MULTIPLE-KERNEL ADAPTIVE SEGMENTATION AND TRACKING (MAST) FOR ROBUST OBJECT TRACKING

Zheng Tang, Jenq-Neng Hwang
Department of Electrical Engineering
University of Washington, Box 352500
Seattle, WA 98195, USA
{zhtang, hwang}@uw.edu

Yen-Shuo Lin, Jen-Hui Chuang
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
linys.cs00g@nctu.edu.tw, jchuang@cs.nctu.edu.tw

ABSTRACT

In a video surveillance system with static cameras, object segmentation often fails when part of the object has similar color with the background, resulting in poor performance of the subsequent object tracking. Multiple kernels have been utilized in object tracking to deal with occlusion, but the performance still highly depends on segmentation. This paper presents an innovative system, named Multiple-kernel Adaptive Segmentation and Tracking (MAST), which dynamically controls the decision thresholds of background subtraction and shadow removal around the adaptive kernel regions based on the preliminary tracking results. Then the objects are tracked for the second time according to the adaptively segmented foreground. Evaluations of both segmentation and tracking on benchmark datasets and our own recorded video sequences demonstrate that the proposed method can successfully track objects in similar-color background and/or shadow areas with favorable segmentation performance.

Index Terms— Adaptive Segmentation, Object Tracking, Multiple Kernels, Background Subtraction, Shadow Removal

1. INTRODUCTION

Because of the low cost of cameras, they are widely used in the world. To effectively assist people to deal with huge amount of videos, more and more intelligent video surveillance systems are getting deployed. Object tracking is a key issue in many surveillance applications. Since most object tracking approaches are based on extracting foreground objects, the failure in foreground object segmentation can severely degrade the performance of tracking. It usually occurs in the following scenario: when an object enters into camera's field of view in which some parts of the body have similar color with the modeled background, object segmentation can easily fail because the corresponding foreground regions are likely to be merged into background or recognized as shadow (the problem of object merging). These will eventually lead to a dead loop that results in increasing errors in tracking. In this paper, a novel object segmentation and tracking system is proposed to deal with this problem. The general idea of the proposed Multiple-kernel Adaptive Segmentation and Tracking (MAST) is to preserve more foreground in segmentation based on region-level similarity between the input image and background, which is calculated using feedback from the preliminary tracking results. Moreover, the optimized segmentation result is further used for updated tracking to improve its accuracy. Our main contributions are highlighted as: (i) The feedback loop originated from preliminary tracking results is used to help preserve segmented foreground when object(s) share color similarity or chromaticity similarity with background. (ii) The region-level feedback based on multiple kernels is used to reduce noise, which can be derived from the kernel histograms built in multiple kernels tracking. (iii) Effective penalty computation is introduced for shadow removal based on the distance between chromaticity histograms of foreground and background.

The rest of this paper proceeds by introducing related work in Section 2. Section 3 presents the proposed MAST approach for robust object tracking. Experimental results and discussions are covered in Section 4. In the end, Section 5 concludes this paper.

2. RELATED WORK

The performance of many object tracking approaches [1]-[4] is highly dependent on foreground segmentation mask. Zhao et al. [1] propose an object detection and tracking system based on image likelihood model. They use head detection and 3D human models to detect objects from foreground segmentation mask. Chu et al. [2]-[3] generalize the constrained multiple-kernel (CMK) tracking that employs projected gradient constrained search to find the best match of the tracked target with occlusion. They further improve the computational efficiency of this method by combining CMK tracking with Kalman filtering, whose prediction of states is made based on the position, velocity and size of the foreground blobs [4]. If the segmentation stage fails, the subsequent tracking of objects across frames will also be adversely affected.

There are many other works concerning adaptation in object segmentation. The pixel-based adaptive segmenter (PBAS) [5] is an example of foreground segmentation using feedback from pixel-level background dynamics. The SuBSENSE method [6], as an improvement to PBAS, allows increased local sensitivity, especially for regions with intermittent dynamic variations. None of these methods takes advantage of the feedback from preliminary tracking to further improve the segmentation and update tracking performance. Furthermore, the threshold decision mechanisms used in these methods are all in pixel level, while our method is in region level. Finally, our proposed method is the only one taking into account the adaptation for shadow removal, since shadow is also a key factor in object segmentation.

3. ADAPTIVE SEGMENTATION FOR TRACKING

Figure 1 shows the overview flow chart of our proposed MAST scheme, where we assume that the modeled background is known (or can be pre-estimated) in a static camera surveillance system. The GMM background model is adopted here, but it can be replaced by other background models. Each module of the flow chart will be elaborated next.

3.1. General Segmentation and Tracking

To extract foreground objects, we perform background subtraction first. Otsu's thresholding method [7] is adopted to dynamically determine the global threshold for each color channel. The thresholding results are combined through intersection. Otsu's method is known for its simplicity and computation efficiency.

Afterwards, we remove the detected shadows in foreground mask based on YCbCr color space using the shadow indicator calculated as,

$$\mathrm{SInd}(x,y)=\begin{cases}1, & (\alpha \le \mathrm{Y^I}(x,y)\,/\,\mathrm{Y^B}(x,y) \le \beta) \\ & \wedge\left(\left|\mathrm{Cb^I}(x,y)\text{-}\mathrm{Cb^B}(x,y)\right| \le \tau_{\mathrm{Cb}}\right) \\ & \wedge\left(\left|\mathrm{Cr^I}(x,y)\text{-}\mathrm{Cr^B}(x,y)\right| \le \tau_{\mathrm{Cr}}\right) \\ 0, & \text{otherwise}\end{cases}, \quad (1)$$

where the superscript *I* denotes the pixel in input image, and *B* for background pixel. α, β, τ_{Cb}, and τ_{Cr} are the parameters set for different color channels. Since these threshold values are set for the whole image, errors are likely to occur in some regions. For instance, if the chromaticity (represented by the Cb and Cr channels) of a portion of the object is similar to the background, they will be easily recognized as shadow. This kind of methods based on separation of chromaticity and brightness has been widely used in many segmentation approaches, such as [8]-[9], for its simplicity and effectiveness. To remove small noise and connect broken parts in foreground mask, morphological operations such as closing, opening, filling gaps are further applied after segmentation. Then, the foreground mask is used as an input to the tracking stage.

The adopted object tracking algorithm, which combines Kalman filtering and CMK tracking [4], can achieve robust performance against occlusion and enjoy the benefit of reusing kernel histograms. Multiple kernels are used to represent several parts of each object, so that when one or some of the kernels are occluded, we can put larger weights to other observable kernels and link all the kernels based on some predefined constraints [10]-[12]. The idea here is to perform Kalman filter prediction first for each object that has been tracked in the previous frame, and then determine whether the object is under occlusion based on similarity of kernel histograms. If not, we choose the segmentation result as the measurement to continue the Kalman filtering. Otherwise, CMK tracking is used to get multiple measurements which will be handled by probabilistic data association. The tracking results are represented as bounding boxes around the tracked objects at their positions, as is shown in Figure 2.

To ensure robust tracking performance, some constraints are imposed to prevent sudden changes of the bounding boxes caused by segmentation failure. The constraints include limited size-change ratio, width-change ratio, and height-change ratio of foreground blobs. This step is important for adaptive segmentation, because it allows some parts of object(s) merged into background (either due to similar color with background or due to shadowing) to be temporarily included in the preliminary tracking result. Hence, some of the missing parts could be recovered through a feedback loop.

3.2. Similarity Computation and Feedback Loop

The key innovation behind our proposed scheme is how we use the feedback loop to optimize segmentation results by adaptively controlling the segmentation thresholds. At first, multiple (elliptical shaped) kernels are generated inside each tracked bounding box based on predefined spatial layout. According to the investigation in [2]-[4],[10] and our own experiments, the spatial layouts of two kernels and four kernels (see Figure 3) produce better results. In all of our experiments, we use only the two kernels for human objects.

The next step is to construct two separate kernel histograms for each kernel. Using the same color space as in segmentation, the YCbCr histogram is first built to measure color similarity between current frame and background for adaptive background subtraction. The 2nd kernel histogram is constructed by using only the Cb and Cr channels to measure the chromaticity similarity for adaptive shadow removal. We use Gaussian kernel function as the weighting for building the kernel histograms, as given in (2),

$$w=\frac{1}{2\pi\sigma_x\sigma_y}e^{-\frac{(x-x_c)}{2\sigma_x^2}}e^{-\frac{(y-y_c)}{2\sigma_y^2}}, \quad (2)$$

where x_c and y_c denote the center coordinate of the kernel, while σ_x and σ_y are set as half of the width and height of the kernel respectively. Kernel histogram is used because the area of the object usually only covers the region around the center of the kernel, which is the part we want to emphasize.

The color similarity and chromaticity similarity are computed by the reciprocals of Bhattacharyya distances [10] between the corresponding kernel histograms, i.e.,

$$colorSimi=\frac{1}{\sum\sqrt{hist^{I}_{YCbCr}(x,y)\cdot hist^{B}_{YCbCr}(x,y)}}, \quad (3)$$

$$chromSimi=\frac{1}{\sum\sqrt{hist^{I}_{CbCr}(x,y)\cdot hist^{B}_{CbCr}(x,y)}}, \quad (4)$$

where both of them are normalized to 0 to 1.

Under the consideration of computation efficiency and smoothness of segmentation, we design a fuzzy Gaussian penalty weighting (pw) function (5), as shown in Figure 4, for adaptively changing the thresholds in segmentation,

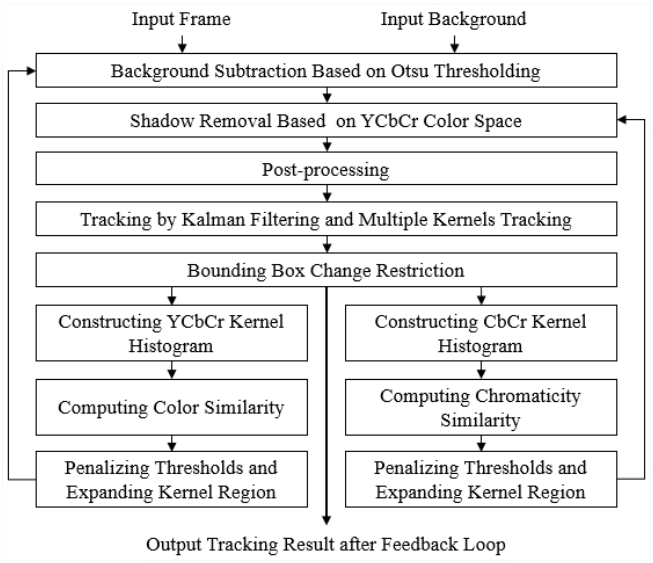
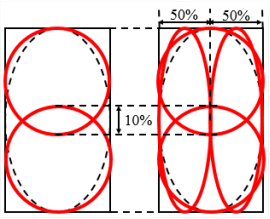


Figure 1. Overview flow chart of the proposed system.

(a) (b) (c)

Figure 2. Segmentation and tracking results for the video sequence *backdoor* at frame #1840 in the CVPR 2014 Change Detection dataset [13]. (a) Background subtraction (white for extracted foreground and black for background). (b) Shadow detection and removal (gray for detected shadow to be removed). (c) Tracking result (bounding boxes in red).



(a) (b)

Figure 3. Spatial layouts of multiple kernels inside the bounding box. Black solid rectangles are bounding boxes and dashed ellipses represent the bounded objects. Red ellipses indicate the locations of kernels. (a) Two kernels. (b) Four kernels.

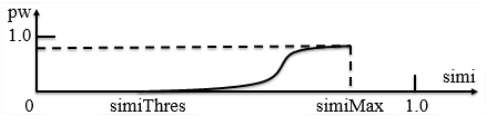


Figure 4. The fuzzy Gaussian penalty weighting (pw) function for adaptively changing the thresholds in segmentation.

$$pw = \begin{cases} e^{-\frac{(simi-1.0)^2}{2\cdot[(1.0-simiThres)/3]^2}}, & simiThres \le simi < simiMax \\ 0, & \text{otherwise} \end{cases}, \quad (5)$$

where *simi* corresponds to the color similarity or chromaticity similarity. The *simiThres* is set as a threshold for the corresponding similarity. If the similarity is lower than this threshold, the general segmentation is considered to be successful, and thus adaptive segmentation will not be triggered. Otherwise, the Otsu's threshold used in background subtraction or the chromaticity thresholds τ_{Cb} and τ_{Cr} in shadow removal will be penalized by multiplying $(1 - pw)$. The adaptive segmentation is then conducted locally inside the corresponding kernel region with the new threshold values. Besides failure in segmentation, the tracking is also likely to fail when the bounding box shifts to a background area, therefore we need to set an upper limit for the similarity (*simiMax*). When the *simi* is beyond this limit, the adaptive segmentation will not be conducted either, and thus we can avoid propagation of tracking errors. Moreover, to recover possible foreground outside the kernel, the kernel region to be re-segmented is expanded by a factor of $(1 + pw/2)$. In summary, higher color/chromaticity similarity will result in smaller thresholds for segmentation inside a further expanded kernel region, and thus more pixels around the tracked objects will be classified as foreground. The final segmentation result is generated by a union combination of first segmentation in the whole frame and local adaptive segmentation in each chosen kernel region.

It should be noted that in the preliminary tracking stage, only Kalman filter prediction and CMK tracking are conducted to derive the positions of bounding boxes. After adaptive segmentation, the tracking algorithm is called again to create final tracking result from the optimized segmentation result with the Kalman filter updated.

4. EXPERIMENTAL RESULTS

Our experimental results include evaluation on the performance of both segmentation and tracking. Compared with the state-of-the-art algorithms, we have tested our proposed method on benchmark datasets and our own videos to emphasize the object merging issue.

4.1. Experiment Setting

The parameters in shadow removal, α, β, τ_{Cb}, and τ_{Cr}, are empirically set as 0.6, 0.8, 12, and 12 respectively. The values of *simiThres* for color similarity and chromaticity similarity are also empirically chosen to be 0.3 and 0.4 respectively. Both of the *simiMax*'s in (5) are set to 0.72. After all, the limits for size-change, width-change and height-change ratios of bounding boxes are given by 0.1, 0.15 and 0.15 respectively. If the boxes are unchanged for more than 2 seconds, there might be some errors, and thus we will let them follow the corresponding foreground blobs immediately.

4.2. Results on Benchmark Datasets

For the evaluation of segmentation performance, we have applied the proposed MAST system to the CVPR 2014 Change Detection dataset [13]. Among all the categories in the dataset, we choose the *shadow* category to test our proposed scheme mainly for object tracking under shadowing scenarios. The comparison results with some of the state-of-the-art algorithms are shown in Table I[1]. Among all the methods in this scenario, the ranking of our performance based on *F-Measure* is in the middle. It is worthwhile to observe that our object segmentation method can achieve such a good ranking in the change detection dataset, since most of other approaches are based on sophisticated pixel-level statistical models. Rather, our method only depends on a simple thresholding algorithm with feedback from region-level similarity of histograms. The reason that we cannot reach the top-level performance is our intention to preserve more foreground than the ground truth. This is necessary for supporting robust tracking.

We have also evaluated our tracking performance using two video sequences, *TwoEnterShop2cor* and *ThreePastShop2cor*, in the CAVIAR Dataset [17]. We manually pick the targets that are occluded during their movements. The comparison of average errors on these sequences are given in Table II. The error calculation is the same as that in [2]-[4]. It is defined by the distance in pixel between the centers of mass of experimental result and the ground truth. The method in [4] uses the general segmentation and tracking approaches mentioned in Section 3.1 with the same parameters setting as ours. We also make a comparison with the method in [6] combined with CMK tracking to show the advantage of our system in supporting robust tracking. Their default parameters are adopted in change detection, and we use the same parameters for CMK tracking. It can be seen that the MAST system improves the tracking performance of the original system that has no feedback loop, and it can generate more robust foreground mask for tracking. It both handles occlusion well and prevents object merging.

Figure 5 shows four representative frames of the tracking results using these three methods. It shows that our system is more robust against drifting when occlusion occurs. Also, the tracking is more accurate when the problem of object merging occurs.

4.3. Results on Our Own Video Sequences

To emphasize our advantage in handling object segmentation and tracking when object(s) have similar color and/or chromaticity with background, we recorded our own video sequences, with resolution of 640x360, in which pedestrians walked around a background area that has similar color and chromaticity with their clothing. Our segmentation and tracking results are provided in Tables III and IV respectively. The segmentation results of MAST and the method in

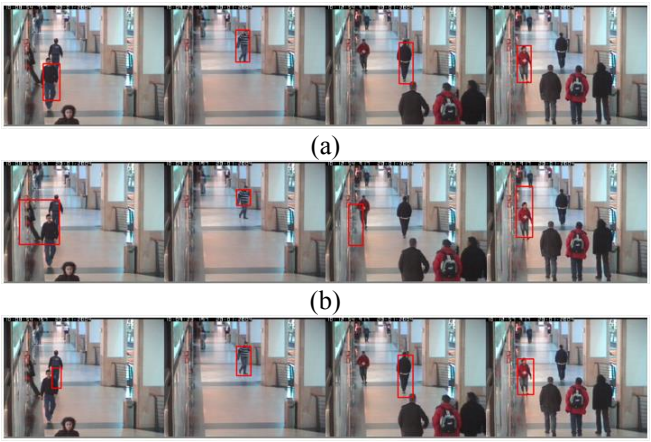
[1] The performance measurement code and the foreground images of the state-of-the-art algorithms are provided on [13]. Moreover, the results in Table I is different from that on [13] because the code only measures a part of the video frames.

Table I. Quantitative comparison of MAST system to several state-of-the-art change detection methods on five measures on the *shadow* scenario of CVPR 2014 Change Detection challenge [13]

	Recall	Spec	FPR	FNR	F
SuBSENSE [6]	0.9419	0.9920	0.0080	0.0581	0.8986
IUTIS-3 [14]	0.9478	0.9914	0.0086	0.0522	0.8984
GMM [15]	0.7960	0.9871	0.0129	0.2040	0.7370
CP3 [16]	0.7840	0.9832	0.0168	0.2160	0.7037
MAST	0.8679	0.9864	0.0136	0.1321	0.7884

Table II. Comparison of average errors of tracking in terms of pixels on two video sequences in CAVIAR Dataset [17]

	MAST	Method in [4]	Method in [6] with CMK tracking
video #1	10.06	26.72	17.27
video #2	9.75	10.74	14.07
Overall	10.18	18.73	15.67



(a)

(b)

(c)

Figure 5. Tracking results on two video sequences in CAVIAR Dataset [17] by (a) the proposed system. (b) Method in [4]. (c) Method in [6] combined with CMK tracking. Frames #755 and #2175 for *video #1: TwoEnterShop2cor*, and frames #642 and #794 for *video #2: ThreePastShop2cor*.

[4] with respect to *F-Measure* score are very close. But our method has significantly higher recall rate, which implies that MAST tends to preserve more foreground for robust object tracking. The performance of method in [6] is less robust than ours, especially when objects go into background areas with similar color. The superiority of MAST in tracking objects under object merging is clearly validated in Table IV. Apart from the comparison of average errors, the other two methods both lose the targets twice in similar background area, while ours can successfully track all objects.

The tracking results in four representative frames are demonstrated in Figure 6. Since the segmentation results by the methods in [4] and [6] lose lots of foreground belonging to the object in similar-color background area, it leads to failures in the tracking stage such as missing the target or bounding only part of the target. The videos of corresponding complete simulation results are made available in http://allison.ee.washington.edu/thomas/mast/.

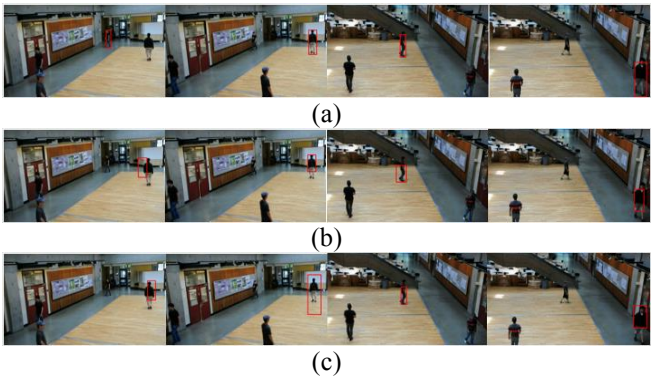
5. CONCLUSION

Table III. Comparison of segmentation performance on five measures on our two video sequences (*video #1* and *video #2*)

	Recall	Spec	FPR	FNR	F
#1-MAST	0.8815	0.9959	0.0041	0.1185	0.8861
#1-Method in [4]	0.8588	0.9974	0.0026	0.1412	0.8910
#1-Method in [6]	0.8569	0.9896	0.0104	0.1431	0.8033
#2-MAST	0.7727	0.9948	0.0052	0.2272	0.7916
#2-Method in [4]	0.7508	0.9965	0.0034	0.2492	0.8026
#2-Method in [6]	0.8938	0.9934	0.0066	0.1062	0.8424

Table IV. Comparison of average errors of tracking in terms of pixels on our two video sequences

	MAST	Method in [4]	Method in [6] with CMK tracking
video #1	17.88	18.26	18.90
video #2	10.30	16.10	15.95
Overall	14.09	17.18	17.43



(a)

(b)

(c)

Figure 6. Tracking results on our video sequences by using (a) MAST system. (b) Method in [4]. (c) Method in [6] combined with CMK tracking. Frame #319 and frame #342 for *video #1*, and frame #329 and frame #611 for *video #2*.

In this paper, we propose an adaptive segmentation and tracking system based on multiple kernels. The purpose is to robustly track objects when they have similar color or chromaticity with the background area. It applies general segmentation and tracking algorithms first. And then kernel histograms are constructed inside each kernel to find their color similarity and chromaticity similarity. The calculated penalty weight is utilized to penalize the thresholds in segmentation and determine the expansion ratio of the corresponding kernel region to redo segmentation. The tracking algorithm is called again to generate the final output using the adaptive segmentation result. From experiments on CVPR 2014 Change Detection dataset, CAVIAR Dataset and our video sequences, it is shown that the proposed system can improve the performance of tracking while keeping fine segmentation result especially when dealing with object merging problem. For future development, since this method is always trying to preserve the foreground of detected objects, it will be helpful to add an object detector into the system to prevent ghosts.

6. ACKNOWLEDGEMENT

This research is supported by research grants from Prism Skylabs Inc. and the Ministry of Science and Technology of Taiwan 104-2917-I-009-022.

7. REFERENCES

[1] T. Zhao, R. Nevatia, and B. Wu, "Segmentation and Tracking of Multiple Humans in Crowded Environments," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1198-1211, 2008.

[2] C.T. Chu, J.N. Hwang, H.Y. Pai, and K.M. Lan, "Robust Video Object Tracking Based on Multiple Kernels with Projected Gradients," *Proc. IEEE Conf. Acoustics, Speech and Signal Processing*, vol. 1, pp. 798-805, Prague, May 2011.

[3] C.T. Chu, J.N. Hwang, H.Y. Pai, and K.M. Lan, "Tracking Human Under Occlusion Based on Adaptive Multiple Kernels With Projected Gradients," *IEEE Trans. Multimedia* , vol. 15, pp. 1602-1615, June 2013.

[4] C.T. Chu, J.N. Hwang, S. Wang, and Y. Chen, "Human Tracking by Adaptive Kalman Filtering and Multiple Kernels Tracking with Projected Gradients," *Proc. ACM/IEEE Int. Conf. Distributed Smart Cameras*, 2011.

[5] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background Segmentation with Feedback: The Pixel-based Adaptive Segmenter," *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, vol. 1, pp. 38-43, 2012.

[6] P. St-Charles, G. Bilodeau, and R. Bergevin, "SuBSENSE: A Universal Change Detection Method with Local Adaptive Sensitivity," *IEEE Trans. Image Processing* , vol. 24, no. 1, pp. 359-373, 2015.

[7] N. Otsu, "A Threshold Selection Method from Gray-level Histograms," *IEEE Trans. Sys., Man., Cyber. on Robotics*, pp. 62-66, 1979.

[8] T. Horprasert, D. Harwood, and L. S. Davis, "A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection," *Proc. IEEE Int. Conf. Computer Vision*, vol. 99, pp. 1-19, 1999.

[9] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric Model for Background Subtraction." *Computer Vision—ECCV 2000*, pp. 751-767, Springer Berlin Heidelberg, 2000.

[10] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564-577, May 2003.

[11] K.H. Lee, and J.N. Hwang, "On-road Pedestrian Tracking across Multiple Driving Recorders," *IEEE Trans. Multimedia*, vol.17, no.9, pp. 1429－1438, Jul. 2015.

[12] K.H. Lee, Y.J. Lee, and J.N. Hwang, "Multiple-kernel Based Vehicle Tracking Using 3-D Deformable Model and License Plate Self-similarity," *Proc. IEEE Conf. Acoustics, Speech and Signal Processing*, pp. 1793－1797, May 2013.

[13] N. Goyette, P. M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A New Change Detection Benchmark Dataset," *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, pp. 1-8, June 2012.

[14] S. Bianco, G. Ciocca, and R. Schettini, "How Far Can You Get by Combining Change Detection Algorithms?" arXiv preprint arXiv:1505.02921 (2015).

[15] C. Stauffer, W. Eric, and L. Grimson, "Adaptive Background Mixture Models for Real-time Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2. 1999.

[16] D. Liang, and S. Kaneko, "Improvements and Experiments of a Compact Statistical Background Model," arXiv preprint arXiv:1405.6275 (2014).

[17] CAVIAR: Context Aware Vision using Image-based Active Recognition, EC founded CAVIAR project/IST 2001 37540, http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.