APPLICATION OF PATTERN RECOGNITION TECHNIQUES FOR *ITS*

Tomokazu Takahashi[*,†], *Ichiro Ide*[*], *Hiroshi Murase*[*]

[*]Graduate School of Information Science, Nagoya University, Japan
ttakahashi@murase.m.is.nagoya-u.ac.jp, {*ide*, *murase*}*@is.nagoya-u.ac.jp*
[†]Japan Society for the Promotion of Science

ABSTRACT

In this paper, we introduce some of our works containing the latest achievements in recognition of large quantities of stored video and low-quality images to support and expand human visual capabilities in the context of driving. We have applied the works to ITS technologies, especially driver-assistance and navigation systems that recognize driving environments with an in-vehicle camera and/or other sensors.

1. INTRODUCTION

Recent years have seen great advances in the development of ITS (Intelligent Transport Systems) technology. For example, driver assistance and navigation with the aid of computers and various sensors are being actively developed. In particular, in-vehicle camera images are commonly utilized, since they contain important visual information. To support and expand human visual capabilities, we have been developing various pattern recognition techniques for large amounts of stored video and low-quality images. We have also been developing applications for ITS recognition techniques as outlined below.

- Enhancement of car navigation systems
  - **Change detection in streetscapes from GPS coordinated omni-directional image sequences** [1] (Section 2)
  - Estimating car location by corresponding series of multiple laser radar data [2]
  - Confirming the existence of stores on an urban map by detecting strip-like signboards [3]
- Weather recognition for driver assistance
  - **Raindrop detection from in-vehicle video camera images for rainfall judgment** [4, 5] (Section 3)
  - **Visibility estimation in foggy conditions by in-vehicle camera and radar** [6] (Section 4)
- Traffic sign and signal recognition
  - Identification of degraded symbols on traffic sign by a generative learning method [7]
  - Recognition of traffic signals in various conditions for safe driving assistance [8]

In this paper, we introduce the works highlighted in bold-face.

2. CHANGE DETECTION IN STREETSCAPES

One aspect of desired ITS technology is to enhance car navigation systems in terms of offering better information to drivers. In these systems, navigation maps are important. However, these need frequent updates because of streetscape changes such as road works and new buildings. Updating a map, however, is expensive, since many people have to actually walk through the city and collect a lot of relevant information.

2.1. Approach

For quick and efficient updating of maps, we have developed a method of automatically detecting changes in streetscapes from omni-directional images taken from cars.

In collecting street image data, we assume that many cars with general GPS and an omni-directional camera run freely without purposefully collecting the data. Consequently, we can obtain a large amount of GPS-coordinated images taken at various times and routes with a low-cost system. We use an omni-directional camera to efficiently capture images in all directions from the cars. To collect position information, high-accuracy GPS such as RTK GPS may be available, but it does not suit our application because of its cost. Therefore, we assume the use of a general GPS or D-GPS fitted to conventional car navigation systems. Such GPS, however, has an approximately 10-m margin of error, meaning that we cannot correctly gather images at identical locations by simply collecting images that have the same coordinates.

For the reasons mentioned above, we need to solve the following three problems.

1. Accurate alignment of images of the same location from images collected by freely running cars at various times.

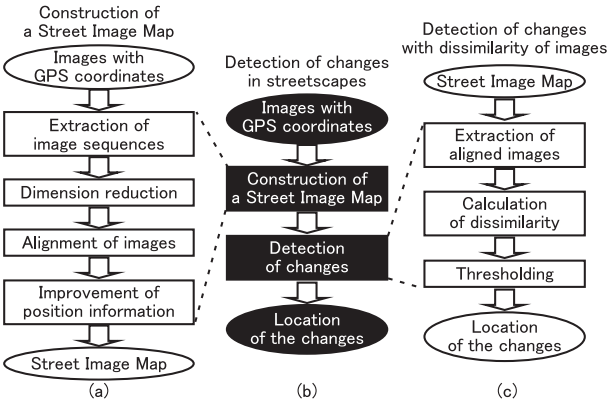

Fig. 1. Detection of changes in streetscapes.

2. Improvement of the position information attached to image frames.

3. Detection of changes in streetscapes from images taken at various times.

To deal with these issues, we have developed a novel method comprising alignment of images and calculation of the difference between aligned image frames. The method is composed of two stages. The first stage accurately aligns a map and street images taken at various times, while the second stage detects changes in streetscapes from the aligned data. In the first stage, we solve the first two problems. Image frames at the same location are aligned by matching image sequences taken along a roughly identified GPS coordinate route. For image matching, we integrate dimension reduction by PCA (Principal Component Analysis) and DP matching (a matching algorithm based on Dynamic Programming), then accurately determine the position information of each frame by calculating average coordinates for the aligned images. We call the aligned image data a *Street Image Map*. Then, in the second stage, we calculate the difference between aligned images taken at various times to detect changes in streetscapes.

2.2. Algorithm

We collect a large number of images with their GPS coordinates, construct a *Street Image Map*, and detect changes in streetscapes according to the process illustrated in Fig. 1.

2.2.1. Collection of images with GPS coordinates

A full implementation of this system involves many cars running freely, each with an omni-directional camera and GPS. The omni-directional camera is attached to the top of the car to capture images in all directions. Coordinates from GPS are taken in synchronization with the images.

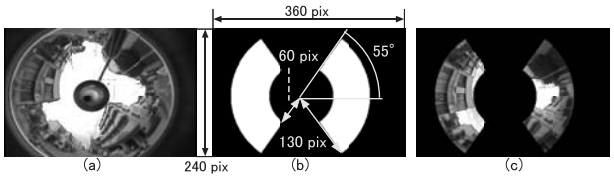

Fig. 2. (a) Omni-directional image, (b) Mask to extract feature vectors (N=25,538), (c) Masked image.

2.2.2. Stage 1: Construction of a Street Image Map

Image frames determined to be at the same location from large amounts of cityscape data are aligned accurately. We call this a *Street Image Map*. Figure 1(a) shows the process. Since the data we collect are taken by a lot of cars running freely, we must extract image sequences taken on the same routes, which we can distinguish with GPS coordinates. As a result, we can obtain images along the same route on various dates.

Next, PCA is applied to reduce the dimensions of the feature vectors of each image frame. This makes it possible to some extent to reduce the amount of calculation, the required storage space, and the influence of illumination changes accompanied by weather changes. The feature vector is an $3N$ dimension vector that has R, G, and B values for pixels in the masked area shown in Fig. 2, where N is the number of pixels in the area. This vector is normalized so that the average of its components should be zero and its norm, one. We limit the area with a mask because pedestrians or other cars may be at the edges, top, or bottom of the omni-directional images, and they can be detrimental to accurate matching. Before reducing dimensions, with PCA we create a lower-dimension eigenspace than the feature space using various city images, then project each frame feature vector to the eigenspace and obtain the sequence of points $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_p\}$ on the eigenspace. Here, p is the number of frames.

Next, we align dimension-reduced images frame by frame. This is the first point of our method. By this process, it is possible to align frames that reflect the same location at various times. DP matching is utilized to absorb temporal expansion and contraction caused by differences of car speed and to achieve alignment through all the images. We apply Eq. 1 below recursively to the two sequences of points on the eigenspace: $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_p\}$ and $\{\mathbf{y}'_1, \mathbf{y}'_2, \cdots, \mathbf{y}'_q\}$, and employ Euclidean distance on the eigenspace $d(i,j) = ||\mathbf{y}_i - \mathbf{y}'_j||^2$ as the dissimilarity.

$$D(i,j) = \min \begin{cases} D(i-1, \quad j \quad) + \omega_1 \cdot d(i,j) \\ D(i-1, j-1) + \omega_2 \cdot d(i,j) \\ D(\quad i \quad, j-1) + \omega_3 \cdot d(i,j) \end{cases} \tag{1}$$

Here, $D(1,1) = d(1,1)$. In an experiment, we set certain values for $\omega_1, \omega_2, \omega_3$. Furthermore, the sequence of frame

number pairs (i, j) chosen up to $D(p, q)$ have been calculated to show matches of two image frames.

The second point of our method is to average GPS coordinates attached to the aligned images with the aim of obtaining accurate position information. It is known that the average of the coordinates measured at a particular location for a long time converges at the true coordinates. Based on this, it is considered that the average coordinates are more accurate than the collected data.

2.2.3. Stage 2: Detection of changes

It is the third point of our method to detect changes in streetscapes from images of a specified location aligned in the first stage. Figure 1(c) shows this process.

The aligned frames of the specified location are sorted in order of time, namely, P_0, P_1, P_2, ... The dissimilarity between P_0 and the other frames P_i $(i = 1, 2, \ldots)$ is calculated by approximately adjusting their positions to reduce the influence of the car's position in the lane. If their variance exceeds a threshold d_T, we determine that there is a change, but if the camera catches a reflection off a large car or if image alignment fails at certain frames, the dissimilarity will rise temporarily. Therefore we restrain it from temporarily rising by median-filter and Gaussian-filter smoothing over the time sequence of the dissimilarities.

2.3. Results and discussion

2.3.1. Stage 1: Construction of a Street Image Map

First, we experimented on the construction of a *Street Image Map*, confirming the accuracy of image alignment. We used 44 data items collected over about a year, and aligned the oldest data with the other data. For the alignment, we extracted a route of about 170 m from the data, and reduced the 64,458 dimensions of the feature vectors to 20 dimensions. In terms of DP matching, the weight factor $(\omega_1, \omega_2, \omega_3)$ is $(2, 1, 2)$ from Eq. 1. This was the best weight in the pilot study.

We judged the results by manual checking. If a frame was aligned to its most similar frame, we judged it to be correct, while if there were some frames more similar than it, we judged it as false. We evaluated the rate of correctly aligned frames in all frames, which resulted in an average of 94.1%. Figure 3 shows a part of frames aligned at a certain location on the route. Their standard deviation of the GPS coordinates attached to the correctly aligned frames at that location was 6.86 m, which is assumed to be the accuracy of the GPS coordinates for measurement over a long time. The average of these standard deviations over all frames was 7.98 m. The data were collected at speeds under 40 km/h, and the frame rate was 30 fps, so the distance between the locations of two consecutive frames was less than 0.4 m. Thus the average coordinate has a 0.4 m margin of error in the car's direction of travel even if the alignment is correct and the position information is accurate. Therefore, it is assumed that if the number of samples increases, we can converge the error of position information to about 0.4 m in the direction of travel.

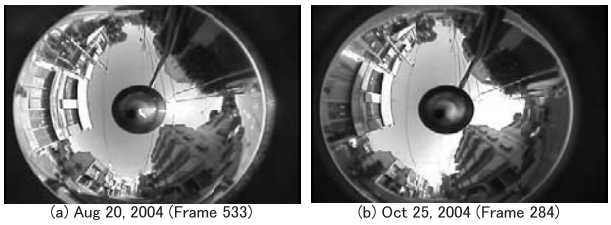

Fig. 3. Example of aligned frames.

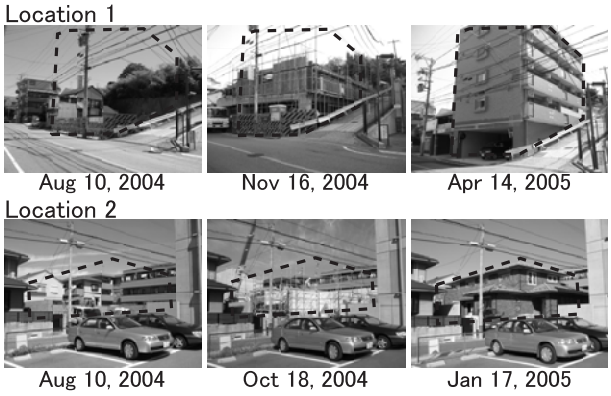

Fig. 4. Example of images at change-detected locations: The area where the streetscape changed significantly is marked by dashed lines.

2.3.2. Stage 2: Detection of changes

We also experimented on detection of changes using the same data used in Section 2.3.1. This time there were four extracted routes, which included four major changes in the streetscapes. Threshold d_T was set experimentally.

According to the results, we correctly detected three changes out of four. Figure 4 shows images of the detected locations. These images reveal that there were indeed changes at the detected locations. For the remaining case, where the change was too small even from human eyes, our method could not distinguish a sufficiently large dissimilarity.

2.4. Conclusion

We have developed a new method that detects changes in streetscapes from many street images taken at various times. Experiments produced the following three results.

- Images were aligned with a high accuracy of 94.1%. It is, therefore, possible to align images taken at the same

location at various times.

- GPS coordinates attached to the aligned images were accurate to between 7 and 8 m. We confirmed that if the alignment was accurate and we collected more data, we could converge the error to less than 0.4 m in the direction of travel.

- Three changes of streetscapes out of four were detected from real-world data.

Future work will include applying a larger amount of data. We consider the Street Image Map will have various applications in addition to detection of changes, for car navigation systems or driving simulation systems that use real-world images.

3. RAINDROP DETECTION FOR RAINFALL JUDGMENT

Since driving in rain is more difficult than in fair conditions, accident rates dramatically increase. Weather changes both temporally and spatially, so we believe that it is important to develop techniques that recognize weather in real time by in-vehicle sensors for driver assistance. Actually, auto-wiping systems are already implemented on some commercial cars for rain recognition, controlled by a so-called "rain sensor." However, the target region for detection covered by the sensor is small, so it does not necessarily reflect the changes in the visibility from a driver's view point. On the contrary, an in-vehicle camera covers most of the driver's visual field since it targets the entire windshield.

3.1. Approach

We have previously proposed a method of detecting raindrops from in-vehicle camera images by template matching using the subspace method, which extracts image features of raindrops and judges rainfall from the detected results [4]. This method suppresses false detection of raindrops by limiting the target region to the sky region, which does not have complex patterns in the background. However, it was ineffective in cases that the ratio of sky region to the entire image is small, such as in an urban district crowded with high buildings or in a tunnel.

Hence, we have developed a new method, using time-series information, that does not require region restriction for stable raindrop detection. This method includes the following features.

1. Automatic extraction of image features of raindrops by using PCA (Principal Component Analysis).

2. Robust detection of raindrops by using time-series information.

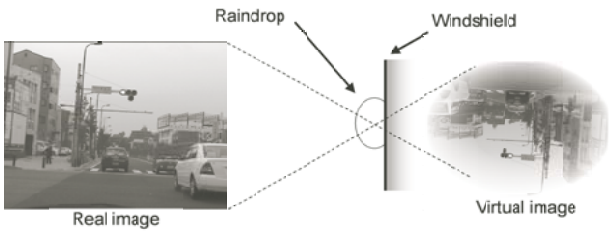

Fig. 5. Image feature of raindrop.

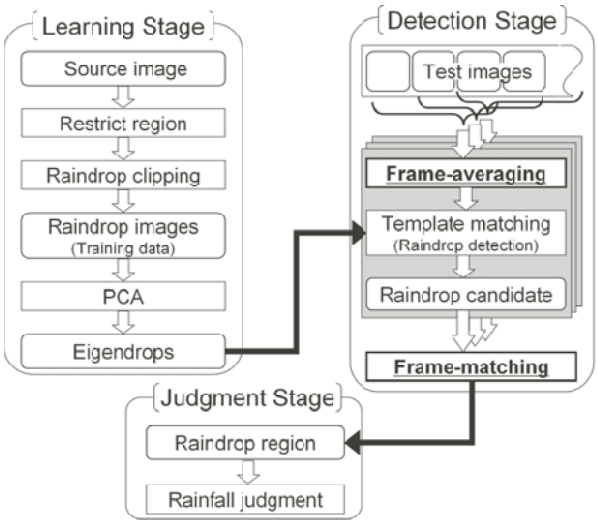

Fig. 6. The flow for rainfall recognition.

Raindrops have a uniform shape; any drop basically appears circular when seen through a windshield, and although a raindrop itself is clear and colorless, it is visible due to the reflection of its background as in Fig. 5. Raindrop texture varies since the background reflecting them varies. However, we believe that raindrops share at least the above features. Such image features are automatically extracted by using PCA.

While positions of raindrops on the windshield do not move in relation to the in-vehicle camera, the external view changes when the car is moving. Because of this, raindrops are emphasized by the change of background. Taking advantage of this phenomenon, we try to improve the detection accuracy by focusing on the temporal change of the image with raindrops, which is difficult to detect from a single frame due to the influence of complex backgrounds.

3.2. Algorithm

3.2.1. Overview of the method

As shown in Fig. 6, our method is composed of three stages: Learning, Detection, and Judgment.

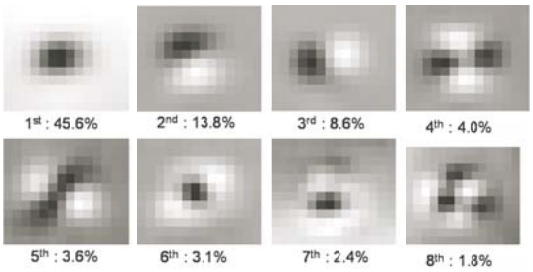

Fig. 7. Eigendrops and their contribution rates.

3.2.2. Learning Stage

First, a rectangular region circumscribing each raindrop is cut manually as a training set from images of a windshield taken in rainy weather. A total of *K* images is prepared for learning. Next, they are normalized in size to width *W* and height *H*, represented as one-dimensional vectors, which are then normalized so that they become unit vectors with means of 0, represented as: $\mathbf{x}_i = (x_1, x_2, \cdots, x_N)^T$, where $N = W \times H$. Let a matrix arranged by *K* randomly selected vectors from the test images be $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_K]$ and its covariance matrix be $\mathbf{Q} = \mathbf{X}\mathbf{X}^T$. The eigenvectors $\{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_R\}$ corresponding to the largest *R* eigenvalues of $\mathbf{Q}$ are selected as the feature vectors. A subspace generated by these eigenvectors are called "eigendrops."

3.2.3. Detection Stage

Raindrops are detected from the test images as follows. First, to emphasize the image features of raindrops, an averaged image is produced from multiple sequential frames obtained from the input video. In the averaged image, we focus on rectangular areas with the size of $W \times H$. Let the area be represented by an one-dimensional normalized vector $\mathbf{a}$. Next, we compute the degree of similarity $S(\mathbf{a})$ of $\mathbf{a}$ with the eigendrops, where $S(\mathbf{a})$ is defined as $S(\mathbf{a}) = \sum_{r=1}^{R}(\mathbf{a}, \mathbf{e}_r)$ ((**x**,**y**): inner product). The area is detected as a raindrop candidate if $S(\mathbf{a})$ is larger than a threshold. The coordinates are detected by computing $S(\mathbf{a})$ throughout the frame by shifting the rectangular area in focus. Finally, raindrop regions are obtained by frame-wise matching of the raindrop candidates.

3.2.4. Judgment Stage

Rainfall is judged by counting the number of raindrops detected during the detection stage. If the number of raindrops in the image exceeds a certain threshold, we judge that it is rainy, and not rainy if it does not.

3.3. Experiments

3.3.1. Setup

We mounted a digital video camera in a car and took the images (30 fps, 640×480 pixels, grayscale). Our method was applied to each frame of the input video sequence. Then the recall and precision ratios for raindrop detection were calculated to evaluate the detection accuracy. In the learning stage, the eigendrops were made from 500 raindrop images. Figure 7 shows eigendrops created from the clipped raindrops. The subspace dimension was six when the eigendrops were made.

3.3.2. Results

Figure 8 shows examples of raindrop detection in some experimental conditions, while Fig. 9 illustrates the recall and precision curves. It is clear that when the number of frames used for averaging increases, although recall improves significantly, precision drops somewhat. Furthermore, when the number of frames used for frame-matching increases, although precision improves, recall falls. The best result was precision of 0.97 and recall of 0.51 when the similarity threshold was 0.70 under five-frame averaging and ten-frame matching.

3.3.3. Discussion

Precision is more important than recall in practice for a windshield wiper controller, since to incorrectly recognize raindrops and let the windshield wiper malfunction must be avoided. However, it is also a problem when recall is too low. While this result was obtained from the entire image, it was not inferior to the result obtained by our previous method that restricted the target region of raindrop detection to the sky region (precision = 0.97, recall = 0.59).

Since the success rate of rainfall judgment using the result of raindrop detection from the sky region showed 89%, the new method should also be able to judge rainfall similarly.

3.4. Conclusion

We have been developed a new method that detects raindrops in background areas using inter-frame information. Experimental results illustrated the method's effectiveness as follows:

- Our method could detect raindrops from an entire image with high accuracy (97% precision and 51% recall) that was almost the same as that of the previous method using only sky regions.
- The success rate of rainfall judgment using the result of our method is considered to be up to 89%.

In future, we will evaluate the method under various rainy weather situations according to time, place, and rainfall.

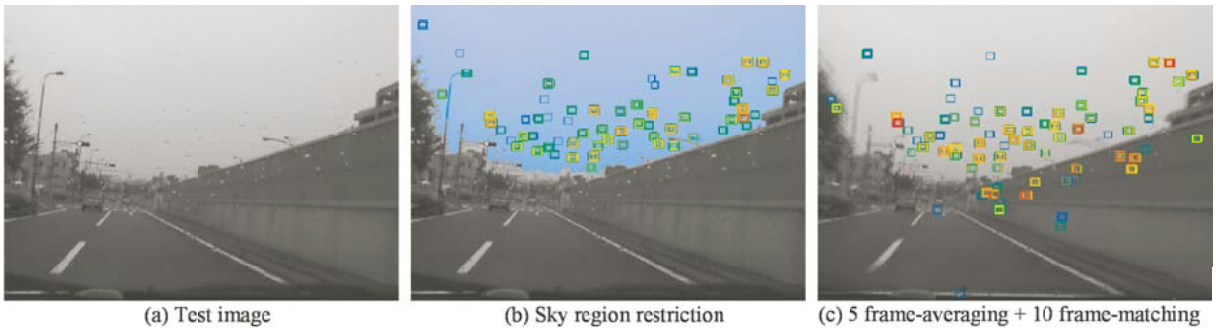

Fig. 8. Result of raindrop detection.

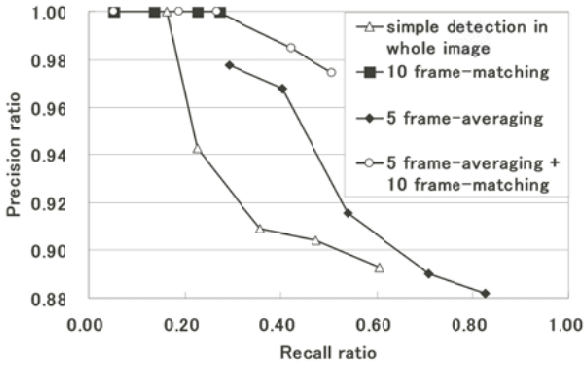

Fig. 9. Accuracy of raindrop detection by the proposed method.

4. VISIBILITY ESTIMATION IN FOGGY CONDITIONS

Fog negatively influences human perception of traffic conditions, making for potentially dangerous situations. Automatic lighting of fog lamps, speed control, and rousing of attention are examples of potential assistance to be realized with respect to fog recognition. Under foggy conditions, it is known that the distance between a preceding vehicle's tail lamp is perceived to be 60% further away than under fair conditions. Furthermore, fog changes significantly both temporally and spatially, and as a result there is a need for real-time detection using in-vehicle sensors. One method that involves installing large numbers of sensors along the roads can be a solution, though it may not accurately reflect a driver's visual condition. It would also be a very expensive system to establish.

4.1. Approach

Considering these problems, we have developed a method that classifies fog density into three levels using in-vehicle camera images and millimeter-wave (mm-W) radar data. The image from an in-vehicle camera reflects the driver's visual conditions, vital when driving. This is the prime advantage of using an in-vehicle camera. We also evaluate the degradation in visibility of images that are captured in foggy conditions, especially by focusing on the change in visibility of a preceding vehicle. It is also necessary to take into account the distance to the targets to determine fog density, because under the same fog condition, nearby objects are easy to see while distant objects are not. Consequently, we use a mm-W radar together with the in-vehicle camera since the radar can measure distance without any negative influence from adverse weather. The method is composed of the following two steps.

1. Extract a visibility feature from an image of a preceding vehicle captured by an in-vehicle camera.

2. Classify the fog density into three levels considering the visibility feature and the mm-W radar data.

4.2. Algorithm

Figure 10 shows the method's flow. Fog density is judged by both the distance to the preceding vehicle and the indicator calculated from the preceding vehicle region as follows.

4.2.1. Clipping the region of a preceding vehicle

First, preceding objects are detected in reference to the distance obtained from the mm-W radar. Moving objects are extracted according to their relative speed with the vehicle with an in-vehicle camera. Next, the position and size of the preceding vehicle region are accurately detected by template matching in the candidate area, referring to the dictionary image.

The accuracy was 90.17% when this method was applied to 4,149 images. All the images include a preceding vehicle. At present, we only consider a specific vehicle as the preceding vehicle, so a dictionary image manually cropped from a captured image was used as a template.

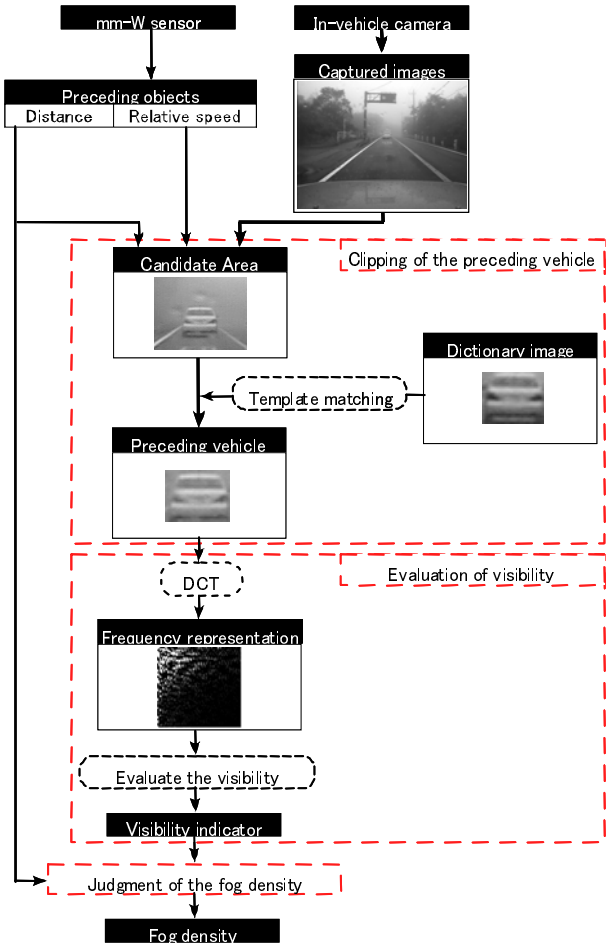

Fig. 10. Flowchart of the proposed method.

4.2.2. *Evaluating the visibility indicator*

We define an indicator that represents the visibility of the preceding vehicle. When fog appears, the outline of a preceding vehicle becomes more indistinguishable than in fair conditions because the captured images become whitish and blurred. This is the point on which we focused. The contrast of images captured in foggy conditions becomes low, and considering this in the frequency domain, we define a visibility indicator based on the image's power spectrum.

Figure 11 shows sample images and corresponding indicator values. An exploratory experiment with human subjects was conducted to investigate the relation between human perceptions of visibility and the indicator. From the result, we confirmed that the preceding vehicle becomes indistinguishable in proportion to the decrease of the indicator value.

4.2.3. *Judging fog density*

Visibility-meters are often used to measure fog density. In our work, however, we focus on the driver's perception rather than absolute physical visibility measures. Our method instead features three levels of fog density: dense, moderate, and light fog conditions, where the judgment of fog density is considered to be the classification of fog density.

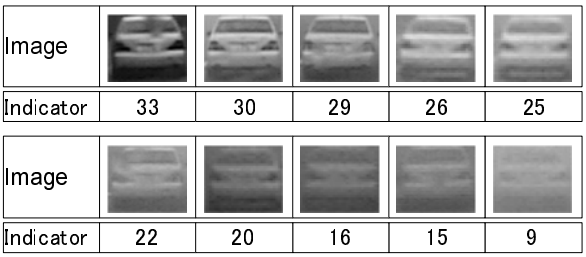

Fig. 11. Sample images and corresponding indicator values.

The classification method is as follows. First, calculate the regression curve that has the minimum squared error to the training data in each class. To classify input data, the distance between the input data and each regression curve is measured. The input data are then classified to a class with the nearest regression curve.

A regression curve is an exponential function referring to Koshmieder's model on the deterioration of brightness. Koshmieder's model is represented as follows:

$$L = L_0 e^{-kd} + L_f(1 - e^{-kd}), \tag{2}$$

where L is the observed luminance, L_0 is the intrinsic luminance of an object, L_f is the luminance of the sky, k is the extinction coefficient of the atmosphere, and d is the distance to the object. Therefore, L deteriorates exponentially according to d when k is fixed. Assuming that the indicator deteriorates according to Koshmieder's model, we use Eq. 2 for the regression curve.

4.3. Experiments

4.3.1. *Setup*

To design the classifier, we need training data with the most appropriate class for each image. This was done by the following procedure, in which we used images captured while

Table 1. Comparison of judgments by the proposed method and by human subjects. The percentages in diagonal elements represent the precision rate of each class.

Subjects \ Method	Light	Moderate	Dense
Light	**100 %**	0 %	0 %
Moderate	13 %	**82 %**	5 %
Dence	0 %	22 %	**78 %**

driving a vehicle. Five sets of images were tested, where one set included ten images that had been selected randomly from the captured ones. Four different subjects, each with a valid driver's license, participated in the experiment. They classified the ten images into three classes for each set, and from the result of this experiment, we obtained the appropriate class for each image, which complies with human perception.

4.3.2. Evaluation method

We compared the judgments by our method and those by human subjects. In the following experiments, the test data set was different from the training data set. The preceding vehicle was always the same vehicle in the experiment and the sky luminance was almost the same when the images were captured. Assuming that L_0 and L_f in Eq. 2 are invariables from this, the regression curve was calculated for each class.

4.3.3. Results and Discussion

The results presented in Table 1 show the confusion matrix for judgment by the proposed method and that by the human subjects. The total precision rate for all classes was 85%. In the experiment, we dealt with only one vehicle. In reality, however, the indicator is affected by the variety in color or shape of vehicles, though the indicator should not be affected by these variances for reliable judgment of fog density. Thus, improvement of the indicator is our next challenge.

4.4. Conclusion

We have developed a method that classifies fog density according to a visibility feature of a preceding vehicle and the distance to the vehicle. We obtained promising results (85% of precision) through an experiment using data collected from an in-vehicle camera while driving a vehicle. From the results, we confirmed that the proposed method can make judgments that comply with human perception.

In future, we will consider an improved visibility feature that does not vary depending on the type or color of a preceding vehicle. In addition, we will consider a situation where there is no preceding vehicle at all.

5. SUMMARY

In this paper, we have briefly introduced three works related to application of pattern recognition for ITS. Future work will involve conducting research on recognition and understanding of video and images. For more details of our works, please visit our website *http://www.murase.m.is.nagoya-u.ac.jp/*.

ACKNOWLEDGEMENT

We would like to thank our colleagues and former/current students, especially Junji Sato, Hiroyuki Kurihata, and Kenji Mori of Nagoya University, and Dr. Yoshito Mekada of Chukyo University.

We also thank Yukimasa Tamatsu and Takayuki Miyahara of DENSO CORPORATION for their cooperative works in Sections 3 and 4 of this paper.

Parts of the works were supported by the Grant-In-Aid for Scientific Research (16300054) and the 21st-Century COE program from the Ministry of Education, Culture, Sports, Science and Technology.

The works are developed based on MIST library, available at *http://mist.suenaga.m.is.nagoya-u.ac.jp/*.

REFERENCES

[1] J. Sato, T. Takahashi, I. Ide, and H. Murase: "Change Detection in Streetscapes from GPS Coordinated Omni-Directional Image Sequences," Proc. 18th International Conference on Pattern Recognition, Vol. 4, pp. 935-938, Aug. 2006, Hong Kong.

[2] N. Shibuhisa, J. Sato, T. Takahashi, I. Ide, H. Murase, Y. Kojima, and A. Takahashi: "Estimating Vehicle Location by Corresponding Multiple Laser Radar Data Series (in Japanese)," Technical Report of IEICE, PRMU2006-52, Jun. 2006.

[3] Y. Nakagawa, T. Takahashi, I. Ide, H. Murase, and Y. Mekada: "Confirming the Existence of Stores on an Urban Map by Detecting Strip-like Signboards (in Japanese)," Proc. Meeting on Image Recognition and Understanding, pp. 1207-1212, Jul. 2006.

[4] H. Kurihata, T. Takahashi, Y. Mekada, I. Ide, H. Murase, Y. Tamatsu, and T. Miyahara: "Rainy Weather Recognition from In-Vehicle Camera Images for Driver Assistance," Proc. IEEE 2005 Intelligent Vehicles Symposium, pp. 204-209, Jun. 2005, Las Vegas.

[5] H. Kurihata, T. Takahashi, Y. Mekada, I. Ide, H. Murase, Y. Tamatsu, and T. Miyahara: "Raindrop Detection from In-Vehicle Video Camera Images for Rainfall Judgment," Proc. International Conference on Innovative Computing, Information and Control, Vol. 2, pp. 544-547, Aug. 2006, Beijing.

[6] K. Mori, T. Kato, T. Takahashi, I. Ide, H. Murase, T. Miyahara, and Y. Tamatsu: "Visibility Estimation in Foggy Conditions by In-vehicle Camera and Radar," Proc. International Conference on Innovative Computing, Information and Control, Vol. 2, pp. 548-552, Aug. 2006, Beijing.

[7] H. Ishida, T. Takahashi, I. Ide, Y. Mekada, and H. Murase: "Identification of Degraded Traffic Sign Symbols by a Generative Learning Method," Proc. 18th International Conference on Pattern Recognition, Vol. 1, pp. 531-534, Aug. 2006, Hong Kong.

[8] F. Kimura, T. Takahashi, Y. Mekada, I. Ide, and H. Murase: "Recognition of Traffic Signals in Various Conditions for Safety Driving Assistance (in Japanese)," Proc. Meeting on Image Recognition and Understanding, pp. 618-623, Jul. 2006.