MULTIPLE-KERNEL BASED VEHICLE TRACKING USING 3-D DEFORMABLE MODEL AND LICENSE PLATE SELF-SIMILARITY

Kuan-Hui Lee, Yong-Jin Lee, Jenq-Neng Hwang
Department of Electrical Engineering,
Box 352500, University of Washington,
Seattle, WA, 98195, USA
{ykhlee, zeroth, hwang}@uw.edu

ABSTRACT

In this paper, we propose a novel vehicle tracking system under a surveillance camera. The proposed system tracks vehicles by using constrained multiple-kernel, facilitated with Kalman filtering, to continuously update the position and the orientation of the moving vehicles. To further reliably track vehicles under partial occlusion or even total occlusion, our tracking algorithm also systematically builds 3-D vehicle model, from which the license plate region is identified and a self-similarity descriptor is further used for low-resolution license plate matching. Experimental results have shown the favorable performance of the proposed system, which can successfully track vehicles under serious occlusion while maintaining the knowledge of 3-D geometry of the tracked vehicles.

Index Terms— Vehicle tracking, Multiple kernels tracking, 3-D vehicle model, Self-similarity descriptor

1. INTRODUCTION

Nowadays, video-based traffic surveillance over vehicles has become a very important research area. By tracking vehicles, it is possible to collect their trajectories in videos for high level analytics, for example, detection and avoidance of vehicle accidents, detection of specific vehicles for theft discovery, collection of traffic statistics information for further decision making, and so on.

Vehicle tracking can be regarded as a specific category of video object tracking, which has been extensively developed and discussed. These object tracking techniques may not be directly applicable for vehicle tracking due to the fact that color information is not very discriminative among different vehicles. Many approaches have thus been proposed specifically for vehicle tracking [1]-[4], mainly based on color histogram or 2-D contours. An intuitive way to extract features for vehicle identity is to modularize a vehicle into a 3-D model. Several techniques which use 3-D vehicle models to locate and recognize the vehicles have also been proposed [5]-[9]. These 3-D model-based works show their efficient performance in vehicle tracking without explicitly dealing with occlusion scenarios.

Another important feature of a vehicle is its license plate. However, due to low resolution and high distortion nature, caused by the dynamic viewpoints of camera, of license plate captured from surveillance videos, a clear license plate image for license plate recognition (LPR) is hardly available in general cases. Although feature descriptors such as Harris corner [10], SIFT [11], and SURF [12], are able to maintain the characteristics of the license plate, these descriptors are sensitive to corners, lines, or intensity, which cannot be extracted properly from the distorted videos.

In this paper, we effectively combine the vehicle tracking technique with 3-D vehicle model into one system and incorporate license plate matching with the SSD, so as to not only track vehicles under occlusion but also maintain the knowledge of 3-D vehicle geometry. The proposed system extended our previously developed video object tracking technique [14] [15], called Kalman-based constrained multiple-kernel (KCMK) tracking. Meanwhile, built upon the approach in [7], the system automatically builds a 3-D vehicle model for each tracked vehicle. On the other hand, we effectively adopt self-similarity descriptor (SSD) [13] to measure similarity between license plate images from the distorted surveillance videos. To reliably extract the SSD, 3-D vehicle model is incorporated to locate the region of the license plate, which is perspectively warped to the view-normalized version and whose SSD can now be used to better identify a vehicle. Facilitated by the 3-D vehicle model, the proposed system solve issues of partial occlusion by 3-D model based multiple kernel tracking, and total occlusion by performing license plate matching in terms of the SSD.

The rest of the paper is organized as follows. In Section 2, the details of the algorithms adopted in our system, including KCMK tracking, 3-D vehicle shape fitting, and SSD, are presented. Section 3 depicts overview of the proposed system and how to integrate KCMK tracking 3-D vehicle model, and license plate identification into one. The experimental results are shown in Section 4, followed by the conclusion in Section 5.

2. ADOPTED ALGORITHMS

2.1. KCMK Tracking

The objective of the constrained multiple-kernel tracking [14] is to retrieve a candidate model, which can be described with multiple kernels with pre-specified constraints among these kernels, so that the maximum similarity can be reached between the tracked objects and the candidate model For M kernels the total cost function $J(\mathbf{x})$ is defined to be the sum of the weighted N individual cost functions $J_i(\mathbf{x})$,

$$J(\mathbf{x}) = \sum_{i=1}^{M} w_i J_i(\mathbf{x}),$$
$$J_i(\mathbf{x}) = 1 - simi_i(\mathbf{x}), \quad w_i = \gamma \times simi_i(\mathbf{x}), \qquad (1)$$

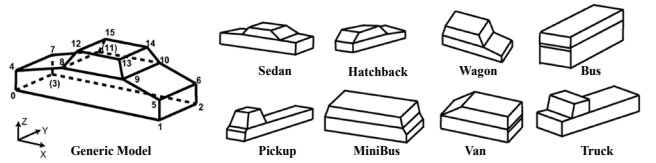
where $simi_i(\mathbf{x})$ is the similarity function at the location $\mathbf{x}$ in the state space domain, the weight w_i is adaptively updated based on the normalized similarity, and γ is an pre-determined empirical constant. Moreover, the constraint functions $\mathbf{C}(\mathbf{x}) = \mathbf{0}$ needs to be considered to maintain the relative locations of the kernels. Therefore, the problem could be further formulated by

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} J(\mathbf{x}), \text{ subject to } \mathbf{C}(\mathbf{x}) = \mathbf{0}. \qquad (2)$$

Being initially predicted by Kalman filtering, the $\mathbf{x}$ of each kernel is further directed to the most similar region but also maintain the constrained conditions, so that the kernels can be tracked successfully [15].

2.2. Vehicle Shape Fitting

Vehicle shape fitting [7] is to generate an approximate 3-D vehicle model deformed from a 3-D generic model. The 3-D deformable model with 16 vertices and 23 arcs, as shown in Figure 1 (a), is defined by 12 shape parameters and 3 pose parameters. These total 15 parameters can be optimized by evaluating the fitness, quantified as fitness evaluation score (FES), between image data and the 2-D projection of a 3-D deformable model, base on an evolutionary computing framework called estimation of multivariate normal algorithm-global ($EMNA_{global}$). Figure 1 (b) shows eight different types of vehicle deformed from the generic model shown in Figure 1 (a).



(a) (b)

Figure 1. (a) Generic model. (b) Different types of vehicle deformed from the generic model.

Figure 2. (a) Initial frame. (b) License plate from the initial frame. (c) 10th frame after the initial one, pixels with the matching score above a threshold are marked in red (as indicated by the blue arrow).

2.3. Self-Similarity Descriptor (SSD)

The self-similarity descriptor (SSD) [13] is based on a relative geometric layout of similarities between neighbor-hoods, so as to match an object in one image with another object with different visual appearance in the other different image as long as they have a similar shape. The visual characteristics of a license plate changes as a vehicle moves and in many cases, characters of a license plate are hardly recognized except the spatial layout between local patches within a license plate. This motivates our use of SSD for matching license plates captured from surveillance cameras.

Figure 2 shows license plate matching using the SSD. From an initial frame (a), we roughly cut off a rectangular template region from the front of a vehicle as a template image (b) and searched the 10th frame for a target license plate by densely comparing the SSD of the template and the patches from the 10th frame. Pixel locations with matching scores above a threshold were marked in red (as indicated by the blue arrow) on a gray image (c). The peaks can be seen around the license plate in the front of a vehicle. It validates our use of the SSD for a license plate even with those characters being hardly recognized.

3. OVERVIEW OF THE SYSTEM

Figure 3 shows the overall procedures of the proposed vehicle tracking system. The first step is to segment the foreground objects, by using the background subtraction technique. Second, the Kalman prediction is applied to the segmented objects and the detection of occlusion is then performed. The system detects if there is an occlusion by checking the predicted states of the tracked objects to see whether they are merged with one another. If there is no occlusion, the tracking results are obtained by measurement selection and thus Kalman updating. Otherwise, the system checks the occlusion condition (partial or total) according to the visibility of the 3-D vehicle model and the similarity of the kernels between the previous and current frames. The 3-D vehicle model built from the previous frame before the occlusion is used for multiple kernel tracking or SSD extraction in the current frame. If it is a partial occlusion, 3-D model based multiple kernel tracking is applied. Otherwise, for the case where a vehicle is totally occluded and reappears in the next couple frames, vehicle features matching using the SSD and 3-D vehicle model is employed to search for the tracked vehicle to resume tracking. Finally, the tracking results are then iteratively used to build the 3-D vehicles for the next frame.

3.1. KCMK Tracking with 3-D Vehicle Model

In KCMK tracking, we regard surface planes in the 3-D vehicle model are kernels [16]. The corresponding vertices of each kernel (plane), annotated by K{•}, are shown in Figure 4, where vertices are defined in Figure 1 (a). Each

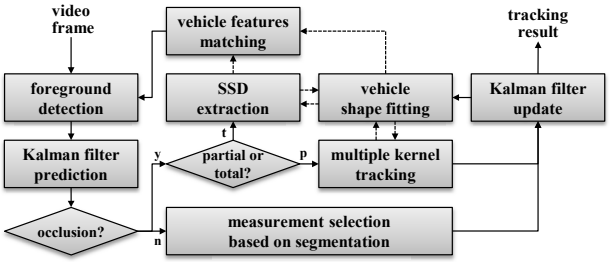


Figure 3. Overall proposed system framework

kernel is a basic component in the tracking procedure. However, either due to the view aspects or occlusions, not all the kernels are reliable. Thus, we adaptively change weight w_i in Equation (1) to assign each kernel with different importance. First, according to the view aspect, only completely visible planes (annotated by $\mathrm{K_v}\{\bullet\}$) projected onto the image frame are used for multiple kernel tracking. In other words, kernels hidden behind are weighted by zero. Second, the similarity of the occluded kernels is zero weighted by Equation (1), so as to mitigate their impact on the tracking.

Unlike the KCMK for human tracking in [14][15] where the used multiple kernels are always vertically aligned, the alignment orientations of the kernels for vehicles vary dynamically in vehicle tracking. Hence, the constraint functions in [13] [14] should be redefined:

$$\begin{cases} \left(x_k - x_{IV}\right)^2 + \left(y_k - y_{IV}\right)^2 = {L_{k,IV}}^2 \\ \dfrac{\left(y_k - y_{IV}\right)}{\left(x_k - x_{IV}\right)} = \tan^{-1}\left(\varphi_{k,IV}\right) \end{cases}, \quad \text{for } k \in \mathrm{K_v}\{\bullet\}, \quad (3)$$

where (x_{IV}, y_{IV}) the location of the $\mathrm{K}\{IV\}$ and $\{(x_k, y_k)\}$ are the locations of the visible kernels $\mathrm{K_v}\{\bullet\}$, $L_{k,IV}$ is the initial distance between $\mathrm{K}\{k\}$ and $\mathrm{K}\{IV\}$, and $\varphi_{k,IV}$ is the initial angle between the kernel axis and the horizontal axis. These constants need to be adaptively updated when either size and/or rotation of a vehicle are greater than the empirical thresholds.

3.2. Occlusion Justification

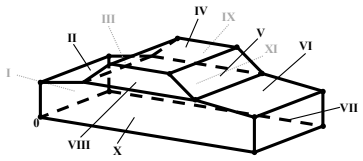
To justify the occlusion condition, two criteria are used in the proposed system: the number of the visible vertex N_v, and the similarity of the visible kernels between the previous and current frames. We define average similarity as

$$avg.simi = \left(\sum_{i=0}^{K_v} simi_i(\mathbf{x})\right) \Big/ K_v \quad , \quad (4)$$

where K_v is the number of visible kernels. More specifically, the occlusion justification is described as follows.

$$\begin{cases} avg.simi > \alpha_1 \text{ and } N_v < \beta_1 \Rightarrow \text{normal case} \\ \alpha_1 \geq avg.simi > \alpha_2 \text{ or } \beta_1 \geq N_v > \beta_2 \Rightarrow \text{partial occlusion} \\ avg.simi \leq \alpha_2 \text{ or } N_v \geq \beta_2 \Rightarrow \text{total occlusion} \end{cases}, \quad (5)$$

where α_1, α_2, β_1, β_2 are the pre-determined thresholds.



K{•}	Vertices	Description
I	0, 3, 4, 7	rear-side
II	4, 7, 8, 11	boot cover
III	8, 11, 12, 15	rear window
IV	12, 13, 14, 15	roof
V	9, 10, 13, 14	windshield
VI	5, 6, 9, 10	engine hood
VII	1, 2, 5, 6	front-side
VIII	8, 9, 12, 13	right window
IX	10, 11, 14, 15	left window
X	0, 1, 4, 5, 8, 9	right-side
XI	2, 3, 6, 7, 10, 11	left-side

Figure 4. Table of kernels in the 3-D vehicle.

3.3. Vehicle Features Matching

In the proposed system, KCMK tracking is applied in normal and partial occlusion cases. In case of total occlusion, vehicle shape and SSD of the license plate extracted from the previous frame are used to track the vehicle in the coming frames. If the features of a vehicle from the previous frame highly match that from the subsequent frame, the vehicles are regarded as identical.

To match the shape features, we calculate the similarity s_{shape} of 12 shape parameters between previous 3-D vehicle model and subsequent ones. If s_{shape} is greater than a threshold τ_{shape}, two vehicles are regarded as of the same shape. Moreover, we also match the SSD features for further verification. Due to sensitivity to scaling and orientation, SSD of the license plate is not robust under varying scale and orientation. Facilitated by the 3-D vehicle model, these features can be extracted easily and be perspectively warped to a view-normalized license plate. The corresponding SSD are extracted and stored before occlusion happens. If there is a newly occurring vehicle in the video, the stored features are compared with new vehicle's features to obtain similarity s_{ssd}. If s_{ssd} is greater than a threshold τ_{ssd}, two SSD features are regarded as the same. If both shape and SSD features are matched, two vehicles are regarded as the same.

4. EXPERIMENTAL RESULTS

The experiment settings are described as follows. In the KCMK tracking part, K-L distance is used for all similarity measures, and the histogram of the object is constructed based on the HSV color space and roof kernel [15]. In the vehicle shape fitting part, the parameters in $\mathrm{EMNA_{global}}$ are N = 2000, R = 100, the threshold the magnitude of gradients for stopping criterion is 4 [7], and f equals to 10; The values of several thresholds are $\alpha_1 = 0.6$, $\alpha_2 = 0.2$, $\beta_1 = 3$, $\beta_2 = 5$, $\tau_{\text{shape}} = 0.05$, $\tau_{\text{ssd}} = 0.7$. The surveillance camera is well calibrated, which implies intrinsic and extrinsic parameters are known.

The efficiency of using 3D shape feature to identify the type of vehicles has been proven in [7]. On the other hand, to show the efficiency of the SSD, we compare 10 different license plates (denoted by 01, 02, …, 10) with one another, and with each license plate extracted from the about 20th frames after its initial frame (denoted by 01', 02', …, 10'). Table I shows the results of the comparison. Each license plate has a relatively higher similarity score when comparing with its corresponding one (gray grids in the table).

In order to demonstrate the performance of our system, we compare the performance with a baseline particle-filter-based tracking approach [6] and apply 3-D deformable model [7] in both systems. Figure 5 shows some snapshots of tracking performance of our dataset 1 in both two approaches. As shown in frame 145 and 147, our proposed system provides better 3-D shape fitting when tracking vehicles 2 and 4 than that of the approach [6]. It is because the estimation of pose parameters in [6] is inaccurate due to the constant variance. Figure 6 shows some snapshots of two approaches when partial occlusion happens. In frame 47, the performance of the proposed systems is as well as the approach [6]. As for frame 52 and 58, our system performs outstandingly, while the approach [6] not only loses tracking but also fails the fitting of the 3-D vehicle model. The reason is that non-occluded kernels are able to be tracked while binding with occluded kernels. Figure 7 shows the results of the dataset 2, which contains total occlusion case. The results show that our system resumes tracking vehicle 0 and maintain the 3-D geometry after the occlusion. The approach [6] is able to track vehicle during occlusion, but cannot maintain 3-D geometry of the vehicle (frame 86). Figure 8 shows the results of two approaches in our dataset 4, which has totally 50 vehicles for testing the robustness of the systems.

To further measure the performance of the system, a quantitative metric is defined for evaluation. We manually build 3-D models of vehicles with best fitted pose parameters to construct the ground truth. Then, the average error of the tracking is defined as the distance between the all vertices of the tracked results and those of the ground truth:

$$Average\ Error = \sum_{i=0}^{15} \left\| x_i - g_i \right\|, \qquad (6)$$

where x_i is the world coordinate of the vertex of the modeled 3-D vehicle and g_i is the world coordinate of the ground truth. Table II shows the overall average error per vehicle in terms of meter. Totally 65 vehicles are tested in our experiments. Our proposed approach performs more accurately on fitting vehicles, especially when vehicles are partially or totally occluded (as reflected by the peak errors in all 4 datasets).

5. CONCLUSION

This paper proposes a novel vehicle tracking system in a single surveillance camera. The proposed system not only utilizes KCMK tracking technique to track vehicles, but also takes advantage of 3-D vehicle model to improve the tracking results. By estimating vehicle geometry based on the well-fitted 3-D model, we are able to obtain more specific and reliable information for further processing.

TABLE I. SIMILARITY SCORE OF THE COMPARISON

	01	*02*	*03*	*04*	*05*	*06*	*07*	*08*	*09*	*10*
01'	0.7610	0.5736	0.5043	0.5568	0.5008	0.5171	0.5910	0.4938	0.5646	0.5636
02'	0.5862	0.7706	0.4839	0.4963	0.4841	0.5052	0.5365	0.4901	0.5341	0.5104
03'	0.4871	0.4580	0.7557	0.5070	0.5363	0.5133	0.5126	0.4336	0.4873	0.4990
04'	0.5949	0.5238	0.5818	0.7719	0.5530	0.5287	0.5994	0.5014	0.5446	0.56657
05'	0.5333	0.5279	0.5600	0.5400	0.7707	0.5519	0.5852	0.4910	0.5361	0.5165
06'	0.5039	0.4890	0.5385	0.4696	0.5544	0.8534	0.5527	0.4834	0.5398	0.5592
07'	0.5910	0.5147	0.5150	0.5569	0.5615	0.5718	0.7606	0.5292	0.5408	0.5271
08'	0.5052	0.4784	0.4617	0.5086	0.4990	0.5087	0.5420	0.7600	0.4994	0.4929
09'	0.5845	0.5235	0.4861	0.4762	0.5007	0.5382	0.5666	0.4730	0.8018	0.5613
10'	0.5410	0.4990	0.5022	0.5083	0.4977	0.5603	0.5362	0.4579	0.5805	0.8415

TABLE II. AVERAGE ERROR (IN TERMS OF METER)

	Proposed system	**Approach in [6]**
Ave. error (m)	3.172	4.729

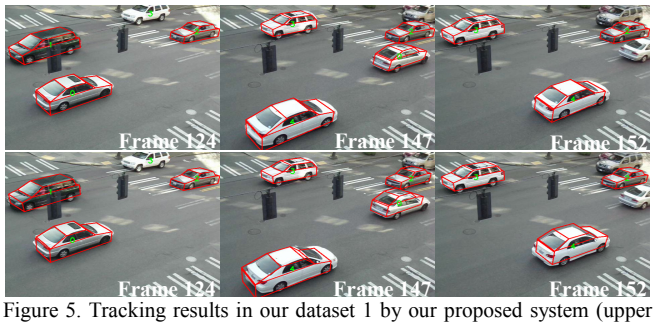


Figure 5. Tracking results in our dataset 1 by our proposed system (upper row), and by particle-filter-based method [6] (bottom row).

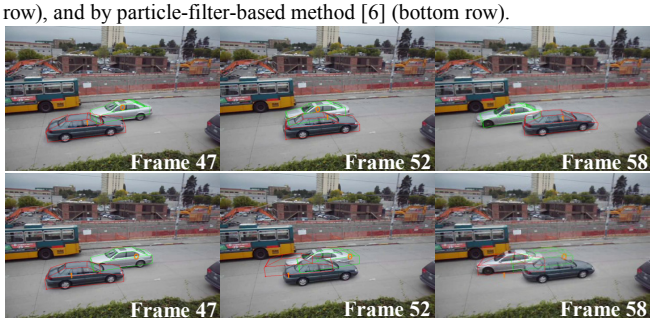


Figure 6. Tracking results in our dataset 2 by our proposed system (upper row) , and by particle-filter-based method [6] (bottom row).

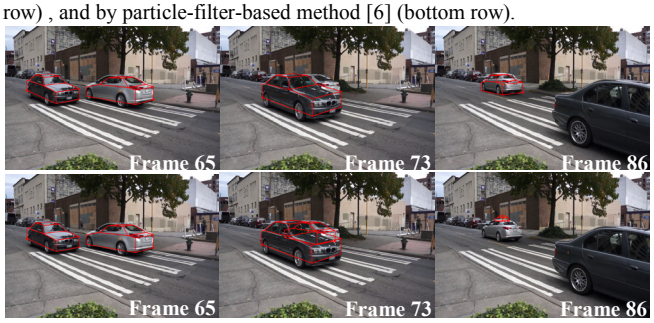


Figure 7. Tracking results in our dataset 3 by our proposed system (upper row) , and by particle-filter-based method [6] (bottom row).

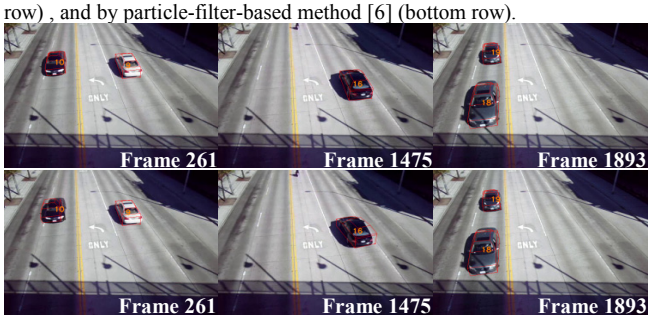


Figure 8. Tracking results in our dataset 4 by our proposed system (upper row), and by particle-filter-based method [6] (bottom row).

6. REFERENCES

[1] D. Koller, J.Weber, and J. Malik, "Towards realtime visual based tracking in cluttered traffic scenes," in *Proc. IEEE Intell. Vehicles Symp.*, pp. 201–206, 1994.

[2] E. B. Meier and F. Ade, "Tracking cars in range images using the condensation algorithm," in *Proc. IEEE Conf. Intell. Transp. Syst.*, pp. 129–134, 1999.

[3] P. L. M. Bouttefroy, A. Bouzerdoum, S. Phung, and A. Beghdadi, "Vehicle tracking using projective particle filter," *Proc. IEEE Int. Conf. AVSS*, pp. 7–12, 2009.

[4] J. Scharcanski, A.B. Oliveira, P. G. Cavalcanti, and Y. Yari, "A particle-filtering approach for vehicular tracking adaptive to occlusions," *IEEE Trans. Vehicular Tech.*, vol. 60, no.2, pp. 381–389, Feb. 2011.

[5] J. Lou, T. Tan, W. Hu, H. Yang, and S. J. Maybank, "3-D model-based vehicle tracking," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1561–1569, Oct. 2005.

[6] Z. Zhang, K. Huang, T. Tan, and Y. Wang, "3D model based vehicle tracking using gradient based fitness evaluation under particle filter framework," *Proc. Int. Conf. Pattern Recog.*, pp. 1771–1774, 2010.

[7] Z. Zhang, T. Tan, K. Huang, and Y. Wang, "Three-dimensional deformable-model-based localization and recognition of road vehicles," *IEEE Trans. Image Process.*, vol.21, no.1, pp.1–13, Jan., 2012.

[8] M. J. Leotta and J. L. Mundy, "Vehicle surveillance with a generic, adaptive, 3-D vehicle model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1457–1469, Jul., 2011.

[9] Y. Li, L. Gu, and T. Kanade, "Robustly aligning a shape model and its application to car alignment of unknown pose," *IEEE Trans. Pattern Anal. Mach. Intell.* , pp. 1860–1876, 2011.

[10]C. Harris and M. J. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, pp. 147–152, 1988.

[11]D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[12]H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features", *Comput. Vis. Image Understanding*, vol. 110, no. 3, pp. 346--359, 2008.

[13]E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," *IEEE Conf. Comput. Vis. Pattern Recogn.*, pp. 1–8, Jun., 2007.

[14]C.-T. Chu, J.-N. Hwang, H.-I. Pai, K.-M. Lan, "Robust video object tracking based on multiple kernels with projected gradients", *IEEE Int'l Conf. Acoustics, Speech & Signal Process.*, pp. 1421-1424, May, 2011.

[15]C.-T. Chu, J.-N. Hwang, S.-Z. Wang and Y.-Y. Chen " Human tracking by adaptive kalman filtering and multiple kernels tracking with projected gradients," *ACM/IEEE Int'l Conf. Distributed Smart Cameras*, pp.1–6, Aug., 2011.

[16]K.-H. Lee, J.-N. Hwang, J. Yu, and K.Lee, "Vehicle tracking iterative by kalman-based constrained multiple-kernel and 3-D model-based localization," *IEEE Int'l Conf. Acoustics, Speech & Signal Process.*, May, 2013.