A New Vehicle Detection Method for Intelligent Transport Systems based on Scene-Specific Sliding Windows

SeungJong Noh[1], Moongu Jeon[1] and Daeyoung Shim[2]
[1]*School of Information and Communication, Gwangju Institute of Science and Technology, Cheomdangwagi-ro, Gwangju, Republic of Korea*
[2]*Department of Architecture, Kwandong University, Gangneung City, Republic of Korea*

Keywords: Intelligent Transport System, Vehicle Detection, Sliding Windows Technique.

Abstract: This paper presents a powerful vehicle detection technique employing a novel scene-specific sliding windows strategy. Unlike conventional approaches focusing on only appearance characteristics of vehicles, the proposed detection method also utilizes actually observable size-patterns of vehicles in a road. In our work, good data to train the size-patterns, i.e., size information of non-interacting moving-blobs are first collected based on the developed blob-level analysis technique. Then, a new region-wise sequential clustering algorithm is performed to train and maintain the size-pattern model, which is utilized to deform shapes of the sliding windows scene-specifically at each image position. All the proposed procedures operate full-automatically in real-time without any assumptions, and allow us to achieve more accurate and computationally efficient detection of vehicles in multiple scales and aspect-ratios. In the experiments on the real-time highway system, we found that performance of the proposed method is excellent in the aspects of detection accuracy and processing time.

1 INTRODUCTION

Although considerable progress has been made in object detection over last decade, real-time vehicle detection in surveillance system still remains a great challenging task. For this functionality to be utilized in the practical intelligent transport systems (ITS), we need to effectively handle the significant appearance variability of vehicles. The appearance variability is mainly caused by 1) severe intra-class variation, 2) unconstrained multiple viewpoints and 3) diverse vehicle subclasses such as sedan, truck, bus, etc. As a solution for these issues, sliding window based approaches are the most widely used because of its superior performance (VOC, 2007). In this technique, detection is treated as localized classification, where we apply a pre-trained classifier function to all image regions and then find locally optimized locations as detection results.

Numerous types of classifiers have been adopted for more elaborate object-class detection. For instance, boosted cascades of Harr-wavelet filters (Viola and Jones, 2001; Mikolajczyk et al., 2004; Tuzel et al., 2007; Brubaker et al., 2008), support vector machine with histogram of gradient features (Dalal, 2006; Dalal et al., 2006), and exemplar shape models (Stenger et al., 2006; Chum and Zisserman, 2007; Gavrila, 2007) were employed. Although these classifiers can successfully model intra-class variation, they still suffer from unexpected viewpoint changes (Su et al., 2009). To overcome this drawback, more sophisticated techniques were proposed such as methods utilizing a set of classifiers for each viewpoints (Thomas et al., 2006; Kushal et al., 2007; Liebelt et al., 2008), and applying 3D model structures (Savarese and FeiFei, 2007; Yan et al., 2007; Su et al., 2009). More recently, viewpoint-specified classifiers using implicit hierarchical boosting (Perrotton et al., 2011), and using a deformable part-based approach (Felzenszwalb et al., 2010) were proposed. However, these approaches are not also suitable for ITS applications, because they require high-resolution vehicle sub-images and high computational costs (Feris et al., 2011b).

In order to better deal with challenges in real road environments, Feris et al. proposed a detection method based on the scene-unique classier (Feris et al., 2011a; Feris et al., 2011b). In their approach, because all training samples are collected per camera-view semi-automatically, not only intra-class variation and but also geometric viewpoint information can be effectively handled. In addition, it over-

Noh S., Jeon M. and Shim D. (2013).
A New Vehicle Detection Method for Intelligent Transport Systems based on Scene-Specific Sliding Windows.
In *Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics*, pages 537-545
DOI: 10.5220/0004631505370545

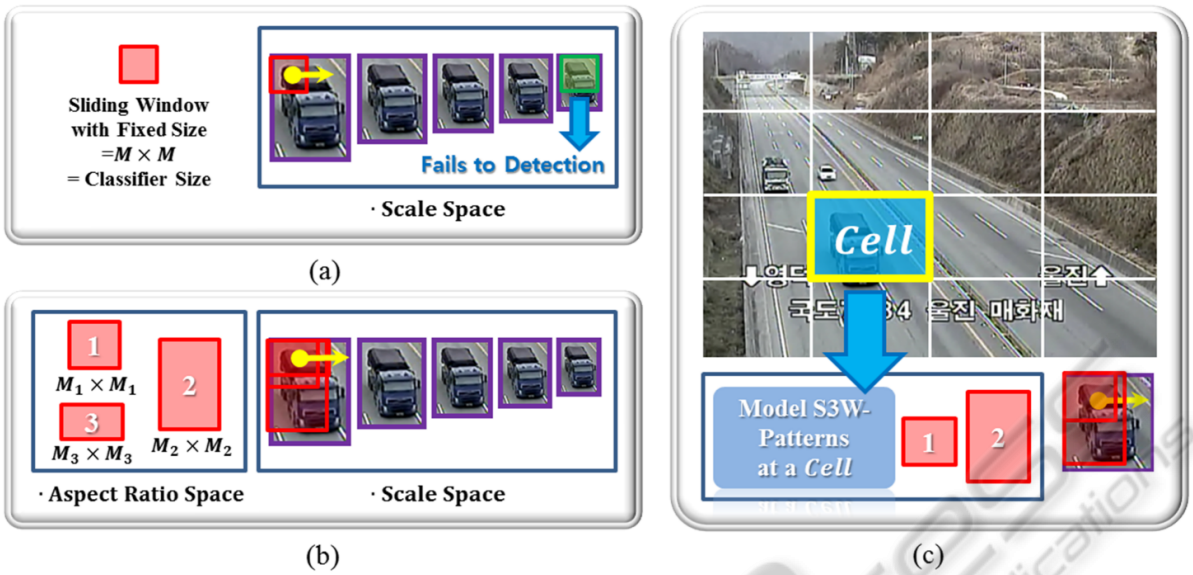


Figure 1: Figures for comparative explanation of the proposed and conventional sliding windows schemes: (a) The standard approach. A sliding window represented by a red-box is scanned for all scale space images. A vehicle having a different aspect-ratio from a classifier can cause inaccurate detection response as shown in a green box; (b) The exhaustive approach. In this method, all numbered red-boxes are exhaustively investigated to address the aspect-ratio issue; (c) The proposed scene-specific scheme. Only trained vehicle-size templates are taken into consideration for detection. White grids describe cell structure employed for S3W-pattern modeling.

comes performance degradation due to various vehicle subclasses by adopting a new search window deforming its shapes according to pre-defined aspect-ratios (Feris et al., 2011b). However, this method also has a limitation that user should manually adjust appropriate aspect-ratio ranges whenever a surveillance camera-scene is changed.

In our work, we propose a novel vehicle detection method based on the scene-specific sliding window, namely S3W. A key contribution of our work is the effective pattern modelling with S3W providing three fundamental advantages. First, it enables a system to learn and update actually observable vehicle aspect-ratio information without any user-settings. Based on this feature, Feris et al.'s method (Feris et al., 2011b) can be easily extended so that appropriate aspect-ratio ranges are automatically determined depending on given scene contexts. Second, S3W-based models allow us to achieve more compact operation. Unlike existing methods based on the conventional sliding window (Viola and Jones, 2001; Breuel, 1992; Keysers et al., 2007; Lampert et al., 2008; Blaschko and Lampert, 2008), the proposed detection method does not investigate detection responses for all scale-spaces (Adelson et al., 1984) since the constructed pattern models offer region-wised cues for real vehicle scale estimation. Finally, the proposed S3W-based image-scan strategy is greatly flexible in choosing classifiers. Ours can be combined with other conventional classifiers (Viola and Jones, 2001; Dalal, 2006; Stenger et al., 2006; Perrotton et al., 2011; Felzenszwalb et al., 2010) without any assumptions. Based on these advantages, the proposed detection technique accomplishes more accurate and computationally efficient performance under diverse real traffic-monitoring environments.

The rest of the paper is organized as follows. *Section* 2 describes employed classifier learning methodologies, and *Section* 3 explains details of the proposed vehicle detection procedures. Several experimental results are given to validate our approach in *Section* 4, and concluding remarks are presented in the last *Section* 5.

2 CLASSIFIER LEARNING

We employ the concept of the scene-unique classifier based detection (Feris et al., 2011b), where all training samples are extracted from specific camera scenes to effectively address intra-class variation and unconstrained viewpoints. To construct a training data set, whenever a frame is captured, we first collect foreground bounding-box images (Noh and Jeon, 2012) and arbitrary sized background images. Then, collected samples are normalized to have 48×48 size,

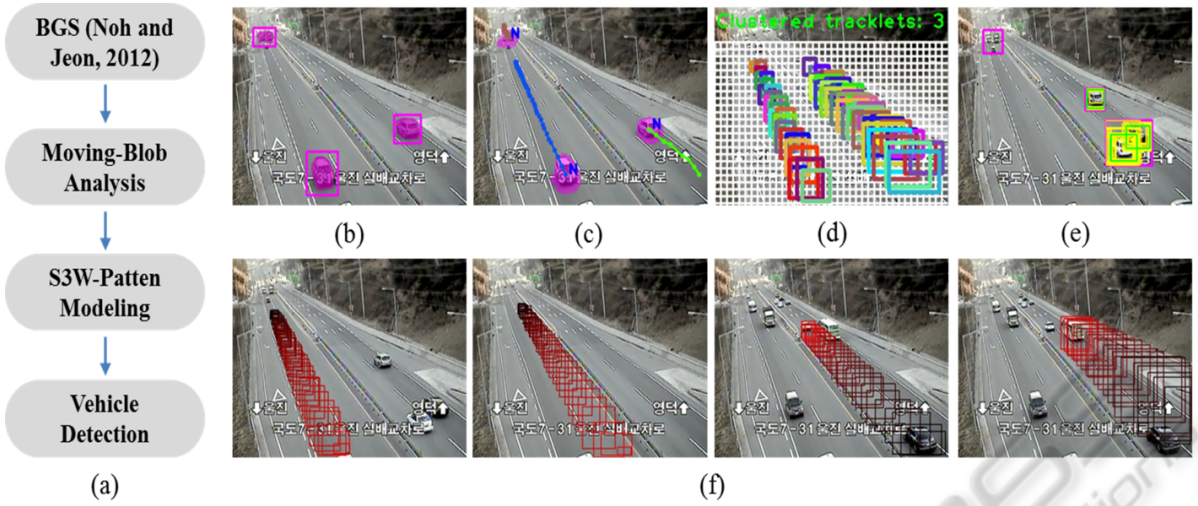


Figure 2: Figures to illustratively present overall procedures of the proposed detection method: (a) A flowchart of the proposed S3W-based approach; (b) Background subtraction results. Violet regions and boxes indicate found foreground regions and corresponding bounding boxes, respectively; (c) Three blob-level tracking histories. Each letter *N* implies that tracked blobs are in the normal state; (d) Constructed S3W-pattern models after three *Tracklet*s are collected. White grids represent applied cell structure and colored boxes show actually trained vehicle size templates at each cell; (e) S3W-based vehicle detection results. Violet, yellow and green boxes indicate foregrounds, initial detection responses and local optimized final detections, respectively; (f) Four good tracking examplars for S3W-modeling. Red boxes represent actually inferred vehicle sizes by selected *Tracklet* instances.

and then false positive samples are removed manually. In our work, these procedures are performed for 10,000 frames resulting 200∼400 initial positive, 100∼150 pruned positive and 1,000∼1,500 negative samples. The final vehicle classifier is produced through the HOG-descriptor based Linear-SVM (Dalal, 2006) in several minutes.

3 PROPOSED METHOD

Standard sliding window approaches have been chosen for many years for accurate object detection (Viola and Jones, 2001; Dalal, 2006; Boykov et al., 2001; Alexe and Ferrari, 2011). Here, a searching window shape is kept constantly, and sub-image hypothesises in all scale space images are considered as candidate detections. However, this methodology cannot ensure high detection accuracy because they do not consider diverse vehicle aspect-ratio information sufficiently (*Figure1(a)*). To overcome this drawback, a more exhaustive approach is proposed (Feris et al., 2011b), in which detection responses in not only scale space but also aspect-ratio space are exhaustively investigated (*Figure*1(b)). Although this has been shown to be quite effective in many applications, it still suffers from two main disadvantages. First, it is not clear how to set optimal aspect-ratio space. Because appropriate aspect-ratio ranges should be determined by expert's experience manually, significant errors can be caused if a camera-view is changed unexpectedly. Second, it is computationally inefficient since classifications should be performed at entire image positions for all scales and aspect-ratios.

The main reason why scale and aspect-ratio space have been adopted for detection is that most of methods focus on training only appearance features of a target object class (Viola and Jones, 2001; Dalal, 2006; Stenger et al., 2006; Perrotton et al., 2011; Felzenszwalb et al., 2010; Feris et al., 2011b). For instance, in the Feris et al.'s work (Feris et al., 2011b), the system should examine scale and aspect-ratio factors as many as possible during actual execution times since size information implied in the training set is completely lost in sample normalization procedure to produce a more compact classifier. We address this issue by proposing a new S3W-based image scan strategy. *Figure1(c)* describes the fundamental concept of the developed method. In our work, classifier evaluations are performed only for several promising vehicle-size patterns inferred from contexts of the given camera scene in real-time. Therefore the proposed approach doesn't require predefined scale and aspect-ratio data.

Our system utilizes three types of scene-specific data structures: 1) a scene-wised vehicle classifier, 2) a region-wised S3W-pattern model and 3) a pixel-wised background model. The classifier is created

(a) (b) (c) (d)

Figure 3: Figures for the blob-level tracking precesses: (a) Examples of isolated and interacting moving blobs in a scene. We collect only isolated blobs for the subsequent S3W-pattern modeling because bounding boxes for merged blobs cannot provide accurate vehicle size information as shown in '*Interacting Blobs*' figure; (b), (c) and (d) describe blob-matching results in the isolation, split and merging state, respectively. Yellow arrows show matching directions, and letters N, M and S imply that corresponding blobs are in N_{state}, M_{state} and S_{state}.

based on the semi-automated learning methodology offline (Feris et al., 2011b), and the background model is generated when a video stream is started based on multiple-cues during 200 initial frames (Noh and Jeon, 2012). On the other hand, the S3W-model is constructed and maintained based on the proposed modeling method in real-time.

In *Figure2(a)*, we give an overview of the proposed detection framework. Whenever a frame is captured after the background modeling stage, the proposed system first creates a foreground mask including all moving blobs in the scene (*Figure2(b)*, (Noh and Jeon, 2012)), and then carries out moving-blob analysis to obtain good-cues to train the S3W-model (*Figure2(f)*, *Subsection3.1*). Next, we conduct S3W-pattern modeling (*Figure2(d)*, *Subsection3.2*) and finally detect vehicles in multiple scales and aspect-ratios through the constructed S3W-model (*Figure2(e)*, *Subsection3.3*).

3.1 Moving Blob Analysis

In this work, we focus on the fact that isolated foreground blobs can provide useful information for S3W-pattern modeling. For instance, widths and height values of bounding boxes for isolated blobs shown in *Figure3(a)* can be effectively utilized to train observable vehicle sizes for a scene. However, interacting blobs, such as blobs merging, occluding each other or split into several parts should be excluded in the modeling stage because it causes inaccurate size data as presented in *Figure3(a)*.

We designed a practical blob-leveling tracking algorithm described in *Table1* to distinguish appropriate blobs and inappropriate blobs for learning. For more specific explanation, let a tracking history (see *Figure2(c)*) be formulized by a 2-tuple $Tracklet =< AUX,\ ST >$, where AUX implies a set of vector $aux_m = (x_m, y_m, w_m, h_m)$ and ST indicates states of $Tracklet$s. Here, x_m, y_m, w_m and h_m means a x coordinate, y coordinate, width and height of a bounding box for a blob, respectively. The variable ST can

Table 1: Blob-Level Tracking Algorithm.

1	**Input**: Blob matching tables
2	• FOR and $BACK$
4	**Output**: Tracking histories at time t
5	• $TH_t = \{Tracklet_{(t,h)} \mid h = 1,\ldots,NH_t\}$
6	**[History Update]**
7	**for** $i = 1,\ldots,NF_{t-1}$
8	**for** $j = 1,\ldots,NF_t$
9	**if** $FOR[i][j] = T$, $BACK[i][j] = T$
10	· Get h involving aux for $blob_{(t-1,i)}$
11	· Add aux for $blob(t,j)$ to $Tracklet_h$
12	· Set ST in $Tracklet_h$ to N_{state}
13	**else**
14	· $NH_t \leftarrow NH_t + 1$, $h \leftarrow NH_t$
15	· Create a new $Tracklet_h$
16	· Add aux for $blob(t,j)$ to $Tracklet_h$
17	**if** $FOR[i][j] = T$, $BACK[i][j] = F$
18	· Set ST in $Tracklet_h$ to M_{state}
19	**else if** $FOR[i][j] = F$, $BACK[i][j] = T$
20	· Set ST in $Tracklet_h$ to S_{state}
21	**else**
22	· Set ST in $Tracklet_h$ to N_{state}
23	**end for**
24	**end for**

be set to normal (N_{state}), split (S_{state}) and merged (M_{state}) states.

In our work, each $Tracklet$ is continuously updated to identify condition of moving blobs until it disappears from the scene. Because the purpose of the proposed tracking technique is not to track object robustly but to identify isolated blobs in a scene, we give up to track moving blobs in merging or splitting events. Instead, a new $Tracklet$ whose ST value is M_{state} or S_{state} is generated (*Table1*). Actions of moving blobs can be easily examined based on simple blob-matching procedures for sequential frames (*Table2*). For instance, blobs at time t are backward matched to blobs at time $t-1$ in the splitting event (*Figure3(c)*), blobs at time $t-1$ are forward matched to blobs at time t in the merging event (*Figure3(d)*),

Table 2: Blob-Matching Algorithm.

1	**Input**: Foreground sets at time t and $t-1$
2	• $F_{t-1} = \{blob_{(t,i)} \mid i = 1,\ldots,NF_{t-1}\}$
3	• $F_t = \{blob_{(t,i)} \mid i = 1,\ldots,NF_t\}$
4	**Output**: Blob matching results
5	• Forward matching table: FOR
6	• Backward matching table: $BACK$
7	**[Initialization]**
8	**for** $i = 1,\ldots,NF_{t-1}$
9	**for** $j = 1,\ldots,NF_t$
10	$\cdot\ FOR[i][j] \leftarrow F$
11	$\cdot\ BACK[i][j] \leftarrow F$
12	**end for**
13	**end for**
14	**[Blob Matching]**
15	**for** $i = 1,\ldots,NF_{t-1}$
16	**for** $j = 1,\ldots,NF_t$
17	// Step1: Forward matching
18	**if** $blob_{(t-1,i)}$ is matched to $blob_{(t,j)}$
19	$\cdot\ FOR[i][j] \leftarrow T$
20	// Step2: Backward matching
21	**if** $blob_{(t,j)}$ is matched to $blob_{(t-1,i)}$
22	$\cdot\ BACK[i][j] \leftarrow T$
23	**end for**
24	**end for**

and blobs at time $t-1$ and t are bilaterally matched if they are isolated (*Figure3(b)*).

To perform the blob-matching method, we first operate the classical connect component algorithm (Dillencourt et al., 1992) on a given foreground mask to label each moving region with $blob_{(t,j)}$, where t implies time that the input frame is captured and j represents a blob index (*Line1-3 in Table2*). Then, blob matching procedures are carried out to produce result tables FOR and $BACK$ (*Line14-24 in Table2*). *KLT* feature-flow based approach (Fusier et al., 2007) is employed to evaluate appearance similarity between two blobs (*Line18, 21 in Table2*). Next, blobs in F_{t-1} are associated with blobs in F_t based on the blob-level tracking algorithm given in *Table1*, and finally *Tracklet*s on a road are add, removed or updated according to the corresponding matching table entries (*Line6-24 in Table1*). These procedures prevent inadequate *Tracklet*s which contain interacting blobs from becoming the normal state. *Tracklet*s with N_{state} are selected as good examplars to train the S3W-model when they are removed if the following condition is satisfied:

$$movedDist \geq min(W_{img}, H_{img}) \times 0.08 \qquad (1)$$

where $movedDist$, W_{img} and H_{img} means a total moved distance of a $Tracklet$, image width and image height, respectively. In *Figure1(f)*, we give several examples of good $Tracklet$ instances.

3.2 S3W-pattern Modeling

In the proposed method, the S3W-model is constructed and maintained based a cell-wised sequential clustering algorithm as shown in *Figure2(d)*, where the cell is an employ grid structure for more compact operation. Let (cx,cy) denotes a cell-coordinate in the adopted grid-system, and W_{cell} and H_{cell} represents the number of the cells in horizontal axis and vertical axis of an image sequence, respectively. In addition, assume that a S3W-pattern model(SPM) is defined as a set of data-structure $SP_{(cx,cy)} = \{cluster^k_{(cx,cy)} \mid k = 1,\ldots,NC_{(cx,cy)}\}$, in which $cluster^k_{(cx,cy)}$ implies the kth involved cluster and $NC_{(cx,cy)}$ means the number of clusters. For each cluster, we continuously update three statistics, i.e., mean vector, covariance matrix and the maximum negative run-length($MNRL$) (Kim et al., 2004). The first two factors are applied to measure Mahalanobis distance(MD) (Mahalanobis and Chandra, 1936) between an input sample and a cluster (*Line14 in Table3*), and the final factor is utilized to clear non-essential clusters.

For each aux_m in a given $Tracklet$, we first calculate its corresponding cell-coordinate (cx,cy) (*Line10-11 in Table3*), and then perform the cluster matching process to find the best-matched cluster (*Line12-15 in Table3*). If there exists a matched cluster, we update its statistics based on the input sample (*Line17-18 in Table3*). Otherwise, we generate a new cluster containing an input vector, an identity matrix and a zero-value as the initial mean vector, covariance matrix and $MNRL$ value, respectively (*Line19-21 in Table3*). Here, because an input sample s_m is defined as (w_m, h_m), mean vectors are also set to be two-dimensional. Whenever 500 aux_m instances are trained, non-essential clusters whose $MNRL$ is larger than 500×0.1 are removed.

3.3 S3W-based Vehicle Detection

3.3.1 Initial Detection

To localize vehicles in unconstrained scales and aspect-ratios, we deform a shape of the sliding window at each image position depending on scene-context. We start more detailed explanation with defining several notations. First, let $ForeBox$, (bx,by) and (bx_{in},by_{in}) means a foreground bounding box, left-top coordinate of the $ForeBox$ in an input frame, left-top coordinate of a sliding window in the $ForeBox$, respectively (*Figure4*). In addition, assume that $\left(w^k_{(cx,cy)}, h^k_{(cx,cy)}\right)$ indicates a mean vector of the k_{th} cluster in a $SP_{(cx,cy)}$ (*Line4 in Table3*).

Table 3: S3W-Pattern Modeling.

1	**Input**: A set AUX in a verified $Tracklet$
2	• $AUX = \{aux_1, \ldots, aux_{NA}\}$
3	**Output**: A cell-wised S3W-Pattern Model
4	• $SPM = \{SP_{(cx,cy)} \mid (cx,cy) \in Image\}$
5	**[Initialization]**
6	• Cell size factor $f_{cell} \leftarrow 10$
7	• Cluster size factor $f_{cluster} \leftarrow 15$
8	**[Cell-Wised Sequential Clustering]**
9	**for** $m = 1, \ldots, NA$
10	// Step1: Getting cell-coordinate for aux_m
11	· $cx \leftarrow \lfloor \frac{x_m f_{cell}}{W_{img}} \rfloor$, $cy \leftarrow \lfloor \frac{y_m f_{cell}}{H_{img}} \rfloor$
12	// Step2: Cluster matching
13	· $s_m \leftarrow (w_m, h_m)$, $idx \leftarrow null$
14	· $id \leftarrow \underset{k}{argmin}\left(MD\left(cluster^k_{(cx,cy)}, s_m\right)\right)$
15	· $matchDist \leftarrow MD(cluster^{id}_{(cx,cy)}, s_m)$
16	// Step3: S3W-model Update
17	· **if** $id \neq null$, $matchDist \geq f_{cluster}$
18	i) Update $cluster^{id}_{(cx,cy)}$ based on s_m
19	· **else**
20	i) $NC_{(cx,cy)} \leftarrow NC_{(cx,cy)} + 1$
21	ii) Generate a new $cluster^{NC}_{(cx,cy)}$ for s_m
22	**end for**

For each window coordinate (bx_{in}, by_{in}) in a foreground bounding box, we first calculate the corresponding cell coordinate (cx,cy) through the following equations:

$$cx = \left\lfloor \frac{(bx_{in} + bx) f_{cell}}{W_{img}} \right\rfloor \quad (2)$$

$$cy = \left\lfloor \frac{(by_{in} + by) f_{cell}}{H_{img}} \right\rfloor \quad (3)$$

where f_{cell} is the applied cell size factor in *Table3*. Next, we produce subimages with size $(w^k_{(cx,cy)}, h^k_{(cx,cy)})$ for all clusters in $SP_{(cx,cy)}$. Each subimage is resized to same size with the trained vehicle classifier, and then classification is conducted. Initial detection responses shown in *Figure2(e)* are generated by performing these procedures for all possible *ForeBox* positions. In this work, we represent a detection response for a window whose location is (x,y) and size is (w,h) as a 2-tuple $R =< r, score >$, where r is a vector $(x,y,w,h)^T$ and $score$ is corresponding classification score.

3.3.2 Non-maximum Suppression

In general, sliding windows schemes cause unnecessary multiple overlapped detections. To overcome this drawback, we employ Dalal's non-maximum suppression technique (Dalal, 2006). In the Dalal's work,

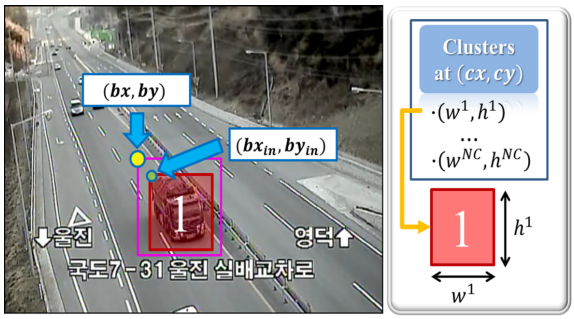


Figure 4: Figures for the S3W-based detection: The violet and red box represents a foreground bounding box and a sliding window, respectively. (cx,cy) indicates a cell coordinate for an image coordinate $(bx + bx_{in}, by + by_{in})$.

non-maximum suppression is treated as a mean-shift algorithm based mode seeking problem. More specifically, each initial detection response $R_i =< r_i, score_i >$ $(i = 1, \ldots, M)$ is continuously moved at each iteration n until it converges to the locally optimized mode based on the following equations:

$$r_i^{n+1} = \left(\sum_i^M w_i(r_i^n) \cdot r_i \right) \quad (4)$$

$$w_i(r) = \frac{t(score_i) exp\left(-ED^2(r, r_i)/2\right)}{\sum_i^M t(score_i) exp\left(-ED^2(r, r_i)/2\right)} \quad (5)$$

$$t(score) = \begin{cases} score & \text{if } score \geq 0 \\ 0 & \text{if } score < 0 \end{cases} \quad (6)$$

where, ED means Euclidean distance between two vectors. Note that a 4×4 identity matrix is employed as the bandwidth matrix because we don't take any assumptions on the vehicle size patterns. In *Figure2(e)*, we describe several examples of the local optimized final detection results by green boxes.

4 EVALUATION

We have conducted systematic experiments on four data sets produced from different view of fixed camera scenes (*Figure5(a)*). A set consists of 10,000 training and 5,000 test sequences captured under real-road environments, and involves 50 ground-truth detections located in randomly selected test sequences. For each test scene, we first train a 48×48 size of vehicle classifier using positive and negative samples extracted from the training sequences offline. The S3W-pattern modeling is operated during the first 10,000 test sequences, and then performance evaluation is performed for the remaining 5,000 frames. More details of the applied scene modeling processes are given in *Figure5*(a).

To determine whether a detection is correct or not, we utilized the following criteria:

$$\begin{cases} \textit{True Detection} & \text{if } F_p \geq 0 \\ \textit{False Detection} & \text{if } F_p < 0 \end{cases} \quad (7)$$

where F_p implies pixel-level F-score to quantitatively measure quality of a detection response. We give the definition of F_p in the below equation:

$$F = \frac{2TP}{2TP + FN + FP} \quad (8)$$

where TP, FP and FN are true positives, false positives and false negative positives, respectively (*Figure5(b)*). Based on these measures, true detection rate(TDR) and detection hit rate(DHR) for the entire data sets were estimated. In addition, we also calculated average detection quality(ADQ) for all true hit detections to clearly demonstrate the superior performance of the proposed S3W-based technique. The final performance score(FPS) of detection algorithms were defined as the harmonic mean of TDR, DHR and ADQ.

We present qualitative and quantitative results on the original sliding windows based approach (Viola and Jones, 2001; Dalal, 2006) and the proposed detection method in *Figure5*(c) and *Figure5*(d). For both techniques, same implementation details were applied except for classifier scan strategies. The exhaustive scale and aspect-ratio space based approach (Feris et al., 2011b) was excluded in our experiments because we considered only full-automatic sliding windows schemes.

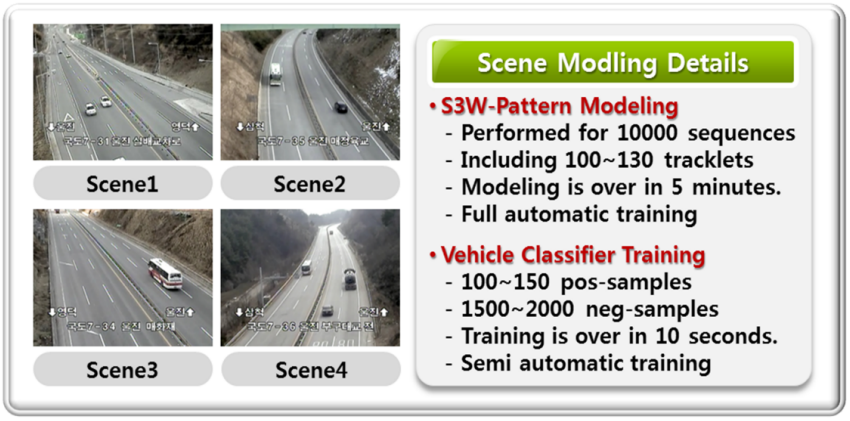
From the qualitative results, first we can see that the proposed method achieves greatly high detection accuracy based on the scene-specific search window deforming its shape according to appearance of vehicles in scenes. On the other hand, because only scale-spaces are investigated for detection, the original scheme fails to localize a target when its aspect-ratio is largely different from that of the classifier. It is also observed that the original approach show a slightly better detection score than ours in the *scene*4 (*Figure5*(e)) in which insufficient vehicle size patterns were provided for the S3W-modeling during initial 10,000 frames. However we can conclude that the proposed technique outperforms the original one because it ensures much more excellent performance(FPS) with smaller numbers of classifier evaluations($\#CE$) on average (*Figure5*(d)).

5 CONCLUSIONS

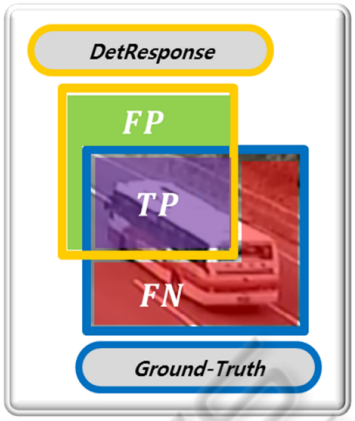
In this work, we have presented a novel vehicle detection method based on a scene-specific sliding window(S3W) technique. Our method does not exhaustively investigate sub-image hypothesises for all possible scales and aspect-ratios, but classify actually observable vehicle size patterns only. Whenever a frame is captured, the proposed system first creates a foreground mask including all moving blobs in the scene (Noh and Jeon, 2012), and then conducts moving-blob analysis to get good-cues to train S3W-patterns. Next, procedures for S3W-modeling are carried out, and finally multiple scales and aspect-ratios of vehicles are localized precisely. Experimental results have demonstrated superior performance to the conventional method. However, we found that the proposed method can cause some false detections in the severely curved regions of the road. In such case, because intra-class variations due to viewpoint changes become much larger, not only size-patterns but also classifiers should be trained scene-specifically. To accomplish this goal, we need to develop moving direction based classifier learning methodologies.

ACKNOWLEDGEMENTS

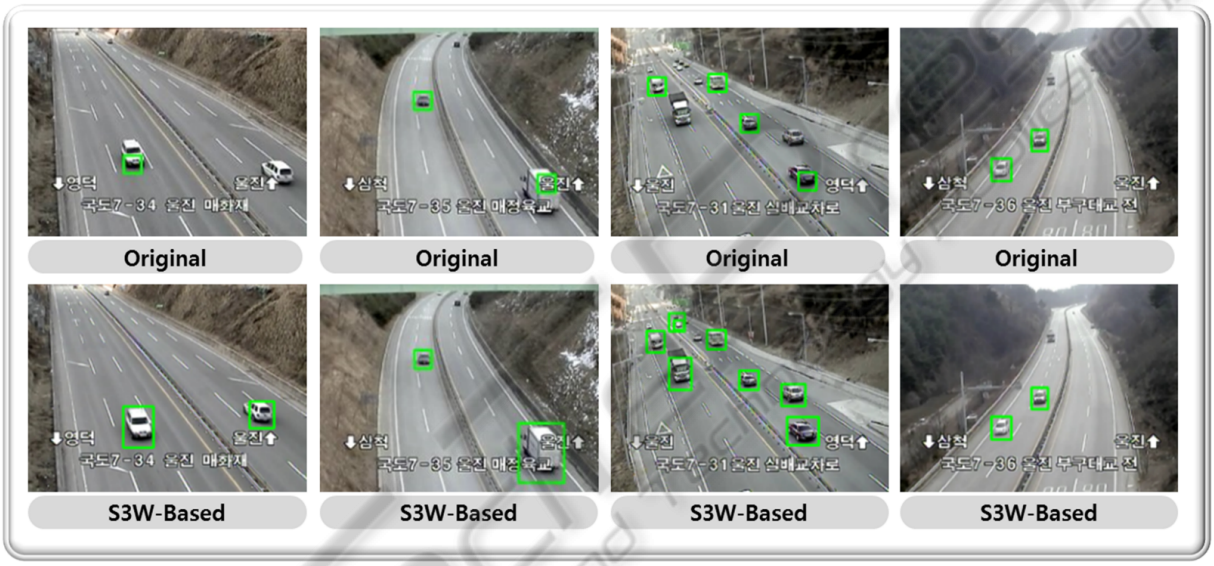
This research was supported by the Pioneer Research Center Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT & Future Planning (grant number 2012-0009462), and by Ministry of Culture, Sports and Tourism(MCST) and Korea Creative Content Agency(KOCCA) in the Culture Technology(CT) Research & Development Program 2012.



(a)



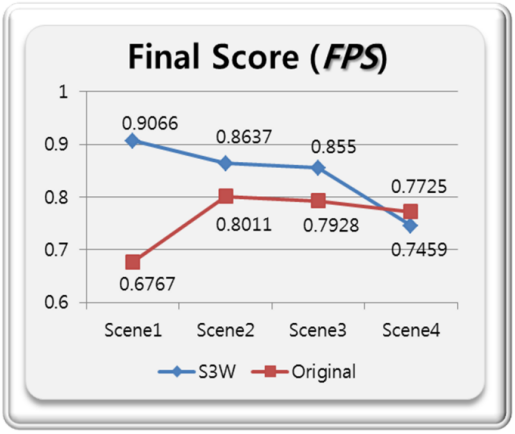
(b)



(c)

		TDR	DHR	ADQ	FPS	#CE	#AC
Scene1	S3W-Based	0.9787	0.9200	0.8332	0.9066	1,032,363	5.75
	Original	0.6957	0.6200	0.7669	0.6767	1,336,203	
Scene2	S3W-Based	0.9534	0.8200	0.8298	0.8637	221,739	4.96
	Original	0.8809	0.7400	0.7947	0.8011	437,347	
Scene3	S3W-Based	0.9722	0.7600	0.8569	0.8550	1,677,361	5.50
	Original	0.9714	0.6800	0.7789	0.7928	2,002,169	
Scene4	S3W-Based	0.8333	0.6400	0.7943	0.7459	130,763	4.97
	Original	0.8717	0.6800	0.7899	0.7725	215,005	
Average	S4W-Based	0.9344	0.78500	0.8285	0.8428	765,557	5.29
	Original	0.8549	0.6800	0.7826	0.7607	997,681	

(d)



(e)

Figure 5: Evaluation settings and results: (a) Tested four road scenes and details of the applied scene modeling procedures; (b) The concept of the utilized detection quality measure. *FP*, *TP* and *FN* means false positives, true positives and false negatives, respectively; (c) Qualitative detection results; (d) Quantitative detection results. *TDR*, *DHR*, *ADQ*, *FPS*, *#CE* and *#AC* indicates the true detection rate, detection hit rate, average detection quality, final performance score, number of classifier evaluations, and average number of clusters for a scene, respectively; (e) Final performance scores for each evaluated camera scene.

REFERENCES

(2007). PASCAL visual object classes challenge.

Adelson, E., Anderson, C., Bergen, J., Burt, P., and Ogden, J. (1984). Pyramid methods in image processing. *Jouranl of RCA-Engineer*, 29(6):33–41.

Alexe, T. D. D. and Ferrari, V. (2011). Learning object classes with generic knowledge. *Journal of IJCV*.

Blaschko, M. and Lampert, C. (2008). Learning to localize objects with structured output regression. In *Proc. ECCV*.

Boykov, Y., Veksler, O., and Zebih, R. (2001). Fast approximate energy via graph cuts. *Journal of PAMI*, 22(11):1222–1239.

Breuel, T. (1992). Fast recognition using adaptive subdivisions of transformation space. In *Proc. CVPR*.

Brubaker, S., Wu, J., J.Sun, Mullin, M., and Rehg, J. (2008). On the design of cascades of boosted ensembles for face detection. *Journal of IJCV*, 77(1-3):65–86.

Chum, O. and Zisserman, A. (2007). An exemplar model for learning object classes. In *Proc. CVPR*.

Dalal, N. (2006). Finding people in images and videos. PhD thesis, Institut National Polytechnique de Grenoble.

Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *Proc. ECCV*.

Dillencourt, M., Samet, H., and Tamminen, M. (1992). A general approach to connected-component labeling for arbitary image representations. *Journal of ACM*, 39(2):253–280.

Felzenszwalb, P., Girshick, R., and Allester, D. (2010). Cascade object detection with deformable part models. In *Proc. CVPR*.

Feris, R., Perrerson, J., Siddiquie, B., Brown, L., and Pankanti, S. (2011a). Large-scale vehicle detection in challenging urban surveillance environments. In *Proc. WACV*.

Feris, R., Siddiquie, B., and Zhai, Y. (2011b). Attribute-based vehicle search in crowded surveillance videos. In *Proc. ICMR*.

Fusier, F., Valentin, V., Bremond, F., Thonnat, M., Borg, M., Thirde, D., and Ferryman, J. (2007). Video understanding for complex activity recognition. *Journal of MVA*, 18:167–188.

Gavrila, D. (2007). A bayesian exemplar-based approach to hierarchical shape matching. *Journal of PAMI*, 29(8):1408–1421.

Keysers, D., Deselaers, T., and Breuel, T. (2007). Optimal geometric matching for patch-based object detection. *Journal of ELCVIA*, 6(1):44–54.

Kim, K., Chalidabhongse, T., Harwood, D., and Davis, L. (2004). Background modeling and sebtraction by codebook construction. In *Proc. ICIP*.

Kushal, A., Schmid, C., and Ponce, J. (2007). Flexible object models for category-level 3d object recognition. In *Proc. ICCV*.

Lampert, C., Blaschko, M., and Hofmann, T. (2008). Beyond sliding windows: object localization by efficient subwindow search. In *Proc. CVPR*.

Liebelt, J., Schmid, C., and Schertler, K. (2008). Viewpoint-independent object class detection using 3d feature maps. In *Proc. CVPR*.

Mahalanobis and Chandra, P. (1936). On the general distance in statistics. In *Proc. NISI*.

Mikolajczyk, K., Schmid, C., and Zisserman, A. (2004). Human detection based on a probabilistic assembly of robust part detectors. In *Proc. ECCV*.

Noh, S. and Jeon, M. (2012). A new framework for background subtraction using multiple cues. In *Proc. ACCV*.

Perrotton, X., Sturzel, M., and Roux, M. (2011). Implicit hierarchical boosting for multi-view object detection. In *Proc. CVPR*.

Savarese, S. and FeiFei, L. (2007). 3d generic object categorization, localization and pose estimation. In *Proc. CVPR*.

Stenger, B., Tayananthan, A., Torr, P. H. S., and Cipolla, R. (2006). Model-based hand tracking using a hierarchical baysian filter. *Journal of PAMI*, 28(9):1372–1385.

Su, H., Sun, M., FeiFei, L., and Savarese, S. (2009). Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *Proc. ICCV*.

Thomas, A., Ferrari, V., Leibe, B., Tuyelaars, T., Schiele, B., and Gool, L. (2006). Toward multi-view object class detection. In *Proc. CVPR*.

Tuzel, O., Porikli, F., and Meer, P. (2007). Human detection via classification on riemannian manifolds. In *Proc. CVPR*.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*.

Yan, P., Khan, S., and Shah, M. (2007). 3d model based object class detection in an arbitrary views. In *Proc. ICCV*.