Proceedings of the 2009 IEEE
International Conference on Information and Automation
June 22 -25, 2009, Zhuhai/Macau, China

Learning Model for Object Detection Based on Local Edge Features

Tang Xusheng, Shi Zhelin, Li Deqiang, Ma Long, and Chen Dan

***Abstract*—We present a learning model for object detection that uses a novel local edge features. The novel features are motivated by the scheme that use the chamfer distance as a shape comparison measure. The features can be calculated very quickly using a look-up table. Adaboost algorithm is used to select a discriminative edge features set from an over-complete local edge features pool and combine them to form an object detector. To demonstrate our method we trained a system to detect car in complex natural scenes using a single shape model. Experimental results show that our system can extremely rapidly detect objects in varying conditions (translation, scaling, occlusion and illumination) with high detection rate. The results are very competitive with other published object detection schemes. The learning techniques can be extended to detect other objects such as airplanes or pedestrian.**

I. Introduction

AUTOMATIC target recognition is still a challenge in computer vision and pattern analysis research. The main difficulty in developing a reliable object detection approach arises from the wide range of variations in images of objects belonging to the same object class even the same object. In addition, various changes in images such as translation, rotation, scaling, occlusion, and illumination, make the process more difficult.

Psychological studies have indicated that line drawings of objects can be recognized as quickly and almost as accurately as photographs [1]. The advantage of using edges as image features is that they can provide robustness to illumination change and sensors change. Furthermore, matching techniques have been developed for edge maps that can handle occlusion, image noise and clutter.

Edge matching schemes such as chamfer matching and Hausdorff matching have proven to be a reliable and efficient method for object recognition [2, 3]. In this context, shapes are sets of points obtained from images using edge detection. Typically, an object is represented in terms of an ideal shape, which is referred to as the model. A new shape is classified as an instance of the object if the generalized distance between the model and the new shape is small.

Most these recognition systems simply use an instance of the object as the model. In that case, it has two major limitations: (1) the performance degrades rapidly with geometric image distortions and their susceptibility to background clutter; and (2) it might be necessary to manually edit the model to correct for noise and remove background clutter. Moreover, in some cases it is not clear what would be a good model for a particular object.

Many recent research works are concentrating on learning object edge models from training examples [4-9]. Our learning model is similar to those of [8, 9] and is briefly reviewed in section 2.

In this paper, we present a learning mode for object detection in images based on local groupings of edge fragments representation. The most discriminant edge fragments features are acquired automatically from training examples using improved chamfer matching as similarity measure. A classifier is then trained, using machine learning techniques, to distinguish between object and non-object images based on this representation. The role of training is to select special features, which are moderately likely at certain places on the object class but rare in the background clutter across the training set. We demonstrate our method by training a system to detect lateral cars in images using a single shape model. As shown in our experiments, the resulting algorithm is able to accurately detect objects in complex natural scenes.

The remaining part of the paper is organized as follows: section 2 contains description of related work; section 3 provides an overview of our approach. Experiments and analysis are conducted in section 4, followed by conclusion in section 5.

II. Related Work

Borgefors[2] proposed a hierarchical chamfer matching algorithm to improve match efficiency, in which a coarse-to-fine search is performed using a resolution pyramid of the image.

Most work on chamfer-based matching has dealt with the case of matching one template against an image. When using a single template, chamfer matching cannot handle large shape variations. Gavrila[4,5] introduced template hierarchy to capture the variety of object shapes. They simply store a large number of positive templates in memory. A new shape is considered an instance of the target object if it is similar to any of the stored shapes. The number of templates needed increases with object complexity.

Felzenswalb[6] show that using examples-based learning techniques one can build a single shape model, which captures the information from all examples. Furthermore, the performance of detection system has been improved due to

Manuscript received January 15, 2009. This work was supported in part by the National Natural Science Foundation of China under Grant 60603097.

Tang xusheng, Shi Zhelin, Li Deqiang, Ma Long, and Chen Dan are all with Shenyang Institution of Automation, the Chinese Academy of Sciences,Shenyang,110016,China (corresponding author to provide e-mail: tribology@ 163.com).

978-1-4244-3608-8/09/$25.00 ©

using the negative examples. The hausdorff distance is used as feature in that paper.

Following his work, the local boundary fragment model is adopted as representations of object shape and chamfer distance is used as feature similarity measure in [7, 8, 9]. The detector is learned with examples-based learning algorithm. They present results that are very competitive with other state-of-art object detection schemes. Our learning model is mainly inspired by these works [8, 9]. The main differences are: the level of segmentation required in training (they all require, we don't need); the different computation of feature matching score (our novel features can be computed rapidly at all scales in constant time using the integral image technique), which leads to our detection speed is much faster than theirs.

III. Approach

In our learning method, models or other parameterizations need not to be established explicitly. The model is estimated from the training examples. When a good model exists, we are guaranteed to find one that provides (with high probability) a recognition rule that is accurate. The intuitive motivation behind our method is that images of different objects within a class have a particular structural similarity - they can be expressed as combinations of common substructures. The training and detection process scheme for our system can be summarized as shown in Fig. 1(DT in the figure denotes distance transform).

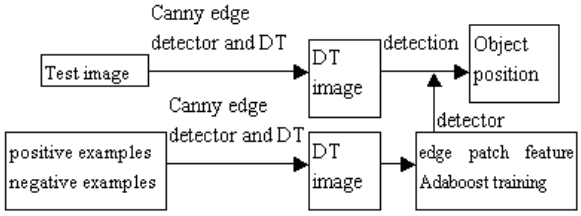

Fig. 1. Outline of training and detection of our approach

A. Chamfer Matching

First, all the edge points of image are detected by Canny edge detection algorithm. Second, a distance transformation [10] is employed on binary edge images to obtain distance images (DT image). L1 distance is used in this paper. Suppose two sets of edges for model and image are represented by T={t} and E={e}.

The template T is translated and positioned over the DT image of E. The chamfer matching score is a function of relative position x:

$$D_{cham}^{(T,E)}(x) = \frac{1}{|T|}\sum_{t \in T} d_E(t+x). \quad (1)$$

where $|T|$ denotes the number of features in T and d_E denotes the distance between feature in T and the closest feature in E.

The chamfer score map forms a distribution of distances of the template features to the nearest features in the image. The lower score is, the better the match between image and template at this location.

In applications, to alleviate problems arising from missing edges in E, a template is considered matched at locations where the distance measure is below a user-supplied threshold θ

$$D_{charm}^{(T;E)}(x) \le \theta \quad (2)$$

B. Edge Patch Feature Based on Chamfer Distance

One theory of biological vision explains object detection on the basis of decomposition of objects into constituent parts. According to this theory, the image-patch (or part-based) feature representation was used in this paper.

We suppose the features of model could be at any position in the model raw image. Our novel raw feature set is then the set of all possible features of the form, where the position, the aspect ratio of the patch and the number of feature are arbitrarily chosen. This raw feature set is (almost) infinitely large. For practical reasons, it is reduced through simply limited the aspect ratio of edge patch (3, 4,5and 6 was selected in this paper). Of course, one can extended by adopting other aspect ratio, which may be add additional domain-knowledge to the learning framework and which is otherwise hard to learn. Examples of edge patch feature for the class of lateral cars are illustrated in Figure 2. Suppose a rectangle r=(x, y, w, h) denotes an edge patch feature. According to (1), each edge patch feature value for the image I can be calculated as follow:

$$f_r = \frac{1}{w*h}\sum_{t \in r} d_I'(t) \quad (3)$$

where $d_I'(t)$ denotes the value in the DT image of raw image I.

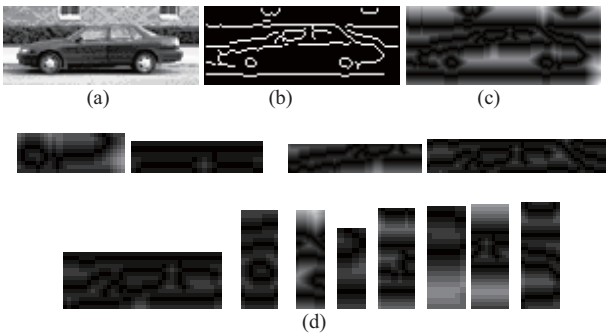

Fig. 2: (a) A sample object image.(b) edge image by Canny detector.(c) DT image of (b).(d) Examples of edge patch feature

Our features can be computed at any position and any scale in the same constant time using integral image [12]. Only 4 table lookups are needed per feature.

Our experiments are performed on images of side views of cars. The training examples of cars are 100*40 pixels in size in the experiments. There are 218,784 edge patch features for a 100*40 detection window size(3, 4,5and 6 of aspect ratio of edge patch were selected in this paper).

C. Learning Method

We train our classifier from a set of examples using AdaBoost algorithm (see figure 3) [14]. It selects a set of "weak" classifiers and combines them into a powerful strong classifier. For each feature, the weak learner determines the optimal threshold classification function, such that the minimum numbers of examples are misclassified. According to Eq.(2)and(3), the weak classifier can be defined as follows:

$$h_j(I) = \begin{cases} 1 & p_j f_j \le p_j \theta_j \\ 0 & otherwise \end{cases} \quad (4)$$

Here a polarity p_j indicating the direction of the inequality sign, i denotes a 100*40 pixel sub-window of an DT image, j denotes an edge patch feature.

The role of training is: feature selection, parameter estimation, and learning a classifier.

- Given the training examples $\{(x_1^p, y_1^p), \cdots, (x_m^p, y_m^p), (x_1^n, y_1^n), \cdots, (x_l^n, y_l^n)\}$ where x_i^p and x_i^n denote the DT image of the ith example, $y_i^p = 1$ and $y_i^n = 0$, the superscript p and n denote the positive and negative samples respectively.
- Start with $w_i = 1/(m+l), i = 1, \ldots, (m+l)$.
- Repeat for t=1,…,T
  a) For each feature, j, train a weak classifier h_j which is restricted to using a single feature. The error $err_j = \sum_{i=1}^{m+l} w_i |y_i - h_j(x_i)|$. Choose the classifier, h_t, with the lowest error err_t.
  b) Update the weights: $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$ where $e_i = 1$ if example x_i is classified correctly, $e_i = 0$ otherwise, and $\beta_t = \frac{err_t}{1 - err_t}$.
  c) Renormalize weights so that $\sum_i w_i = 1$.
- The final strong classifier is $H(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \ge \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}$ where $\alpha_t = \log \frac{1}{\beta_t}$

Figure 3: The Adaboost algorithm for classifier learning. Each round of boosting selects one edge patch feature from candidate features.

D. Detection Hypothesis Using the Learned Classifier

Having learned a classifier, object can be detected in an image by applying the learned classifier to all the potential windows of the image. Objects in images may appear at different scales depending on the depth of the object with respect to the camera. Therefore it is important to search the image at different scales in order to detect all objects. To keep the number of features of the classifier fixed, we iteratively scale the image by a specified amount.

Another important issue to resolve is the multiple detections that may occur for the same object in the scene due to the invariance of the classifier to small translations and scales of an object. Therefore, in using a classifier to detect, it is necessary to have another processing step. A simple strategy is: detected windows are partitioned into disjoint (non-overlapping) groups, and each group gives a single detection located at the centric of the corresponding original detections. A similar method has been used in [11, 12].

IV. Experiment

We applied our learning techniques to build a system that can detect car on the street scenes.

A. Database

In order to allow comparisons with other published methods, we have chosen to work on publicly available databases: the “lateral cars” by UIUC [13]. The database contains training set and test set. The training set contains 550 positive examples, all of size 100x40 pixels. We used the positive examples unmodified. In order to select the effective negative examples for detection, the bootstrapping method[11] is used for selecting negative samples, that is, the false accept samples are later served as negative samples during the training process. The test set contains two test sets. Test set I consists of 170 images containing 200 cars; the cars in this set are all roughly the same size as in the training images. Test set II consists of 108 images containing 139 cars; the cars in this set are of different sizes, ranging from roughly 0.8 to 2 times the size of cars in the training images. They are of different resolutions and include instances of partially occluded cars, cars that have low contrast with the background, and images with highly textured backgrounds.

B. Evaluation Criteria

Here, we strictly followed the evaluation criteria described in [13] to evaluate our approach[1]. The evaluation criteria are introduced briefly as follow:

(1) *Recall-precision curve*, where

$$Recall = \frac{TP}{nP}, Precision = \frac{TP}{TP + FP} \quad (5)$$

where *TP*,*FP*,*nP*, denotes number of true positives, number of false positives and total number of positives in data set, respectively.

[1] Both the data sets we have used and the evaluation routines are available from http://l2r.cs.uiuc.edu/~cogcomp/Data/Car/.

(2) *F_measure*, where

$$F\_measure = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \quad (6)$$

The *F_measure* summarizes the trade-off between recall and precision, giving equal importance to both.

C. Evaluation Results

Our car detector is finally built with 802 features. We applied our car detector to test set I and set II. To reduce computational costs, the sub-window was moved in step of 3 pixels and 2 pixels in the horizontal and vertical directions, respectively. For test set II, the images were scaled to sizes ranging from 0.48 to 1.2 times the original size, each scale differing from the next by a factor 1.2; a 100*40 sub-window was the moved over each scaled image.

Fig.4 shows our recall-precision results on test set I together with other published results [8,13,15]. The performance of system on test set I are also compared. The results are list in Table 1.

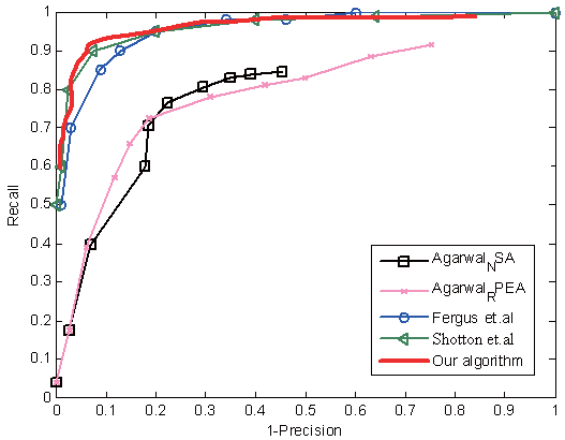

Fig. 4: Comparison Recall-precision results to other method [8,13,15] on single-scale Test Set I

TABLE 1: COMPARISON PERFORMANCE TO OTHER METHOD ON SINGLE-SCALE TEST SET I

Ref.	Best F-measure(%)	Average time (s/frame)	Hardware of system
[13]	~77.1	2.5	Two Sun Ultra SPARC-II 296MHz
[15]	88.5	-	-
[8]	92.8	10	3GHz Pentium4
Our	**92.5**	**0.02**	**1.86GHz Core2**

The detection rate of our algorithm is very competitive with other published results. From Table 1 we can see the speed of our system is much faster than and the detection rate is equivalent to the best published results [8]. The reason for this is twofold. Firstly, to evaluate the oriented chamfer distance that adopted in [8] spend much time. To evaluate our improved chamfer distance are only needed 4 table lookups using integral image. Secondly, the detector in [8] was run over each image twice since the test images had cars present facing left and right. Our detector is insensitive to the car present facing direction.

Due to the detection speed the algorithms in[8] can not deal with in more challenging multi-scale environments. We only compare our result with results of the method presented in [13] on multi-scale test set II. Fig.5 shows the results. The performances on test set II are also compared and the results are list in Table 2. The multi-scale results in [13] are considerably poorer than ours. We also see that the highest F-measure drops from 92.5% in the single-scale case to 84.7% in the multi-scale case. This certainly leaves much room for improvement in approaches for multi-scale detection.

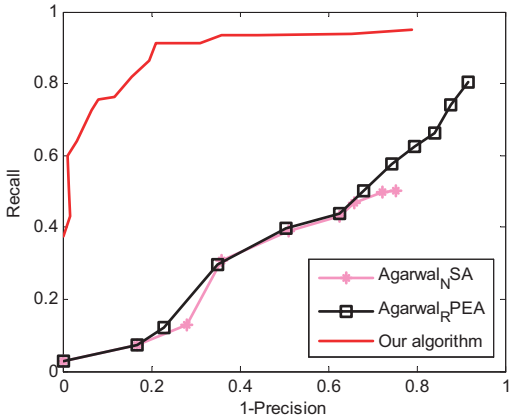

Fig. 5: Comparison Recall-precision results to other method[13] on Multi-scale Test Set II

TABLE 2: COMPARISON PERFORMANCE TO OTHER METHOD ON MULTI-SCALE TEST SET II, CONTAINING 139 CARS

Ref.	Best F-measure(%)	Average time (s/frame)
[13]	44	-
Our	**84.7**	**0.4**

Fig.6 shows the output of our detector on some sample test images.

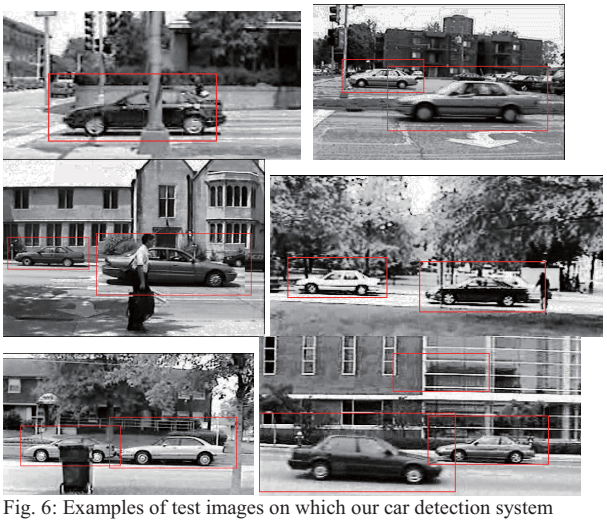

Fig. 6: Examples of test images on which our car detection system achieved perfect detection results.

V. Conclusion

We have presented a learning model for object detection based on object edge shape. Expressive features are acquired automatically and capture information about the parts in an image and the spatial relations among them. An efficient learning algorithm over this feature space then learns a good classifier from a training set.

We have shown that our method works successfully on a test set of images containing side views of cars. Our algorithm achieves high detection rates on real-world images with a high degree of clutter and occlusion, while it is capable of detecting target extremely fast (20ms per image on 1.86GHz Core2). Our framework is easily extensible to other objects that have distinguishable two-dimensional edge shape such as airplanes or pedestrian.

References

[1] I.Biederman, J.Gu., "Surface versus edge-based determinants of visual recognition," Cognitive Psychology, vol.20, pp.38-64, 1998.

[2] G.Borgefors, "Hierarchical chamfer matching: a parametric edge matching algorithm," IEEE Trans. On Patten Analysis and Machine Intelligence, vol. 10, pp. 849-865, Nov. 1988.

[3] D.P. Huttenlocher, G. A. Klanderman, and W. J. Ruklidge, "Comparing images using the Hausdorff distance," IEEE Trans. On Patten Analysis and Machine Intelligence, vol.15, pp.850–863, Sep. 1993.

[4] D. M. Gavrila. "Pedestrian detection from a moving vehicle," in Proc. 6th European Conf. on Computer Vision, Dublin, Ireland, 2000, pp. 37–49.

[5] D.M. Gavrila, "A Bayesian,Exemplar-Based Approach to Hierarchical Shape Matching," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.29, pp.1-14 , Aug. 2007.

[6] P.F.Felzenszwalb, "Learning Models for Object Recognition," In Proc. IEEE Conf.on Computer Vision and Pattern Recognition, Kauai, HI, United states, 2001, pp.1056-1062.

[7] E.Seemann,B.Leibe,and K.Mikolajczyk.etc., "An evaluation of local shape-based features for pedestrian detection," In Proc.BMVC,2005

[8] J.Shotton,A.Blake,and R.Cipolla, "Contour-based learning for Object Detection," In Proc. ICCV, Beijing, China, 2005, pp.503-510.

[9] A.Opelt,A.Pinz,and A.Zisserman, "A Boundary-Fragment-Model for Object Detection," In Proc. 9th European Conference on Computer Vision, Graz, Austria ,2006, pp. 575-588.

[10] P.F.Felzenszwalb, D.P. Huttenlocher, "Distance transforms of sampled functions," Technical report, Cornell, 2004.

[11] H. A. Rowley, S. Baluja, and T. Kanade. "Neural network-based face detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, pp. 23-38, Jan. 1998.

[12] P. Viola, M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Kauai , Hawai, USA, 2001, pp 511-518.

[13] S. Agarwal, A.Awan,and D.Roth, "Learning to Detect Objects in Images Via a Sparse, Part-Based Representation," IEEE Tran. On Patten Analysis and Machine Intelligence, vol.26, pp.1475-1490, Nov. 2004.

[14] Y.Freund, R.Schapire, "Experiments with a new boosting algorithm," In Proc. Of the Thirteenth Int. Conf. on Machine Learning, Bari, Italy, 1996, pp.148-156.

[15] R.Fergus,P.Perona,and A.Zisserman, "Object class recognition by unsupervised scale-invariant learning," In Proc. IEEE Conf.on Computer Vision and Pattern Recognition, Madison, WI, USA, 2003,pp. 264-271.