2015 International Conference on Virtual Reality and Visualization

Adaptive Template for Parsing Object of Indoor Scene Image

Changqun Xia[1], Jie Xu[2], Qing Li[3], Yu Zhang[1], Jia Li[1], Xiaowu Chen[1]*
[1]*State Key Laboratory of Virtual Reality Technology and Systems*
School of Computer Science and Engineering, Beihang University, Beijing, China
[2]*National Computer Network Emergency Response Technical Team Coordination Center of China*
[3]*Beijing Union University, Beijing, China*

***Abstract*—In this paper, we propose an adaptive template for semantic labeling of indoor scene objects and estimating their oriented bounding facets (OBFs). The proposed adaptive template encodes prior geometric information of objects based on statistics of the training images. Given an input image, we utilize the adaptive template on the detected bounding boxes to initialize the raw labeling and OBF estimation of objects. To refine the initial results, multiple cubes/faces that follows geometric principles of adaptive template are generated to make up OBFs hypotheses. Each of the OBFs hypotheses is scored by the consistency matched with its corresponding semantic labeling result. The OBFs hypothesis that has the highest matching score with the corresponding labeling result is selected as the final parsing result. We evaluate our method on the bed, sofa and tea table categories, on both real and rendered indoor scenes. The experimental results show that our method has improved performance compared with the state-of-the-art detectors, and can give reasonable 3D interpretations of objects.**

***Keywords*-Image scene understanding; Semantic labeling; Geometry parsing**

I. Introduction

Indoor scene understanding resulting in geometry estimation or semantic labeling is one of the most fundamental problems in computer vision. Previous works have mostly focused on the processing of outdoor scenes. Indoor scenes, on the other hand, have received relatively less attention. This is due in part to certain unique challenges the indoor scene object parsing problem presents, including poor illumination, diversity in the scene and a lack of distinctive features [1].

Recently, interests in indoor scene analysis and modeling have been feuded by the ease of capturing RGB+D images with the aid of devices such as Kinect. The availability of depth information, though noisy at times, makes the analysis problem more tractable. Model- or data-driven approaches have been proposed for indoor scene labeling [1] and object recognition [2], [3], [4], [5]. Prior knowledge evidently plays the key role in improving the accuracy of these solutions; it may even be a necessity.

Compared to depth images, conventional RGB images are still much easier to acquire and manipulate (e.g., when preparing for training data). The ability to understand an indoor object from a single image allows us to tap into and utilize the vast repository of existing images. Existing works on indoor scene understanding from single images have addressed the problems of sparse geometry estimation [6], [7], [8], [9], [10], recovery of room layout and spatial relations between objects [6], [7], [10], as well as object detection [8]. However, these works mostly focus on the recognition of objects, resulting in a rough geometry estimation of object, much less the geometry estimation of object faces. They generally utilize cubes or boxes to approximate objects, whereas regardless of the object diversity in appearance and shape.

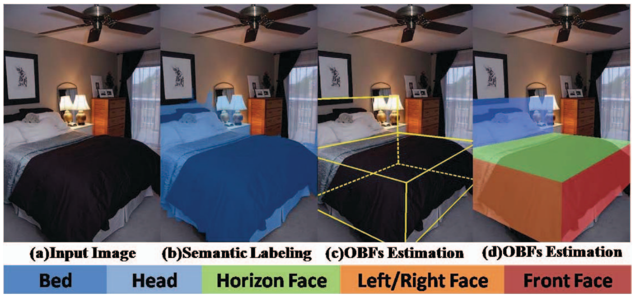

Figure 1. The motivation of our method. (a) The input image, (b) semantic labeling result, (c) and (d) are both the OBFs estimation results (in different visualization). In (c), yellow lines indicate the 3D structure lines aligned to the object. In (d), different color illustrates different faces of an object.

In this paper, we also address the problem of indoor scene understanding from a single image, while seeking an understanding at a more granular level, i.e, object-level. Specifically, we predict a *semantic* labeling of the object in an input scene image and estimate the geometry of each labeled object by inferring a set of *oriented bounding facets* (OBFs). The semantic labels and geometric estimation can be directly applied to subsequent scene modeling taskes such as object rearrangements, replacements, and resizing.

Our algorithm is naturally template-guided and utilizes annotated data consists of semantically labeled indoor scene images and OBF labeling of objects contained in the image dataset. We pose the scene image analysis problem as an *object parsing* problem, where the solution is guided by an adaptive template. The adaptive object template, incorporat-

*Corresponding author, E-mail:chen@buaa.edu.cn

978-1-4673-7673-0/15 $31.00 © 2015 IEEE
DOI 10.1109/ICVRV.2015.56

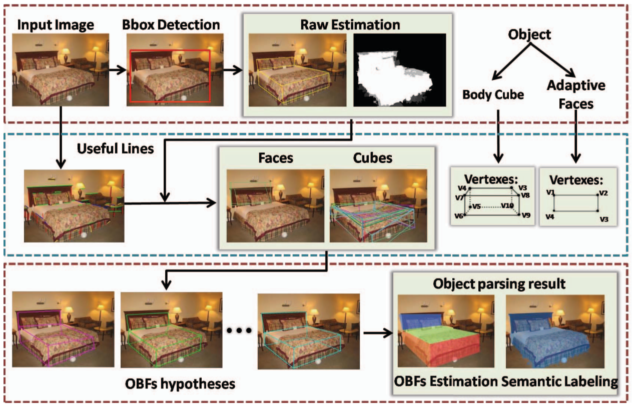

Figure 2. The overview of our method. Our adaptive template severs as guidance for our method including four steps: bounding box detection, raw estimation, useful lines exaction, OBFs estimation.

ed with a body cube and adaptive faces, are derived from annotated image dataset. Given an input image, we extend the deformable part model of [11] to estimate the bounding box of objects. Based on these bounding box detections, we generate a raw estimation of semantic labeling and rough localization of object template vertexes. Based on the raw estimation of OBFs, we generate multiple OBFs hypotheses incorporating the extracted useful line segments. Utilizing precise semantic labeling for each OBFs hypothesis, the final object-parsing result is inferred through matching degree between semantic labeling and OBFs estimation.

The main contributions of this paper include: 1) It provides a new perspective to parsing object through adaptive template, resulting in semantic labeling as well as OBFs estimation of object. 2) An adaptive object template incorporating geometry information. Due to the adaptive property, our template handles well OBFs estimation, especially when the objects are diverse in appearance and shape.

II. Related work

Recently, several literatures propose parsing grammar or interpreting graph for image scene understanding [12], [13], [14], [15], [16]. In these literatures, their presented representative units, such as block [13], facade [14] or line segment [16], to imply shape and geometric information. According to spacial relationship or grammar rules, these units are used to parse the image scene into a hierarchical structure in terms of semantic or geometric. Inspired by these parsing methods, we propose an adaptive object template for object parsing, which integrates semantic labeling and OBFs estimation into a unified framework. The work of Gupta et al.[13] is similar to ours to some extent. They parse the image scene into a 3D graph taking steps of semantic labeling, depth ordering, geometry estimation and support recovery. However, they deal with the outdoor scene image, not the indoor scene object. A rich literatures related to indoor scene understanding pay much attention on the spatial layout estimation [6], [16], [9], [2], [17] but less on the object parsing which serves as a main purpose of our work, especially using a single image without depth information. The most similar works to ours are those of [8], [10], [7].

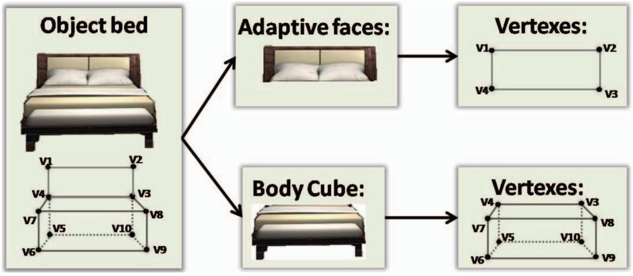

Figure 3. Illustration of our template. Take object *bed* for example, the entire object is the root node, it consists of a body cube and an adaptive face which is the head of bed. Every face and cube are represented by vertexes.

Hedau et al.[8] develop a 3D cuboid box to represent the indoor scene objects. The orientation of an object is estimated based on the orientation of the room as they with the assumption that the faces of object cuboids are parallel to the walls. Thus they adopt a searching strategy by sliding a 3D cuboid to fit the object and generate object hypotheses. To score these hypotheses, they apply a trained SVM as well as 2D detectors. A difference compared to our work lies in that they do not predict the class of an object while we do semantic labeling of objects. Another difference from [8] is that our object template has adaptive faces to indicate the orientation and distinguish structures of different categories.

Lee et al.[7] use the volumetric constraints to generate hypothesis of both indoor scene and objects. In terms of object geometric representation, they adopt a coarse 3D parametric cuboid to reason the spatial layout of an object without recognizing the object class. It share the same difference with [8] from ours as our method can reason the spatial orientation of an object as well as its semantic category.

Subsequently, Hedau et al.[10] extend backrest to the generic 3D representation *box* for objects like sofas and chairs. Though their extension can detect certain category objects, they still do not identify the recognition and semantic labeling. Their object candidates are searched by sliding the 3D box and scored using trained SVM classifier with local contrast image features, however,our refinement of object OBFs estimation exploits the semantic labeling result and OBFs of object template.

III. Object Parsing

Our goal is to parse the indoor scene object into precise semantic regions and oriented bounding facets. We propose adaptive object templates to model the oriented bounding facets of objects. The overview of our method is shown in Figure 2. We extend the deformable part model of [11] to estimate the bounding box of objects. Based on these bounding box detections, we generate a raw estimation of semantic labeling and rough localization of object template vertexes. With the raw estimation of OBFs, we generate multiple OBFs hypotheses incorporating the extracted useful line segments. Then, we score precise semantic labeling and OBFs for each hypothesis. The score is obtained by using reliability of geometry and matching degree between semantic labeling and OBFs, hypothesis with the best score is the final object-parsing result.

A. Adaptive template

Rather than utilizing cubes or boxes to approximate objects, we propose a hierarchical template incorporating adaptive parts as well as geometric information of object.

As shown in Figure 3, this template contains three levels which are implying the geometric information of an object from coarse to fine. The first level is a root node which denotes the entire object. In the second level, the node is a part-based component of cube or face. Due to the shape diversity, some part may be absent for objects from a category. For example, some chairs have backs while others don't. Thus we append an occurrence attribute to indicate whether the part is in the object or not, which makes our object template adaptive. It is observed that a rectangle face and a cube consist of fixed number of vertexes (4 and 8 respectively). Once we have located vertexes, the bounding boxes and orientations of cubes or faces can be localized. Hence, in the third level, the node represents the vertex. The part component and the entire object can be comprised of the vertex nodes.

To annotate the training images, we develop a template vertexes annotation tool. Using this tool, we label the objects with their category, vertexes, as well as the occurrence attribute for each vertex. For one category template, we learn the position distribution of each vertex in the training images. To expand the number of our sample, we flip the object to get a symmetrical sample.

Our adaptive template is utilized in the following steps of raw estimation, and OBFs estimation. Specifically, low level information of template guides the generation of cubes and faces, and the high level information of template guides raw estimation and OBFs estimation.

B. Object detection

To implement an exact parsing for an object, we first utilize the current object detection technique, which is improved apparently in recent years. Since the objects in our dataset have diverse appearance, to train discriminative object detectors, we preprocess our training images by clustering them into several subsets through a two-level clustering scheme. At first level, relative locations of labeled object vertexes with similar length-width ratio are fed as features into spectral clustering algorithm, since the detector is sensitive to them. We obtain $M(M < 5)$ clusters after this step. At the second level, we further subdivide the M clusters by object poses. Although labels like left, right and front are provided for each object in some datasets like VOC[18], they are too rough to distinguish from objects under arbitrary views. In addition, as mentioned in section II, there are optional faces of object, so we use the locations of vertexes on the present faces as features to differentiate poses of objects. As shown in Figure 3, taking an object *bed* for example, ten vertexes are indexed, where vertex *3* and *10* are stationary vertexes while vertex *1* and *2* are vertexes on the head of bed, i.e, an optional part.

By this two-level clustering, we split objects in the same category label into N clusters, denoted as N classes, where objects in the same class have similar height-width ratio, similar distribution of vertexes and poses. In addition, considering the horizontal symmetry, we add the flipped location of vertexes into the second level features to handle horizontal symmetry. Figure 4 *(a),(b)* shows example training images and the corresponding part-based models for each cluster.

The detectors are trained and refined through an EM-like process. In each iteration, we train detector on the clustered images. The trained detector is then used to score each training instance. If an object in cluster A has a higher score respecting to the detector trained on cluster B, we move the object from **A** to **B**. Generally 3 iterations can produce enough good results.

C. Raw estimation

With bounding box of objects and the guidence of adaptive template, we can estimate a raw semantic labeling and OBFs for objects.

As mentioned above, we have clustered training dataset into N classes. The similar geometry information of objects in a class, locations of vertexes, is regarded as prior geometric property. Each class has a corresponding adaptive object template. For images in a class, we statistic the relative location of each vertex of objects in their bounding boxes. For each data sample, we flip the object to get a symmetrical sample, thereby expand the number of sample.

In order to avoid misleading of occasional samples, we sort the vertexes of samples by their relative location inside the bounding box. Only the samples in the central seventy percentages of the sorted data list are accepted as the statistical samples as the prior. Then we calculate the mean values of relative locations of samples. Figure 4 *(c)* shows the relative locations of vertexes for each class.

Given an input image with detection result by part-based model, we localize the vertexes in the bounding box with location prior as initial vertexes, thus we can get initial faces of object guided by the low level information of our template. With over-segmentation of this image, we compute

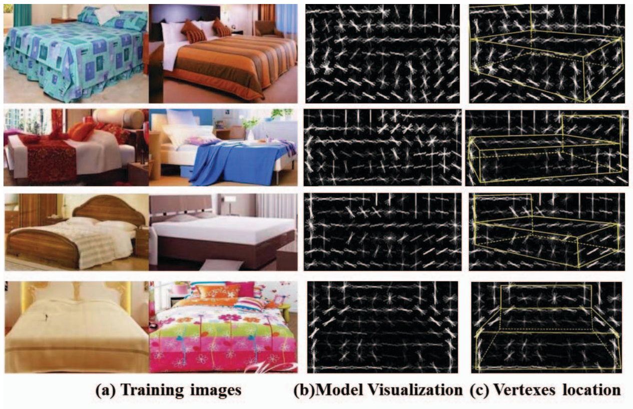

Figure 4. Visualization of our part-based model and geometric prior for each class of object *bed*. Four class types are shown here. The first two columns give some example images from training set of each class, where images in each row are symmetric object examples. The third column shows the visualization of part-based model and the fourth column visualizes the geometry prior which are relative locations of vertexes statisted on each class.

the area rate amid these faces for each superpixel and remove superpixels which rates are lower than a threshold.

We compute the possibility of a superpixel belonging to the object. Here, m is the number of pixels both in superpixel i and faces, n denotes the number of pixels in superpixel i, the rate of superpixel i inside faces is $rate_i = \frac{m}{n}$. Based on this rate, we compute a probability map for an object as raw semantic labeling. The probability of pixel p is :

$$S_{p\in i} = \begin{cases} 0 & \text{if} \quad rate_i \leq \theta_1 \\ 1 & \text{if} \quad rate_i \geq \theta_2 \\ rate & \text{other} \end{cases} \tag{1}$$

In our implementation, θ_1 is 0.1 and θ_2 is 0.65. For each object template, we compute the matching score between initial vertex locations and raw semantic labeling. The matching score is the mean score of pixels which have higher pixel score than a threshold (0.8). The template which has the best matching score with probability map is selected as the best initialization. Then we get the initial vertexes of OBFs and raw semantic labeling.

D. Line segments exaction

We over-segment the input image into superpixels using algorithm of [19] and exact line segments using the Matlab toolbox by Kovesi [20] which runs Canny edge detector, links edge pixels and fits line segments. Then we produce line segments by a merging processing as described in Algorithm 1. There are some notations we used: slope distance DS_{ij} between line l_i and line l_j, the endpoint distance DEP_{ij} between lines l_i and l_j, and the difference of two lines l_i and l_j D_{ij}. $ep(l_i)$ is the endpoint of line l_i.

$$D_{ij} = \lambda_1 DS_{ij} + \lambda_2 DEP_{ij} \tag{2}$$

Algorithm 1 Merging Line Segments

Require: $L = \{l_1 ... l_n\}$:initial line segments
α, β: thresholds

Ensure: $M = \{m_1 ... m_k\}$:merged lines

```
$flag = 1$;
Merge line segments
while flag do
    $L_{merge} = \{l_i, l_j \mid DS_{ij} \leq \alpha \bigcap DEP_{ij} \leq \beta\}$
    if $L_{merge}$ is empty then
        $flag = 0$;
    else
        $[l_i, l_j] = \min\{D_{ij} \mid (l_i, l_j) \in L_{merge}\}$
        $l_{new} = merge\{l_i, l_j\}$, updata $L$
    end if
end while
M=L;
```

where

$$\begin{aligned} DS_{ij} &= |slope\,(l_i) - slope\,(l_j)| \\ DEP_{ij} &= \min\left(Distance\left(\forall ep\,(l_i)\,, \forall ep\,(l_j)\right)\right) \end{aligned}$$

Since vaninshing points can be computed by line segments[21], and are important in indoor scene geometry estimation, we classify line segments to corresponding vanishing points and we list the geometric principles will be used in following steps:

i. If a line belongs to one vanishing point, it should pass this vanishing point or pass the vanishing point through a small rotation angle.

ii. Parralle lines belong to the same vanishing point and adjacent lines belong to different vanishing point.

iii.Opposing faces should be parallel, which means that they should be classified to the same vanishing point. Adjacent faces should be classified to different vanishing points.

iv. Faces of a cube should be classified to three different vanishing points.

E. OBFs estimation

Having raw estimation and useful lines we can process the OBFs estimation of objects.We adopt a bottom-up and top-down strategy to perform OBFs estimation. Figure 5 shows the pipeline of our OBFs estimation. We have already generate line segments in the image as bottom node and a raw OBFs and semantic labeling as our top hierarchy node, we generate OBFs hypotheses in four steps as following.

Firstly, we search useful lines with initial lines and initial vertexes. We search the candidate lines for every lines of object around them within a small deformation that we define as a small rotation angle and a short translation. Guided by the raw semantic labeling, we narrow down the search range that we only search lines passing our raw

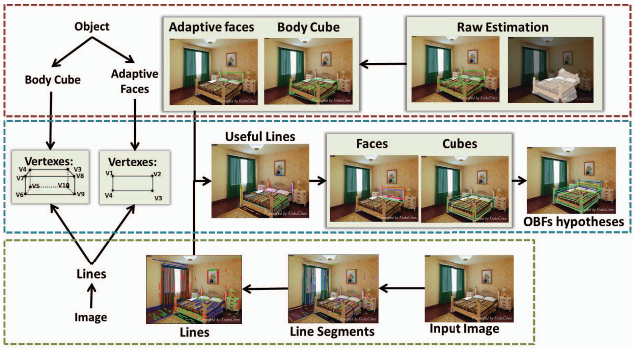

Figure 5. Illustration of OBFs hypotheses estimation. We generate cubes and faces by useful lines and raw semantic labeling and OBFs, OBFs hypotheses are generated by combination of cubes and faces.

semantic labeling with an average of pixels' scores over a threshold (0.3). For each line, we generate their candidate set consist of these similar lines and their initial lines linked by initial vertexes as useful lines.

Secondly, we generate faces and cubes by line sets. For a face, we have four candidate line set. Take a combination of four lines from these four set, we compute intersection point by 2 adjacent lines as vertexes of face. If four lines allow geometric principles and generate a face without large variety, this face will be added into face candidates. For a cube, there are six outer lines. To generate a cube candidate, it may be a little complicated that we use three steps for generation. First, we localize the six outer vertexes on the cube similar to faces generation. Then we need to fix the vertexes inside cube. To allow the individual lines to deform slightly, for example we compute location of $V7$ in Figure 6, we firstly compute possible equations of lines $l_{(7,6)}$, $l_{(7,4)}$ and $l_{(7,8)}$, here we take line $l_{(7,6)}$ as example.

Before the calculation, we need to ascertain which vanishing point line $l_{(7,6)}$ belongs to. We calculate the average intersection angles between initial lines of object and every vanishing point vp_i of image shown in Figure 6, we ascertain each line the vanishing point vp_i with the intersection angle as small as possible satisfied geometric principles. Though we can directly calculate the equtation of line $l_{(7,6)}$, we try to use the parallel relationship to revise it which can get a better result when vanishing point are not correct and more accordant when the side of object actually have a roll angle:

$$\begin{aligned} l_{(7,6)} = \alpha F_{parallel}\left(l_{(7,6)}|V4, V5, V8, V9\right) \\ +\beta F_{vp}\left(l_{(7,6)}|V6\right) \end{aligned} \tag{3}$$

Here, we set $\alpha = 0.7$ and $\beta = 0.3$. Given location of $V6$, F_{vp} computes the equation of $l_{(7,6)}$ by the equation of line $l_{(vp_i,6)}$. In Figure 6, we difine $Dis(V_i, V_j)$ is the distance between vertex V_i and vertex V_j, then $F_{parallel}$ is defined as:

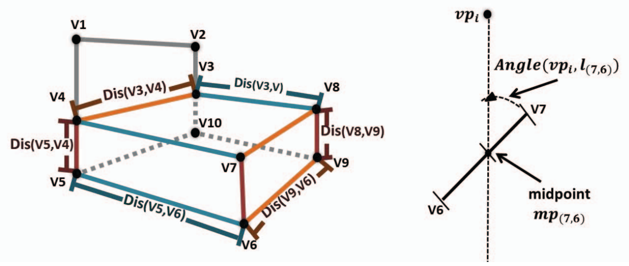

Figure 6. Example for localization of vertexes inside cube. We show opposite sides of cube in same color.

$$F_{parallel}\left(l_{(7,6)}|V4, V5, V8, V9\right) = \lambda_1 l_{(4,5)} + \lambda_2 l_{(8,9)} \tag{4}$$

$$\lambda_1 = \frac{Dis(V5, V6)}{Dis(V5, V6) + Dis(V9, V6)}, \tag{5}$$

$$\lambda_2 = \frac{Dis(V9, V6)}{Dis(V5, V6) + Dis(V9, V6)} \tag{6}$$

Then we compute crossover point for each pair of lines $l_{(7,6)}$, $l_{(7,4)}$ and $l_{(7,8)}$ and use the mean value of the locations of crossover points as the location of vertex $V7$. Thirdly, we need to combine faces and cubes into objects hypotheses. Since the influence factor is their common edge, we define a matching score:

$$score = \exp\left\{-\left(\lambda_1 D_{ij} + \lambda_2 D_{vertex}\right)\right\} \tag{7}$$

$$D_{vertex}(l_i, l_j) = \frac{\sum\left(|Loc_{V_{i1}} - Loc_{V_{j1}}| + |Loc_{V_{i2}} - Loc_{V_{j2}}|\right)}{2} \tag{8}$$

Here, l_i is the common edge of the face while l_j is the edge of the cube, V_{i1} and V_{i2} are the end point of line l_i, V_{j1} and V_{j2} are the end point of line l_j, Loc_V means the location of point V, D_{ij} is the difference between two lines l_i and l_j calculated by (2).

For each cube candidate, we search for the face with gratest matching probability to generate corresponding object hypothesis.

In the end, having object candidates, we can finally infer the best OBFs estimation and exact accurate semantic labeling from candidates. Repeat the process of computing semantic labeling in section III-C on each object candidates and the corresponding semantic labeling result can be easily computed by labeling the pixels in the probability map with a score larger than a threshold(0.8) to find the best object parsing result. The score is defined as average precision of semantic labeling and OBFs.

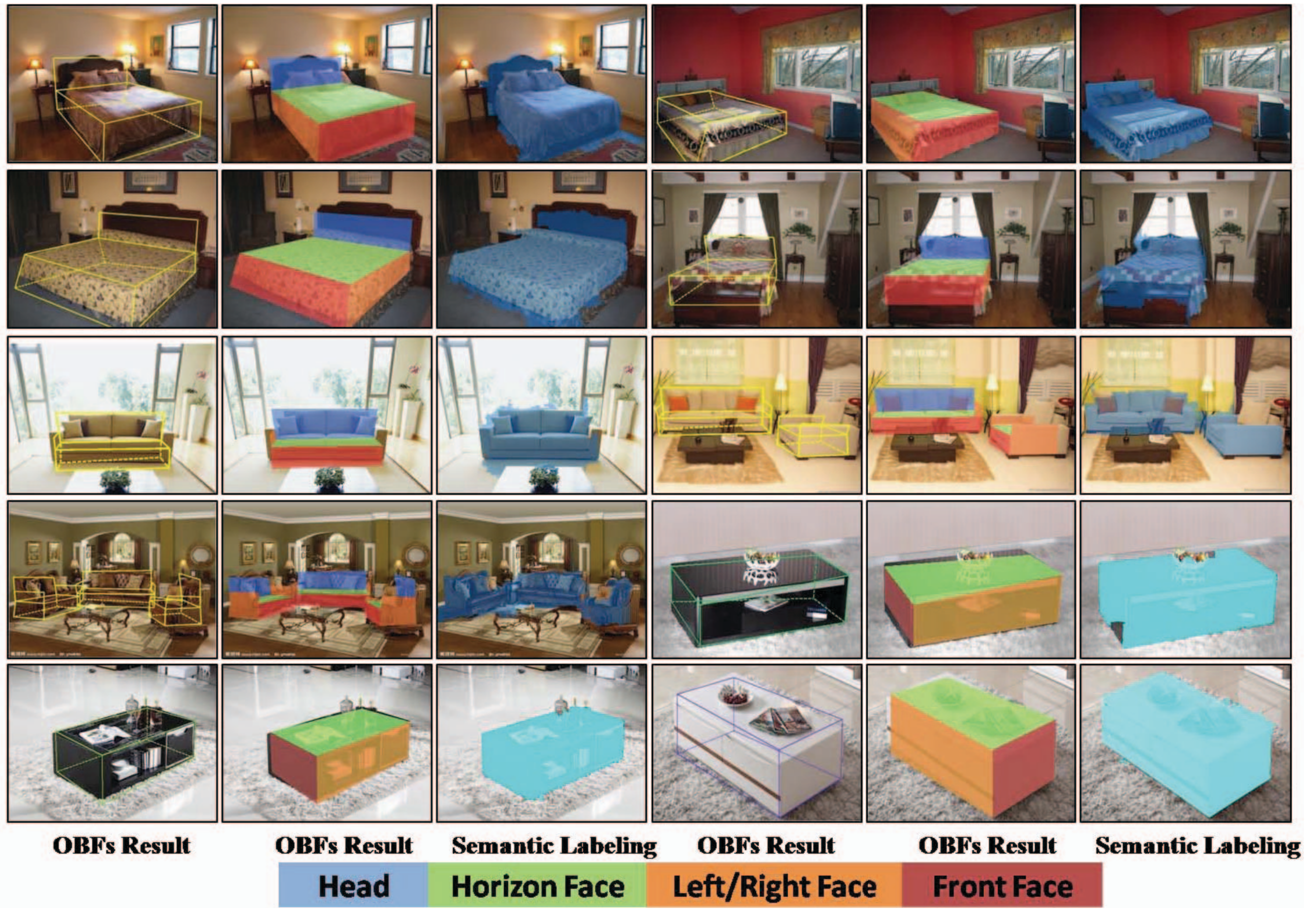

Figure 7. Our results of both bedroom and living room. Column 1 and 4 are the results of OBFs estimation, visualized in aligned object. Column 2 and 5 are also the results of OBFs estimation, which visualized in colored faces. Column 3 and 6 are the results of semantic labeling, where lagend bar is shown below.

IV. EXPERIMENTS AND RESULTS

We perform our method on different size of training set including 100, 200, 300 and 400 images. With the increasing of training images, the performance and training time both increase. To balance between training time and performance, we finally choose 212 images as our training images which are automatically clustered into 12 classes. We test our method on 149 images of bedroom scenes, including bedroom and living room scenes. In the test images, there are 72 images from Hedau et al.[8], which are mostly pictured from real scenes. To extend our performance, we add other 77 elegant rendered images scenes searched from Google. Figure 7 shows some results results of both bedroom and living room, including bed, sofa and tea table categories. We also compare geometry estimation of our result with the published images of Hedau et al.[8] in Figure 8.

Since our cubes and faces generation are guided by the lines in image and the adaptive properties of our template, the edges of our OBFs results are more precise to the real boundary of objects. Our method works well on objects which have not adaptive parts, such as the top two examples in the fourth column of Figure 7. Besides, our object-level detection template leads to parse multiple objects in the image, while theirs [8] do not. For example, we can handle with the multiple sofas in Figure 7 and two beds in the third column in Figure 8.

To evaluate the object detection accuracy , we compare out method with the state-of-the-art algorithm of Felzenszwalb et al.[11], as shown in Table I. The evaluation criteria $Average\ Precison\ AP(object)$ are similar to VOC Challenge, $map(result)$ represents the bounding box result of OBFs estimation and $map(gt)$ represents the ground truth of the bounding box.

Besides, we evaluate the geometric labeling in terms of the overlap of each face. R_F represents the result of our OBFs estimation and GT represents the ground truth respectively. Since we have indexed faces of an object, we compute the average precisions of each face $AP(face_i)$.

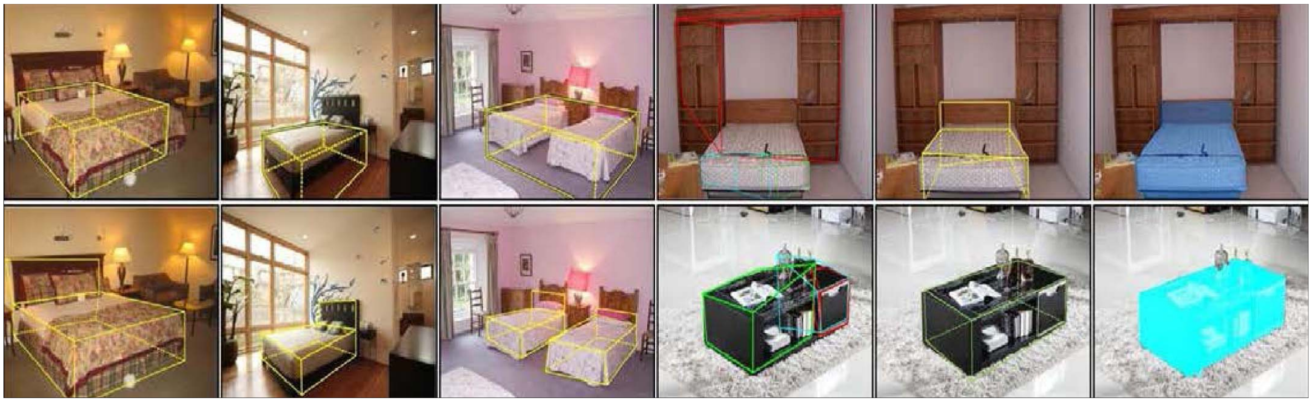

Figure 8. Geometry estimation comparison. In the first three columns, we compare our results (the second row) with that of Hedau et al.[8] (the first row). In the last threee columns, we compare our results (the fifth and sixth columns) with that of Xiao et al. [22] (the fourth column).

$$AP(face_i) = \frac{R_F = face_i \bigcap GT = face_i}{R_F = face_i \bigcup GT = face_i} \tag{9}$$

$$AP(all) = \frac{\bigcup_i (R_F = face_i \bigcap GT = face_i)}{\bigcup_i (R_F = face_i \bigcup GT = face_i)} \tag{10}$$

Method	Indoor dataset [8]		Our dataset	
	Bed	Cube	Bed	Cube
Method [11]	0.636	-	0.641	0.568
Our method	0.768	-	0.766	0.818

Table I
AVERAGE PRECISION. THE ”CUBE” MEANS OBJECTS WHICH HAVE CUBIC SHAPE, SUCH AS TEA TABLE.

	Average Precision			
	Head	Left/right	Horizontal	Front
Indoor Dataset[8]	0.502	0.541	0.478	0.467
Our Dataset	0.495	0.541	0.456	0.428

Table II
AVERAGE PRECISION(AP) OF EACH FACE.

As shown in Table.II, the AP is higher in the $head$ and *left/right side* than in the *Horizontal* and *front* face. The poor precision in *Horizontal* and *front* face is caused by the misleading lines on the bed, such as the pattern on bedsheet or the boundary of pillow and quilt. For the $left/right$ side, although there are redundant lines, our semantic labeling can restrain the geometric labeling to match the proper lines abandoning the outlier ones. Due to this adaptive property, we gain a higher AP on the head of bed.

V. CONCLUSIONS

In this paper, an adaptive template is proposed for parsing object of indoor scene image. Rather than capturing the cuboid of object by estimating camera parameters and detecting cubes, we address the problem of object parsing by template-guided inference of *oriented bounding facets* (OBFs) and semantic labeling of object.

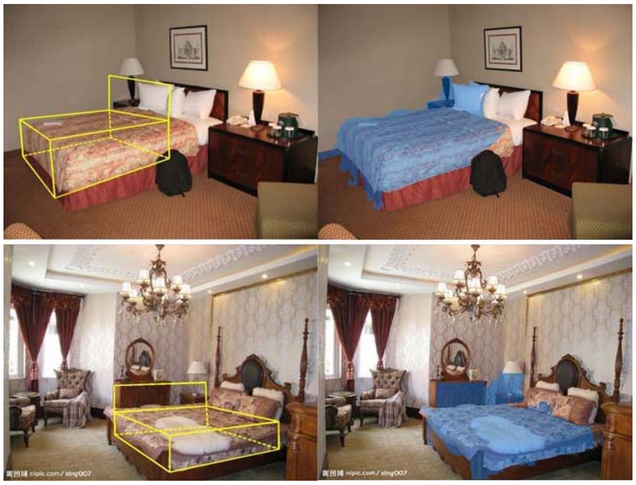

Figure 9. Failure cases. These examples are failed due to poor extracted lines.

Failing cases in Figure 9 suggests that the performance of OBFs estimation is sensitive to the goodness of useful line extraction. As a result, our method may fail in classifing objects into correct classes when encountered with objects bounding with non-straight lines and some misleading lines of other objects. Likewise, we agree with many researchers [8], [10] that features from images are helpful to evaluate the homogeneity of texture in a face, so we aim to extend our object model to learn more features of images from richer data sources and extend our adaptive template to other categories in future work.

ACKNOWLEDGMENT

This work was partially supported by NSFC (61421003) & (61370113), 863 Program (2013AA013801), and SRFDP (20131102130002).

References

[1] X. Ren, L. Bo, and D. Fox, "RGB-(D) Scene Labeling: Features and Algorithms," in *Proc. of CVPR*, 2012.

[2] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Proc. of ECCV (5)*, 2012, pp. 746–760.

[3] L. Nan, K. Xie, and A. Sharf, "A *search-classify* approach for cluttered indoor scene understanding," *ACM Trans. Graph.*, vol. 31, no. 6, p. 137, 2012.

[4] Y. M. Kim, N. J. Mitra, D.-M. Yan, and L. J. Guibas, "Acquiring 3d indoor environments with variability and repetition," *ACM Trans. Graph.*, vol. 31, no. 6, p. 138, 2012.

[5] T. Shao, W. Xu, K. Zhou, J. Wang, D. Li, and B. Guo, "An interactive approach to semantic modeling of indoor scenes with an rgbd camera," *ACM Trans. Graph.*, vol. 31, no. 6, p. 136, 2012.

[6] V. Hedau, D. Hoiem, and D. A. Forsyth, "Recovering the spatial layout of cluttered rooms," in *Proc. of ICCV*, 2009, pp. 1849–1856.

[7] D. C. Lee, A. Gupta, M. Hebert, and T. Kanade, "Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces," in *Proc. of NIPS*, 2010, pp. 1288–1296.

[8] V. Hedau, D. Hoiem, and D. A. Forsyth, "Thinking inside the box: Using appearance models and context based on room geometry," in *Proc. of ECCV (6)*, 2010, pp. 224–237.

[9] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert, "From 3d scene geometry to human workspace," in *Proc. of CVPR*, 2011, pp. 1961–1968.

[10] V. Hedau, D. Hoiem, and D. A. Forsyth, "Recovering free space of indoor scenes from a single image," in *Proc. of CVPR*, 2012, pp. 2807–2814.

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[12] F. Han and S. C. Zhu, "Bottom-up/top-down image parsing with attribute grammar," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 1, pp. 59–73, 2009.

[13] A. Gupta, A. A. Efros, and M. Hebert, "Blocks world revisited: Image understanding using qualitative geometry and mechanics," in *Proc. of ECCV (4)*, 2010, pp. 482–496.

[14] P. Zhao, T. Fang, J. Xiao, H. Zhang, Q. Zhao, and L. Quan, "Rectilinear parsing of architecture in urban environment," in *Proc. of CVPR*, 2010, pp. 342–349.

[15] H. Zhang, T. Fang, X. Chen, Q. Zhao, and L. Quan, "Partial similarity based nonparametric scene parsing in certain environment," in *Proc. of CVPR*, 2011, pp. 2241–2248.

[16] Y. Zhao and S. C. Zhu, "Image parsing with stochastic scene grammar," in *Proc. of NIPS*, 2011, pp. 73–81.

[17] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic, "People watching: Human actions as a cue for single view geometry," in *Proc. of ECCV (5)*, 2012, pp. 732–745.

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[19] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[20] P. D. Kovesi, "MATLAB and Octave functions for computer vision and image processing," Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia, available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.

[21] C. Rother, "A new approach to vanishing point detection in architectural environments," *Image and Vision Computing*, vol. 20, no. 9-10, pp. 647–655, 2002.

[22] J. Xiao, B. C. Russell, and A. Torralba, "Localizing 3d cuboids in single-view images," in *NIPS*, 2012, pp. 755–763.