Hough Voting with Distinctive Mid-Level Parts for Object Detection

Xiaoqin Kuang, Nong Sang*, Feifei Chen, Runmin Wang, and Changxin Gao

Science and Technology on Multi-spectral Information Processing Laboratory,
School of Automation, Huazhong University of Science and Technology,
Wuhan, China, 430074
{kxqkuang,nsang,ffchen,runminwang,cgao}@hust.edu.cn

Abstract. This paper presents an efficient method for object detection in natural scenes. It is accomplished via generalized Hough transform of distinctive mid-level parts. These parts are more meaningful than low-level patches such as lines or corners and would be able to cover the key structures of object. We collect the initial sets of parts by clustering with k-means in WHO space and train LDA model for every cluster. The codebooks are generated by applying the trained detectors to discover parts in whole positive training images and storing their spatial distribution relative to object center. When detecting in a new image, the energy map is formed by the voting from every entry in codebook and is used to predict the location of object. Experiment result shows the effectiveness of the proposed scheme.

Keywords: Object detection, Hough Voting, Mid-level Parts, LDA.

1 Introduction

The detection task aims at locating the same object as the given training images in natural scenes. Due to the large intra-class variations in structure or viewpoint and appearance, a single linear classifier over HOG feature vectors can hardly perform well for generic object detection.

In order to handle the large intra-class variation, exemplar-svm [1] uses multiple components instead of single monolithic detector. However, this method has a huge computational complexity to training a separate SVM for each positive example. Method based on parts can solve the problem to a certain extend since parts are easier to compute than whole object and additionally, they can be shared in different instances which would decrease the complexity. Part-based methods [2, 3, 4, 8, 10, 11, 12, 13] become popular in the field in recent years. It is also robust to solve the partial occlusions in detection.

The work in [2] discovers parts with partial correspondence by annotating important matching points between instances of a category. They use the sematic graph to propagate the correspondence and augment part in learning procedure as well.

[*] Corresponding author.

S. Li et al. (Eds.): CCPR 2014, Part I, CCIS 483, pp. 305–313, 2014.

However, the annotation work is time-consuming and hard to do when facing more new categories. Finding parts automatically is important.

The implicit shape model(ISM) in [3] generates codebooks by clustering patches of interest point. The location of patches occurring in object is stored to reflect the spatial distribution of codebook entries. However, this method uses low-level patches as codebook; plenty patches would be found and some of them with little structure information are ineffective to vote.

In method of Hough Forest [4, 5], patches were sampled with uniform probabilities from positive and negative training images. They construct a random forest and each node stores the statistics of class and spatial information. Every leaf node plays the role of codebook to cast probability vote for position for test images. The more densely they are sampled, the more accurate the detection is. But among the sampled patches, many are slightly different or not distinctive enough to vote efficiently.

We intend to find a way of collecting fewer patches but with more information which is typical to reflect the appearance of object class. Mid-level parts have the advantage to be informative. And with the Hough voting method the individual parts are integrated to estimate position of the entire object for detection.

We follow the training method in [6] with iterative algorithm but use LDA model as in [7] instead of SVM to train classifiers of distinctive parts. The authors in [8] propose that a discriminative part should occur frequently from the category it is learnt but rarely from others. They use the Entropy-Rank Curves to measure the discriminative ability. We adopt this idea to select parts in training as well.

Once the collection of part detectors are trained, we apply them to training positive set and get complete parts as codebook for each detector stored with a set of scores and offsets respect to the object centroid.

The experiment result by the proposed scheme achieves good detection results in UIUC-car data sets. The block diagram of the proposed system is shown in Fig. 1.

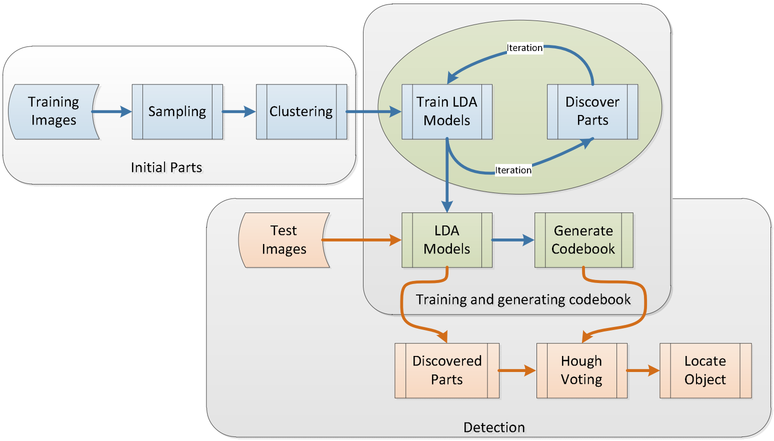

Fig. 1. Block diagram of our method

2 Learning Distinctive Parts

In this section, we describe the procedure of how to find the distinctive parts for codebook. A codebook is a vocabulary of local appearances that are characteristic to reflect the structure or viewpoint of the known object. It is impossible to annotate all distinguished parts manually additionally which is time-consuming and hard to label the same structure exactly for all training images.

2.1 Initial Parts

The ISM [3] uses interest point detectors such as Harris to find interest patches initially and clusters them to generate codebook. However, the low-level patches are easy to repeat like corners or lines. It needs plenty patches to provide a dense cover of an object and perform effectively.

In order to use fewer patches but informative to represent the object appearance, we use mid-level parts to generate codebook. The parts are densely sampled from training images and the ones with low gradient are leaved out. For the object with multiple viewpoints or positions, the parts harvested would cover the different status.

We use clustering method to obtain initial part clusters. The k-means algorithm[9] is employed because of its computational simplicity. Naturally, the clusters are rough and impure due to this unsupervised clustering. So the training scheme is followed which uses the initial clusters to train models for every cluster. It aims at collecting patches purer and more consistent.

2.2 Training

Every training image is equally containing distinctive parts, so we sample densely to get all possible sub-windows and collect hundreds of thousands of parts. We use the augmented HOG feature with gradient orientations of dimensionality d = 31 as in [10]. The dimensionality of feature vector is $\mathrm{N}d$ for the part with N cells.

We cluster the collected patches with k-means in whitened histograms of orientations (WHO) space[7]. The feature vector x is transformed to $\hat{x} = \Sigma^{-1/2}(x - \mu_0)$ in which Σ ($\mathrm{N}d \times \mathrm{N}d$ matrix) and μ_0 ($\mathrm{N}d$ dimensinal) are covariance and mean according to all background features. Clustering with whitening feature can remove the correlations common in natural images and leave behind only discriminative gradients. Each cluster is the initial group for training by LDA. The LDA model is a linear classifier over x with weights given by $\omega = \Sigma^{-1}(x_{mean} - \mu_0)$ and x_{mean} is the mean feature of each cluster.

So as to improve the consistency of each model, we use the cross-validation training scheme. The training set is divided into two set as train-set and validation-set. Firstly clustering in train-set and training each cluster with LDA model to get the initial classifiers, and then using the classifiers to detect in validation-set to find the corresponding parts. Iterate the process until converge (the clusters stay the same) or the iteration comes to the set maximum. The clusters with small number of parts are eliminated since they occur rarely to characterize the appearance of object. Thus we obtain the part classifier for every cluster and these classifiers will serve as part detectors at runtime.

2.3 Codebook Generation

When the classifiers have been trained, we use them to detect in the whole positive training set to generate the codebook. A codebook is a vocabulary which collects the parts with the same local appearance of one part in an object and repeatedly occurs in training images. Parts of all codebooks can cover entirely the characteristic of the object.

We have obtain M detectors $\{D_m\}_{m=1}^{M}$, in which $D_m = (w_m, b_m)$ and w, b are the learned parameters with LDA model in the training procedure. Each detector finds a set of instances $\{P_n = (f_n, d_n, s_n)\}_{n=1}^{N}$ similar to it in positive training images with scores to reflect the similarity between them. Here, f is the feature vector, $d = (d_x, d_y)$ is the offset respect to object and s is the score of every entry with $s = w^T * f + b$ corresponding to a detector. Hence, one detector and one group of parts detected by it construct a codebook $C_m = (D_m, \{P_{m,n}\}_{n=1}^{N_m})$. The detectors have different abilities to discover parts in image so each part group has different number N_m. Fig. **2** shows an example of procedure of generating the codebook. The parts selected are discriminative structures reflect key properties of cars.

Mid-level parts are highly characteristic structures which are more complicate than the low-level patches. They repeat less frequently than patches with just lines or corners in low level. In one training image there are just a few parts that are very similar to the detector. So when collecting parts for codebook, every training image needs to store only several parts with high scores. Pool all parts found in the whole training set, sort them with scores and store a certain amount of parts with high scores.

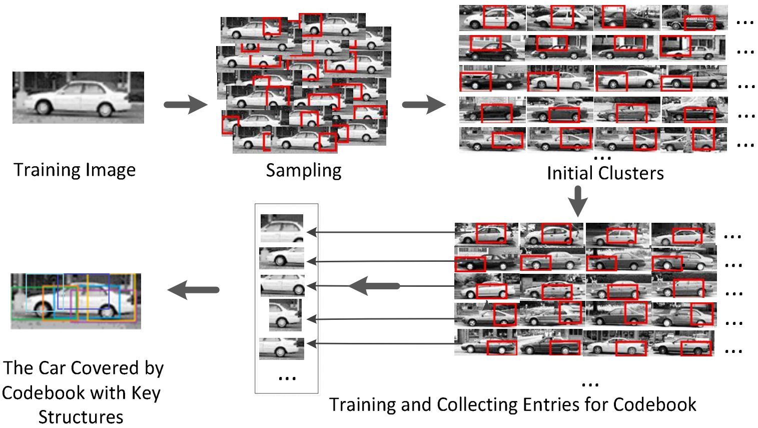
Training Image
Sampling
Initial Clusters
The Car Covered by Codebook with Key Structures
Training and Collecting Entries for Codebook

Fig. 2. The procedure of generating the codebook. Training images are densely sampled and clustered. Initial clusters are trained with LDA model. The codebooks store spatial distribution and scores of parts collected from whole training images by trained models.

3 Detection with Hough Voting

Codebooks are used to locate the boundary box of one class object by Hough transform in test image. Each entry in the codebook will vote with its score to predict the position respect to the position of one detected part by one classifier.

Given a test image, we apply the trained classifiers to find parts in every location. Apparently, parts with low scores do not match the model well and parts with the high scores are more confident to be valuable for voting. The advantage of mid-level parts is that it can use fewer numbers of patches to detect, so a small percentage of high score parts are kept.

Let f_m be the feature vector of one part extracted by detector D_m from location $\ell_m = (x_{0\mathrm{m}}, y_{0\mathrm{m}})$, we obtain a score $sc(D_m|f_m, \ell_m)$ and the learned spatial distribution $\mathrm{sp}(x, y|C_m)$ predicting the position (x, y) by codebook C_m. Then the votes from all parts and entries from codebooks are summed up to form an energy map, the scores of each point (x, y) in map can be expressed as:

$$\mathrm{e}(x, y|f_m, \ell_m) = sc(D_m|f_m, \ell_m)\, sp(x, y|C_m, \ell_m) \tag{1}$$

The first term is the score that the part detected by a detector which is independent to the location. The second term is the learned spatial distribution and there are n parts expressed as $\{P_{m,n}\}$ existing in codebook C_m. Thus the equation is written as:

$$\mathrm{e}\big(x, y\big|f_{m,}, \ell_m\big) = sc(D_m|f_m) sp(x, y|C_m, \ell_m) \tag{2}$$

$$sp(x, y|C_m, \ell_m) = \textstyle\sum_n sp\big(x, y\big|P_{m,n}, \ell_m\big) \tag{3}$$

However, in order to adapt to the shape deformation and be robust to structure variation, we use the Parzen-window estimate to obtain continuous voting space. Then the equation above is:

$$sp(x, y|C_m, \ell_m) = \textstyle\sum_n sp(x, y|P_{m,n}, \ell_m) * \frac{1}{2\pi\sigma_x^2\sigma_y^2} \exp\left(\frac{\left(x - x_0 - d_{x_n}\right)^2}{-2h\sigma_x{}^2} + \frac{(y - y_0 - d_{y_n})^2}{-2h\sigma_y{}^2}\right) \tag{4}$$

Here, (σ_x, σ_y) is the variance of the predicting position (d_x, d_y) of all entries saved in one codebook. The parameter h is relative to voting contribution of every point. The smaller it is, the clearer is to observe the every voting contribution. In order to reduce the fluctuation of voting points, we set $h = 1$ to obtain a more smooth distribution.

Each detector discovers a set of parts as $\{f_{m,i}\}_{i=1}^{m_i}$, then all parts from the complete detectors will form the final energy map as:

$$\mathrm{E}(x, y) = \textstyle\sum_m \sum_i \mathrm{e}(x, y|f_{m,i}, \ell_{m,i}) \tag{5}$$

Fig. 3 presents the procedure of detection using codebook and Hough transform. Parts are firstly discovered by LDA model (Fig.3-A), and then the entries in codebook corresponding to each part cast votes to the location of object center (Fig.3-B). All the votes are summed up to form an energy map reflecting the hypothesis (Fig.3-C), and the objects are located in peaks of map (Fig.3-D). There may be multiple overlapping detections on the same object, so we use non-maximum suppression to keep one hypothesis for every instance.

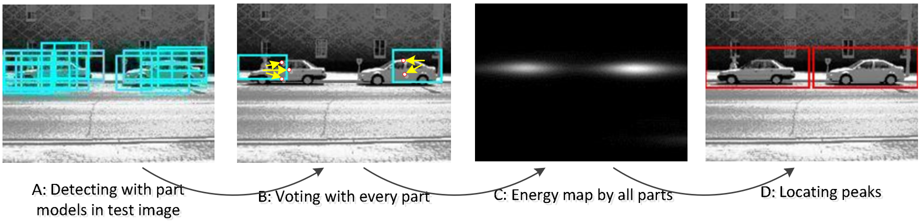
A: Detecting with part models in test image
B: Voting with every part
C: Energy map by all parts
D: Locating peaks

Fig. 3. The example of detection procedure

4 Experiments

In order to evaluate the performance of the Hough voting with mid-level parts, we apply the method on open dataset UIUC Cars and compare the result with the similar approaches.

The UIUC Cars single-scale test set contains 170 images with 200 side views of cars of approximately the same scale. The images are low contrast with some cars partially occluded and multiple objects would occur. The training set contains 550 training cars of size 100×40 and 500 negative training examples of the same size.

We adhered to the experimental evaluation criteria based on bounding box overlap as previous works. The hypothesis with center (x, y) will be accepted if it is in the ellipse of the annotation center coordinate (x_0, y_0) with size (w, h) that:

$$\frac{|x-x_0|^2}{(0.25w)^2} + \frac{|y-y_0|^2}{(0.25h)^2} \leq 1 \tag{6}$$

We accept one hypothesis as correct detection for every instance and treat others as false positive.

In the experiment, each part extracts a 1116D HOG feature vector by concatenating the 31D vector of 6×6 cells. We have trained 53 detectors using 6 scales parts with size of 26×26, 24×48, 24×96, 32×48, 32×96, 35×35. The car can be covered entirely by detected parts as shown in Fig. 2. Applying the trained detectors to discover parts from test image and forming an energy map of voting. The peaks in the map are hypothesis of object.

Table 1. Comparison of our results on the UIUC-Single car database with other methods

Methods	ISM. No MDL	ISM +MDL	Hough Forest	HF Weaker supervision	Our approach
PR-EER	91%	97.5%	98.5%	94.4%	99.5%

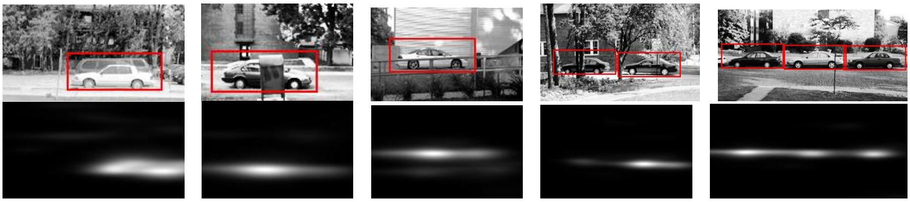

Fig. 4. Examples of detection results. The top row is the location of peaks in energy map of the down row. Objects with some occlusions or multiple instances are detected correctly.

Table 1 shows the result comparison of several similar Hough voting algorithms with recall-precision equal error rate (EER). Our method achieves an impressive 99.5% EER (corresponding to 200 out of 202 detections with 2 false positives) for UIUC-Single database. Fig. 4 is examples of some detection results.

The methods of ISM with MDL verification and Hough Forest (HF) have comparable performances with ours. However, the former one needs segment annotation additionally which is consuming labor; the accuracy of HF interrelated to the sample density so that it needs a large number of patches. Our codebooks generated using trained models are more accurate than clustered only in ISM. What's more, they are discriminative and more informative than patches in HF so that fewer parts are needed to vote for predicting.

As a method based on parts, the size of parts have an impact to the detection performance. We analyze the impact of different scale part and the combination of multiple parts. Result is shown in Fig. 5. The case of two scales has parts with size 24×48 and 32×48, the case of three scales has parts with size 24×48, 32×48 and 35×35, the case of six scales has all scales as presented above.

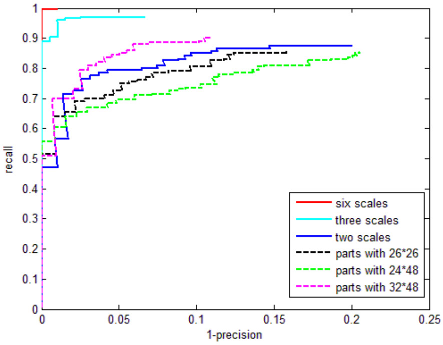

Fig. 5. The impact of different scales of parts to the recall-precision equal error rate (EER) of our method on UIUC-Single dataset

We can infer that the codebook with parts too small or too large will both not perform well. A small part contains less distinctive structures so that the smaller a part is, the more similar parts will be discovered by detectors and more noise interference will be produced from negative objects. On the other side, codebook with large size parts would obtain fewer but more informative parts. However, if the size is too large, then very few parts will be found in image so that the number cannot support a reliable vote.

There is a tradeoff between the distinctive and the number of parts according to the size. Codebook of parts with 32×48 would generate more informative patches to vote while codebook of parts with 26×26 would generate more number patches to vote, so both of them perform better than codebook of parts with 24×48. But the results of all of them are not quite well though.

A solve of this is to use multiple scales to compensate the impact of the two factors. We can observe from Fig. 5 that the performance by codebook of parts with six scales is better than codebook of parts with three scales, and much better than codebook of parts with fewer scales.

5 Conclusion

We proposed a Hough voting method based on codebook of distinctive mid-level parts for object detection. The codebook is generated via LDA training after clustering with kmeans in WHO space. This scheme helps to gain more accurate and consistent structure entries for every codebook. The energy map is formed by voting from the codebook. Experiment results show the effectiveness of our method compared to several similar algorithms.

Acknowledgement . This research is supported by the Natural Science Foundation of China (No.61105014, No.61302137, No.61401170), and the Natural Science Foundation of Hubei Province (2013CFB403).

References

1. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-SVMs for object detection and beyond. In: IEEE International Conference on Computer Vision, pp. 89–96 (2011)
2. Maji, S., Shakhnarovich, G.: Part discovery from partial correspondence. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 931–938 (2013)
3. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. International Journal of Computer Vision 77(1-3), 259–289 (2008)
4. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1022–1029 (2009)
5. Gall, J., Yao, A., Razavi, N., Van Gool, L.: Hough forests for object detection, tracking, and action recognitions. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(11), 2188–2202 (2011)
6. Singh, S., Gupta, A., Efros, A.A.: Unsupervised Discovery of Mid-Level Discriminative Patches. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 73–86. Springer, Heidelberg (2012)
7. Hariharan, B., Malik, J., Ramanan, D.: Discriminative Decorrelation for clustering and classification. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 459–472. Springer, Heidelberg (2012)

8. Juneja, M., Vedaldi, A., Jawahar, C.V., Zisserman, A.: Blocks that shout: Distinctive parts for scene classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 923–930 (2013)
9. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
10. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1627–1645 (2010)
11. Yao, C., Bai, X., Liu, W., Latecki, L.J.: Human Detection using Learned Part Alphabet and Pose Dictionary. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 251–266. Springer, Heidelberg (2014)
12. Wang, X.G., Wang, B.Y., Bai, X., Liu, W.Y., Tu, Z.W.: Max-Margin Multiple Instance Dictionary Learning. In: Proceedings of the 30th International Conference on Machine Learning, pp. 846–854 (2013)
13. Dollár, P., Babenko, B., Belongie, S., Perona, P., Tu, Z.: Multiple component learning for object detection. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 211–224. Springer, Heidelberg (2008)