

Local-feature Based Vehicle Recognition System Using Parallel Vision Board

YOSHIDA Tatsuya, KAGESAWA Masataka, IKEUCHI Katsushi

Institute of Industrial Science, University of Tokyo

4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, Japan

phone:+81-3-5452-6242, fax:+81-3-5452-6244

E-mail:kagesawa@iis.u-tokyo.ac.jp, ki@iis.u-tokyo.ac.jp

ABSTRACT

This paper describes a robust method for recognizing vehicles. Our system is based on local-feature configuration, and we have already shown that it works very well in infrared images. The algorithm is based on our previous work, which is a generalization of the eigen-window method. This method has the following three advantages: (1) it can detect even if part of vehicles is occluded. (2) it can detect even if vehicles are translated due to running out of the lanes. (3) it does not require us to segment vehicle areas from input images. It is true that we have first developed our system with infrared images, but it is not essential for our system to employ infrared images. In this paper, applying our system on images of super wide-angle, we have shown that our system is effective to optical images, performing two outdoor experiments. Our system is good at detecting locations of vehicles, hence it will be useful for not only vehicle detection but also such application, ETC, DSRC or so, that system needs to know with which vehicle it communicates.

Keywords: Vehicle recognition, Image processing, Eigen-space, Eigen-window, Parallel image processor

INTRODUCTION

The main purpose of vehicle detection was to measure the number of vehicles at each sensing point for flow estimation and prediction. As a result, point-oriented sensors, i.e., ultrasonic sensors or loop detectors, have been often used. Recently, image-processing sensors have become practically available in ITS applications. Not only can those sensors measure the number of vehicles but they also can measure velocity; they also have the potential to detect traffic accidents.

We have already proposed an algorithm to recognize vehicles based on local features. The algorithm can solve one of the common problems in image-based vehicle recognition systems: it can recognize vehicles partially occluded by other vehicles or obstacles. We have shown the effectiveness of our system in infrared images, but in this paper, we describe the performance of our system in usual optical images of super wide-angle. Since our system is view-based and views of an object in super wide-angle images are different each other if locations of the object are different, it is very good at detecting location of the object.

We have implemented vector quantization method on IMAP-vision board. In order to explain the algorithm, we need to start with the eigen-window method.

RECONITION ALGORITHM

Our basic algorithm is a generalization of the eigen-window method, originally developed by Ohba and Ikeuchi^{(2),(3),(4)}.

Eigen-window technique

The following flow shows how the eigen-window system works:

1. Make the database set in advance.

(a) Make the set of training images.

(b) Extract local feature points from each training image.

(c) Compress the set of feature points using eigen-space technique

(d) Reduce the number of local features according to the criteria of uniqueness and reliability.

(e) Make the database set which consists of pairs of a compressed local feature and the location of the feature point in the training image.

2. Compare input images with the database set.

(a) Extract local feature points from input image.

(b) Find the closest feature points in the database for each local feature point.

(c) Make such a vote that two feature points in the input image are voted to the same point if, and only if, their relative position is the same in both the training image and the input image.

(d) Detect according to the result of the vote.

The original algorithm is based on the eigen-space method⁽¹⁾, in which we calculate eigen values of a covariant matrix. The eigen-window method uses small windows as features for object recognition. Due to this

window method, the algorithm can handle images that contain partially occluded objects.

We here explain the essential parts; eigen-space technique, local feature selection part and voting operation part.

Eigen-space technique

Let $\{z_i\}$ be M images of size $N=m \times n$. Each z_i can be regarded as a N -dimensional vector. We can assume that the sum of all z_i is equal to zero because if not we just replace z_i into $z_i - c$, where c is the average of $\{z_i\}$. Consider the covariant matrix Q of $N \times N$:

$$Q = [z_1, \dots, z_M][z_1, \dots, z_M]^t$$

Then Q can be written with some orthogonal matrix T as follows:

$Q = T X T^t$, where X is a diagonal matrix whose components are non-negative. That is to say, (i,j) -component of X , X_{ij} , is equal to 0 if i not equal to j , is equal to e_i , one of the eigen-values of Q , if i equals to j . The matrix T corresponds to an orthogonal transformation in N dimensional vector space, and in the coordinate transformed by T , Q can be written as a diagonal matrix. It is known that the first several eigen-values of Q are large, but others are very small. This fact shows that in this coordinate, the first k coordinates are important. If we replace T into $T' = P_k T$ of $k \times N$ matrix, where P_k denotes the projection matrix into the first k coordinates, and consider $\{T' z_i\}$ instead of $\{z_i\}$, then the covariant matrix of $\{T' z_i\}$ is almost the same as Q , but $T' z_i$ are regarded as k -dimensional vectors. Hence when we need to find the closest image z_i for a given input image z , we need to find the closest image $T' z_i$ for $T' z$. Usually k is about 5-10, whereas N is large (N will be 65536 if the dimension of an image is 256x256), thus this technique makes matching very fast.

Local feature selection

In the eigen-window system, the local features are selected according to three criteria of detectability, uniqueness and reliability.

Detectability is the criterion that each selected window is actually characteristic. We apply a corner detector for each image to satisfy the detectability criterion.

Uniqueness is the one that each selected window is unique among all characteristic windows. In other words, we remove such characteristic window w that there is a characteristic window w' which is similar to w .

Reliability is the one that each window is stable with slight movement of the object. We say that a window w is stable with slight movement of the object if the following is hold:

Let w be a window of an object O in pose X , and w' be any window of the object O in any pose X' such that the poses X and X' are almost the same and the window w' of O in pose X correspond We now have the database of

characteristic windows $\{(w_i, T_i; x_i, y_i)\}$, where w_i is the image of the window, T_i is the number of the training image to which w belongs, and (x_i, y_i) is the location of w in the image of number T_i . Let $(w; x, y)$ be a window which is located at (x, y) in an input image J . The base space of our voting system is $Z \times R \times R$, where Z corresponds to the number of training images and $R \times R$ corresponds to off-sets in a training image. If w is similar to w_i , then we put a vote onto $(T_i; x - x_i, y - y_i)$.

s to w . Then w is similar to w' .

We again remove all windows that is not stable. In this way, we obtain a reduced set of local features.

Voting operation

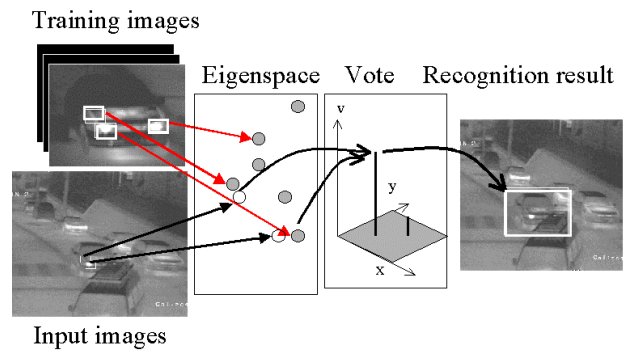


Figure 1: Eigen-window technique

For all windows in J , we repeat this vote. If the number of votes on a point (I, x, y) is r , it means that there are r local features in a training image I such that their relative position in the training image is the same as that in the input image. Hence, for each point in the base space (I, x, y) on which the number of vote is large enough, our system tells that there is an object in the image I with the off-set (x, y) . Note that the off-set $(0, 0)$ means that the location in the input image is the same as that in the training image.

Thanks to this voting operation, our system has the following properties:

1. It might recognize occluded objects when there is a large enough number of their local-features which are not occluded in input images.
2. It can recognize all objects in input images.
3. It can detect the objects even if the location in an input image is different from that of the training image.
4. It does not require us to segment the vehicle area from the input images.

Vector Quantization Algorithm

The eigen-window method is very good at recognizing objects, but it takes much time for recognition. Actually it takes 30 seconds or more for processing single image on a Sun workstation SS-20. In order to make our system fast, we decided to implement our algorithm on a hardware parallel vision board, which is called IMAP vision board. But it is not very easy to implement the

method on a hardware vision board in general, because hardware boards do not support floating point calculation, which is required to obtain eigen-values and vectors of the covariant matrix. Hence we have implemented another algorithm on the IMAP board. The new algorithm is called vector quantization algorithm(3). Its essential part is the same as that of eigen-window system, but it is customized for fast integer calculation as follows:

1. We apply an edge detector to obtain binary images from both input and training images.
2. Feature points are not “detectable points” but stable points, whose hamming distance between the blocks in their neighbourhood is small.
3. We use code features instead of eigen-vectors: a code feature is the centre of a group which consists of similar feature points. In other words, we create a set of some average local features which plays a role of eigen-vectors in eigen-window method.

The step 1 makes our system faster. In the step 2 and 3, we need only integer calculation, thus we can implement this algorithm on hardware image processing board.

Actual implementation is as follows:

1. Make the database set in advance.
 - (a) Make the set of training images: All training images are binarized with an edge detector.
 - (b) Extract stable local feature points from each training image: A window w of $(2n+1) \times (2n+1)$ centred in (x,y) is called a stable local feature point if the following value is small:

$$\max\{H(D(x,y), D(x+i, y+j)); -n \leq i, j \leq n\},$$
 where $D(x,y)$ is the binary vector correspond to the window of $(2n+1) \times (2n+1)$ centred in (x,y) and H denotes the Hamming distance between two binary vectors.
 - (c) Compress the set of feature points into code features: For given n , the number of code features, we divide the whole space of binary vectors into n segments so that each segment includes the same number of local feature points. Taking the center of all local feature points in a segment, as binary vector, we obtain the code feature in the segment.
 - (d) Make the database set which consists of pairs of a compressed corresponding code feature and the location of the feature point in the training image.

2. Compare input images with the database set.
 - (a) binarize each input image with an edge detector.
 - (b) Extract local feature points from the binarized image.
 - (b) Find the closest code feature in the database for each local feature point.
 - (c) Make such a vote that two feature points in the input image are voted to the same point if, and only if, their

relative position is the same in both the training image and the input image.

- (d) Detect according to the result of the vote.

VEHICLE RECOGNITION EXPERIMENTS

We have already confirmed that our system works very well in infrared images, in which we can obtain a lot of local features in day and night. But we know that our system works well whenever we have enough number of local features, even in optical images.

We believe that one advantage of vision system is its wide area of sensing. In order to maximize the merit, we adopted a super wide-angle camera mounted on road. The figure 2 is a sample image with this camera.

We have performed two kinds of outdoor experiment.

The first one is to confirm the robustness of our system against occlusion. Two vehicles are modeled in advance, and recongize the vehicles in such bad environment that



Figure 2: Wide-angle images in our experiments

their shadow is dropped on road and a bicycle gets through between the vehicles. In traditional vehicle recognition systems, it is hard to recongize two vehicles, but our system has succeeded. The second one is to confirm that our system can track vehicles with multiple training images. We have prepared nine training images on various locations in advance, and we have experimented if our system can recognize where vehicles are. In both experiments, our system has worked well.

Vehicles with obstructions

In this experiment, we have prepared two vehicles, of black and white. We at first made models of two vehicles in vector quantization. We have picked up about 55 local features from each training image, and packed them into 30 code features. For detecting the model 1, black vehicle, we put threshold to 8 votes. For the model 2, white vehicle, we put 11. The reason why we modified the threshold is that some of local features in black vehicle are unstable, thus, the number of effective local features in black vehicle seemed to be decreased. The figure 3 shows the code features selected and their parts in training images.

We have processed 575 images in some of which there were no vehicles. There were no false alarms: our system didn't recognize vehicle in the images without

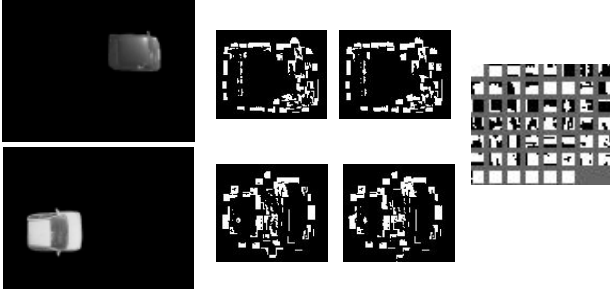


Figure 3: Models of two vehicles (Training images, local features, code features, set of code features)

vehicles. The recognition ratio for black vehicle was about 80%, and that for white vehicle was 100%. The result of this experiment is shown in the table 1. One reason why recognition ratio for black vehicle is worse is

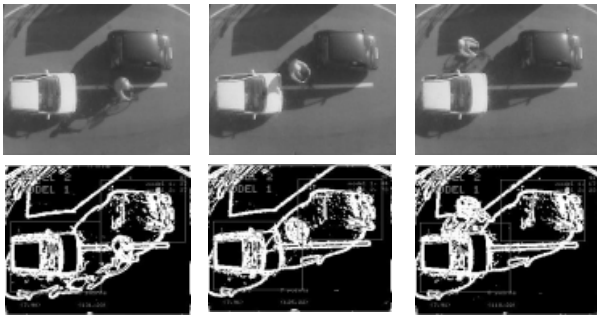


Figure 4: Some recognition results

that black vehicle does not have enough number of stable local features. Another reason is that the shadow of black vehicle occludes some of its edges. In other words, from the view of image processing, black vehicle is occluded partially in this experiment. But our system has succeeded recognizing both vehicles. If we would have better devices, such as cameras with high dynamic range or so, then we would obtain more local features clearly and our system would be able to detect black vehicles better. The figure 4 shows some of recognition results on our system

| | Recognized | Failed | Ratio |
|----------------|------------|--------|-------|
| No vehicles | 100/100 | 0/100 | 100% |
| Model1 (black) | 207/251 | 44/251 | 82% |

| | Success | Failure | Ratio |
|-----------------|---------|-----------|-------|
| No vehicles | 119/119 | 0/119 | 100% |
| White vehicle | 227/233 | (5+6)/233 | 95% |
| Model 2 (white) | 173/173 | 0/173 | 100% |

Table 1: Results on the first experiment

Vehicle tracking experiment

Now we have performed another experiment in which vehicle location in the super wide-angle camera image is identified. We have modeled vehicles of nine different locations on road in advance, then for each input images,

Table 2: Results on the second experiment

our system check if there is a vehicle on road, and where the vehicle runs. The figure 5 shows the nine training images. We select 60 local features on each image to obtain 20 code features. We set the threshold to 13 votes for all training images. The figure 6 shows the extracted feature points in this experiment.

We have processed 352 images. 233 of them include the white vehicle. The table 2 shows the result of this experiment. There were 5 false alarms and 6 detection failure. In 227 images out of 233 input images, our

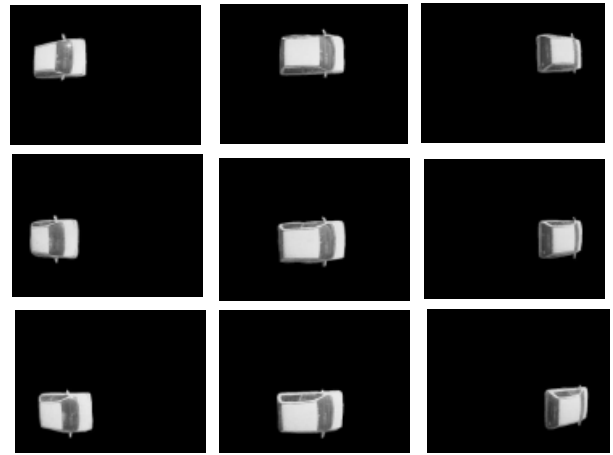


Figure 5: nine training images

system can recognize the white vehicle with exact location.

These results show that our system is effective for tracking vehicles, hence our system is useful to position-depending vehicle recognition.

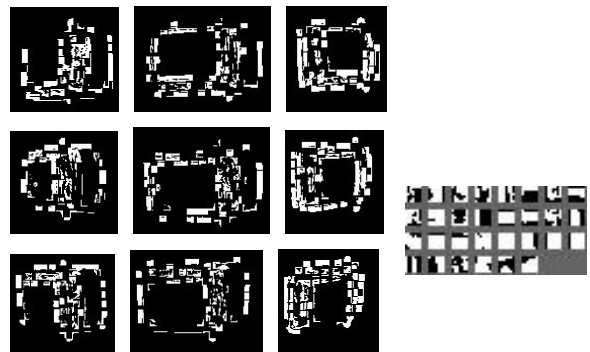
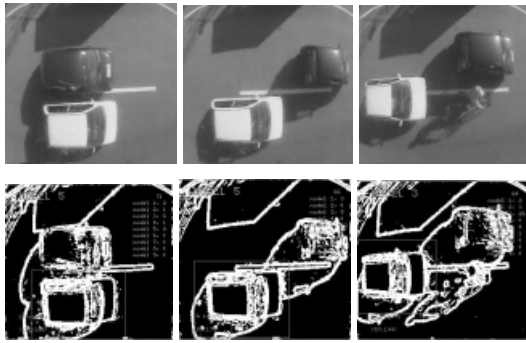


Figure 6: Feature points in nine training images and code feature set

(6) K.Kagesawa, A.Nakamura et al(2000) "Vehicle Type Classification in Infra-red Image Using Parallel Vision Board", *ITSWC 2000, Torino*.



CONCLUSION

We have confirmed throughout two outdoor experiments that our local-feature based method is effective to vehicle recognition with optical images. In the first experiment, two vehicles and a bicycle appeared in the same image and the shadow of a vehicle drops on road. In the second one, nine pre-set locations of vehicle is give in advance

Figure 7: Example of recognition

and our system detect where the vehicle in input images are. In the first experiment, the accuracy of our system is about 80-90%, and in the second one, it is over 95%. Since our system is view-based, these results suggest that our system is very good at detecting location of vehicles. In particular, this kind of system will be useful for such DSRC application that system needs to know to which vehicle it communicates with. We are now planning to combine this method with our vehicle classification method ⁽⁶⁾ to obtain intelligent vehicle recognition system.

REFERENCES

- (1) H.Murase and S.K. Nayar (1995) "Visual Learning and Recognition of 3-D Objects from Appearance," *International Conference on Computer Vision*.
- (2) K. Ohba and K. Ikeuch (1997) "Detectability, Uniqueness, and Reliability of Eigen-Windows for Stable Verifications of Partially Occluded," *IEEE Pattern Analysis and Machine Intelligence*, vol.19, No.9, pp.1043-1048.
- (3) John Krumm (1997) "Object Detection with Vector Quantized Binary Features," *Proc. of Computer Vision and Pattern Recognition '97*, pp.179-185.
- (4) Kohtaro Ohba and Katsushi Ikeuchi (1997) "Detectability, Uniqueness, and Reliability of Eigen-Windows for Robust Recognition of Partially Occluded Objects'," *IEEE Pattern Analysis and Machine Intelligence*, Vol.19, No.9, pp.1043-1048.
- (5) K.Kagesawa, S.Ueno et al (1999) "Vehicle Recognition in Infra-red Images Using Parallel Vision Board", *ITSWC '99, Toronto*.