Structuring Visual Words in 3D for Arbitrary-View Object Localization

Jianxiong Xiao, Jingni Chen, Dit-Yan Yeung, and Long Quan

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{csxjx,jnchen,dyyeung,quan}@cse.ust.hk

Abstract. We propose a novel and efficient method for generic arbitrary-view object class detection and localization. In contrast to existing single-view and multi-view methods using complicated mechanisms for relating the structural information in different parts of the objects or different viewpoints, we aim at representing the structural information in their true 3D locations. Uncalibrated multi-view images from a hand-held camera are used to reconstruct the 3D visual word models in the training stage. In the testing stage, beyond bounding boxes, our method can automatically determine the locations and outlines of multiple objects in the test image with occlusion handling, and can accurately estimate both the intrinsic and extrinsic camera parameters in an optimized way. With exemplar models, our method can also handle shape deformation for intra-class variance. To handle large data sets from models, we propose several speedup techniques to make the prediction efficient. Experimental results obtained based on some standard data sets demonstrate the effectiveness of the proposed approach.

1 Introduction

In recent years, generic object class detection and localization has been a topic of utmost importance in the computer vision community. Remarkable improvements have been reported in the challenging problem of true 3D generic multi-view object class detection and localization [1,2,3]. In this work, we focus on the problem of automatically determining the locations and outlines of object instances as well as the camera parameters by reconstructing 3D visual word exemplar models. The objects in the test images can be at arbitrary view and the camera parameters are completely unknown. Under this setting, object detection and localization is a very challenging problem.

1.1 Related Work

Most existing approaches for object detection focus on detecting an object class from some particular viewpoints by modeling the appearance and shape variability of objects [4]. These approaches, however, are only limited to a few predefined viewpoints. On another research strand, several powerful systems focus on detecting specific objects in cluttered images in spite of viewpoint changes [5,6,7].

Although the reported results are impressive, they can only find specific objects shown in the training images.

In the context of multi-view generic object class modeling and detection, different models with geometric and appearance constraints have been proposed. Thomas *et al.* [1] developed a system for detecting motorbikes and sport shoes by establishing activation links and selecting working views. Savarese *et al.* [2] also proposed a model for 3D object categorization and localization by connecting the canonical parts through their mutual homographic transformation. Without a real 3D model, both methods have to use complicated mechanisms for approximately relating the structural information of the training views or different parts of the objects with simplified assumptions. These indirect representations cannot capture the complete spatial relationship of objects, and may fail to recognize objects when the test images are taken from quite different viewpoints from the training images. In this sense, a real 3D model plays an essential role in further improving the performance of multi-view object class detection. A closely related work is [3], which creates a 3D feature model for object class detection. However, in the process of matching between a test image and the 3D model, their method is computationally costly because it directly operates with a SIFT descriptor and has to enumerate a large space of viewing planes. Another closely related work is [8], which renders a synthetic model from different viewpoints and extracts a set of poses and class discriminative features. During detection, local features from real images are matched to the synthetically trained ones. However, since the features are extracted from a synthetic database, they may deviate significantly from those extracted from real-world images. Moreover, the camera poses are still estimated by searching for the registration of 3D models to images.

1.2   Our Approach

In this paper, we propose an exemplar-based 3D representation of visual words for arbitrary-view object class detection and localization. This model produces a powerful yet simple, direct yet compact representation of object classes. During the training process, our method removes the unknown background of images and obtains the region of interest for class instances. Also, given a test image of arbitrary view containing single or multiple object instances, our algorithm detects all the instances and outlines them precisely. For finding the viewing angle, instead of enumerating all the possible viewpoints of the 3D model, it accurately estimates both the intrinsic and extrinsic camera parameters in an optimized way. Moreover, with exemplar models, our method can also handle shape deformation with intra-class variance. To handle large data sets, several speedup techniques are also proposed to make the prediction more efficient.

2   Automatic Training of 3D Visual Word Models

This section presents the training procedure for the automatic training of 3D visual word models from a set of images taken around each object with unknown background and unknown camera parameters.

2.1 Creating Visual Words and Learning Word Discriminability

Local image patches are the basic building blocks of 2D images. In practice, we choose the Hessian-Laplace detector [9] to detect interest points on a set of images and the SIFT descriptor [10] to characterize local features, described by a set of 128-dimensional SIFT vectors. These SIFT vectors are then vector-quantized into visual words by k-means [11]. Each visual word is a cluster center of the quantized SIFT vectors. In our work, this procedure is performed over an image set containing two types of images. One type contains images taken around the objects for reconstructing 3D models. The other type contains the training images from the PASCAL Visual Object Classes (VOC) challenge [12]. We take the visual words as descriptors for the interest points in both 2D images and 3D models.

For a particular class, not all the visual words play the same role in detection. For a particular visual word w, its weight to an object class C_i is learnt by a ratio discriminability function [13],

$$D_i\left(w\right) = \frac{\#\ \text{images in}\ C_i\ \text{containing}\ w}{\#\ \text{images in image set containing}\ w}. \tag{1}$$

The word weight measures the relevance of w with respect to the object class C_i. The higher the value of $D_i\left(w\right)$, the more discriminative the visual word w is. For each object class, we only preserve the top 512 most discriminative visual words for its 3D models.

2.2 Creating 3D Visual Word Models

With these visual words, several exemplar models for each object class are created. For each exemplar model $\langle\mathcal{M}, \mathcal{M}^+\rangle$, the training procedure is shown in Fig. 1.
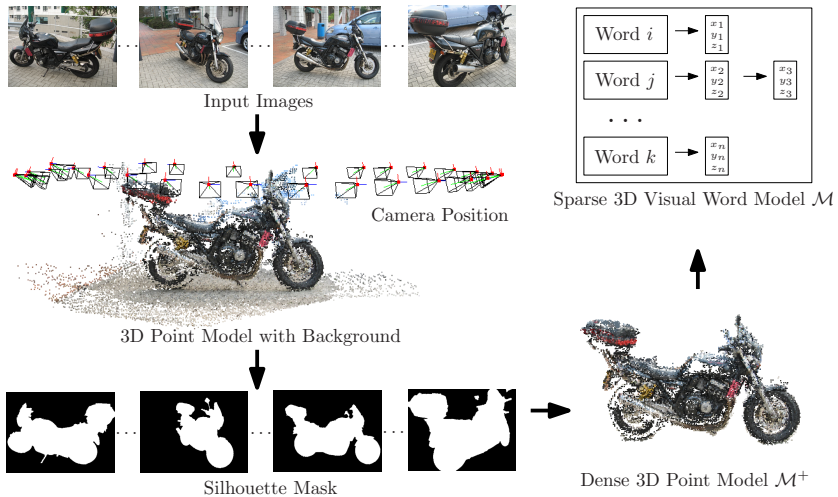

Fig. 1. Training Procedure for an Exemplar Model

In the first step, the input multiple-view images are used for 3D reconstruction by the standard Structure from Motion algorithm [14]. Specifically, the unordered input images are matched in a pairwise manner by the visual words. Taking these sparse pixel-to-pixel correspondences as seeds, a dense matching is obtained by [15]. Then, for three images with more than six mutual point correspondences, a projective reconstruction is obtained by [16]. We merge all the triplet reconstructions by estimating the transformation between those triplets with two common images as in [17]. Finally, the projective reconstruction is metric upgraded to Euclidian reconstruction. In each step, bundle adjustment is used to minimize the geometric error. Since our training data do not contain any label information about the object location in the image, not only the target object but also the background of the scene is reconstructed. However, we only want to preserve the 3D model for the target object.

Hence, in the second step, a graph-cut based method [18] is used to automatically identify image regions corresponding to a common space region seen from multiple cameras. Briefly, we assume that the background regions present some color coherence in each image and we exploit the spatial consistency constraint that several image projections of the same space region must satisfy. Each image is iteratively segmented into two regions such that the background satisfies the color consistency constraints, while the foreground satisfies the geometric consistency constraints with respect to the other images. An EM scheme is adopted where the background and foreground model parameters are updated in one step, and the images are segmented in the next step using the new model parameters. Because the silhouette is just used to filter out the background of the 3D model, it does not need to be very precise. In most situations, the above automatic extraction results are satisfactory. In other cases, an interactive method [19] can be used. In our experiment, 8.5% of the silhouettes are annotated manually by [19].

After we have extracted the silhouette of the target object, we filter out all 3D points with projection outside the silhouette of the object and the set of remaining 3D points is the model $\mathcal{M}^+$. To facilitate fast indexing and dramatically accelerate the detection, we record some 3D points in a hash table model $\mathcal{M}$, with visual words as keys and the 3D points with coordinate (x, y, z) as content. The 3D points in the hash table model $\mathcal{M}$ are from the sparse matching seeds of $\mathcal{M}^+$ and correspond to the top 512 most discriminative visual words.

3   Object Localization and Camera Estimation

Given a new image with single or multiple instances, the task is to detect the locations of objects from a particular class, outline them precisely and simultaneously estimate the camera parameters for the test image. With the trained 3D exemplar models, our method can estimate arbitrary pose of the target object with no restriction to some predefined poses. The flow of the testing procedure is shown in Fig. 2 and Alg. 1.
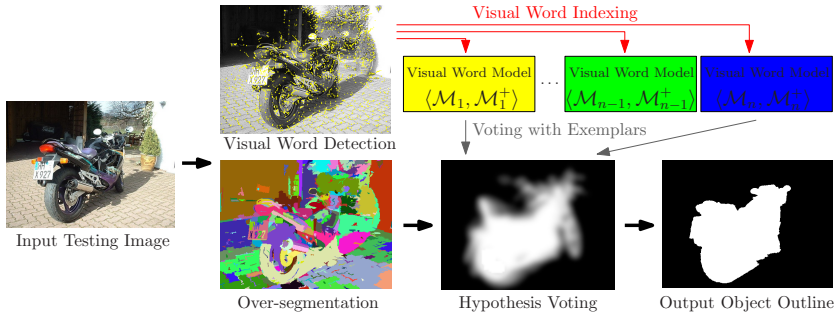
Fig. 2. Testing Procedure

Algorithm 1. Simultaneous Object Localization and Camera Estimation

1. Over-segment the test image I.
2. For each small region R_i in the over-segmentation and each exemplar model $\langle \mathcal{M}_j, \mathcal{M}_j^+ \rangle$,
   (a) get all 2D and 3D correspondence pairs $\mathcal{S}_{ij}$ inside the region R_i
   (b) compute the camera projection matrix P_{ij} by SVD
   (c) project the 3D point model $\mathcal{M}_j^+$ and vote in the image space for hypothesis.
3. Take the cumulative voting score as data cost and image gradient as smoothness in MRF to extract the outline O.
4. Use all 2D and 3D correspondence pairs $\mathcal{S}^*$ inside each connected component R' of the outline O to compute the final camera matrix P^*.

3.1 Visual Word Detection and Image Over-segmentation

We follow the same procedure as in training to find local interest points in a test image by the Hessian-Laplace detector [9] and characterize the local features by a set of 128-dimensional SIFT vectors [10]. Each SIFT descriptor is then translated into its corresponding visual word by finding the nearest visual word around it. If the Euclidean distance between the SIFT descriptor of the interest point and that of the nearest visual word is two times larger than the mean distance of that cluster from its centroid, that interest point is deleted. The mapping from SIFT descriptor to visual word descriptor makes the matching between 2D image interest point and 3D visual word model very efficient by just indexing with the visual word as key.

The target object in the test image may be embedded in a complicated background that will affect the overall performance of detection and localization. Over-segmenting the test image can help to improve the accuracy of object detection and get a much more precise outline of the object. It will also be useful for camera hypothesis estimation in the testing stage. Traditionally, over-segmentation is done by the watershed or mean-shift algorithm. In this work we adopt the over-segmentation technique by [20], which is very efficient and also stable with parameters to control the region size.

3.2   Visual Word Indexing and Hypothesis Voting

Suppose a test image I is over-segmented into n regions and there are m exemplar models. For each small region R_i in I and each 3D visual word model $\mathcal{M}_j$, all correspondence pairs of 2D interest point $\mathbf{u}_k$ inside R_i (from the test image I) and 3D point $\mathbf{X}_k$ (from the 3D visual word model $\mathcal{M}_j$) that have the same visual word descriptor are collected:

$$\mathcal{S}_{ij} = \{\mathbf{u}_k \leftrightarrow \mathbf{X}_k \,|\, w(\mathbf{u}_k) = w(\mathbf{X}_k), \mathbf{u}_k \in R_i, \mathbf{X}_k \in \mathcal{M}_j\}$$

Given N correspondence pairs between the 2D projections and 3D points, the camera pose can be directly estimated by a linear unique-solution N-point method [21] with SVD as the solver.

To improve the robustness of the above method, we refine it to automatically filter out some obvious error correspondences in $\mathcal{S}_{ij}$. The filtering algorithm is based on the following locality assumption: The 3D points $\{\mathbf{X}_k\}$, with 2D projection $\{P_{ij}\mathbf{X}_k\}$ inside the same small over-segmentation region R_i, should be also close to each other in 3D space. This assumption empirically holds since the over-segmentation algorithm tries not to cross depth boundaries. With this assumption, we first compute the average 3D position $\bar{\mathbf{p}}$ of the 3D points in $\mathcal{S}_{ij}$. Then we filter out the correspondence pairs whose 3D points are far away from $\bar{\mathbf{p}}$. Specifically, we compute the mean $\bar{d}$ and standard deviation σ from the distances between $\bar{\mathbf{p}}$ and all the 3D points in $\mathcal{S}_{ij}$. Then if the distance between a 3D point of a particular correspondence pair and $\bar{\mathbf{p}}$ is greater than $\bar{d} + 2\sigma$, this correspondence pair is removed from $\mathcal{S}_{ij}$.

Since the camera matrix P_{ij} is estimated from a local over-segmentation region R_i, it is likely to be degenerated if the 3D points are nearly planar. Hence, to further improve the camera estimation robustness, instead of the sparse visual word model $\mathcal{M}_j$, we make use of the dense 3D point model $\mathcal{M}_j^+$ to increase the number of 2D to 3D correspondences for camera estimation. In detail, each 2D interest point $\mathbf{u}_k$ to 3D point $\mathbf{X}_k$ correspondence $\mathbf{u}_k \leftrightarrow \mathbf{X}_k$ in $\mathcal{S}_{ij}$ is taken as the seed, and the pixels in the neighborhood of $\mathbf{u}_k$ in R_i are greedily matched with the points in the neighborhood of $\mathbf{X}_k$ in the model $\mathcal{M}_j^+$. In this way, a new set $\widehat{\mathcal{S}}_{ij}$ of 2D to 3D correspondences can be obtained. $\widehat{\mathcal{S}}_{ij}$ contains much more correspondences that can characterize the local geometry changes and hence can greatly improve the camera estimation robustness. With the new correspondence pair set $\widehat{\mathcal{S}}_{ij}$, the camera matrix P_{ij} is computed in the same way as before.

After estimating the camera matrix P_{ij}, we project the whole 3D model $\mathcal{M}_j^+ = \{\mathbf{X}_k^+\}$ onto the test image with projections $\{P_{ij}\mathbf{X}_k^+\}$ and vote in the image space for the hypothesis P_{ij}. In detail, we lay over the test image I a regular grid with the same resolution as the image. For each $\mathbf{X}_k^+ \in \mathcal{M}_j^+$, the value of the cell in position $P_{ij}\mathbf{X}_k^+$ will increase by one. Therefore, for each over-segmentation region R_i, there is one vote for each exemplar model $\langle \mathcal{M}_j, \mathcal{M}_j^+ \rangle$. Because each over-segmentation region R_i has its vote, our method is insensitive to occlusion since other un-occluded regions can still vote for the occluded regions. To increase the

effective regions for each point $\mathbf{X}_k^+$, the neighboring grid cells of $P_{ij}\mathbf{X}_k^+$ also have scores from $\mathbf{X}_k^+$ weighted with a 2D isotropic Gaussian. In our case, the variance is set to be 0.5% of the width of image I.

However, if most parts of the small region are not the object of interest, the estimated camera projection matrix will be completely useless. In order to capture the difference, the hypothesis P_{ij} is associated with a score $c\,(R_i, \mathcal{M}_j)$ indicating the confidence of the vote:

$$c\,(R_i, \mathcal{M}_j) = \left(\mathrm{median}_{\mathbf{X}_k \in \mathcal{M}_j, \mathbf{u}_k \in I, w(\mathbf{u}_k)=w(\mathbf{X}_k)}\left\{\|\mathbf{u}_k - P_{ij}\mathbf{X}_k\|\right\} + 1\right)^{-1}. \quad (2)$$

The smaller the re-projection error $\|\mathbf{u}_k - P_{ij}\mathbf{X}_k\|$, the higher the confidence. Here, $\mathbf{u}_k$ and $\mathbf{X}_k$ form a correspondence pair. However, the 2D and 3D visual word correspondence is not necessarily a bijection. Several 2D interest points $\{\mathbf{u}_k\}$ in the test image may have the same visual word w, and hence may correspond to several 3D points $\{\mathbf{X}_k\}$ in $\mathcal{M}_j$. For such multiple matched pairs $\{\mathbf{u}_k\} \leftrightarrow \{\mathbf{X}_k\}$, the re-projection error is computed as the minimum distance between any 2D interest point $\{\mathbf{u}_k\}$ and the projection $\{P_{ij}\mathbf{X}_k\}$ of any 3D visual word $\{\mathbf{X}_k\}$.

3.3   Outline Extraction and Camera Matrix Re-estimation

The over-segmentation regions are used to construct a Markov random field (MRF) graph. The smoothness cost is defined as the $L2$-norm of the RGB color difference between the background and the target object, as in [22]. The corresponding voting score is normalized and taken as the data cost in the MRF. An implementation of the graph cut algorithm from [23] is used for optimization and getting the outline O. Inside the outline O, we can obtain several connected components $\{R_i'\}$. We use all corresponding pairs inside each connected component region R_i' and the best matched 3D visual word model $\mathcal{M}^*$ to re-estimate the camera matrix P^* by the same method as in Sec. 3.2. Here, the best matched 3D visual word model $\mathcal{M}^*$ for that connected component region R' is the one with the highest cumulative voting score summing up all over-segmentation regions R_i in R', i.e.,

$$\mathcal{M}^* = \arg\max_{\mathcal{M}_j} \sum_{R_i \in R'} c\,(R_i, \mathcal{M}_j).$$

In fact, for each target object in the test image, what we want to estimate is its relative pose and the camera parameters. Since each 3D point model has its own coordinate system, the camera so estimated is specific to that coordinate system. If multiple object instances exist in the test image, multiple cameras, one for each object instance, should be estimated in the respective coordinate system for the corresponding 3D point model. These multiple cameras do not violate the principle that there is only one camera for each image according to the perspective camera imaging theory. Because they are at different coordinate systems and will align to be exactly one camera (only theoretically when there is no noise) in the real-world coordinate system. In our case, we are more

concerned about the relative pose between each object and the corresponding camera. Hence, we do not try to align the multiple cameras for multiple objects.

Now, for multiple object instances from the same object class in the same test image, if the objects do not overlap with each other, the outline O will have several connected components $\{R'_i\}$, and several best matched models $\{\mathcal{M}^*_j\}$ as well as several estimated cameras $\{P^*_{ij}\}$. If the objects overlap greatly with each other, the object outlines can still be estimated correctly although the cameras cannot be estimated well. For objects from different classes, exemplars from different classes will vote on different grids. The voting score is normalized as the data cost in MRF, and multi-label graph-cut can be used to find the optimal outline for each class. After that the same procedure as in the single class case is used to estimate the cameras for each class separately.

3.4   Acceleration

Unlike previous 2D voting based methods, our method is computationally more expensive due to the larger data size. The bottleneck is, for each region R_i and each 3D model $\mathcal{M}_j$, there is one SVD operation to compute the camera parameters and many matrix multiplications to project all 3D points onto the 2D grid. However, for different over-segmentation regions and different 3D exemplar models, there is no computational dependency. So it is possible to do parallel computing for different hypotheses. Here, we make use of a commercial programmable graphics hardware, a graphics processing unit (GPU), to speed up the testing procedure. The SVD algorithm is implemented as in [24] which mainly includes two steps: bidiagonalization of the given matrix by applying a series of householder transformations, and diagonalization of the bidiagonal matrix by iteratively applying the implicit-shifted QR algorithm. In practice, after the camera matrix is computed from SVD, the projection matrix in GPU is set to be the same as the camera matrix, and the 3D model is rendered on the GPU while the frame buffer is set to have the same resolution as the test image.

To speed up and handle intra-class variance, for each class, we use only some most similar 3D exemplar models for hypothesis voting. For a rigid class with small intra-class variance, the voting values from the top five most similar exemplar models are added together to improve the robustness. For a very deformable object class such as a person class, however, we use only one most similar exemplar model rather than five for computation.

4   Experiments

There is a training data set with motorbikes and shoes provided by Leuven [1]. However, this data set is not specialized for 3D reconstruction, since the baseline is too large to achieve a reliable two-view matching for structure from motion. In fact, in our experiment, only two motorbike models can be successfully reconstructed from this data set. Due to the lack of an appropriate multi-view database for 3D reconstruction for the purpose of object class detection, we construct a
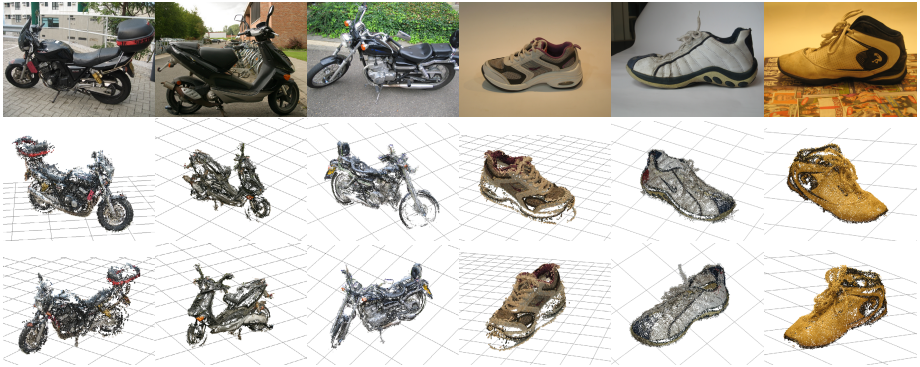
Fig. 3. Some 3D exemplar models. The first row shows one of the training images for each model. The second and third rows show two example views of the corresponding 3D point models.
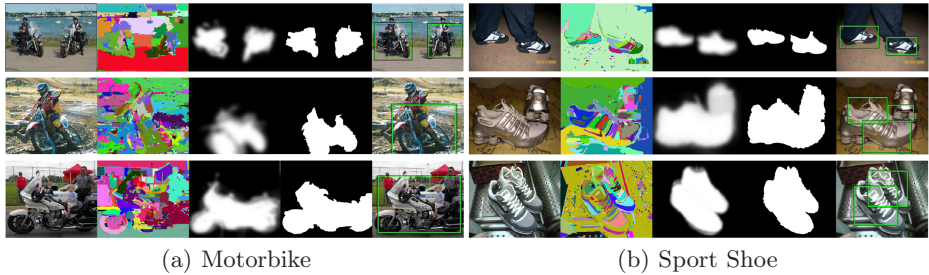

(a) Motorbike   (b) Sport Shoe

Fig. 4. Some output examples. For each subfigure, the first column contains the input test images, the second column contains the over-segmentation images, the third column contains the voting results, the fourth column contains the outlines of the detected objects, i.e., the final result of our method, and the fifth column contains the result from [1].

3D object database with 15 different motorbikes and 30 different sport shoes. For each object, about 30 images with resolution 800×600 are taken around it and the camera parameters are completely unknown. Fig. 7 shows some sample images of our data set. Our exemplar models are mainly trained based on this data set. Hence, including the two motorbikes reconstructed from Leuven's data set [1], there are 17 motorbike exemplar models and 30 shoe exemplar models in our experiments. Some 3D exemplar models are shown in Fig. 3.

For a test image with resolution 480×360, it takes about 0.1 second for over-segmentation, 6.1 seconds for hypothesis voting, and 0.5 second for outline extraction on a desktop PC with Intel Core 2 Duo E6400 CPU and NVIDIA GeForce 8800 GTX GPU. For voting, we use the five most similar exemplar models. Fig. 4 shows some results of over-segmentation, hypothesis voting and outline extraction. Our method can handle occlusion very well, such as the persons

(a)   (b)   (c)   (d)

Fig. 5. Example results of camera estimation. The left of each subfigure is the input test image, and the right is the best matched 3D exemplar model with the estimated camera for the test image shown as the top view in 3D space. The camera is drawn by lines.

on the motorbike. The estimated camera positions of some test images are also shown in Fig. 5.

4.1   Evaluation and Comparison

For comparison with [1] and [2], although our model is obtained from different training data using different kinds of supervision, it can be evaluated on the same test set. We adopt the same evaluation protocol as in the PASCAL VOC Challenge, which is also used in [1,2]. Precison/recall curves are used to evaluate the performance of localization.

We adopt the same 179 images from the 'motorbikes-test2' set provided by the PASCAL VOC Challenge 2005 [25] for testing. Fig.6(a) shows a substantial improvement of our method compared to [1]. Although our performance in terms of precision is similar to that of [2], we regard it as satisfactory, given the fact that the number of exemplar models is not large enough in our motorbike experiment.

For more comparison, we use Leuven's multi-view sports shoes data set for testing [1]. The result is shown in Fig. 6(b). Observing that our proposed method is significantly better than [1], we believe that this is partially due to the larger and better training data that we used. [2] did not report results on Leuven's multi-view sports shoes data set and the shoes in their own data set are mainly leather shoes. Hence, we do not compare with [2] on shoes.
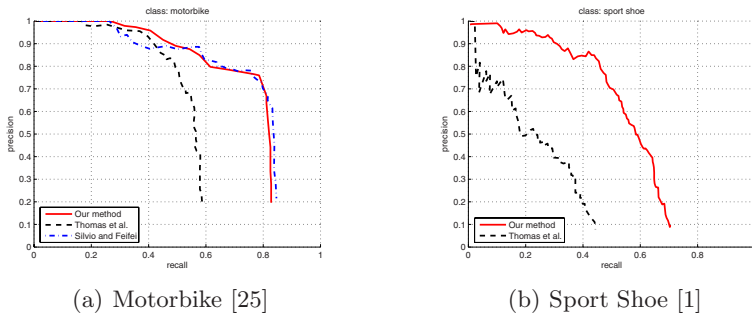

(a) Motorbike [25]   (b) Sport Shoe [1]

Fig. 6. Precision-recall Curves

Fig. 7. Sample images from our 3D object category data set

4.2 Discussions

Our approach may be seen as a significant extension of many previous works. The PASCAL VOC 2007 Detection Task winner [26] can be seen as the 2D version of our method, although their method uses histogram due to the lack of explicit structural information. [3] is a much simplified version of our method and does not take the efficiency issue into consideration, while [8] approximates our 3D visual word model by synthetic data, and both of them determine the camera matrix through searching. To handle large intra-class shape variance, state-of-the-art representations such as [27] rely on deformable part models. Extending the deformable models to 3D is feasible but quite complicated. In our method, instead of explicitly modeling the deformation, we use an exemplar-based method to characterize the intra-class variance.

On the other hand, our method extensively uses many standard state-of-the-art methods for different problems in computer vision as building blocks, making it easy to implement and achieve good performance. The Structure from Motion algorithm [14] from the multiple view geometry community is used to reconstruct the 3D positions for the visual words. Efficient over-segmentation [20] from the image segmentation community is used to outline the region in which visual word matching is collected for hypothesis voting. A max-flow based MRF solver [23] from the energy minimization community is used to extract the object boundary. Moreover, a graphics hardware GPU is used to accelerate the voting procedure including camera estimation using SVD.

5 Conclusion

We have proposed a novel and efficient method for generic object class detection that aims at representing the structural information in their true 3D locations. Uncalibrated multi-view images from a hand-held camera are used to reconstruct the 3D visual word models in the training stage. In the testing stage, beyond bounding boxes, our method determines the locations and outlines of multiple objects in the test image, and accurately estimates the camera parameters in an optimized way. To handle large data sets, we propose several speedup techniques to make the prediction efficient. However, as a limitation of our method, more specific training data needs to be collected than many previous methods. Future work includes conducting more experiments with more object classes such as person classes, and extending our method to estimate the camera parameters for highly overlapping objects.

References

1. Thomas, A., Ferrari, V., Leibe, B., Turtelaars, T., Schiele, B., Gool, L.V.: Towards multi-view object class detection. In: IEEE Conference Computer Vision and Pattern Recognition, vol. 2, pp. 1589–1596 (2006)
2. Savarese, S., Fei-Fei, L.: 3D generic object categorization, localization and pose estimation. In: IEEE International Conference on Computer Vision, pp. 1–8 (2007)
3. Yan, P., Khan, S., Shah, M.: 3D model based object class detection in an arbitrary view. In: IEEE International Conference on Computer Vision, pp. 1–6 (2007)
4. Dorko, G., Schmid, C.: Selection of scale-invariant parts for object class recognition. In: IEEE International Conference on Computer Vision, vol. 1, pp. 634–639 (2003)
5. Ferrai, V., Tuytelaars, T., Gool, L.V.: Simultaneous object recognition and segmentation by image exploration. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 40–54. Springer, Heidelberg (2004)
6. Lowe, D.: Local feature view clustering for 3D object recognition. In: IEEE Conference Computer Vision and Pattern Recognition, vol. 1, pp. 682–688 (2001)
7. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In: IEEE Conference Computer Vision and Pattern Recognition, vol. 2, pp. 272–277 (2003)
8. Liebelt, J., Schmid, C., Schertler, K.: Viewpoint-independent object class detection using 3D feature maps. In: IEEE Conference Computer Vision and Pattern Recognition (2008)
9. Mikolajczyk, K., Leibe, B., Schiele, B.: Multiple object class detection with a generative model. In: IEEE Conference Computer Vision and Pattern Recognition (2006)
10. Lowe, D.: Object recognition from local scale-invariant features. In: IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 (1999)
11. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: IEEE International Conference on Computer Vision, vol. 2, pp. 1470–1477 (2003)
12. Everingham, M., Zisserman, A., Williams, C.K.I., Van Gool, L.: The PASCAL Visual Object Class challenge 2006 (VOC 2006) results (2006)
13. Dorkó, G., Schmid, C.: Object class recognition using discriminative local features. IEEE Transaction on Pattern Analysis and Machine Intelligence (2004)
14. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
15. Xiao, J., Chen, J., Yeung, D.Y., Quan, L.: Learning two-view stereo matching. In: European Conference on Computer Vision (2008)
16. Quan, L.: Invariant of six points and projective reconstruction from three uncalibrated images. IEEE Tranactions on Pattern Analysis and Machine Intelligence 17(1), 34–46 (1995)
17. Lhuillier, M., Quan, L.: A quasi-dense approach to surface reconstruction from uncalibrated images. IEEE Transaction on Pattern Analysis and Machine Intelligence 27(3), 418–433 (2005)

18. Lee, W., Woo, W., Boyer, E.: Identifying foreground from multiple images. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part II. LNCS, vol. 4844, pp. 580–589. Springer, Heidelberg (2007)
19. Xiao, J., Wang, J., Tan, P., Quan, L.: Joint affinity propagation for multiple view segmentation. In: IEEE International Conference on Computer Vision, pp. 1–7 (2007)
20. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International Journal of Computer Vision 59, 167–181 (2004)
21. Quan, L., Lan, Z.: Linear n-point camera pose determination. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(8), 774–780 (1999)
22. Li, Y., Sun, J., Tang, C.K., Shum, H.Y.: Lazy snapping. ACM Transaction on Graphics 23(3), 303–308 (2004)
23. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. IEEE Transaction on Pattern Analysis and Machine Intelligence (2004)
24. Galoppo, N., Govindaraju, N.K., Henson, M., Bondhugula, V., Larsen, S., Manocha, D.: Efficient numerical algorithms on graphics hardware. In: Workshop on Edge Computing Using New Commodity Architectures (2006)
25. Everingham, M., et al.: The 2005 PASCAL Visual Object Class challenge. In: Selected the 1st PASCAL Challenges Workshop (2005)
26. Chum, O., Zisserman, A.: An exemplar model for learning object classes. In: IEEE Conference Computer Vision and Pattern Recognition, pp. 1–8 (2007)
27. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multi-scale, deformable part model. In: IEEE Conference Computer Vision and Pattern Recognition (2008)