Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, MO, USA, October 3-7, 2009

Pedestrian Detection by On-board Camera Using Collaboration of Inter-layer Algorithm

Bipul Kumar Sen, Kaichi Fujimura, Shunsuke Kamijo
Institute of Industrial Science, University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo, Japan
{sen, fujimura}@kmj.iis.u-tokyo.ac.jp, kamijo@iis.u-tokyo.ac.jp

Abstract— **In this paper we present a robust pedestrian detection algorithm in low resolution on-board monocular camera image sequences of cluttered scenes. At first a motion based object detection algorithm is developed to detect foreground objects by analyzing horizontal motion vector. A cascade structure of rejection type classifier is utilized for our pedestrian detection system. Initial stage of cascade, simple rule based classification techniques are used to separate pedestrian from obvious road side structural object and later part of the cascade, a more complex algorithm which is a combination of Histogram of Oriented Gradients(HOG) and Support Vector Machine(SVM) based classification techniques are utilized to separate pedestrian from non-pedestrian objects. Finally, the image segments are tracked by our Spatio-Temporal Markov Random Field model(S-T MRF). Results show that our algorithms are promising for pedestrian detection in cluttered scenes.**

Keywords- people detection, the S-T MRF model, pedestrian tracking.

I. INTRODUCTION

In the context of driving assistance systems, pedestrian detection in a real world environment is a complicated task. From the existing large number of pedestrian detection techniques, they can be classified into two groups: texture and motion based. Texture based approach utilize the appearance feature that is extracted using Haar-wavelet [1], egde template [4] and histogram of oriented gradients [5], etc. Papageorgiou and Poggio[1] and Oren et al.[2] utilize the extracted Haar-wavelet feature to describe pedestrian front and back side image and SVM classifier to validate the candidate region from static image frame. Gavrilla and Munder [3] and Gavrilla et al. [4] detect region of interest (ROI) based on pedestrian edge template matching followed by a verification stage based on Neural network architecture. Munder et al. [12] describes multicue(i.e. shape, texture, depth) object model within a Bayesian framework for detection and tracking based on particle filtering. Shashua et al. [6] presented a system which break region of interest into sub-regions, create a local vector representation per sub-region and feed this processed sub-region vector to adaboost classifier for verification. A similar approach presented by Dalal and Triggs [5] which uses the fact that the shape of object can be represented by a distribution of local intensity gradients or edge direction. For classification, SVM is trained using gradient histogram features from pedestrian and non-pedestrian classes.

On the other hand, motion based techniques rely on short-term motion by estimating optical flow. Cutler and davis [7] focus on human periodic walking motion pattern as a main cue for pedestrian detection. Sidenbladh [8] technique is based on collecting examples of human and non-human motion pattern and learns SVM with RBF kernel to create a human classifier. Viola et al. [9] presents a detection algorithm which combines motion and appearance information to build a robust model of walking human. The detector is trained using adaboost to take advantage of both motion and appearance information to detect a walking person. Curio et al. [10] detect walking pedestrians at road intersection. Their initial detection process is based on a fusion of texture analysis, model-based contour features matching of pedestrians, and inverse- perspective mapping (binocular vision). Additionally, motion patterns of limb movements are analyzed to classify pedestrian from other objects. Elzein et al. [11] detect region of interest by computing optical flow only in regions selected by frame differencing. By computing optical flow velocity their algorithm generate a group of pixel which having a small time-to-collision probability. Finally the selected regions of interest are searched to find pedestrian using manually selected Haar-wavelet features. Although above mentioned research looks promising with some extents, more research will be needed to adapt this driving assistance system as a life saving tool in a moving vehicle. Our research is aiming to fulfill the requirement to recognize pedestrian in real time.

In this paper, we present inter-layer collaborative pedestrian tracking algorithm where initial foreground segmentation is done by motion based object detection algorithm. A cascade structure of rejection style classifier introduced by viola et al. [9] is utilized to separate pedestrian and non-pedestrian object. Finally, Spatio-Temporal Markov Random Field model(S-T MRF) [13] based tracking is performed to track pedestrians.

II. INTER-LAYER COLLABORATION ALGORITHM

For pedestrian tracking from a moving vehicle, it is very important that the tracking technique in the single area is robust. We have developed an inter-layer collaboration algorithm for robust pedestrian tracking. Generally, algorithm hierarchy for cognitive systems can be separated into two groups of layers such as Signal Processing Layer and Semantics Layer. Signal Processing Layer is responsible for transforming the electronic signal such as image data into designated data having significance in the real-world such as vehicle trajectories. Semantics Layer is responsible for transforming data obtained from Signal Processing Layer into

978-1-4244-5521-8/09/$26.00 ©2009 IEEE

semantic symbols verbally recognizable for human. Physics Layer in Signal Processing Layer should contain algorithms that refer purely intrinsic information within images themselves. In other words, the algorithms in Physics Layer should not refer to any information such as form of vehicles and human, coordinate geometry between the image and the real world, and topology of vehicle lanes. For example, the method for background image acquisition can be regarded as Physics Layer algorithms, and the algorithm is generally applicable to a variety of objects and scenes. On the other hand, the algorithms in Morphology Layer should refer to external information that cannot be obtained from images themselves. Shape models of vehicles, pedestrians, buildings and road infrastructures are examples for such the external information. The S-T MRF model can be regarded as a Physics Layer algorithm, and it can track various objects against occlusion. The S-T MRF model also track objects in motion camera images segmenting them from motion background image. However, poles or trees on the road would be detected as objects having different motion from the motion background image. We developed object classification algorithms for Morphology Layer in order to distinguish pedestrians from such the road infrastructures. The combination of the algorithms in the two layers would achieve good performance, and the algorithms of object classification have become quite simple due to the help of the S-T MRF model. The fig. 1 shows the algorithm hierarchy and their collaboration in details.

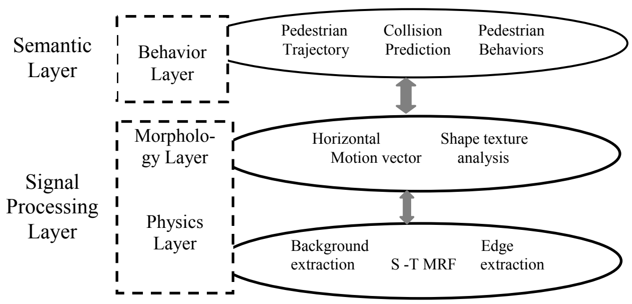


Figure 1. Inter-layer Collaboration System

The S-T MRF model in physics layer is responsible for stabilizing the algorithm and keep performance a certain level, and algorithms in Morphology Layer are responsible for maximizing the performance.

A. Motion Based Object Detection

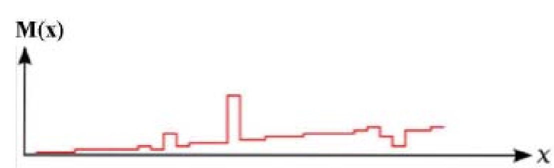
We employed a motion-based method for object detection which focuses on the difference of motion between foreground objects and onboard background images. Image regions of detected foreground compose regions of interests (ROIs), which indicate the candidates of pedestrians. Following the ROIs detection, texture patterns in the ROIs is analyzed to classify pedestrians from the other objects like polls, trees, bushes and other foreground facilities.

Walking pedestrians can be classified due to the difference in their motion from the motion of background infrastructures such as buildings. In addition, even the motion of the standing pedestrians and foreground facilities should be different from the background infrastructures due to the difference in their distance from the onboard camera. Therefore, foreground facilities such as polls, trees, and signboards would be detected at the same time as pedestrians. The method of motion-based object detection is described in this section, and the method of object classification will be described in the following section.

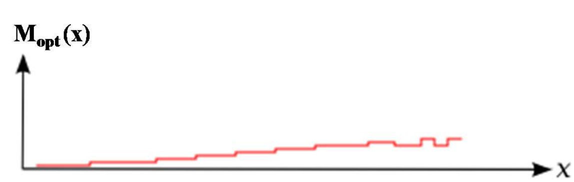
In general, the motion of the onboard background image is formulated by considering the motion of the onboard camera itself. In this paper, we assumed that the motion of the onboard background image can be approximated by the linear formula along the horizontal axis. It is because that onboard camera is moving horizontally, and the motion of the background image varies depending of distance between the camera and the background infrastructures.

Here, x-axis and y-axis in the image is taken as horizontal and vertical axis. An image block is defined as a group of 8 x 8 pixels, and images are divided into a group of image blocks. A matching line to approximate the background motion in a linear formula consists of neighboring blocks in the same vertical height as shown in Fig.2 (a). The motion of the background image is approximated by linear regression model with respect to each matching line. Blocks having different motion from the approximated linear formula should be detected as the foreground objects.

(a) An example of an image and a matching line



(b) Initial error in motion estimation



(c) Motion optimization by linear regression

Figure 2. Initial errors and potimization results of motion

The motion vector $M(x,y)$ is estimated about each block along the matching line by popular block matching technique: here (x,y) represents the image coordinate of the block. The motion vector $M(x,y)$ is estimated in the pixel unit. Since the major motion of the onboard camera is assumed to be in the horizontal direction, the matching region is widely searched along horizontal axis. On the other hand matching region is searched in the small motion along the vertical axis in order to cancel disturbance of the onboard camera in the vertical direction. For example, matching range in the horizontal direction was taken to be 24 pixels into the both directions, and matching range in the vertical direction was taken to be 5 pixels into the both direction. Fig.2 (b) shows the results of motion detection about the matching line indicated in Fig.2 (a). For a given matching line, y is a constant and $M(x,y)$ can be abbreviated by $M(x)$. $M(x)$ is a scalar value representing the motion in the horizontal direction, since the motion in the vertical direction is due to the disturbance of the onboard camera.

Initial result from block matching with respect to each block may contain an error in the motion estimation. It is because texture in the background building tends to have iterative textures and poor textures. Therefore, we employed the optimization algorithm to correct such the errors in motion estimation to obtain the most optimal motion approximation.

［Algorithm for motion optimization］

Step.1: For all the blocks along the given matching line, the motions $M(x)$ are individually estimated by the block matching technique. Here, for a given y, $M(x,y)$can be abbreviated by $M(x)$.

Step.2: Motion fitting function $L(x)$ is estimated by as a linear regression model of $M(x)$.

Step.3: For all the x，compare $M(x)$ and $L(x)$. If $|M(x) – L(x)|$ is larger than the threshold ML_{th}, $M(x)$ is assumed to be $M_{tmp}(x)$: $L(x) - ML_{diff} \le M_{tmp}(x) \le L(x) + ML_{diff}$. The sum of absolute value of image difference $D_{tmp}(x)$ for the block is then estimated according to the motion $M_{tmp}(x)$. If $D_{tmp}(x) < D_{th}$, take $M_{tmp}(x)$ as a corrected value for $M(x)$

Step.4: If the any error correction was done in the Step.3, go back to Step.2. If no error correction was done, proceed to Step.5.

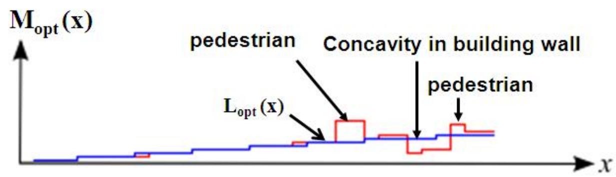
Step.5: Register $M(x)$ to $M_{opt}(x)$, and $L(x)$ to $L_{opt}(x)$ as the final result.

［The end of algorithm］

Relationship between $M(x)$ and $M_{opt}(x)$ are shown in Fig.2 (b) and (c). Although, some errors occurred in motion estimation in Fig.2 (b), the most of those errors were corrected in Fig 2 (c).

Fig.3 shows a case in which a matching line for motion detection is set in the height, y coordinate, of pedestrian existing. In this figure, difference between $L_{opt}(x)$ and $M_{opt}(x)$ appears to indicate foreground objects of different motion from the onboard background image. Such the foreground objects correspond to pedestrians of entrance concavity of a building, and the map distribution of the objects is fed to the S-T MRF model.

(a) An image and a matching line



(b) Difference in motion between onboard background image and foreground objects

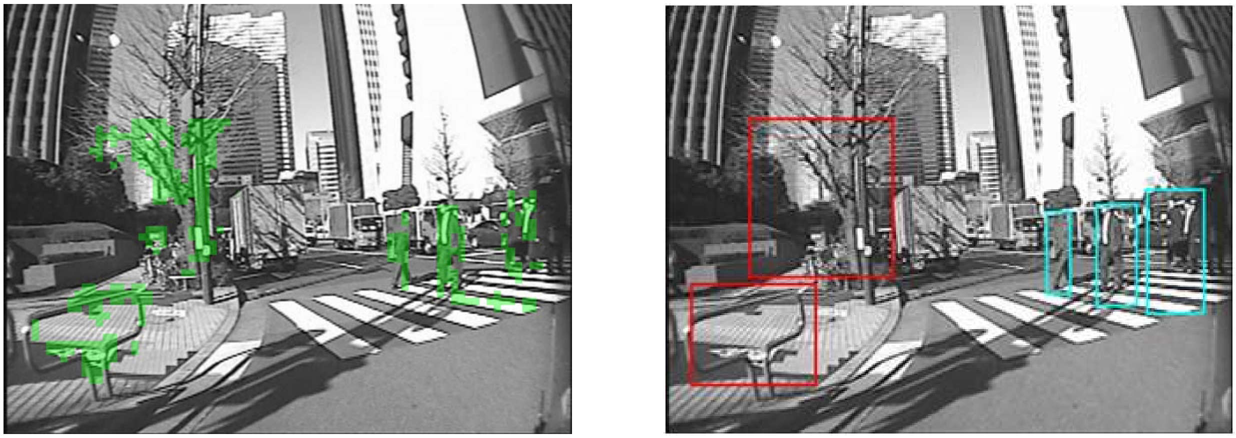
Figure 3. Optimization of motion estimation along a matching line

Fig.4 (a) shows blocks detected by the above motion-based foreground object detection. As in this figure, region of different motion from background image is determined in the block unit, 8 x 8 pixels. Those detected blocks were then grouped into clusters with individual labels, and the map distribution of the cluster labels is fed to the S-T MRF model to initialize the Object-map. The S-T MRF then proceeds to image segmentation in the following image frames. Here, image segmentation by the S-T MRF in the successive image frames means the tracking of the object shown in fig. 4 (b).

B. *Object Classification Algorithms*

The motion based object detection algorithm successfully segments pedestrians from a moving vehicle with few false positive segments. Most false positive segments are due to poles and organic structures, such as trees. Such false positives are, however, easily rejected in a classification stage. The presented segmentation algorithm is intended to be used as a component in a detection/classification framework.

In this section, we will discuss about cascade classifier that uses more than one stage of classifier to reject non-pedestrian elements as many as possible. After several stages of non-pedestrian object rejection, we finally get the filtered image segments which are most like to be pedestrian objects. In our cascade framework, simple detectors (rule based classifier) are placed earlier in the cascade while more complex detector with large number of feature vectors is used in the later part of the cascade. The cascade classification stages are as follows.



(a) Detected blocks by motion difference

(b) Object tracking and classification

Figure 4. Results of motion-based object detection

1) *Cascade stage 1*: *Strong vertical object classification*

Usually, pedestrian edges are not as vertical and straight as object like poll or window segments, very thin vertical object etc. To eliminate strong vertical object, we have used Hough Transformation [14] algorithm to find straight line which has very dense edge points (DEP). After analyzing adjacent texture distribution of the finding lines which are not complex in nature is discarded by this earlier stage detector.

A further stage of poll like object detector is utilized which is discussed in following algorithm.

[Algorithm: poll like object separation]

Step 1: Input Gray Scale Image Segment and transform as sobel edge image.

Step 2: Find N number of straight Lines using Hough Transformation that has Dense Edge Point (DEP) greater than a threshold.

Step 3: If texture distribution between any two parallel straight line is not complex, it will be treated as poll like object and rejected

[End of Algorithm]

2) Cascade stage-2: *Bush like complex textured object classification*

Usually bush like image segments contain high density of edge and distribution of edges are nearly uniform in all over the image. The following algorithm discuss about bush like complex textured object rejection methodology.

[Algorithm: Bush like object separation]

Step 1: Find the statistics of edge distribution

A fix window of size h×w is slide all over the edge image by shifting k-pixels in x and y direction. Finally, statistical variance (σ^2) is calculated over average edge distribution matrix E=M×N, where M = ((image height – h)/k) +1 and N = ((image width – w)/k) +1.

Step 2: Find the statistics of texture complexity.

Similarly, texture complexity of fix window h×w is observed as positive if average edge count for this local window is very high. Finally, a high texture occupancy matrix is generated by sliding window by k-pixels in x and y direction.

Step 3: Find the average edge density of the edge image.

Step 4: Experimentally, by setting appropriate thresholds in all the previous steps, if statistical variance is very low, high textured areas are occupied in most portion of the segmented image and average edge density is very high, it will be treated a complex textured object and rejected.

[End of the algorithm]

3) *Cascade stage-3*: *HOG + SVM based pedestrian r*

At this stage, a complex object classification is employed by using Histogram of Oriented Gradient(HOG) as a feature extraction algorithm and linear support vector machine(SVM) as a learning machine to make a linear decision boundary between pedestrian and non-pedestrian objects. Filtered image segments are variable in size. As a matter of fact, we have employed a HOG + SVM based pedestrian searching algorithm by sliding a variable sized window over the segmented region of interest (ROI). Following algorithm discuss about the HOG + SVM searching algorithm in detail. The searching algorithm have two parts: part 1 indicated algorithm for training methodology and part 2 indicated pedestrian detection algorithm.

[Algorithm: HOG + SVM pedestrian recognition]

[Part 1: Trainer]

Step 1: Input N number of normalized positive and negative window image of size $W_i \times H_j$; $\{W_{i=1 \text{ to } 5} \times H_{j=1 \text{ to } 5}\}$= {64 × 128, 120 × 56, 112 × 56, 104 × 48, 96 × 48}.

Step 2: Set HOG [5] Parameters: cell = 8 pixels, block = 2 x 2 cells, overlap of block = 1 cell, Number of Orientation

Bin = 9, no weighted voting and L_1-sqrt block normalization. Generate Histogram of Oriented (HOG) features for those training sets.

Step 3: Train SVMLight[15] with this five sets of Training Images labeled as positive and negative. This training phase will generate five (5) train model, TM_n where n= 1 to 5.

[End of part 1]

[Part 2: Detector]

Step 1: Read a list of Train model, TM_n where n = 1 to 5.

Step 2: Initialize HOG algorithm with the same set of parameters specified in part 1, step 2.

Step 3: Scale the ROI of size Width × Height pixels. In our experiment height remain fixed as 144 where width varies according to height. Generate HOG Feature Vectors for input Test image/ROI.

Step 4: Searching Begins

For n= 1 to 5

a) Slide window of size $W_n \times H_n$ over ROI by shifting k-pixels in x and y direction.

b) SVM classifier: SVM (HOG ($W_n \times H_n$), TM_n) = pedestrian/non-pedestrian.

c) Generate list file containing decision value (0 or 1) of SVM classifier

End of for loop

Step 5: Cluster based Decision Making algorithm: Final decision as pedestrian /non-pedestrian.

[End of part 2]

[End of algorithm]

Fig. 5 demonstrates the block diagram of HOG + SVM base pedestrian detection system. Usually pedestrian sizes inside ROI vary randomly. One traditional method to find the appropriate size of pedestrian is to scale the ROI in every possible size and search the pedestrian in every scale. Processing time for this kind of searching algorithms is very high and not feasible for real time pedestrian detection applications. To overcome the limitation of those traditional systems, we have proposed HOG + SVM searching algorithm as discussed in the above. Instead of scaling the ROI, it will remain fixed and we search the different size of pedestrian by using five different training models. The five different training models indicate five different size of pedestrian (128×64, 120 × 56, 112 × 56, 104 × 48, 96 × 48). Following algorithm will describe our cluster based decision making technique which is the final step of HOG + SVM searching algorithm.

[Algorithm for Cluster Based Decision Making]

For any particular ROI, the detection area of HOG/SVM searching algorithm tends to be clustered in a particular region of pedestrian images. This clustering tendency is true in all five scales. On the other hand, the detection area for a non-pedestrian ROI tends to be random. By using this simple cue, we can separate pedestrian with non-pedestrian.

[End of algorithm]

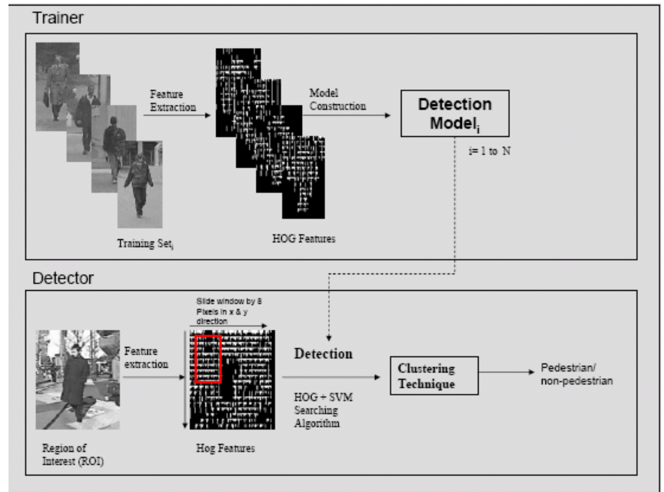


Figure 5. Block diagrams our pedestrian detection system.

(a) Hit area(red region) is very high for positive image

(b)Hit area is random for negative image

Figure 6. Cluster based decision making in scale 128×64(left), 120 × 56, 112 × 56, 104 × 48, 96 × 48(right)

Fig. 6 demonstrated the cluster based classification algorithm for pedestrian and non-pedestrian images. The rectangle gray region (Top left of each image) map the whole ROI. The Red region indicated successfully detected region by HOG/SVM searching algorithm. It is seen from fig. 6(a) that HOG/SVM detected pedestrian consecutively in all different searching scale. As a result, it is successfully detected as a pedestrian image. On the contrary, fig. 6(b) indicated successfully rejection by clustering technique.

C. *S-T MRF as a Physics Layer Algorithm*

Segmentation of the object region in the spatio-temporal image is equivalent to tracking the object against occlusion (see fig. 7).This is the principle idea of the S-T MRF model. Usually, the spatial MRF segments an image pixel by pixel. However, since the usual video cameras do not have such high frame rates, objects typically move ten or twenty pixels among consecutive image frames. Therefore, neighboring pixels within a cubic clique will never correlate in terms of intensities or labeling. Consequently, we defined our Spatio-Temporal Markov Random Field model (the S-T MRF model)[13] so as to divide an image into blocks as a group of pixels, and to optimize the labeling of such blocks by referring to texture and labeling correlations among them, in combination with their motion vectors. Combined with a stochastic relaxation method, our S-T MRF optimizes object boundaries precisely, even when serious occlusions occur.

Here, a block corresponds to a site in the S-T MRF, and only the blocks that have different textures from the background image are labeled as part of the object regions. In an image consists of 640x480 pixels, and a block is 8x8 pixels; such a distribution map of labels for blocks is referred to as an Object-Map. S-T MRF estimates current Object-map $X(t)=y$; given previous Object-map $X(t-1) = x$, previous image $G(t-1;i,j) = g(i,j)$, and current image $G(t,i,j) = h(i,j)$. Detail explanation of S-T MRF tracker can be found in [13].

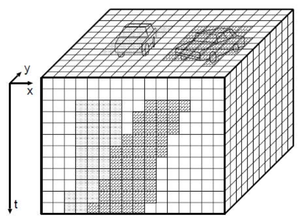


Figure 7. Segmentation of Spatio-Temporal Images

III. EXPERIMENT

In this section we will discuss about test and training datasets used in our experiments. We also discuss about result analysis in detail.

A. Datasets

1)Training Sets

Image data of pedestrian with various background and posture are used as training data for SVM. To train our HOG + SVM detector we selected 500 pedestrian from MIT database which is available from *http://cbcl.mit.edu/software-datasets/PedestrianData.html* and 2416 images of pedestrian from INRIA training database which is available from *http://lear.inrialpes.fr /data.* To get negative training data, we have selected person-free image frames from our database. By sliding a fixed size window over these non-person images, we automatically generate our negative data and randomly pick some negative data that are best candidate for negative training. In this paper, we have prepared five sets of training images of size 128×64, 120 × 56, 112 × 56, 104 × 48, 96 × 48 respectively.

2)Test images

We have taken video frames from CCD car mounted wide angle (75^0) camera placed on left and right side of vehicle. Most of the video sequences are taken from the crowded places in shinjuku, Tokyo. We have analyzed three gray scale video sequences of size 640x480 pixels containing a large number of low resolution pedestrians as well as other non-pedestrian objects. By using our motion based object detection algorithm and S-T MRF algorithm, a list of region of interest(ROI) are extracted from all the sequences. An ROI rectangle is scaled into height 144 pixels whereas width changes according to height by maintaining aspect ratio. Such scaled ROI will be examined by our cascade of classifier to validate the region as pedestrian or non-pedestrian. A sample positive and negative images that is used by HOG + SVM pedestrian detection system is show in fig. 8.

(a) Positive region of interest

(b) Negative region of interest

Figure 8. Sample test region of interest

B. HOG Descriptor

In this paper, all the training images are scaled or cropped in such a size that it will be even multiple of 8 pixels. For HOG descriptor, a block of 8 × 8 pixels are defined to extract local HOG feature. An orientation of gradients is estimated by an edge operation to each pixel, and the orientation is quantized into nine measures. For our experiments, no weighted voting mechanism is used. 64 quantized orientations for 8 × 8 pixels are plotted into a histogram with respect to nine measures. This histogram is translated into nine dimensions at which values of elements represent nine magnitudes in histogram. To overcome bad effect of local variation in illumination and foreground-background contract, local contract normalization is utilized. In this paper, a descriptor block of 16 × 16 pixels consisting of four 8 × 8 pixels cells are defined, and a descriptor vector of 36 dimensions is obtained by connecting four 9 dimension vectors. If an image size is 64 × 128, a sequence of descriptor blocks are obtained by shifting 8 pixels into the direction of raster scan, and a sequence of 7 × 15 descriptor blocks are obtained. Each descriptor blocks are normalized by L_1-sqrt norm (L_1-sqrt, f =sqrt ($v/(\|v\|+e)$)). Thus, a vector of 3780 dimensions is obtained connecting 105 normalized descriptor vectors with respect to an image of size 64 × 128.

C. Results

For result analysis we analyzed two (2) left camera video sequences and one (1) right side camera video sequences. The numbers of frames are 646 and 304 for left side camera video sequence and 1345 for right side camera video sequences respectively. Generally, pedestrians are in dangerous situation when they cross road at intersections and drivers taking left and right turn at those intersections. The analyzed video sequences are mostly road intersection video sequence where a large number of pedestrian are crossing road.

To quantify detector performance we have used two performance metric recall rate $\left(\frac{\#\ of\ Detected\ Pedestrian}{\#\ of\ Pedestrian}\right)$ and false alarm rate $\left(1-\frac{\#\ of\ Detected\ Pedestrian}{\#\ of\ all\ \det ected\ Object}\right)$. Table 1 shows our results.

TABLE 1. PERFORMANCE ANALYSIS

Classification Algorithm	SVM + HOG	Cascade
Total Number of Pedestrian object	122	122
Total number of non-pedestrian objects	382	382
Recall rate	0.84	0.84
False alarm rate	0.32	0.26

We have analyzed the performance of our pedestrian detection system in two different scenarios. At first experiment, we have analyzed performance by skipping first two stages of cascade and utilize only HOG + SVM as a basis for our classification algorithm. Later, we utilized all the three stages of cascade to find the performance benefit from the previous experiment. The ROI provided by our motion based object detection algorithm is successfully tracked for a certain number of frames by S-T MRF tracker and each object is identified by a unique object ID as shown in fig. 9. From table 1, it is found that 122 pedestrian and 382 non-pedestrians are tracked by our S-T MRF tracker from three test video sequences. After the classification stage, recall rate of 84% is found for both of our experiment. With using cascade of classifier, false alarm rate is improved from 32% to 26%. Fig. 9 shows the S-T MRF tracking results. Red bounding box of non-pedestrian object is successfully separated by our classification algorithms. Fig. 10 shows some of cases which can not handle by our pedestrian detection system.

IV. CONCLUSION

In this paper, we introduce the idea of inter-layer collaboration based pedestrian detection and tracking algorithm and demonstrate how it can be used for pedestrian detection in a highly cluttered scene. Using a cascade of classifier, our pedestrian detection system is very efficient. Specially, initial stages of rule based classifier filtered out most of the road side organic and structural object that makes later stage classifier efficient for pedestrian recognition. By taking the benefits of inter-layer collaboration, our S-T MRF tracker keeps system performance in acceptable level.

ACKNOWLEDGMENT

The research of this paper is founded by Semiconductor Technology Academic Research Center (STARC).

REFERENCES

[1] C. Papageorgiou, T. Poggio, “A trainable system for object detection,” International Journal of Computer Vision 38 (1) (2000) 15–33.

[2] M. oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio., “Pedestrian detection using Wavelet template,” In Proceedings of IEEE CVPR 97, pages 193-199,1997.

[3] D. M. Gavrila, J. Giebel, and S. Munder, “Vision Based Pedestrian detection: The PROTECTOR system,” in proc. IEEE Intell. Veh. Symp., June 2004.

[4] D. M. Gavrila, and S. Munder, “Multi-cue pedestrian detection and tracking from a moving vehicle,” Int J. Comput. Vis., 2007.

[5] Navneet Dalal, Bill Triggs, “Histograms of oriented gradients for human detection,” IEEE Conference on Computer Vision and Pattern Recognition 2 (2005) 886–893.

[6] Amnon Shasua, Yoram Gdalyahu, Gaby Hayun, “Pedestrian detection for driving assistance systems: single-frame classification and system level performance,” IEEE Intelligent vehicles Symposium, 2004.

[7] R. Cutler, L. Davis, “Robust real-time periodic motion detection: analysis and applications,” IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 781–796.

[8] H. Sidenbladh, “Detecting human motion with support vector machines,” Proceedings of the 17th International Conference on Pattern Recognition, 2:188-191, 2004.

[9] P. Viola, M.J. Jones, and D. Snow, “ Detecting pedestrian using patterns of motion and appearance,” IEEE International Conference of Computer Vision, 2:734-741,2003.

[10] Cristóbal Curio, Johann Edelbrunner, Thomas Kalinke, Christos Tzomakas, and Werner von Seelen, “ Walking Pedestrian Recognition.,”IEEE Transaction on Intelligent transportation Systems, Vol.1, No. 3, September 2000.

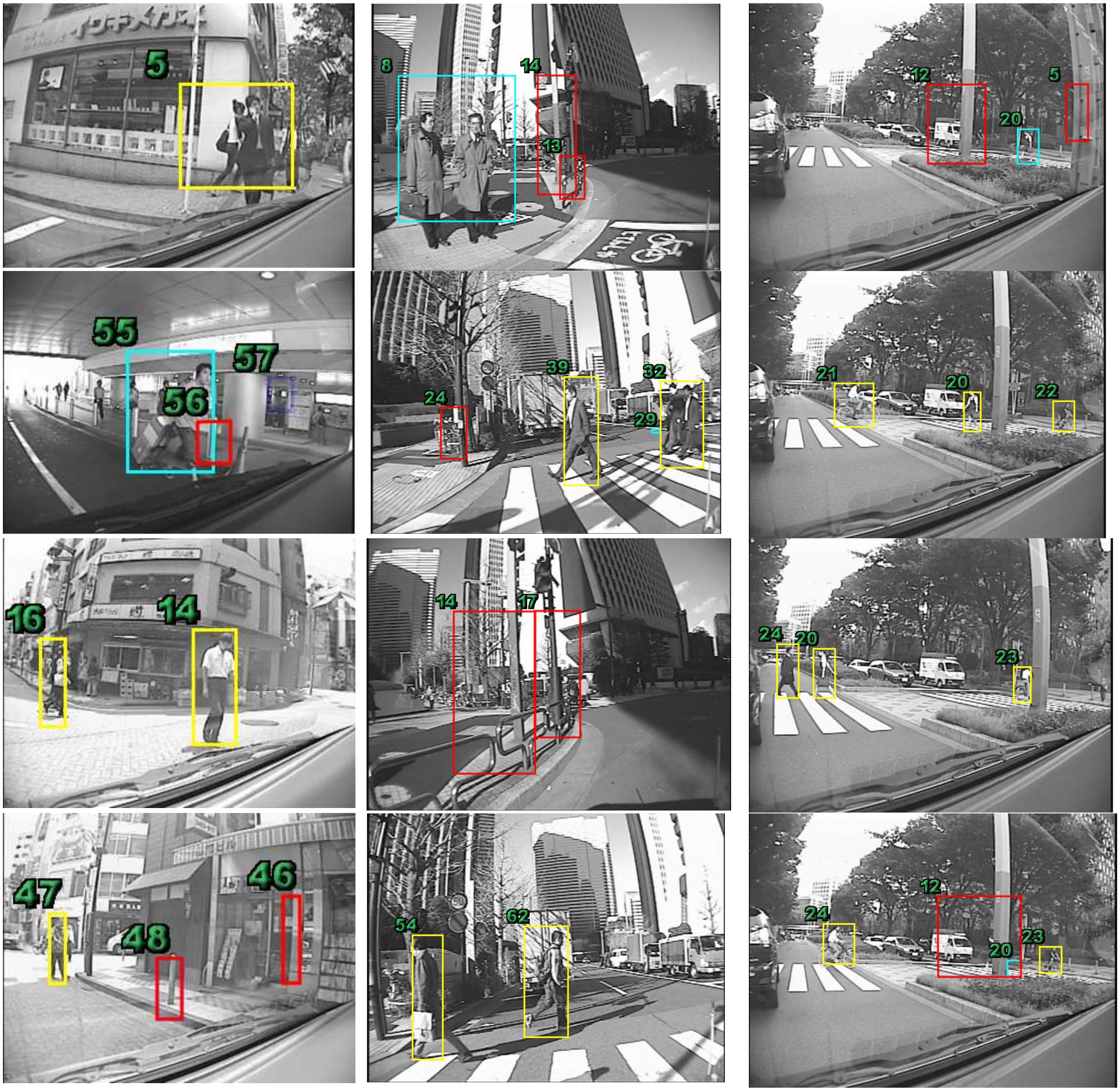
[11] Elzein, H. Lakshmanan, S. Watta, P. , “A motion and shape Based pedestrian Detection algorithm,” Proceedings. IEEE Intelligent Vehicles Symposium, 2003.

[12] Munder, S. Schnorr, C. Gavrila, D.M. , “Pedestrian Detection and Tracking Using a Mixture of View-Based Shape–Texture Models,” IEEE Transactions on Intelligent Transportation Systems, Volume: 9, Issue: 2, page(s): 333-343 , June 2008.

[13] S.Kamijo, M.Sakauchi, “Simultaneous Tracking of Pedestrians and Vehicles in Cluttered Images at Intersections,” 10th World Congress on ITS, Madrid, November.2003, CD-ROM

[14] Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Comm. ACM, Vol. 15*, pp. 11–15 (January, 1972).

[15] T. Joachims. Making large-scale svm learning practical. In B. Schlkopf, C. Burges, and A. Smola, editors, Advances in Kernel Methods - Support Vector Learning. The MIT Press, Cambridge, MA, USA, 1999.



(a) sequence 1 (b) sequence 2 (c) sequence 3

Figure 9. S-T MRF tracking result. Red bounding box-rejected false detections, yellow and sky colored bounding box -correct detections

(a) pedestrian rejected (b) one pedestrian undetected (c) poll object detected as pedestrian

Figure 10. Fail to handle by our pedestrian detection system