

Learning 3D Object Templates by Hierarchical Quantization of Geometry and Appearance Spaces

Wenze Hu
Department of Statistics, UCLA
wzhu@stat.ucla.edu

Abstract

This paper presents a method for learning 3D object templates from view labeled object images. The 3D template is defined in a joint appearance and geometry space composed of deformable planar part templates placed at different 3D positions and orientations. Appearance of each part template is represented by Gabor filters, which are hierarchically grouped into line segments and geometric shapes. AND-OR trees are further used to quantize the possible geometry and appearance of part templates, so that learning can be done on a sub-sampled discrete space. Using information gain as a criterion, the best 3D template can be searched through the AND-OR trees using one bottom-up pass and one top-down pass. Experiments on a new car dataset with diverse views show that the proposed method can learn meaningful 3D car templates, and give satisfactory detection and view estimation performance. Experiments are also performed on a public car dataset, which show comparable performance with recent methods.

1. Introduction

This paper presents a method for learning 3D object templates, more specifically, 3D car templates from view labeled images. The 3D templates are defined in a large continuous and compositional space, which are factorized into geometry and appearance spaces. We propose to use AND-OR trees to further quantize and represent the two spaces separately. In this way, the learning problem is posed as an optimization problem in a discrete, structured space, which can be solved efficiently by dynamic programming.

We use an information theoretic measure, namely the information gain, to evaluate candidates of the 3D template and its parts. The information gain for each candidate part is pooled over images of different views. Because meaningful part templates must be aligned across different views, learning them require fewer images than learning a set of view specific templates.

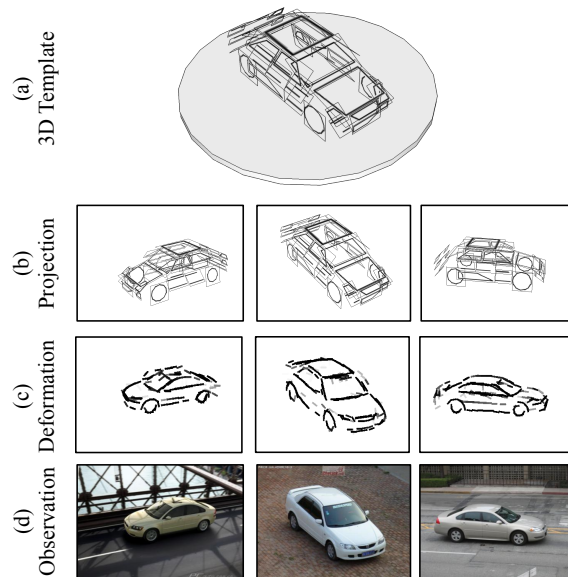


Figure 1. Overall view of the proposed object representation. (a) A learned 3D car template, which is composed of planar part templates. (b) At each specific view, the learned 3D template is projected to derive a 2D template. (c) 2D templates are then deformed to match observations, which are images shown in (d).

Fig.1 shows a learned 3D car template and its deformed projections on object images. The 3D template is composed of planar part templates. Geometry of a part template refers to its size, position and orientation. Appearance of a part template is represented by deformable Gabor filters in images, which are hierarchically grouped into line segments and geometric shapes in 3D space.

We collected a new car image dataset, where viewpoints are more diverse and evenly distributed (see Fig.6). Experiments on this dataset show that the proposed approach can learn meaningful 3D car templates, draw boundaries of object instances on different views, and give satisfactory performance in detecting cars and their poses. Experiments are also done on a popular dataset [14], which show comparable performance with a recent method [9].

Contributions of this paper are three fold: 1.) We propose a sparse compositional and deformable 3D object representation; 2.) An AND-OR tree structure is introduced to express part template compositions, with which an efficient algorithm can be implemented to learn 3D object templates directly from view labeled images; 3.) A new car dataset with diverse and labeled views is presented.

2. Related Literature

Most models in the 3D object recognition literature can be categorized into two classes:

Object centered models. Early work proposes representing 3D objects as a composition of volumetric parts, such as Geons [2] by Biederman *et al.* and 3D primitives [3] by Dickinson *et al.* However, it is difficult to learn and recognize those 3D volumes because of the ambiguity in generating 3D shape proposals from real images. In this paper, we propose to resolve this ambiguity by deformable template projection and information gain pooling, which forms a loop between the 3D representation and corresponding image observations across views.

Recently, many papers [19, 1, 7, 10] proposed recognizing objects using point based 3D models, with appearance as SIFT [11] descriptors, or its quantized version [4]. The SIFT descriptor is good at creating point correspondences. However, by only extracting SIFT features, other salient image information is neglected, such as object boundaries. The proposed model is complementary to these models, since it mainly relies on sketch information from images.

Viewer centered models. These models [17, 15, 12] usually do not assume a global 3D model and are easier to learn, because they do not explicitly enforce appearance consistency across large view discrepancies. Nevertheless, less constraints also means more data are required to learn a robust model for each view.

In terms of model hierarchy, most of the models mentioned above are flat models. For models with parts or feature groups, they are either prefixed by creating a grid on object images [9, 12], or individually clustered without optimizing the global objective of the model [15].

The proposed learning framework uses AND-OR tree structures, which are similar to that used for general knowledge representation in [13]. In computer vision, the proposed AND-OR tree resembles the And-Or Graph by Zhu and Mumford [20], yet instances in the AND-OR tree do not necessarily represent an object interpretation or parse graph, and the embedded grammar in our tree is a context free grammar.

The statistical model used in this paper is consistent with the active curves model [5] and active basis model [18]. In image space, line segments in our part template are realized by a subset of active curves [5], which are deformable templates of straight line segments.

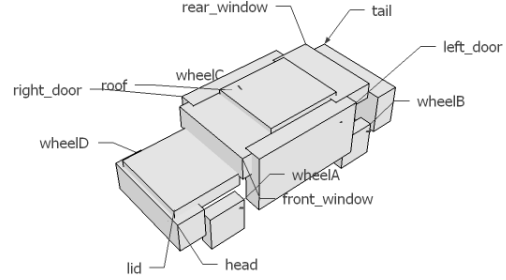


Figure 2. Volumes of interest (VoI) extracted from a 3D CAD model.

3. Template Space Quantification

The space of proposed part templates can be decomposed into geometry space and appearance space. Points in geometry space are parameterized by part template positions, orientations and sizes. Appearance space is compositionally defined from a set of geometric shapes, such as trapezoids, which are decomposed into line segments and further into Gabor elements.

A 3D object template is a composition of 3D part templates, thus it corresponds to a point in the product space of geometry and appearance spaces. We apply three measures to decompose, quantify and organize the space into a hierarchy of geometry and appearance AND-OR trees, so that the template learning problem can be posed as a search problem in a discrete and structured space.

3.1. AND-OR Tree for Part Geometry

The geometry space is first decomposed or reduced by extracting volumes of interest (VoI) as volumes where part templates may exist. This is achieved by parsing semantic object part annotations from a car CAD model file. Sizes of each VoI is rounded to multiples of a unit volume size, which is set to 6 by 6 by 6 inches. Depth direction of each VoI is also defined, which is the VoI side direction facing outward from object center. Extracted VoIs are shown in Fig. 2.

Each VoI is further divided into a set of overlapping sub-volumes, which are used as bounding volumes for the placement of part templates. By placing a 3D grid into VoI, these overlapping sub-volumes can be defined as volumes with vertices on the grid points. For each volume, possible part templates are assumed to be either inscribed or on its frontal surface, so that their possible sizes, positions and orientations can be defined. Examples of the grid and sub-volumes are shown on the nodes in Fig.3(a). As appearance of part templates has yet to be defined, panels are used to illustrate the geometry of part templates, relative to sub-volumes. The distance between nearest grid points is the unit length, which is set to 6 inches. The frontal sur-

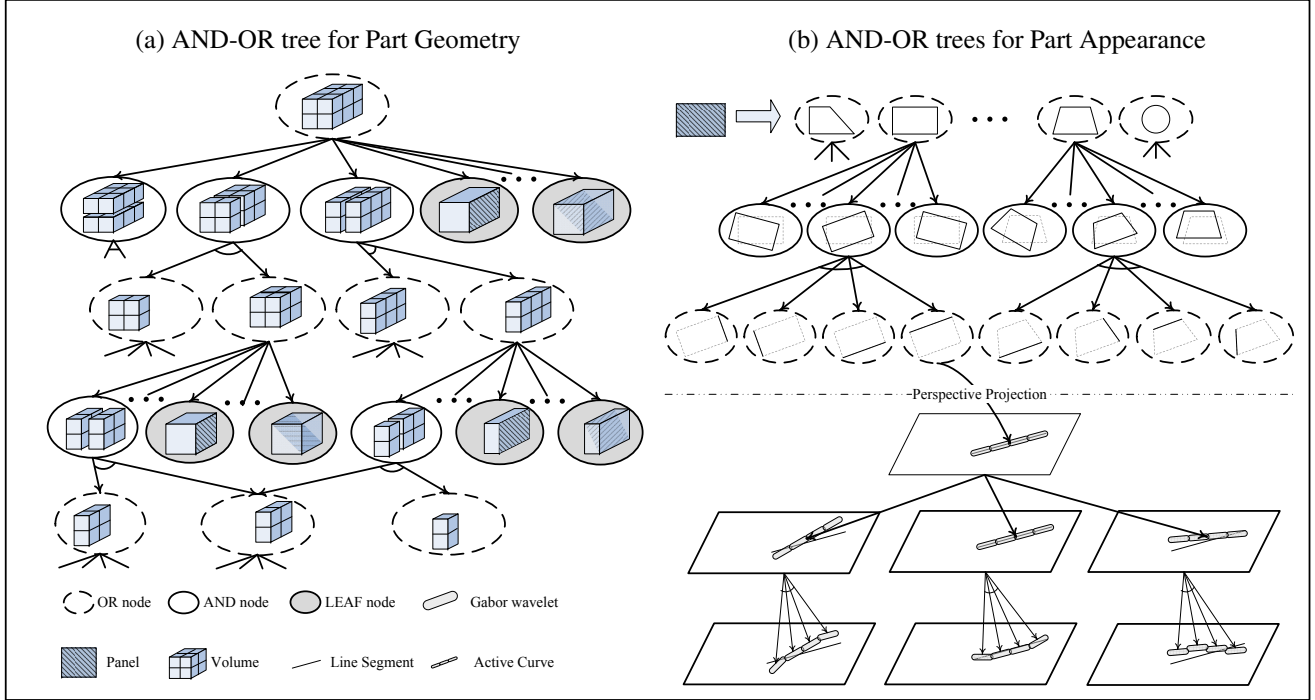


Figure 3. (a): AND-OR tree for part geometry, where AND nodes represent combinations of two sub-volumes occupying larger sub-volumes, OR nodes connect to multiple AND nodes representing possible combinations for the same sub-volume, and leaf nodes represent panels inscribing their parent volumes. (b): Each panel represents geometry of a part template, and are connected to another AND-OR tree for part appearance. Here AND corresponds to composition and OR corresponds to deformation. It extends the geometry AND-OR tree to image spaces since its leaf nodes are Gabor filters.

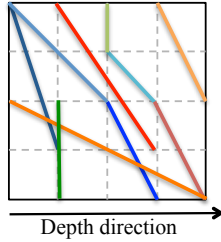


Figure 4. After quantizing VoI, possible panels still preserve varieties in positions and orientations and sizes. Examples of quantified line segments in a box are drawn here to show the analogy.

face of a volume can be decided by the depth direction of the VoI. Though we only allow a restricted set of panels to exist, they still represent large variations in positions, sizes and orientations. To illustrate this situation, an example of this quantization approach on line segments in a 2D box is shown in Fig.4.

The third measure is to organize these sub-volume combinations using an AND-OR Tree. We assume that bounding volumes of final part templates do not overlap, and fully occupies the VoI. With this assumption, the number of possible combinations is still large, but these full combinations may share partial-combinations. Sharing suggests possible reuse of computations on partial combinations, which also

suggests the use of AND-OR tree for identifying and organizing these combinations.

The AND-OR tree can be generated recursively by partitioning volumes and representing partitions by AND-OR node pairs, such as the one shown in Fig.3(a). The OR node connects to all the AND nodes which slice the volume represented by this OR node into two sub-volumes. The OR node also connects to two sets of leaf nodes, where on each node a panel is placed by either inscribing the volume or on the surface perpendicular to the depth direction. Each AND node connects two OR nodes, with each representing one of the two smaller sub-volumes occupying the current sub-volume. This tree starts from a root OR node representing the VoI, and keeps growing until the sub-volumes are divided to a size limit. Currently, the size limit is set to 2 by 2 by 1 of unit size, where 1 is along the depth direction.

Starting from the root node, and by keeping only one child at OR nodes, all selected leaf nodes form a partition of a VoI composed of non-overlapping sub-volumes. This set of leaf nodes represents a combination of panels, which defines the geometry portion of a 3D template candidate.

3.2. AND-OR Tree for Part Appearance

Using panels as bounding boxes, appearance of part templates can be further defined. As shown in Fig.3 (b), pos-

Template type	Appearance	Parameters	Deformation range
Circles		$C = \{\text{center } x, \text{radius } r, \text{number of line segments } n\}$	Translate in $0.1w \times 0.1h$ area.
Trapezoids (rectangle is a special case)		$T = \{\text{parallel line pair } L_1, L_2\}$	Translate in: $0.1w \times 0.1h$ area. ± 11 degree rotation.
Parallel lines		$P = \{\text{parallel line pair } L_1, L_2\}$	Same as above.
Line segments (on panel)		$L = \{\text{center } X, \text{orientation } \theta, \text{length } U\}$ in world coordinate system	No deformation, deformation are expressed in image space.
Active curves (on image)		$I = \{\text{Gabor at center } b_\theta, \text{number of Gabors } u\}$	± 3 pixel translation, ± 11 degree rotation.
Active Gabors		$b = \{\text{center } x, \text{orientation } \theta, \text{scale } s\}$	± 2 pixel translation, ± 11 degree rotation.

Table 1. List of entities used in our representation, their parameters and deformation range.

sible appearance for each part template is also represented by an AND-OR tree, where AND represents composition and OR represents deformation. Layers of AND nodes decompose the part templates into line segments, which are projected to active curves and are further decomposed into Gabor filters. Layers of OR nodes represents the 3D deformation of part templates and 2D deformation of active curves and Gabor wavelets.

As is introduced in Section.1, a part template is a planar template whose appearance form a geometric shape. These geometric shapes include circles, trapezoids, parallel line pairs and line segments, where trapezoids and parallel line pairs each include 6 sub-types shown in column 2 of Table.1.

Shapes in templates are parameterized and decomposed according to these parameters. For example, parameters of a trapezoid is those of its parallel line pair, and each line is parameterized by its orientation, length and centering position. Parameters for the other two line segments can be deduced from the parallel line pair, with which we can decompose the template into four line segments. Parameters for all part templates are shown in column 3 of Table.1.

By restricting sketch shape types, we reduce the space of possible template appearance. We believe the current shape set is enough to represent a variety of vehicle types, such as sedan, vans and pickup. In principle, more shape types could be added to model an even broader range of man made objects categories.

Given the height h , width w and center c of a panel, part

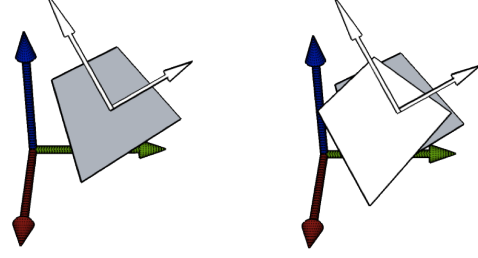


Figure 5. An example of the 3D deformation for part templates. We allow the template to rotate round panel center and translate along the axis direction.

template instances are generated by quantizing parameter spaces in the following way: 1.) Circles: center at c , radius $r = 0.45 \times \min(h, w)$. 2.) Trapezoids: fix the longest line length to be $0.9w$, and the parallel line pair is placed at $1/6h$ away from corresponding side of panel. Lengths of short line are instantiated from $0.9w$ to $0.5w$, decremented by 3 inch. 3) Line pairs: same as Trapezoids. 4) Line segments: center at c , length equals to $0.9w$.

With instantiated parameters, line segments in part templates can be projected onto images, and associated with active curves. In current context, an active curve is a collection of weakly overlapping Gabor wavelets, consecutively placed along the projected line segment. By projecting all the line segments on to a specific view, the 3D object template can be converted to a 2D object template composed of active curves, which should resemble object appearance in that view.

The part templates are deformable, in order to fit geometric shapes to the corresponding object image sketches. At part level, templates can perform in-plane translation and rotation (see Fig.5), which is called 3D deformation. The template is allowed to rotate ± 11 degrees, and translate $\pm 0.1w$ and $\pm 0.1h$ along the corresponding direction, so that there are totally 27 deformations at this level. Projected active curves and Gabor wavelets are also allowed to deform in 2D. Specific 2D deformation ranges are listed in column 4 of Table.1, and their meanings are specified in [5]. Through 3D and 2D deformations, templates of abstract geometric shapes can be adapted to various shapes, where a few examples are shown in row b-d in Fig.1.

After a part template is selected in the learning stage, further selection at OR nodes on the appearance AND-OR tree generates its deformed sketches on images.

4. Template Evaluation by Information Gain

We propose to use information gain to evaluate part template and object template candidates. This is an information theoretic measure that takes into account both the significance of a template in specific views, and the frequency it appears across different views. Besides, by the probabilistic

image model presented below, it can be computed easily as summation of scores of templates sketches.

Denoting an image as \mathbf{I} and its view as ω , we want to build target image distribution $p(\mathbf{I}, \omega)$. We start from a reference distribution $q(\mathbf{I}, \omega)$, which is tilted to approach $p(\mathbf{I}, \omega)$ by updating marginal distributions on the part of image covered by our template:

$$p(\mathbf{I}, \omega | \mathbf{T}) = q(\mathbf{I}, \omega) \prod_{n=1}^N \frac{p(\mathbf{I}_{\Lambda_{T_n}} | T_n, \omega)}{q(\mathbf{I}_{\Lambda_{T_n}} | T_n, \omega)}, \quad (1)$$

where \mathbf{T} is a 3D template composed of N part templates $\{T_n\}_{n=1}^N$, Λ_{T_n} refers to the pixel indexes covered by T_n . In Eqn.(1), the part templates are assumed to be independent, which is valid if the object is of a convex shape and part template projections do not overlap. For part templates not visible at current view ω , $p(\mathbf{I}_{\Lambda_{T_n}})$ is set to be equal to $q(\mathbf{I}_{\Lambda_{T_n}})$. Probability ratios for visible part templates are further decomposed.

As line segments inside a part template do not overlap with each other, the probability ratio of pixels covered by a part template can be further factorized into the product of that covered by its constituent line segments.

$$\frac{p(\mathbf{I}_{\Lambda_T} | T, \omega)}{q(\mathbf{I}_{\Lambda_T} | T, \omega)} = \prod_{k=1}^K \frac{p(\mathbf{I}_{\Lambda_{L_k}} | L_k, \omega)}{q(\mathbf{I}_{\Lambda_{L_k}} | L_k, \omega)}, \quad (2)$$

where L_k denotes the k -th line segment inside T . By projection, likelihood ratio of a line segment is defined to be equal to that of the corresponding active curve:

$$\begin{aligned} s &= \log \frac{p(\mathbf{I}_{\Lambda_L} | L, \omega)}{q(\mathbf{I}_{\Lambda_L} | L, \omega)} = \log \frac{p(\mathbf{I}_{\Lambda_l} | l)}{q(\mathbf{I}_{\Lambda_l} | l)} \\ &= \sum_{g=1}^G \log \frac{p(r_g)}{q(r)} = \sum_{g=1}^G [\lambda h(r_g) - \log Z], \end{aligned} \quad (3)$$

where l is the active curve for L under view ω , r_g is the response of the g -th Gabor wavelet along l , and λ and $\log Z$ are parameters of a corresponding exponential model. By assuming the Gabor response distribution on reference image is position and view independent, a general $q(r)$ is used to replace $q(r_g)$ in Eqn.(3).

Eqn.(3) is denoted as s , because in active curves model, this is also called the score of an active curve hypothesis. Note that as view ω varies, orientation and length of the projected active curve also varies, so that the number of Gabor wavelets G changes across views. The function $h(r)$ performs a sigmoid transform that saturates large Gabor responses. Theoretical underpinnings of this transformation can be found in active basis model [18].

Combining steps above, and on a view labeled M image training set $\{\mathbf{I}_m, \omega_m\}_{m=1}^M$, the information gain of a template $\mathbf{S} = \{S_i\}_{i=1}^N$ between the model distribution and reference distribution can be computed by pooling active curve

scores over all images:

$$\begin{aligned} \text{IG}(\mathbf{S}) &= \iint p(\mathbf{I}, \omega | \mathbf{T}) \log \frac{p(\mathbf{I}, \omega | \mathbf{T})}{q(\mathbf{I}, \omega)} d\mathbf{I} d\omega \\ &\approx \sum_{m=1}^M \log \frac{p(\mathbf{I}_m, \omega_m | \mathbf{S})}{q(\mathbf{I}_m, \omega_m)} \\ &= \sum_{m=1}^M \sum_{n=1}^N \sum_{k=1}^{K_n} s_{mnk} \end{aligned} \quad (4)$$

where s_{mnk} refers to k -th active curve score on n -th part template of m -th image. Also note that score for invisible part templates are zero.

5. Learning 3D Template by AND-OR Search

By connecting (coupling) each panel in leaf node of geometry AND-OR trees with its instantiated appearance AND-OR tree, and connecting the geometry AND-OR trees representing each VoI using an AND parent node, a big AND-OR tree that represents a large set of 3D templates can be constructed. Within this set, the 3D template with maximum information gain can be computed on the tree by one bottom-up pass and one top-down pass. This constitutes the AND-OR search algorithm introduced below.

5.1. AND-OR Search Algorithm

The bottom-up pass starts from sum and max operations at active basis level, followed by sum and max operations at active curves level, where scores of all active curves are computed over each image. These scores are saved in form of score maps, and details of these sum-max operations can be found in [5].

Computing information gains of part templates involves another round of sum-max operations that connects to layers of sum-max operations for the score of active curves. Specifically, information gain of each deformed part template on each image is computed as sum of the scores of its projected active curves. Maximum of these scores is then assigned to the score of the part template on current image. Information gain of a part template is then computed as the sum of these maximum scores over images.

By now, all leaf nodes of geometry AND-OR trees are loaded with information gains, and AND-OR search can be continued on these trees by computing information gains of non-leaf nodes in bottom-up fashion. On each AND node, the information gain is equal to the summation of that on its child OR nodes. On each OR node, information gain is computed as maximum information gain of its child nodes.

A corresponding series arg-max operations from root node to leaf nodes in geometry AND-OR trees retrieves the part template combination leading to this maximum information gain, which is the desired 3D object template.

Further arg-max operations retrieve the deformed part templates, deformed line segments and deformed active basis as sketches on each training image.

As operations within each layer of the tree can be computed independently, they can be done in parallel, which makes the algorithm more efficient.

For specific VoIs, we further learn alternative part template combinations, in order to encode large structural variations within an object category. We use the K-means clustering framework to alternatively impute image cluster labels and learn part template combinations for image clusters iteratively. In the following experiments, we learn two clusters for each of the head, tail and the four wheel VoIs.

5.2. Optimality of AND-OR Search

The sum-max procedure above is in fact a dynamic programming algorithm, which assures the searched information gain is global maximum over all possible part template compositions represented by the AND-OR tree. This is because the information gain is defined recursively along the AND-OR tree, and children AND nodes of a same OR node are independent. With the two conditions, the optimization problem can be recursively decomposed as optimal combination of sub optimization problems. For example, denoting IG_i as information gain at i -th node, we have:

$$\begin{aligned} & \max IG_i^{\text{OR}} \\ &= \max_{j \in ch(i)} \max IG_j^{\text{AND}} = \max_{j \in ch(i)} \max \sum_{k \in ch(j)} IG_k^{\text{OR}} \\ &= \max_{j \in ch(i)} \sum_{k \in ch(j)} \max IG_k^{\text{OR}} \end{aligned} \quad (5)$$

where function $ch(i)$ returns indexes of children nodes of node i . We can get similar recursion starting at an AND node.

The recursion in the AND-OR search above starts from the AND node connecting all the VoIs, and stops at leaf nodes on the appearance AND-OR trees, which corresponds to the convolution of images with specific Gabor filters. Thus the AND-OR search finds optimal combination of 3D part templates by directly pooling evidence over object images from different views.

6. Inference Scheme

6.1. Template Projection and Testing

Given a specific view, a 3D deformable template can be projected to a 2D deformable template, with 3D in-plane deformation of each part template realized by 27 alternative image templates. After projection, the sliding window method is employed, using these deformable 2D templates to perform detection in that view. In each window, dynamic programming is used to infer the max 2D template score

over all possible deformations. Alternative part combinations for VoIs are also treated as deformations: we simply project both cases to a specific view, and the one with highest score prevails.

To perform inference on multiple views, we enumerate discrete views in the view sphere, and use the approach above to perform object detection in each view. To generate discrete views, we fix the internal camera parameters by assuming a general focal length, and discretize the external parameter space of pan, tilt, and camera distance to the world origin. For simplicity, we assume roll angle of the camera is zero. After scanning all windows on enumerated views, object detection windows are reported using non-maximum suppression.

6.2. Feature Weight Adjustment

The proposed learning method builds a sparse object model, which provides good features for object recognition. However, to achieve high recognition performance on image datasets, the reference distribution $q(r)$ should be recalibrated to compensate the error induced by the position invariant assumption. This leads to adjusted weights on the scores of active curves. To this aim, we lump the scores of line segments on the learned template into a feature vector, and use linear SVM to re-train the weights of these features.

6.3. Hypothesis Verification by Color Histogram

In experiments, we found that templates only using sketches tend to generate false positives on highly structured areas, such as fence or brick walls. We use color information to further suppress these false positives, by re-testing on high score windows using both sketch and color information.

To this aim, we evenly sample patches in each part template, and concatenate their color histograms into the feature vector for linear SVM. We allocate 8 bins on each of the 3 color channels, so that for each part template another 24 dimensions are added to the feature vector. Note that the color and sketch features are concatenated into one vector and their weights are trained together.

7. Experiments

7.1. Dataset

There are some widely used datasets emphasizing 3D object recognition [8, 14], but most of them only provide images from a few specified views or limited ranges of views. This could potentially trivialize the problem of 3D object recognition, as models may simply memorize object appearance in these views, thus essentially cast the problem into a multi-class object recognition problem.

We introduce a new dataset¹ of car images, featuring

¹Available at: <http://www.stat.ucla.edu/~wzhu/CVPR12>

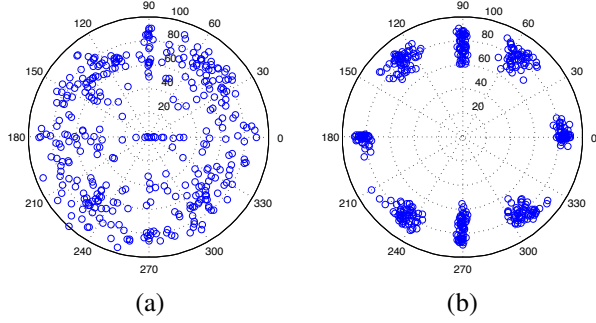


Figure 6. View distribution of our dataset (a) and the 3D car dataset (b) in [14]. The angular direction represents pan angle and radius direction represents tilt angle.

a large variety of views, which are collected uncontrolled from Internet, such as the ones in Fig.1. For each image, we label object view using annotation software provided in the project page of [6]. We also labeled views for all images of car dataset in [14], and show both of them in Fig.6.

In the following experiments, we use 160 of the 360 images dataset as training data, and the rest as testing data.

7.2. Learning Object Templates

Fig.1 shows the learned template for car images. From the template, we can clearly interpret some part templates as wheels, windows. Even some detailed parts such as head-lights and grills can also be recognized. This is benefited by that fact that appearance of the proposed part templates is composed of large and regularized shapes. Plus, the combination of these individual part templates forms a car shape, which demonstrates that 3D templates represented by AND-OR Tree include meaningful ones, and they can be searched through by the proposed algorithm. Deformed templates also demonstrate that the proposed deformation model can adapt the regularized shapes to its variants observed on images.

7.3. Object Recognition Experiments

On our newly collected dataset, we run the inference steps in Section 6 to perform object detection experiment. We search pan angle at 15° interval in $[0^\circ, 360^\circ]$, tilt angle at 5° interval from $[5^\circ, 90^\circ]$, and 8 camera distances for each pan and tilt angle combination.

We show the object detection performance by precision recall curves as shown on the left of Fig.7, where windows with intersection over union area ratio greater than 0.75 are considered positive detection. Specifically, we show the performance of our model using and without using the color features mentioned above. From the curves, we can see that adding color features help the object recognition performance.

For correctly detected instances, we also plot the histogram of view estimation errors on pan angles, which are

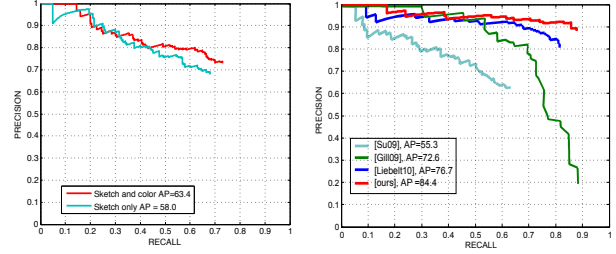


Figure 7. **Left:** Object detection performance on the proposed dataset. **Right:** Object recognition performance on the 3D car dataset [14]. All curves except the red one are from [9].

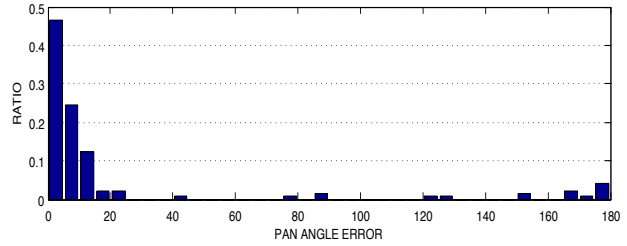


Figure 8. Pose estimation error on our newly collected dataset

shown on Fig.8. From the plot, we can see that majority of the instances are detected at the correct angles. We notice that a few of estimates are totally flipped from head to tail, this suggests we should model more details of head and tails at higher resolutions, as the general shape of cars at flipping views are similar.

We also tried our method on the 3D car dataset in [14]. We use the learned model from the experiment above, and retrain feature weights using training images in this dataset. Object detection performance are evaluated as precision recall curves, which are shown on the right of Fig.7, together with the rest of curves from [9].

We also show the performance of pose estimation task using confusion matrix, together with that from [9] in Fig.9. From the results, we can see that our model achieves higher performance in terms of object detection and comparable performance in pose estimation. Note that according to the evaluation criteria in [16], only correctly detected samples in the testing set are accounted into the confusion matrix. Our model achieved higher detection rate, so more images are accounted into the confusion matrix.

By comparing the confusion matrix, we also find that accuracies for different poses are not as uniform as that in [9]. We believe this is because: 1.) our feature weights are shared across views and 2.) in the re-weighting step, the linear-SVM only optimizes class labeling errors, regardless of the pose estimation performance. With our continuous view formulation, we believe a structured-SVM that optimize both class label and view should eliminate this problem. This is worth investigation in subsequent study.

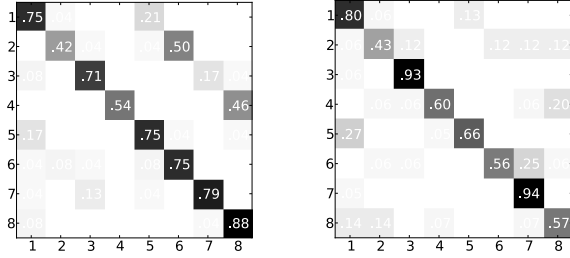


Figure 9. Confusion matrix for pose estimation in dataset [14]. Left: results from [9], AP = 0.70. Right: ours, AP = 0.69.

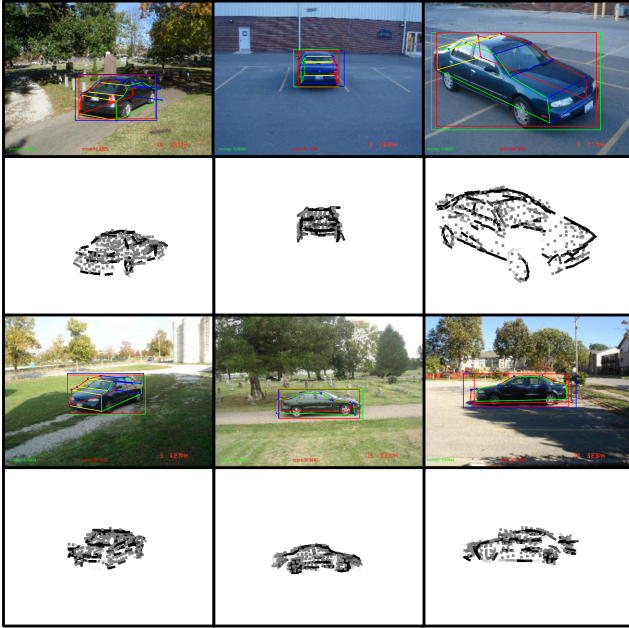


Figure 10. Sample experiment results. Green rectangle: the ground truth object bounding box. Blue rectangle: reported bounding box. Red rectangle: their intersection. 3D wireframe shows estimated object pose. Dots in templates show positions of sampled color patches.

8. Discussion

In this paper, we propose a 3D object representation using part templates of geometric shapes, and a method for learning 3D object templates from images by quantizing spaces. Experiments show that the proposed method can learn meaningful 3D car templates from view labeled images, and give comparable performance in object detection and pose estimation.

Future work includes investigating a better feature re-weighting method and an efficient bottom-up inference algorithm.

Acknowledgement

This project is supported by NSF IIS 1018751, ONR MURI N00014-10-1-0933, DARPA MSEE grant FA8650-11-1-7149 and NSF DMS 1007889. The author would also like to thank Brandon Rothrock for his insightful suggestions.

References

- [1] M. Arie-Nachimson and R. Basri. Constructing implicit 3d shape models for pose estimation. *ICCV*, 2009.
- [2] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–117, 1987.
- [3] S. Dickinson, A. Pentland, and A. Rosenfeld. From volumes to views: An approach to 3-d object recognition. *Computer Vision and Image Understanding*, 1992.
- [4] E. Hsiao, A. Collet Romea, and M. Hebert. Making specific features less discriminative to improve point-based 3d object recognition. In *CVPR*, 2010.
- [5] W. Hu, Y. N. Wu, and S.-C. Zhu. Image representation by active curves. In *ICCV*, 2011.
- [6] W. Hu and S.-C. Zhu. Learning a probabilistic model mixing 3d and 2d primitives for view invariant object recognition. In *CVPR*, 2010.
- [7] A. Kushal, C. Schmid, and J. Ponce. Flexible object models for category-level 3d object recognition. In *CVPR*, 2007.
- [8] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *CVPR*, 2003.
- [9] J. Liebelt and C. Schmid. Multi-view object class detection with a 3d geometric model. In *CVPR*, 2010.
- [10] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint independent object class detection using 3d feature maps. In *CVPR*, 2008.
- [11] D. G. Lowe. Object recognition from local scale-invariant features. *ICCV*, 1999.
- [12] N. Payet and S. Todorovic. From contours to 3d object detection and pose estimation. In *ICCV*, 2011.
- [13] J. Pearl. *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley Longman Publishing Co., Inc., 1984.
- [14] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *ICCV*, 2007.
- [15] H. Su, M. Sun, F.-F. Li, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *ICCV*, 2009.
- [16] M. Sun, H. Su, S. Savarese, and F.-F. Li. A multi-view probabilistic model for 3d object classes. In *CVPR*, 2009.
- [17] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, and L. Van Gool. Towards multi-view object class detection. In *CVPR*, 2006.
- [18] Y. N. Wu, Z. Si, H. Gong, and S.-C. Zhu. Learning active basis model for object detection and recognition. *IJCV*, 2009.
- [19] P. Yan, S. M. Khan, and M. Shah. 3d model based object class detection in an arbitrary view. In *ICCV*, 2007.
- [20] S.-C. Zhu and D. Mumford. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2006.