term (7) in the layout model (6)), so without appearance model the inferred object shapes are usually just local disturbance of the shape constraint and cannot affect the high-level information provided by the layout model and consequently reduce the power of our layout model. So similar to "Forest" shape prior, appearance cues also contribute to inferring the blocking relationship. However, we have to note that the assumption of distinguishable clothes appearance is not always correct. In those cases, blocking information from the blocking model is useful (the relative object location information used in the blocking model does not depend on local appearance).

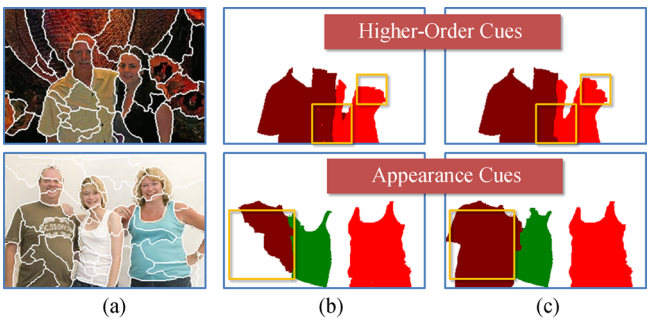


Fig.12. Qualitative comparison when different informative cues are turned on and off. (a) Input image. (b) Without the cues. (c) With the cues.

A noticeable improvement (2.5%) can be seen when incorporating higher-order information. Its impact is similar to the appearance cue, which is aiding the layout model (6) by proposing more sophisticated object shapes to adjust the shape prior resampled by the layout model.

Interleaved Inference. The inference algorithm is performed iteratively and the accuracy is increased consistently. The increment in the first iteration is the most dramatic (about 1.8%) and decreases while converging. The inference involves multiple graph cuts (in alpha-expansion) for the large graph in segmentation, so we set the maximal iteration number as 3 due to the heavy computational load. By incorporating appearance and higher-order information in the segmentation framework, the resulting object shape could refine the supplied shape constraints. There may be doubt why the refined shape constraints are necessary. The reason is that sophisticated shapes can reduce ambiguities in appearance. For example, refer to the region marked by white rectangle in the second row of Fig.11. Persons in this picture wear clothes with similar colors, but the blocking model incorporates relative object location in-

formation and prefers a larger size of the unblocked person, so our full model can generate accurate segmentation even in such cases. In addition, in (9), we prefer larger covering of object foreground in the overlapping region, so our blocking model can handle strange shape (illustrated in the last row of Fig.11).

Comparison with Original Method. Our full model improves 1.3% compared with the previous method[29]. Some qualitative results are shown in Fig.13. The new algorithm reduces disorders due to arms and faces by using a rough skin filter and is more robust to appearance ambiguities by iterating between the object layout and bottom-up segmentation.

Fig.13. Comparison experiments with original method[29]. (a) Input image. (b) Original result. (c) New result.

Comparison with GrabCut. We also compare our method with GrabCut[4] (implemented by OpenCV[40]). The initial rectangle for GrabCut is generated based on the global clothing template (rectangles in Fig.14). Without blocking information, GrabCut might fail for blocked people when the appearance is ambiguous. Severe errors may occur when people wear clothes with color distribution similar to that of the background, e.g., the second row in Fig.14. Our algorithm improves over the GrabCut by updating the shape information while updating the appearance.

Computational Efficiency. Our blocking model decomposes the full ordering problem into local pair-wise ones, so the proposed algorithm can deal with a large number of objects. For each iteration in Fig.7, the optimization of (6) runs in $O(OM^2T_B)$, where T_B is the iteration number of belief propagation; the optimization of (2) runs in $O(O(K+R)^3T_\alpha)$①, where T_α is the iteration number of alpha-expansion. We can see that both of the time complexities are linear with the object number O. Note that object boxes are large enough to cover the whole object region, so a possible acceleration

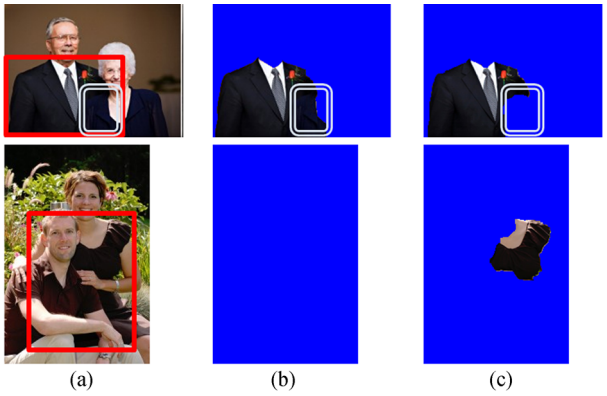


Fig.14. Comparison experiments with GrabCut. The initial rectangle (red one) is generated by our algorithm. (a) Input image. (b) CrabCut. (c) Our method.

is applying the algorithm only in object boxes (extended with a narrow band outside object boxes for background model estimation) instead of the entire image. In this way, objects far from each other are separated and most background pixels are excluded. But the actual running time of our complete algorithm still varies with the object size and object number. Typically, for a 180×180 object, our algorithm runs in about 3.5 seconds on an Intel® P4 2.33 GHz machine. The most time-consuming part is the iterative α-expansion algorithm which takes 2.9 seconds approximately. GrabCut[4] with three iterations (the last two iterations are initialized with mask for acceleration)② takes about 1.8 seconds. Our original method[29] takes about 2.1 seconds since it uses less min-cut/max-flow iterations. For more objects, the running time of the proposed method linearly increases with respect to the object number.

Examples with a larger number of objects are shown in Fig.15. As can be observed, even with heavy occlusions, these objects can still be accurately segmented. More results can be found in the supplementary documents.

6.3 Pedestrian Segmentation in Video

When segmenting occluded objects, an important problem is that largely-occluded objects are difficult to be detected and located. Our algorithm manages to deal with that by isolating the object locating algorithm. In previous experiments, we use context information (face) to locate object (clothing) and in this subsection we explore more general cases where a tracking algorithm is used to locate occluded objects.

①When different algorithms for min-cut/max-flow are used, the time complexity might be different. However the linear relationship to the number of objects stays the same.

② Note that the running time of GrabCut is also affected by the image size, so we apply further acceleration by only using a narrow band outside the initial box as background seeds.

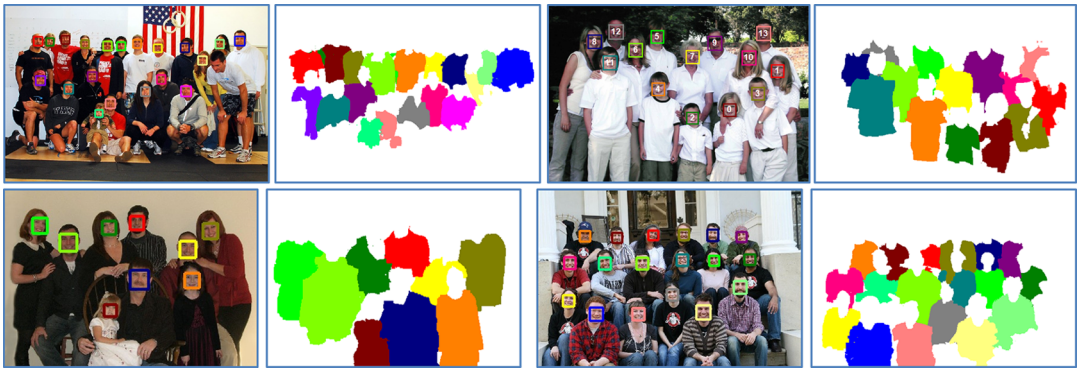


Fig.15. Clothing segmentation results for group images with more people.

We demonstrate our algorithm in the TUD-Crossing dataset which contains side views of pedestrians with various occlusions between them. People are detected using the algorithm in [43] and some occluded ones are located by [44]. Note that there are still occluded ones being missed. We evaluate our segmentation algorithm without any temporal information, i.e., segmentation is performed in each frame individually.

The training data is collected from two sets. One is from detector training data from [15] which is provided with annotated pedestrian shapes. The other is collected from the TUD-Crossing video in which pedestrian occlusions are covered. We annotate every five frames starting from the third frame (avoiding the provided test data from [46]). We apply five-fold cross-validation to tune the model parameters using the algorithm in Fig.8.

Dealing with pedestrians has challenges different from clothing segmentation. The first one is that the appearance models of clothes and pants may be very different, so a single color model is not appropriate. We apply a localized color model similar to the idea in [46] except that we use a multi-label optimization formulation to solve it. Specifically, for each person its foreground and surrounding background are both partitioned into two parts horizontally. The four parts are modified based on current shape and local appearance cues iteratively. The localized color model is built from the final partitions. Another challenge is that graph-cut based segmentation algorithm is prone to yield shorter boundaries. This bias is particularly harmful in pedestrian segmentation due to the concave leg parts. We refine the segmentation result using a postprocess as in [47] which flips the label of superpixels by voting for the desired label with the nearest neighbors of all pixels in them. The shape and color model is updated accordingly and a final alpha-expansion is used to get the pixel-level segmentation results.

Fig.16 shows the final segmentation results produced by our algorithm in TUD-Crossing and TUD-Campus sequences. More results can be found in the supplementary documents. We obtain accurate pedestrian segme-

Fig.16. Pedestrian segmentation in video. The frames in (a) and (b) are from TUD-Crossing sequence and those in (c) are from TUD-Campus sequence.

ntation even if occlusions and pose variations occur. Note that diverse color distributions between clothes and pants are also well handled. Detailed quantitative results are shown in Table 2. It can be observed that all parts contribute to improving the segmentation result. Higher-order cues seem to be less useful than that in clothing segmentation. This is because the superpixels in these low-resolution and blurred frames are not so accurate. Appearance cues are slightly enhanced due to the effectiveness of the localized color model. A comparison with other state-of-the-art methods is reported in Table 3, where BR is the level set tracker[45] initialized with detector bounding box and LAM+HP is the localized appearance model initialized and optimized with shape probability map provided by Hough Forest detector[46].

Table 3. Segmentation Performance for TUD-Crossing

Method	Precision (%)
BR[45]	83.1
LAM+HF[46]	92.1
Our full model	94.3

6.4 Object Detection

Our algorithm is initialized by object detection/location algorithms whose precision affects the segmentation accuracy of the proposed method. First, we have to emphasize that no matter what detection algorithms are applied, they must be consistently used in the training and testing stages. We estimate the effect of the detection algorithm in Fig.17 where detected boxes are drifted or rescaled by several pixels. The experiments show that small drift and rescaling does not hurt the performance too much, but when they become larger, accuracies drop quickly. Rescaling has less affection on clothes than on pedestrian. A possible reason is that the object boxes for clothes are estimated from face and they are not so accurate even in the training stage, so the algorithm is more tolerant of inaccurate detection box.

In addition, we will show that although our algorithm is initialized by an external object locator, it is still possible for our algorithm to verify the detection hypotheses. There are a number of approaches that combine segmentation and detection (recognition). Here we focus on segmentation as a mechanism to improve detection performance. The most related work is [48] where a shape classifier is learned to verify the detection hypotheses. However, in our problem, segmentations of true positive candidates can be very different due to inter-object occlusions. The intuition of our solution here is that while training the shape classifier for object shapes, we incorporate the information from their neighbors.

We apply our algorithm to images where ground truth locations of objects are known and obtain a set of true-positive and false-positive segmentations. We then learn a random forest shape classifier to distinguish them. Instead of using binary shape masks as weak features, we use a tri-label one to take the neighbor shapes into consideration. Specifically, segmentation of object o is transformed into a tri-label mask $\{l_i^o\}$:

$$l_i^o = \begin{cases} 0, & \text{if } z_i = 0, \\ 1, & \text{if } z_i = o, \\ 2, & \text{if } z_i \neq 0 \text{ and } z_i \neq o. \end{cases} \tag{13}$$

The decision stump for each node is derived from the proportion of label l in rectangle R, where $l \in \{0, 1, 2\}$ and R are randomly generated. The verification results are shown in pedestrian segmentation in Fig.18. It can be seen that even some blocked pedestrian segmentations seem different from regular human, our algorithm still classifies them as true-positive candidates, while the real false-positive ones are rejected due to their unexpected shapes (marked by black rectangles).

7 Conclusions

In this paper, we proposed a simultaneous algorithm

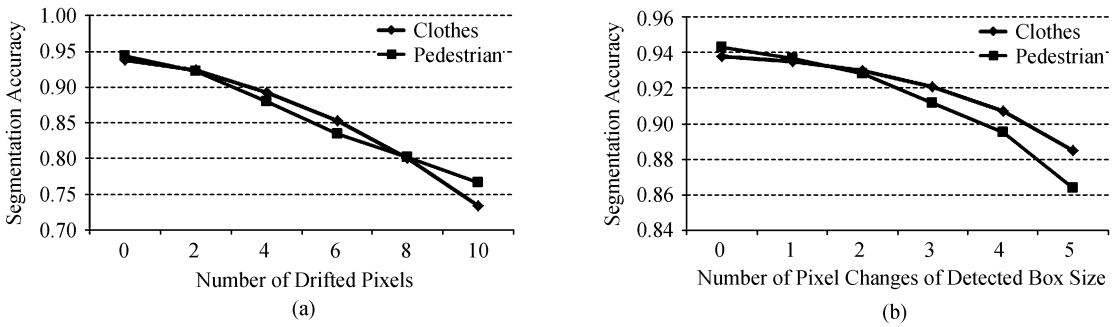


Fig.17. Detection affection on segmentation accuracy.

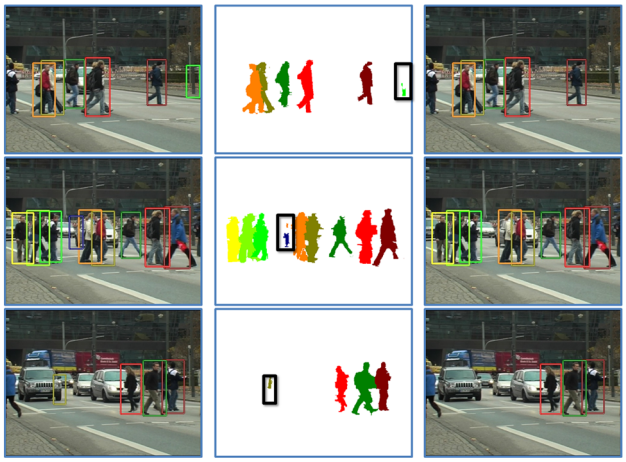


Fig.18. Detection verification results of our algorithm. (a) Input detections. (b) Segmentation. (c) Verified detections.

to segment multiple highly-occluded objects. Our key idea is inferring the blocking relationship and using this high-level information to guide the segmentation. To achieve this, we proposed to combine high-level and low-level information in a unified framework. The high-level layout model provides sophisticated shape prior to reduce appearance ambiguities with the consideration of blocking relationship between objects. The low-level segmentation model integrates higher-order constraints from superpixels and image appearance features to adjust the object shapes. The two models are optimized iteratively and converge to more robust segmentation results.

The proposed algorithms has been demonstrated in specific objects. In the future, we will apply our approach to segment more general objects and thus explore effective algorithms to predict the occlusion relationship between objects from different categories.

References

[1] Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2002, 24(5): 603-619.

[2] Shi J, Malik J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2000, 22(8): 888-905.

[3] Deng Y, Manjunath B. Unsupervised segmentation of color-texture regions in image and video. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2001, 23(8): 800-810.

[4] Rother C, Kolmogorov V, Blake A. "GrabCut"—Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graphics*, 2004, 23(3): 309-314.

[5] Pham V Q, Takahashi K, Naemura T. Foreground-background segmentation using iterated distribution matching. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2011, pp.2113-2120.

[6] Viola P, Jones M. Robust real-time face detection. *Int. J. Computer Vision*, 2004, 52(2): 137-154.

[7] Dalal N, Triggs B. Histogram of oriented gradients for human detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 27-July 2, 2004, Vol.1, pp.886-893.

[8] Felzenszwalb P F, Girshick R B, McAllester D, Ra-manan D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2010, 32(9): 1627-1645.

[9] Andriluka M, Roth S, Schiele B. Pictorial structures revisited: People detection and articulated pose estimation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2009, pp.1014-1021.

[10] Borenstein E, Ullman S. Combined top-down/bottom-up segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2008, 30(12): 2019-2125.

[11] Kumar M P, Torr P, Zisserman A. Objcut: Efficient segmentation using top-down and bottom-up cues. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2010, 32(3): 530-545.

[12] Levin A, Weiss Y. Learning to combine bottom-up and top-down segmentation. *Int. J. Computer Vision*, 2009, 81(1): 105-118.

[13] Zhu L, Chen Y, Yuille A. Learning a hierarchical deformable template for rapid deformable object parsing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2010, 32(6): 1029-1043.

[14] Gallagher A, Chen T. Understanding groups of images of people. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2009.

[15] Andriluka M, Roth S, Schiele B, People-tracking-by-detection and people-detection-by-tracking. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2008.

[16] Yang Y, Hallman S, Ramanan D, Fowlkes C. Layered object detection for multi-class segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2010, pp.3113-3120.

[17] Leibe B, Schiele B. Interleaved object categorization and segmentation. In *Proc. British Machine Vision Conference*, September 2003.

[18] Winn J, Jojic N. Locus: Learning object classes with unsupervised segmentation. In *Proc. the 10th Int. Conf. Computer Vision*, October 2005, Vol.1, pp.756-763.

[19] Winn J, Shotton J. The layout consistent random field for recognizing and segmenting partially occluded objects. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2006, pp.37-44.

[20] Hoiem D, Rother C, Winn J. 3D layoutCRF for multi-view object class recognition and segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2007.

[21] Gould S, Rodgers J, Cohen D, Elidan G, Koller D. Multi-class segmentation with relative location prior. *Int. J. Computer Vision*, 2008, 80(3): 300-316.

[22] Wu B, Nevatia R. Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. *Int. J. Computer Vision*, 2009, 82(2): 185-204.

[23] Gao W, Ai H, Lao S. Adaptive contour features in oriented granualar space for human detection and segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2009, pp.1786-1793.

[24] Gallagher A C, Chen T. Clothing cosegmentation for recognizing people. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2008.

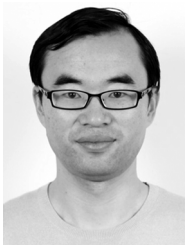
[25] Vu N, Manjunath B. Shape prior segmentation of multiple objects with graph cuts. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2008.

[26] Ladicky L, Sturgess P, Alahari K, Russell C, Torr P H. What, where and how many? Combining object detectors and CRFs. In *Proc. the 11th European Conf. Computer Vision*, September 2010, pp.424-437.

[27] Kohli P, Ladicky L, Torr P H. Robust higher order potentials for enforcing label consistency. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2008.
[28] Maire M, Yu S X, Perona P. Object detection and segmentation from joint embedding of parts and pixels. In *Proc. Int. Conf. Computer Vision*, November 2011, pp.2142-2149.
[29] Wang N, Ai H. Who blocks who: Simultaneous clothing segmentation for grouping images. In *Proc. Int. Conf. Computer Vision*, November 2011, pp.1535-1542.
[30] Lafferty J, McCallum A, Pereira F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. the 18th Int. Conf. Machine Learning*, June 28-July 1, 2001, pp.282-289.
[31] Wang N, Ai H, Lao S. A compositional exemplar-based model for hair segmentation. In *Proc. the 10th Asian Conf. Computer Vision*, November 2010, Vol.3, pp.171-184.
[32] Ladicky L, Russell C, Kohli P, Torr P H. Associative hierarchical CRFs for object class image segmentation. In *Proc. Int. Conf. Computer Vision*, September 27-October 4, 2009, pp.739-746.
[33] Boykov Y, Jolly M. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proc. the 8th Int. Conf. Computer Vision*, July 2001, Vol.1, pp.105-112.
[34] Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. *Machine Intelligence Learning*, 2006, 63(1): 3-42.
[35] Murphy K P, Weiss Y, Jordan M I. Loopy belief propagation for approximate inference: An empirical study. In *Proc. the 15th Conf. Uncertainty in Artificial Intelligence*, July 30-August 1, 1999, pp.467-475.
[36] Breiman L. Random forests. *Machine Learning*, 2001, 45(1): 5-32.
[37] Koller D, Friedman N. Probabilistic Graphical Models: Principles and Techniques (1st edition). MIT Press, 2009.
[38] Kolmogorov V, Zabih R. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004, 26(2): 147-159.
[39] Boykov Y, Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004, 26(9): 1124-1137.
[40] Bradski G. The opencv library. In *Dr. Dobb's Journal of Software Tools*, 2000, http://www.drdobbs.com/open-source/the-opencv-library/184404319, July 2013.
[41] Huang C, Ai H, Li Y, Lao S. High performance rotation invariant multiview face detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2007, 29(4): 671-686.
[42] Eichner M, Ferrari V. We are family: Joint pose estimation of multiple persons. In *Proc. the 11th European Conf. Computer Vision*, September 2010, Part 1, pp.228-242.
[43] Duan G, Ai H, Lao S. A structural filter approach to human detection. In *Proc. European Conf. Computer Vision*, September 2010, Part 6, pp.238-251.
[44] Xing J, Ai H, Lao S. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2009, pp.1200-1207.
[45] Bibby C, Reid I. Robust real-time visual tracking using pixel-wise posteriors. In *Proc. European Conf. Computer Vision*, October 2008, pp.831-844.
[46] Horber E, Rematas K, Leibe B. Level-set person segmentation and tracking with multi-region appearance models and top-down shape information. In *Proc. Int. Conf. Computer Vision*, November 2011, pp.1871-1878.
[47] Brox T, Bourdev L, Maji S, Malik J. Object segmentation by alignment of poselet activations to image contours. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2011, pp.2225-2232.
[48] Ramanan D. Using segmentation to verify hypotheses. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2006.

Nan Wang received the B.S. degree in computer science and technology from Tsinghua University, Beijing, in 2008. He has won the Excellent Student Scholarships of Tsinghua University in 2004∼2008, 2011, and 2012. He is currently a Ph.D. candidate at Tsinghua University. His research interests include computer vision, machine learning and pattern recognition, with special focus on object specific segmentation.

Hai-Zhou Ai received the B.S., M.S., and Ph.D. degrees in computer application all from Tsinghua University, China, in 1985, 1988, and 1991, respectively. He spent the period 1994∼1996 in the Flexible Production System Laboratory at the University of Brussels, Belgium, as a postdoctoral researcher. He is currently a professor in the Computer Science and Technology Department at Tsinghua University. His current research interests are human image processing, biometrics, and visual surveillance. He has published more than 80 papers in peer-reviewed journals and international conferences. He is a senior member of the IEEE.



Feng Tang is a senior researcher in Hewlett-Packard Laboratories, Palo Alto, from 2009. He obtained his Ph.D. degree in computer engineering in University of California, Santa Cruz in December 2008, master' degree and bachelor's degree from State Key Lab of CAD&CG, Zhejiang University in 2004 and 2001 respectively. His research interests include computer vision, machine learning and multimedia. He received the best paper award for Multimedia Modeling Conference in 2011.