Korean Journal of Remote Sensing, Vol.24, No.3, 2008, pp.235~243

Mean-Shift Blob Clustering and Tracking for Traffic Monitoring System

Jae-Young Choi and Young-Kyu Yang †

College of IT, Kyungwon University, Seongnam-si, Gyeonggi-do, 461-701, KOREA

Abstract : Object tracking is a common vision task to detect and trace objects between consecutive frames. It is also important for a variety of applications such as surveillance, video based traffic monitoring system, and so on. An efficient moving vehicle clustering and tracking algorithm suitable for traffic monitoring system is proposed in this paper. First, automatic background extraction method is used to get a reliable background as a reference. The moving blob(object) is then separated from the background by mean shift method. Second, the scale invariant feature based method extracts the salient features from the clustered foreground blob. It is robust to change the illumination, scale, and affine shape. The simulation results on various road situations demonstrate good performance achieved by proposed method.

Key Words : background extraction, mean-shift, feature point, clustering, SIFT, tracking.

1. Introduction

Traffic monitoring is very essential element regarding to Intelligent Transport System because it makes a collection of traffic information in real-time and is used for traffic control. Magnetic loop detector is common method for traffic monitoring and control, but it is inflexible and requires digging grooves in the road in spite of inexpensive cost. Moreover, it can not give a variety of traffic information because it is considered as point detector.

In case of vision based system, it is more suitable than magnetic one and is able to detect, track, recognize, count, estimate speed, classify and extract information from the input images. It does not disturb traffic while installed and is easy to modify, too. Therefore, it can be used for road surveillance and control with low cost (Choi *et al.*, 2007).

In spite the apparent advantages of vision based approaches there are still many challenges. Especially, tracking is not an easy task as it encounters many problems as follows:

- Target object is often cluttered / occluded
- Target changes in size/scale/direction
- Environments are varied / changed
- Real-time computational cost is trade-off

In this paper, an efficient object clustering algorithm that extracts background automatically and segments moving object with mean shift method is proposed. The extracted and updated background

Received June 2, 2008; Revised June 16, 2008; Accepted June 20, 2008.

† Corresponding Author: Young-Kyu Yang (ykyang@kyungwon.ac.kr)

image is used in subsequent analysis to detect object. For accurate tracking of the segmented object, we also use a feature based tracking instead of the searching and matching entire image. The scale invariant feature transform can improve the performance which is robust to changing the intensity (illumination), shape, and scale of moving object. Through the feature matching based on scale invariant keypoints, the candidate features are compared between consecutive frames by Euclidean distance. The features from the same blob will follow similar trajectory, and the target's space location can be estimated. We consider five different matching scenarios with regarding to the change of the number of blobs. Therefore the system decides whether the blob is exact object or not. The suggested approach is able to measure the traffic parameters such as vehicle count, speed, and class over the entire trajectory of blob.

The remainder of this paper is organized as follows; Section 2 outlines the framework of the proposed system. Section 3 describes the background extraction and foreground blob clustering algorithm. Section 4 explains tracking blobs on feature space. Section 5 and 6 contain the experimental results and conclusion.

2. System Overview

1) Space-Time Tracking

There are three main strategies for space-time segmentation and tracking (DeMenthon *et al.*, 2002.); the first strategy attempts to discover spatial structures and extend them in temporal dimension; the second one discovers temporal structures and groups them in spatial dimension; and the third strategy treats the spatial and temporal dimensions equally as shown in Fig. 1.

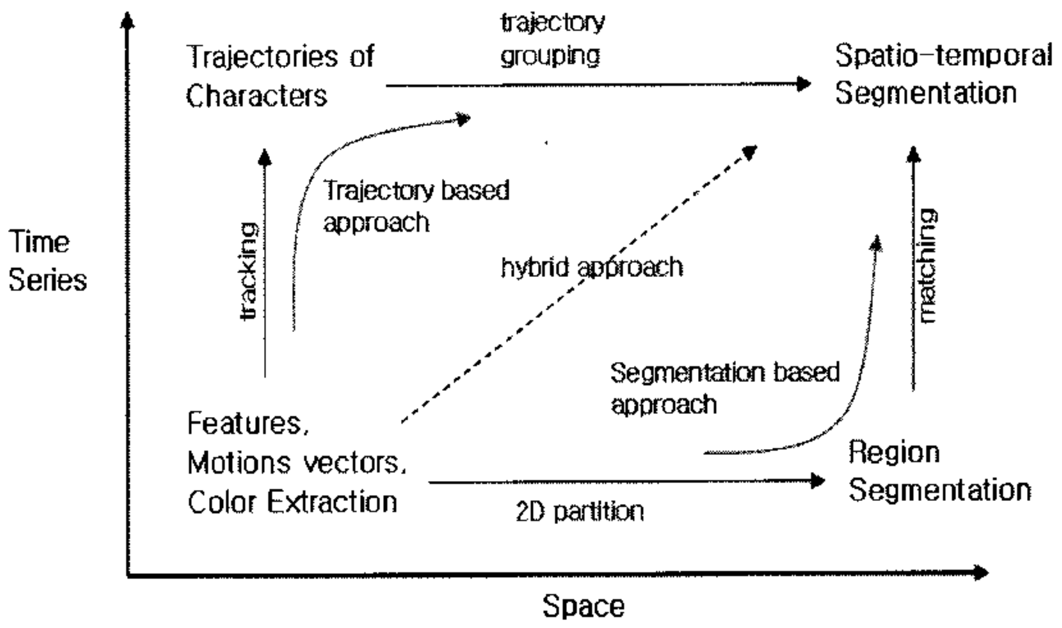

Trajectories of Characters
trajectory grouping
Spatio-temporal Segmentation
Time Series
tracking
Trajectory based approach
hybrid approach
matching
Segmentation based approach
Features, Motions vectors, Color Extraction
2D partition
Region Segmentation
Space

Fig.1. Various approaches for object segmentation and tracking.

Recently, a hybrid tracker has achieved impressive success in robust tracking.

The simple method for moving object detection and extraction in sequence frames is that compares contiguous images and subtracts the image from the previous image in order to eliminate background and get moving region within two images. However, it is influenced by environmental change and various speed of vehicle. Another method is template matching technique. It also has problem that is difficult to choose a proper template to find object because vehicles have different shape and size. After detecting foreground object, there are many methods in the literatures. Region based tracking is popular technique if background subtraction method was used for detecting vehicle. This process, however, makes the task of segmenting individual object difficult in case of under congested traffic conditions, vehicles partially occlude each other instead of being spatially isolated.

Feature based tracking method tracks subfeatures such as distinguishable points on the object. The advantage of this approach is that even in the presence of partial occlusion or deformation of shape, it could detect some of the remains of visible features on the moving object. This paper uses the features on the clustered blob to complement region based object tracking.

2) Framework

Fig. 2 depicts the framework of our proposed method. From the input digitized image, automatic background extraction method is used to get a reliable background as a reference. The moving object blob is then separated from the background by mean shift method.

The scale invariant feature based method extracts the salient features from the clustered foreground blob. SIFT can extract distinctive features from the image to be used to match different views, intensity, and scale.

Through the tracking process, the center of features in same blob is obtained, and makes a trajectory as connected line among the sequential frames. It is possible to measure the traffic parameters over the entire trajectory and the speed calculated by frame rate. The next sections will describe in detail.

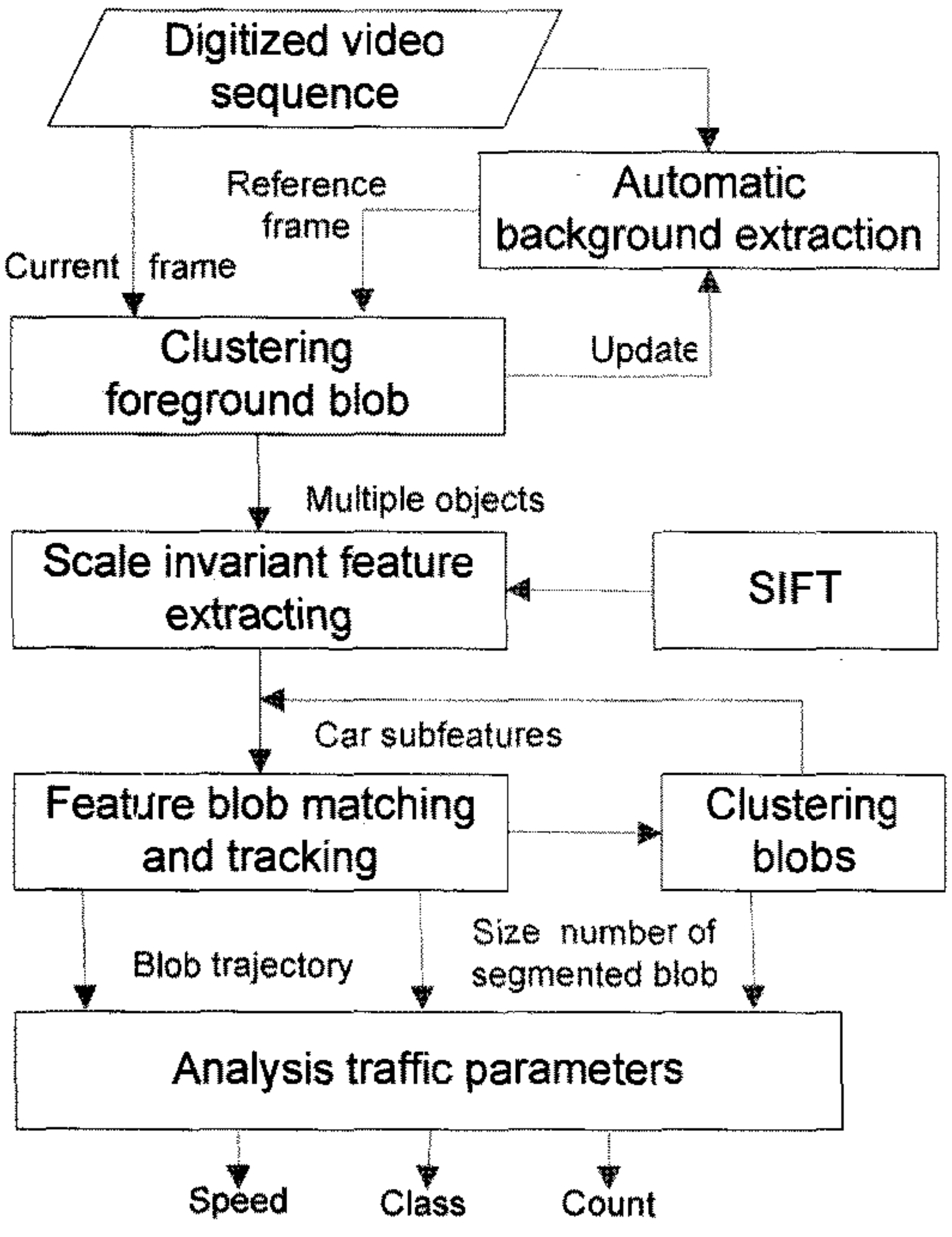

Digitized video sequence
Reference frame
Automatic background extraction
Current frame
Clustering foreground blob
Update
Multiple objects
Scale invariant feature extracting
SIFT
Car subfeatures
Feature blob matching and tracking
Clustering blobs
Blob trajectory
Size number of segmented blob
Analysis traffic parameters
Speed
Class
Count

Fig. 2. Framework for proposed method.

3. Object Clustering

1) Background Extraction

Accurate moving object segmentation will greatly improve the performance of next processes such as tracking, classification, and analysis.

Under the situation all foreground objects move, the pixel in the input image is possible to be background if it is not changed during several frames. Using above assumption, we gather the unchanged pixel to make a reliable background.

$F_{i,j}(t)$ denotes the input frame at time t, and i,j represents the pixel position. Eq. (1) shows the expression of frame-to-frame difference where λ represents the interval of time between frames. If we use consecutive frames to subtract each other, the foreground object also subtracts its own occluded region. Sometimes, the error occurs in case that the foreground region is not changed as a background one. Therefore, the suggested algorithm uses a proper time interval in order to avoid occlusion of same foreground on sequential frames. λ is considered with the environment such as traffic and change of illumination. In our test, the traffic density is normal, and we select frames with 0.6 fps, approximately. However, it is need to deal with adaptive value λ in accordance with road condition as a perspective development.

$$D_{i,j}(t) = \left| F_{i,j}(t) - F_{i,j}(t+\lambda) \right| \tag{1}$$

$$BG_{i,j}(t) = \begin{cases} D_{i,j}(t) & if \;\; I_{i,j}(t) < T_f \\ 0 & otherwise \end{cases} \tag{2}$$

In Eq. (2), if $I_{i,j}(t)$, grayscale intensity of $D_{i,j}(t)$, is less than T_f, the value of pixels at (i, j) is not change during the time interval λ. T_f is the threshold to make a decision whether the pixel is candidate background or not. Eq. (3) describes update of background where the symbol '|' is the bitwise OR logical operator.

$$BG_{i,j}(t+\lambda) = BG_{i,j}(t+\lambda) \,|\, BG_{i,j}(t) \tag{3}$$

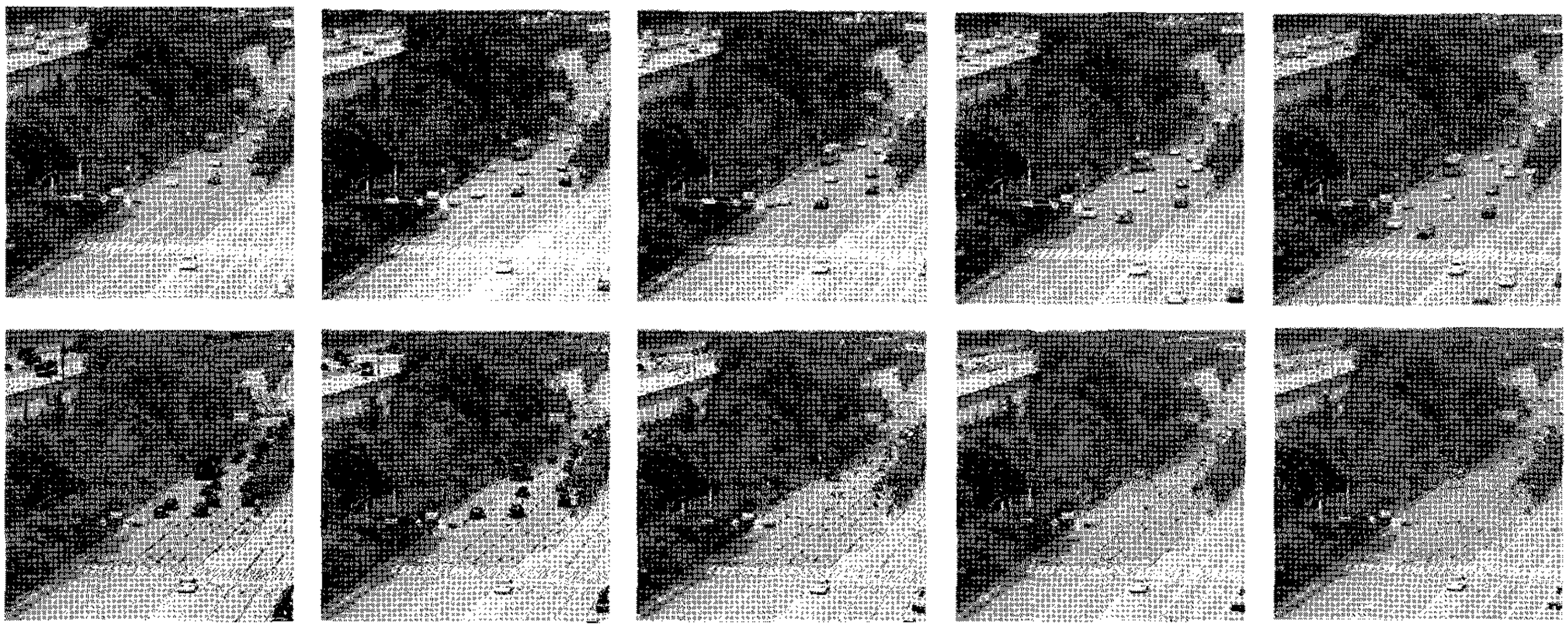


Fig. 3. Background extraction; (top): Input frame with time $t+n\lambda$, (bottom): Results of the each iteration.

The process will stop updating the reference background only when the pixels do not change much through comparing with prior updated background. That is, the procedure repeats each step till $|BG_{i,j}(t+\lambda) - BG_{i,j}(t)| < T_b$ where T_b is the threshold value of background error.

In Fig. 3, the top row shows the input image. The bottom of the Fig. 3 illustrates the results through Eq. (1)-(3). Only 5 iterations, we can obtain well result. Contrasting to the existing method, this approach does not need to empty a scene to get the reference background.

2) Mean Shift Clustering

The mean shift clustering algorithm first applied to image segmentation by Comaniciu and Meer in 1997 (Comaniciu *et al.*, 2002; Meer *et al.*, 2001), whereas the original idea was proposed in 1975 by Fukunaga and Hostetler (Fukunaga *et al.*, 1975).

This algorithm is designed to find modes, centers of the regions of high concentration, of data represented as arbitrary dimensional vectors. The major steps in the computation of the algorithm as follows; (Comaniciu *et al.*, 1997)

1. Choose the radius r of the search window.
2. Choose the initial location of the window.
3. Compute the mean shift vector and translate the search window by that amount.
4. Repeat till convergence.

The mean shift vector is described in Eq. (4).

If y_j is instead of x, and $\{y_j\}_{j=1,2,...}$ denotes the sequence of successive locations of the kernel $G(x)$, the Eq. (4) can be the weighted mean at y_j.

$$m(x) = \left[\frac{\sum_{i=1}^{n} x_i g\left(\frac{\|x-x_i\|^2}{h}\right)}{\sum_{i=1}^{n} g\left(\frac{\|x-x_i\|^2}{h}\right)} - x \right] \qquad (4)$$

In case of the color image segmentation like our application, the RGB color image is mapped into the $L^*u^*v^*$ color space model. The mean shift method clusters this multi-dimensional data set by associating each point to a peak of the data set's probability density. Fig. 4 depicts the results of clustering blob using mean shift segmentation algorithm by using window size 7.

One of the drawbacks in using mean shift algorithm is choice of the fixed bandwidth. That is, once the kernel bandwidth decided by the size of window, it is invariable in the entire tracking process. A certain method modified the bandwidth in ±10% intervals, the tracks object separately using three

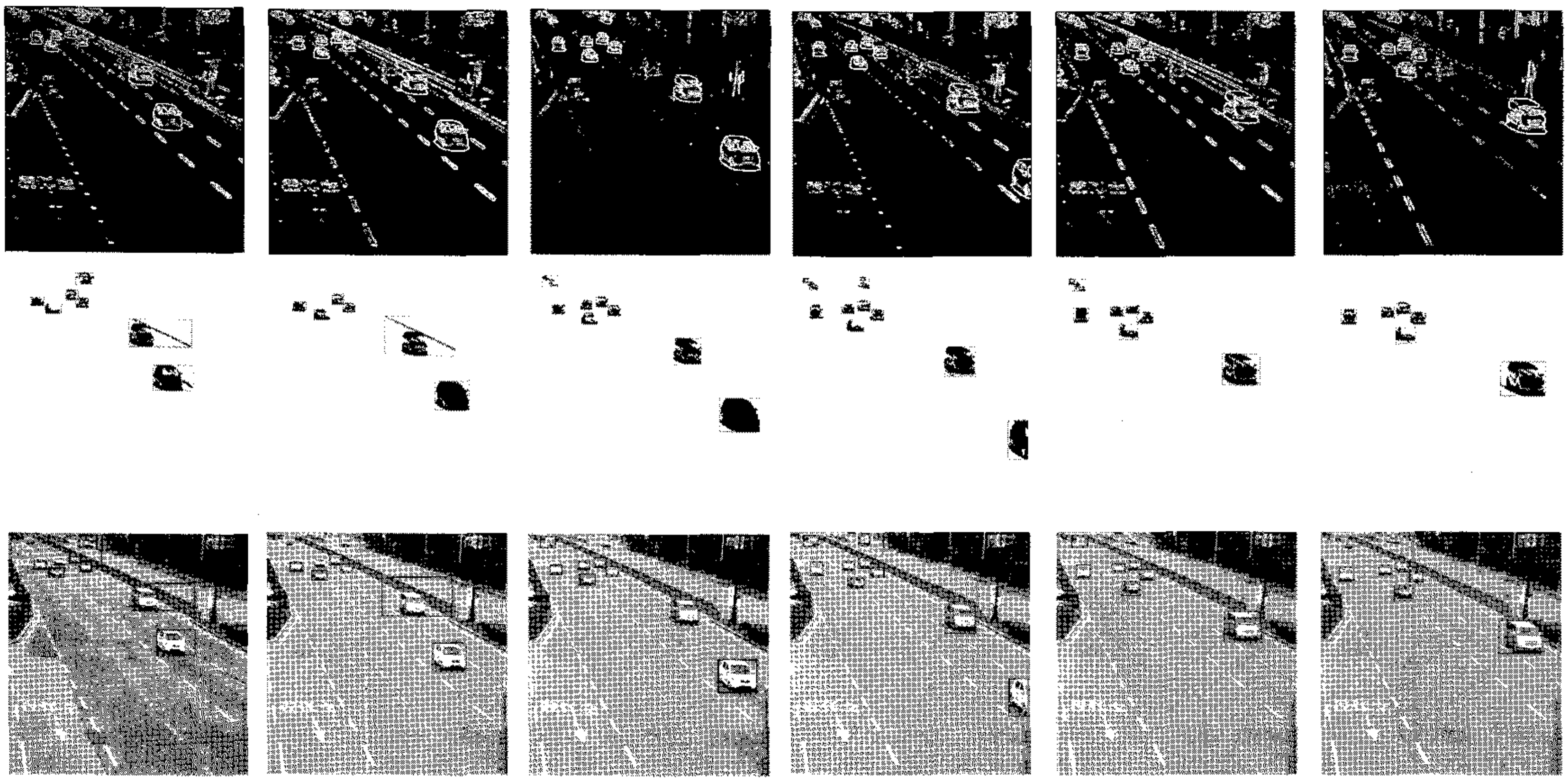


Fig. 4. Clustering foreground blob; (top): gradient map for clustering, (middle): Results of the clustering blob, (bottom): Segmented blob's position boxes.

different kernel bandwidths in the current frame (Comaniciu *et al.*, 2003). However, the scale of the object often changes more in time.

Recent method introduces tracking technique using the scale space with mean shift idea (Collins, 2003). This technique, however, needs whole search the next input frame to calculate similarity measurement using estimated target probability density function(pdf) if the target in next frame is not laid overlap the previous search boundary.

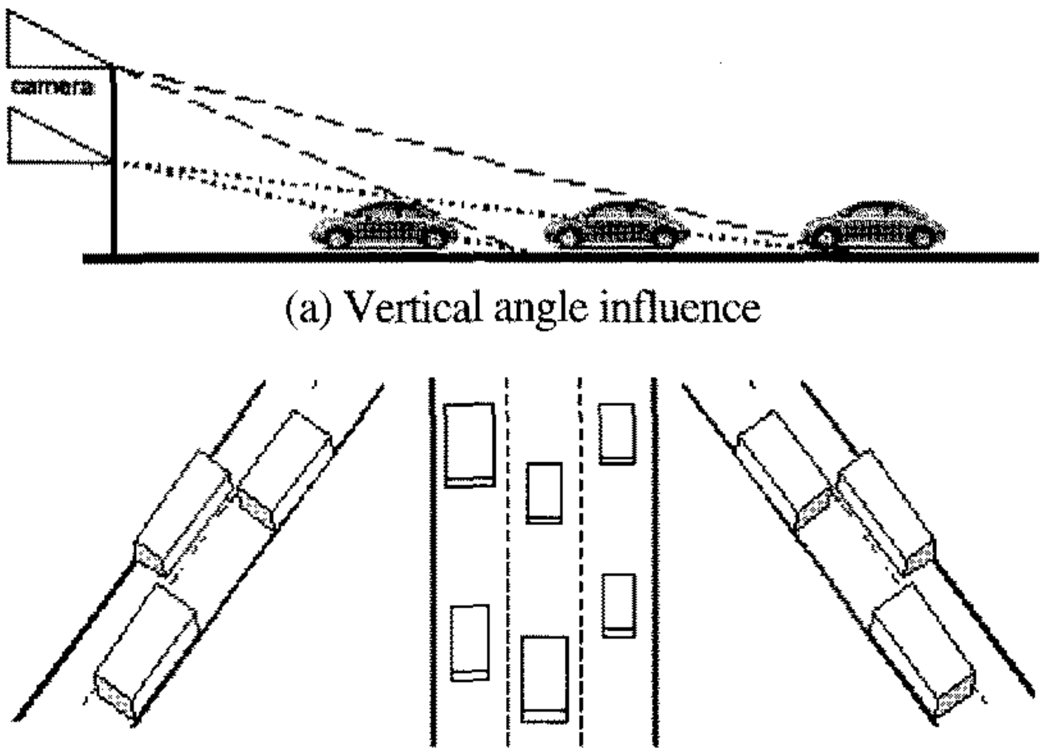
The advantage of using segmented blob by our proposed method is that reduce the cost of SIFT because it does not need to check features on the whole image in order to make SIFT keypoints.

4. Tracking Blobs on Feature Space

1) Features (keypoints) extraction

The clustered blobs are detected continuously via moving blob extraction by mean sift step, and tracked by using SIFT algorithm. SIFT can extract distinctive features from image to be used to matching different views, color, and shapes (Lowe, 2004).

Fig. 5 illustrates the different view with regard to the camera installation position. For example, when the camera is high and near the center of the road, a homography can be defined to map the road and height of vehicles does not affect to track because object's appearance does not change. In contrast, when the camera is located at the low or beside road, the object's view is changing dynamically according



(a) Vertical angle influence

(b) Horizontal angle influence

Fig. 5. Influence related angle of camera.

to the movement of vehicle (Kanhere *et al.*, 2005). By using scale invariant feature transform in this paper, we can overcome those restrictions.

The SIFT descriptors are constructed from two scale spaces; the Gaussian scale space of the input image $I(x, y)$ as in Eq. (5) and difference of Gaussian as in Eq. (6) where g_σ is an isotropic Gaussian kernel of variance σ^2. Scale space is function $F(x, y, \sigma) \in R$ of a spatial coordinate $x, y \in R^2$ and a scale coordinate $\sigma \in R_+$. Since a scale space $F(\cdot, \sigma)$ typically represents the same information at various scales $\sigma \in R$, its domain is sampled in a particular way in order to reduce the redundancy.

$$G(x, y, \sigma) \cong (g_\sigma * I)(x, y) \tag{5}$$

$$\begin{aligned} D(x, y, \sigma) &= G(x, y, k\sigma) - G(x, y, \sigma) \\ &\cong (k-1)\sigma^2 \nabla^2 G \end{aligned} \tag{6}$$

Using scale-space method the image is progressively Gaussian blurred (smoothed) in level σ_n, and produces a new series of spaces with the difference of Gaussians (DOG). It provides a close approximation to the scale-normalized Laplacian of Gaussian as shown in above Eq. (6).

Input image will produce several thousand overlapping features to identify potential interest points (keypoints) that are invariant to the scale and orientation. From the extrema in scale space the keypoints are chosen and assigned orientation as shown in Fig. 6(right).

The orientation θ of a keypoint (x, σ) is obtained as the predominant orientation of the gradient $\angle \nabla G(x_1, x_2, \sigma)$ in a window around the keypoint (Lindeberg, 1998). The boundary of segmented object does not affect to the descriptor since the responses along the image boundary or corner of object are rejected (Mikolajczyk *et al.*, 2005; Kadir *et al.*, 2001). To reduce the effects of illumination change, the feature vector is also normalized to unit length.

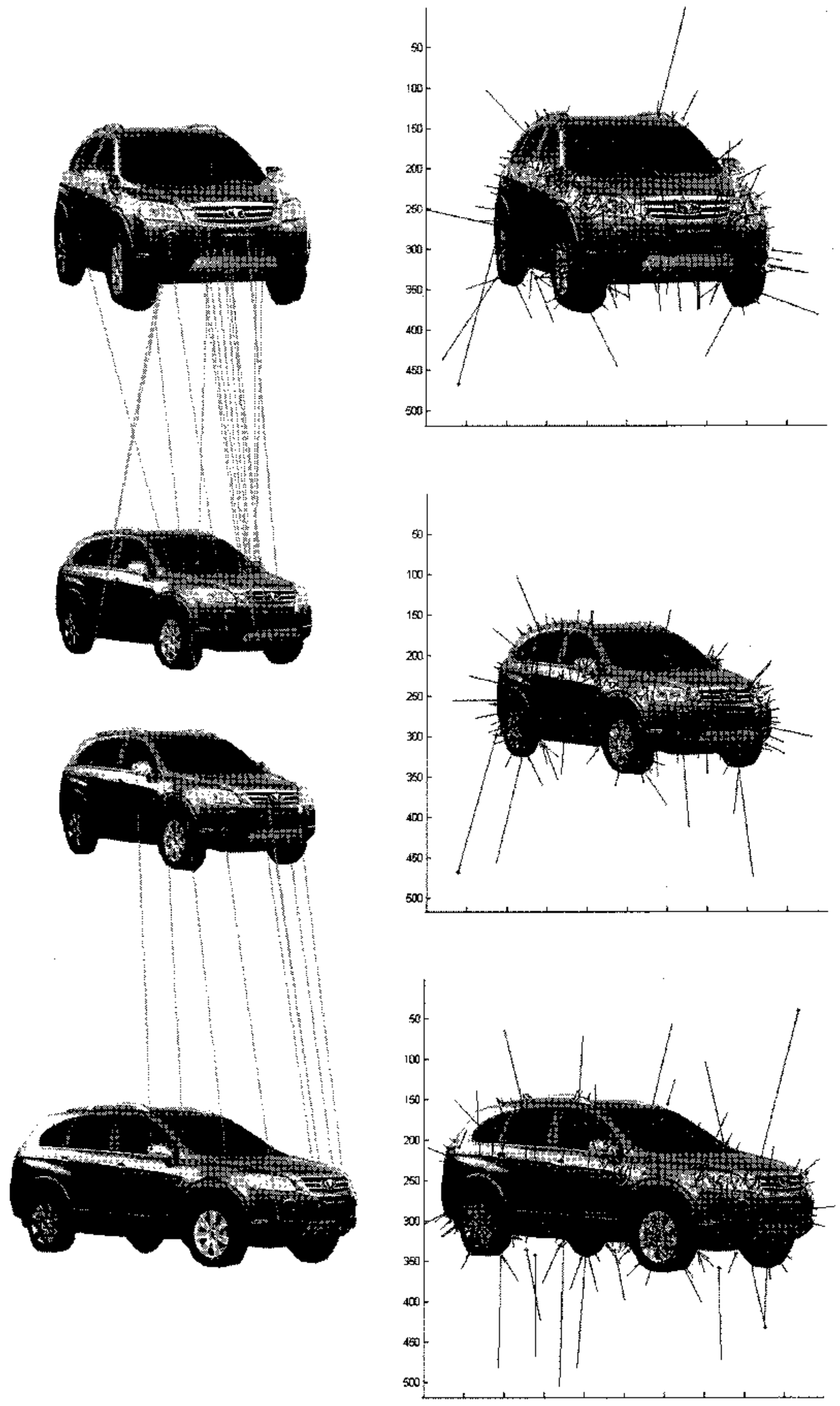


Fig. 6. Examples of matching features according to change of appearance; (left) Matching each feature by SIFT, (right) Description of features (keypoints).

2) Tracking blobs using features

Through matching process, the candidate features are compared between two frames by Euclidean distance. In fact, the feature descriptor consists of a histogram of gradient orientations augmented and spatial location. To classify features, each keypoint is matched independently along frames. Once each keypoint ϕ_i of N keypoints belongs to the same blob $B(j)$, then the keypoint is member of group $\Phi^{(j)}$ and get a same j^{th} label as shown in Eq. (7). Moreover, we can calculate the central coordinate, $\hat{l}^j$ of group $\Phi^{(j)}$.

$$\Phi^{(j)} = \{\phi_i \in B(j) \mid 1 \le i \le N\} \tag{7}$$

Therefore, features from the same vehicle will follow similar trajectory, and we can obtain the trajectory connected with the location of $\hat{l}^j$. The system considers a trajectory as an array of consecutive lines as $T^j = \{\hat{l}_i^j\}_{i=1,..,n}$ of blob $B(j)$ within n frames using spatial offset.

In achieving the matching and tracking vehicle between consecutive frames, we consider several different match scenarios as follow;

1. 'one to one' : it is good for establishment of performance and does not have any problem.
2. 'one to many' : this case causes the trouble that the vehicle of previous frame is matched with more than two vehicles on the next frame. That is, the features belong to previous one region are matched with two or more labeled next region. If the object splits across two similar amount of feature groups during a tracking, the algorithm ascribes the divided objects to the different object, whereas it ignores the case that few features mismatch with another blob. Therefore the system decides to the implicit occlusion, and generates two tracking trajectories for multiple vehicles even if those were occluded each other at initial point.
3. 'many to one' : it is possible to happen where most features of more than two previous objects belong to one next object since a vehicle is occluded by another vehicle or closes to an adjacent vehicle. In case of occlusion, the trajectory is preserved until the detection of the corresponding occlusion split. Otherwise, this trajectory is deleted from the list.
4. 'one to none' : this case occurs if a vehicle disappears from the scene or if the noise or shadow has been tracked. This case is resolved by eliminating object from the list if there is no same object on the next frame.
5. 'none to one' : the new one may be noise, and it is deleted. However, if the size of new object is larger than criterion, the object is added to the tracked object list as a new vehicle.

Left column of Fig. 6 illustrates matching keypoints which are compared between descriptors (see the right column of Fig. 6) with Euclidean distance, and tracking using those matched features. Although the size of vehicle is different (the middle car is smaller than the others a little), the feature from the space with the difference of Gaussians(DOG) can be matched well.

5. Experimental Results

In this paper, the digital camera installed over the several road with different environment to detect successive images, and 512 × 512 pre-digitized images was tested off-line. The images have various characteristics such as multiple moving vehicles, vehicle changed lane and occluded, and so on.

Note that the test images are performed on Matlab 7.0 platform, and the computation of our system required a little bit time per frame without code optimization. We think our approach will be definitely faster and can be adapted to the real time system by optimal programming and parallel processing.

To test the performance of the proposed method, our data consist of four different road situations such as Fig. 7.

In section 3.1, we should determine the threshold

Fig. 7. Various road environments.

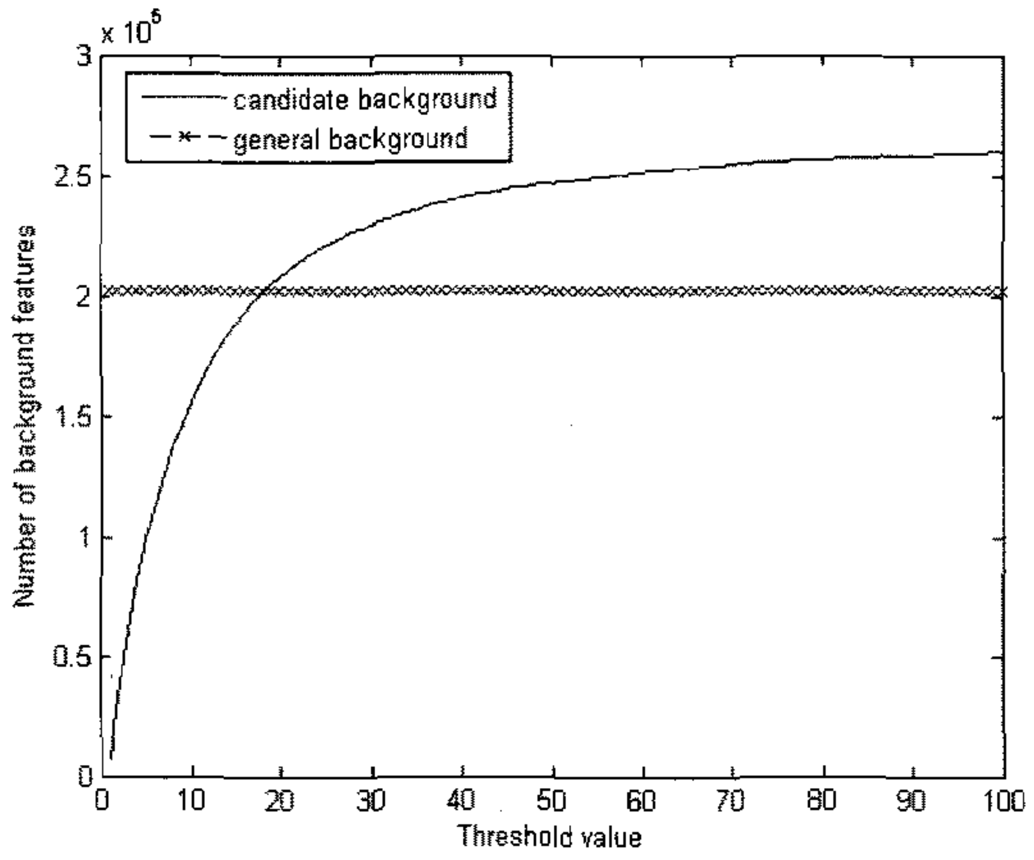


Fig. 8. Determination of threshold.

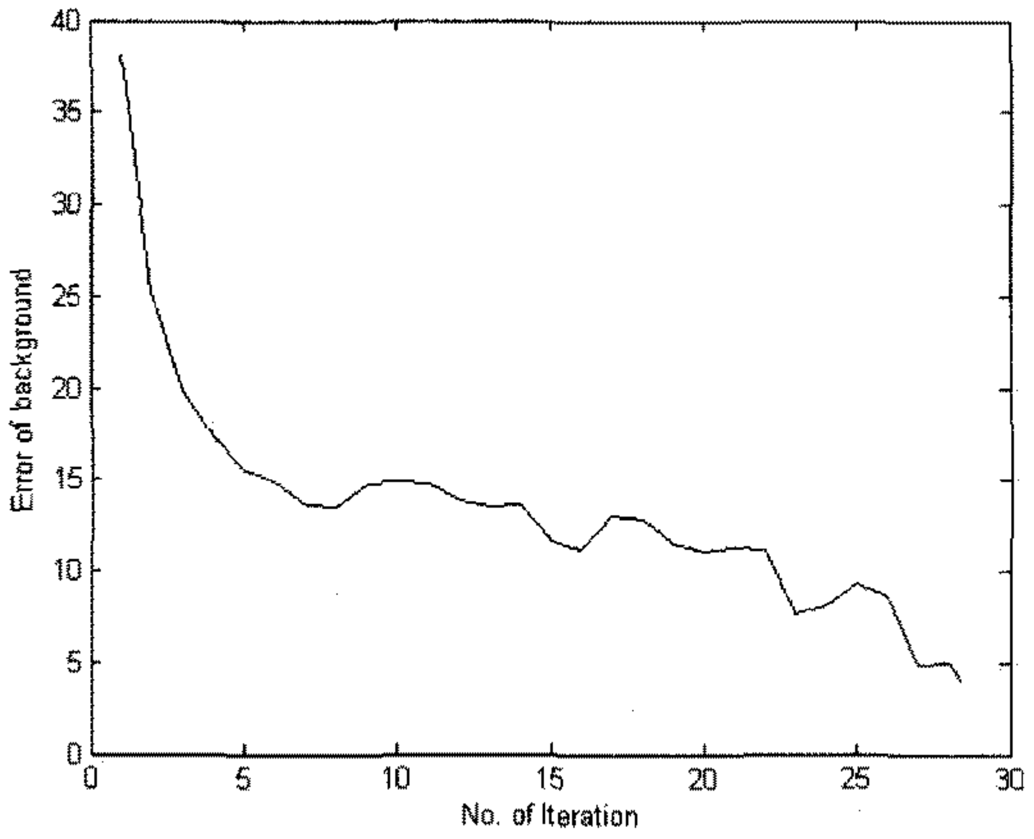


Fig.9. Error rate of background according to number of iteration.

value T_b in order to stop the background update process. As shown in Fig. 8, we calculate the amount of general background features, and decide the threshold value. The value of T_b applies same value to the whole test images from different road cases as 17. Sometimes, the threshold value depends on road situation such as amount of traffic flow.

To evaluate background extraction, we compare the results with ground truth as depicted in Fig. 9. The error rate is difference between our result and reference background. After 5 iterations, the error of output result is less than 15 pixel value. As a result, the proposed method is more efficient to extract background and has good performance with low computational cost.

Although the higher frame rate could get the higher accuracy of tracking, there is no meaning in our system on the tracking part. Therefore, in order to increase computational performance higher, it is able to reduce the number of sampling frames using tradeoff relationship.

Table 1. Evaluation of vehicle detection

Items	Detection rate [%]	False alarm rate [%]
Segmentation	83.6	7.5
Vehicle detection	95.4	2.6

Table 1 summarizes the numerical evaluation of test images. Some of the false alarms are caused by split blobs, sometimes the different region from the vehicle center body such as hood, trunk, and shadow is not merged with the tracked blob. In contrast, under the heavy traffic, it happened the vehicles are grouped together as a bigger blob. Nonetheless, most of vehicles are detected at the end of trajectory as you can check on Table 1 since it is easy to distinguish each blob. Such kind of mistakes will be reduced if the position of camera is higher.

6. Conclusion

The main contribution of the paper is to introduce a novel framework for efficient extracting and tracking of moving objects. Moreover, no pre-processing operations are carried out on the vehicle patterns. So, it does not need training sequence in prior.

We are developing the automatic background extraction method and the feature based tracking instead of tracking entire region using SIFT on image based system. The scale invariant feature transform can improve the performance which is robust to changing the illumination, shape, and scale of moving object. Through tracking, the features from the same blob will follow similar trajectory, and the location of blob can be estimated. We consider five different matching scenarios with regarding to the change of the number of blobs. Therefore the system can detect the exact object. If the algorithm is improved to overcome a few limitations, the proposed method will be very useful for the image based applications such as traffic monitoring system.

Acknowledgement

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)(IITA-2008-C1090-0801-0040).

References

Fukunaga, K. and L. D. Hostetler, 1975. The estimation of the gradient of a density function, with applications in pattern recognition, *IEEE Trans. Information Theory*, 21(1): 32-40.

Comaniciu, D. and P. Meer, 2002. Mean shift: a robust approach toward feature space analysis. *IEEE Trans. PAMI*, 24(5): 603-619.

Meer, P. and B. Georgescu, 2001. Edge detection with embedded confidence, *IEEE Trans. PAMI*, 23(12): 1351-1365.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints, *Int'l J. Computer Vision*, 60(2): 91-110.

Lindeberg, T., 1998. Feature detection with automatic scale selection, *Int'l J. Computer Vision*, 30(2): 77-116.

Kadir, T. and M. Brady, 2001. Scale, saliency and image description, *Int'l J. Computer Vision*, 45(2): 83-105.

Comaniciu, D..and P. Meer, 1997. Robust analysis of feature spaces: Color image segmentation. *IEEE conf. Computer Vision and Pattern Recognition*, 750-755.

Choi, J. Y., J. W. Choi, and Y. K. Yang, 2007. Improved tracking of multiple vehicles using invariant feature-based matching, *Lecture Note in Computer Science*, 4815: 649-656.

Collins, R., 2003. Mean-shift blob tracking through scale space, *IEEE conf. Computer Vision and Pattern Recognition.*

Mikolajczyk, K. and C. Schmid, 2005. A performance evaluation of local descriptors, *IEEE Trans. PAMI*, 27(10): 1615-1630.

DeMenthon, D. and R. Megret, 2002. Spatio-temporal segmentation of video by hierarchical mean shift analysis, *Proc. Statistical Methods in Video Processing Workshop*, Denmark.

Kanhere, N. K., S. J. Pundlik, and S. T. Birchfield, 2005. Vehicle segmentation and tracking from a low-angle off-axis camera, *IEEE conf. Computer Vision and Pattern Recognition.*

Comaniciu, D., V. Ramesh, and P. Meer, 2003. Kernel-based object tracking, *IEEE Trans. PAMI*, 25(5): 564-577.