

A Novel Unmanned Aerial Surveillance Scheme

Nevrez İmamoğlu
Dept. of Medical System Engineering
Chiba University
Chiba, Japan
nevrez@gmail.com

Mehmet Önder Efe
Dept. of Computer Engineering
Hacettepe University
Ankara, Turkey
onderefe@gmail.com

Aydın Eresen
Dept. of Electrical Engineering
Texas A&M University
USA
aeresen@gmail.com

Okyay Kaynak
Dept. of Electrical and Electronics Engineering
Boğaziçi University
İstanbul, Turkey
oktay.kaynak@boun.edu.tr

Abstract—An aerial surveillance method is proposed for a predefined single object tracking. The algorithm takes advantage of template matching to be able to track the selected object in video sequences. The selected templates are chosen from the images by utilizing gradient operation on the Gabor Wavelet representations. The algorithm achieves the tracking task even in the cases where i) the observer is moving, ii) the target object is moving, iii) video record is low quality with noise and iv) there are high illumination changes.

Keywords— Gabor Wavelets, feature extraction, pseudo-Zernike moments, template matching, tracking and surveillance

I. INTRODUCTION

Tracking is one of the prime tasks in surveillance applications, which can be achieved by various sensors (e.g. fiber sensors, pyroelectric sensors, infrared sensors, hyperspectral sensors, ccd/cmos cameras, and etc.) according to the requirements of objectives for the tracking process. Therefore, several sensor types have been employed in researches to implement task specific surveillance applications either for monitoring, tracking or localization tasks, [1-2]. Among these various tasks, one of the prime application fields of surveillance systems is the tracking of a moving object where the targets to be tracked are generally vehicles or humans. These systems are able to track objects, estimate the possible next position or find the next position among the varying scenes provided by a video source. The aim is to track the object of interest within the consecutive video frames with the best accuracy and computational performance. Generally, better results are obtained with algorithms that introduce more computational intensity than those with low complexity. Real-time recognition and tracking applications make it unavoidable to consider these issues as the system has to be implemented with high frame rates with best possible accuracy.[3-5].

The contribution of this paper is to demonstrate that a precise tracking on monochromatic video sequences could be achieved by utilizing texture features of the templates by computing weighted SSD to find out best matching template. To be able to achieve this, a better representation of the image is used for feature extraction instead of original

image to increase the performance of tracking procedure. In this paper, the aim is to track a predefined object which is a vehicle moving and parking in the video. Gradient-based Gabor Wavelet (GW) representation of the image is used to select templates for template matching with pseudo-Zernike features. The paper does not consider the cases where there are occlusion and rotational motion changing the 3D projection of the object significantly, in which these cases can be handled by utilizing kalman or particle filter algorithms to define and track vehicle states. Since the aim is to demonstrate the performance of the proposed template matching model, this work do not include state estimation algorithms during tracking, and instead, a simple estimation is used for possible candidate position in the next frame. However, tracking is maintained under the presence of rotational motion which slightly changes the 3D projection of the targeted object and this is a challenge addressed appropriately here. Handling the rotational motion in full is coupled with the scale invariance property which is not aimed at here as the template size is fixed initially.

The paper is organized as follows: Section 2 explains the procedural steps and the proposed techniques including gradient-based GW representation and feature extraction algorithms. Section 3 discusses the experimental results. Concluding remarks are given at the end of the paper.

II. PROCEDURAL STEPS OF THE PROPOSED SYSTEM

The approach proposed in this paper starts with the representation of the image as a mixture of GW including different orientations. Having obtained the GW representation of the image, the next step is to take the gradient of the GW represented image, which yields two components as the changes on x and y axis of the 2D data. The magnitude of the changes will be used to select the template, which is to be sought and marked on the consecutive frames.

To define the template, a point is selected which is a part of the object body. This point is called the pivot point of the base template used to match the object on other frames among the possible templates. This point is set as the center of the template for the base features necessary for tracking.

The selected base template will be used as a reference in matching to the most similar template in the following frames. In this work, the pivot point is pre-selected on the object before the algorithm runs, i.e. the tracking system is intentionally informed about the object to be tracked initially.

Having determined the base template, the most similar template is sought in the next frame by using the feature set provided through the pseudo-Zernike moments. Assuming that the object displacement is not large between consecutive frames, which is equivalent to assume a satisfactory frame rate, the proposed approach searches the best matching template in some neighborhood of the previous template center. In addition, after the sixth frame, an expected position is calculated by examining the three previous displacements on x and y directions to find out the expected position. Hence, template matching is applied around this expected position. This procedure improves the performance of the system both in speed and in accuracy. As a final step, the center point of the best matching template is marked on the running images on the video.

A. Image Representation

Gabor Wavelets are useful tools in image processing applications as they provide local information with various orientation and frequency scales. GW enables representation of the image with spatial localization, orientation selectivity, and frequency characteristics making them useful in various applications to discover hidden information in images [6, 7]. In our work, the parameters are selected according to the experimental results discussed in [6, 7]. The analytical description of the Gabor wavelet is given in (1) [6, 7].

$$\psi_{\mu,\nu}(x,y) = \frac{\|k_{\mu,\nu}\|}{\sigma^2} e^{-\frac{\|k_{\mu,\nu}\|^2((x-x_c)^2+(y-y_c)^2)}{2\sigma^2}} \times \left[e^{j(\text{Re}(k_{\mu,\nu})(x-x_c)+\text{Im}(k_{\mu,\nu})(y-y_c))} - e^{-\frac{\sigma^2}{2}} \right] \quad (1)$$

where μ is the orientation, and ν is the scale of the Gabor wavelet. x_c and y_c are the center point of the 2D GW, σ^2 denotes the variance of GW, $j = \sqrt{-1}$, $\text{Re}(k_{\mu,\nu})$ is the real part of the $k_{\mu,\nu}$, $\text{Im}(k_{\mu,\nu})$ is the imaginary part of the $k_{\mu,\nu}$, and $\exp(-\sigma^2/2)$ denotes a coefficient to provide zero mean. The definition in (1) is used as a mother wavelet in which changing ν and μ yields GW masks with different scale and orientation. Besides, in (1), $k_{\mu,\nu}$ is the wave vector for the mother wavelet which enables to give various GW masks by changing ν and μ , and described as in (2). In the notation shown, it should be noticed that an argument (x,y) is used when a particular pixel is meant, otherwise the entire entity is of interest.

$$k_{\mu,\nu} = k_\nu e^{j\phi_\mu} \quad (2)$$

where k_ν is the frequency and ϕ_μ orientation value defined in (3) and (4) respectively. In below equations, α is the maximum frequency of GW defined for the system, f is the spacing factor for each scaling factor ν in the frequency domain. In this paper, parameters are selected for GW as $\nu = 2$, $\mu = \{0,1,2,3,4\}$, $\sigma = 2\pi$, $\alpha = \pi/2$, and $f = \sqrt{2}$. Real parts of the obtained GW are shown in Fig. 1 with given parameters in surface and image graph respectively. Subplots of Fig. 2 are imaginary parts of the obtained GW with given parameters in surface and image graph respectively. Figs. 1-2 illustrate that changing μ produces different masks with the orientations defined in (4).

$$k_\nu = \alpha / f^\nu \quad (3)$$

$$\phi_\mu = \pi\mu / 8 \quad (4)$$

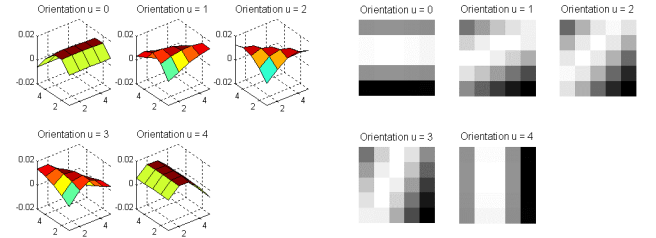


Fig. 1. Left: Surface graphs of the real parts of GW with the scale $\nu = 2$ and window size 5×5 . Right: Images representation of the real parts of GW with the scale $\nu = 2$ and window size 5×5

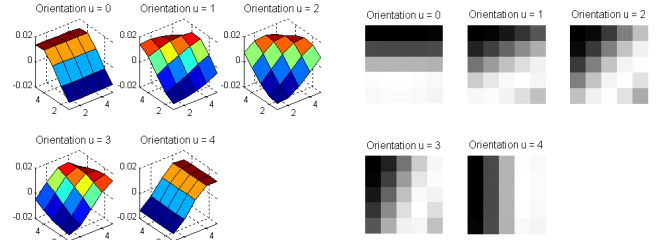


Fig. 2. Left: Surface graphs of the imaginary parts of GW with the scale $\nu = 2$ and window size 5×5 . Right: Images representation of the imaginary Parts of GW with the scale $\nu = 2$ and window size 5×5

Representation of the image with GW is satisfied by the convolution of each GW obtained with ν and μ values on the image as given in (5) [7, 6].

$$C_{\mu,\nu}(x,y) = I(x,y) * \psi_{\mu,\nu}(x,y) \quad (5)$$

where $*$ stands for the convolution operator, $I(x,y)$ is the original gray scale image, and C is the result of the convolution operation. Representation of the image is achieved for each scale ν by summing the magnitude of the all convolution results expressed as in (6).

$$R^\nu(x,y) = \sum_{\mu=0}^4 \|C_{\mu,\nu}(x,y)\| \quad (6)$$

where R^ν is the GW representation of the original image for the scale value $\nu=2$ with selected orientations $\mu = \{0,1,2,3,4\}$.

Fig. 3 is a frame taken from the video, Left subplot of Fig. 4 illustrates the mesh graph of the original image in Fig. 3, and right subplot of Fig. 4 depicts the mesh graph of its GW representation with selected parameters from the video

used in this paper. In the right subplot of Fig. 4, it is seen that GW representation suppresses the spurious content available in the left subplot of the same figure. The video source used in this paper is used also in [8].

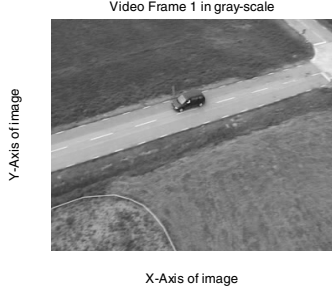


Fig. 3. Gray scale image frame from video

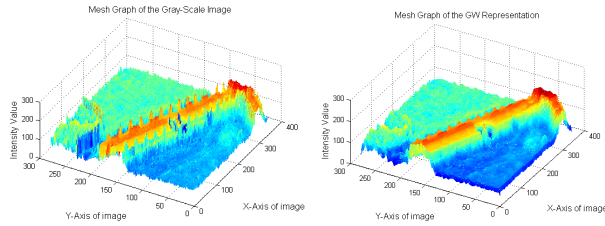


Fig. 4. Left: Mesh Graph of the Image, Right: Mesh Graph of the GW Represented Image

The changes on R^v , the GW represented form, correspond to the details contained by the image and are obtained through the use of gradient operation defined in (7) for 2D [9]. And, the spatial discretization of the operator is described in (8) [9].

$$\nabla R^v := \begin{bmatrix} g_x \\ g_y \end{bmatrix} := \begin{bmatrix} \frac{\delta R^v}{\delta x} \\ \frac{\delta R^v}{\delta y} \end{bmatrix} \quad (7)$$

$$\nabla R^v(x,y) \equiv \begin{bmatrix} g_x(x,y) \\ g_y(x,y) \end{bmatrix} \equiv \begin{bmatrix} \frac{(R^v(x+1,y) - R^v(x-1,y))}{2} \\ \frac{(R^v(x,y+1) - R^v(x,y-1))}{2} \end{bmatrix} \quad (8)$$

where $g_x(x,y)$ is the derivative along x axis at point (x,y) and $g_y(x,y)$ is the derivative along the y axis at point (x,y) . Useful information for our purposes is contained in the magnitude of the gradient vector, which is computed as given in (9).

$$M(x,y) = \|\nabla R^v(x,y)\| = \sqrt{g_x(x,y)^2 + g_y(x,y)^2} \quad (9)$$

where $M(x,y)$ is the magnitude of the gradient R^v at point (x,y) . The variable M contains critical information used in representing the image. An example of gradient data M is shown in Fig. 5.

The procedure after image representation is the feature extraction from the selected template. Templates are selected from the gradient based GW represented image

which is the result of the gradient operation in (8) and magnitude operation in (9). Some difference in between the two figures is visible at the first glance and in the sequel we describe how this difference is reflected to the tracking performance.

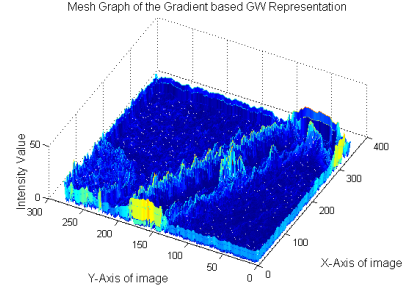


Fig. 5. Mesh graph of the gradient-based GW representation of the image

B. Feature Extraction

Feature extraction is a substantially important step of the proposed approach. There are many feature extraction techniques used for recognition and classification purposes [3-5, 10, 11] and the references therein, where it is possible to see different levels of performance with different techniques yet the search of the methods resulting in the best set of features that can be extracted from a given raw data is an active field of research. Since the extracted features determine the performance and accuracy of the applications heavily, choosing the most suitable technique is a critically important stage to implement a system displaying a good level of performance.

In this work, at every feature extraction step, two windows are selected from the template to describe a more distinctive Region of Interest (RoI) as shown in the left subplot of Fig. 6. First window is the template itself, and the second window is the half size of first window located with the same center point as first window. Hence, templates include two features belonging to their first and second window, respectively.

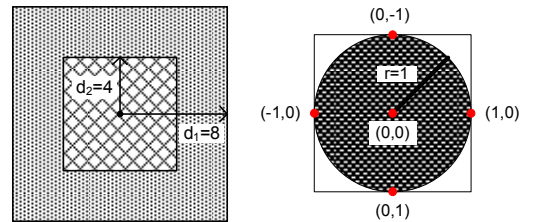


Fig. 6. Left: Double window to represent the template. Right: Template for feature extraction

Because of its invariance and robustness to illumination changes, pseudo-Zernike moments are selected for the extraction of features. Since pseudo-Zernike moments are more robust to noise than Zernike moments and they represent characteristics of the objects more distinctively than other moment based extraction methods [10, 11]. We

prefer utilizing the information obtained through pseudo-Zernike moments that are calculated around a unit circle where $x^2 + y^2 \leq 1$ is satisfied with x and y being the pixels around -1 and $+1$ (See the right subplot of Fig. 6). In this paper, pseudo-Zernike moments described by (10) are calculated as discussed in [10, 11].

$$Z_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x,y) V'_{nm}(x,y) \quad (10)$$

where Z_{nm} is the pseudo-Zernike moment of order n and repetition m defined for pseudo-Zernike polynomials, $V_{nm}(x,y)$ is the pseudo-Zernike polynomials expressed in (11), and $V'_{nm}(x,y)$ is the conjugate of $V_{nm}(x,y)$.

$$V_{nm}(x,y) = V_{nm}(\rho, \theta) = R_{nm}(\rho) e^{jm\theta} \quad (11)$$

$$\rho = \sqrt{x^2 + y^2}, \theta = \tan^{-1}\left(\frac{y}{x}\right) \quad (12)$$

and $R_{nm}(\rho)$ is the radial polynomial given by (13), [8].

$$R_{nm}(\rho) = \sum_{s=0}^{n-|m|} (-1)^s \frac{(2n+1-s)!}{s!(n+|m|+1-s)!(n-|m|-s)!} \rho^{n-s} \quad (13)$$

where $0 \leq |m| \leq n$, $n \geq 0$. In this study, m is also used as $m \geq 0$, and it should be noticed that $V_{nm}(x,y) = 0$ when $x^2 + y^2 > 1$ is satisfied, and this condition provides us to extract features of the template from a circular area instead of square or rectangular one.

Using the aforementioned computations, pseudo-Zernike moments are calculated for both first and second window of the selected template. Hence, we obtain two features for two windows. In this work, two features are merged to acquire one feature vector which describes the selected template.

C. Template Matching Process

After obtaining the feature vector for the selected template, the next step is to find best matching template on the next window. Centers of the possible matching templates are selected within a range around the center point of the base template. Features of the all possible templates are extracted as described in (10)-(13), and to be able to find best matching template, SSD in (14) is used between the base template and the possible templates.

$$d(x,y) = \eta \times \left((\mathbf{f} - \mathbf{t}(x,y))^T (\mathbf{f} - \mathbf{t}(x,y)) \right) \quad (14)$$

where (x,y) is a point on the image corresponding to the center of the possible templates, $d(x,y)$ is the SSD between the feature vector \mathbf{f} of the base template and the feature vector \mathbf{t} of the template at (x,y) of the possible template. A scaling coefficient, $\eta = 10^{-4}$, is used to obtain tractably small values. The best matching template is the one corresponding to the minimum value in the vector of SSD values, d . Hence, the template center points (x,y) with minimum SSD is the selected point on the object being tracked.

In conjunction with the described feature extraction process, continuous update of the feature vector is required to make the system robust against illumination changes and

rotational motion of the object, which causes high changes on texture because of the 3D projection of the object on image frame. To avoid a possible failure in the tracking process, instead of updating at each frame, base feature vector for template matching is updated after every seven frames to decrease the risk of instability and false tracking points that are not related to tracked object. Additionally, to increase the performance of template matching, we start the search of best matching template around an expected point which is calculated after the sixth frame. Expected point is computed by examining three previous changes on x and y axis. Hence, the expected change, which will be added to the previous selected template center point to give the next search point as in (15), is determined.

$$E\{X\} = \frac{\sum_{k=1}^3 kX_k}{\sum_{k=1}^3 k} \quad \text{and} \quad E\{Y\} = \frac{\sum_{k=1}^3 kY_k}{\sum_{k=1}^3 k} \quad (15)$$

where $E\{X\}$ and $E\{Y\}$ are the expected changes on x and y axis respectively, X_k and Y_k are the previous frames indexed by k , i.e. $k=1$ means the oldest and $k=3$ means the closest previous frame. Hence, as shown in (15), the weight corresponding to the closest previous frame is larger than the others, meaning that its effect is more than the effects of others on choosing the next search point. Based on (15), the search point in the following frame can be expressed as in (16).

$$S(x,y) = T(x_c + E\{X\}, y_c + E\{Y\}) \quad (16)$$

where $S(x,y)$ is the calculated search point on the search frame, and $T(x_c, y_c)$ is the center point of the selected template on previous frame.

In addition, after calculating $d(x,y)$ around the expected search point $S(x,y)$, instead of using \mathbf{d} to find out the best matching template, we use a weighted distance measure to give a meaning to $S(x,y)$. Weighting operation on $d(x,y)$ is made using a Gaussian like function as given in (17).

$$g(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-x_c)(y-y_c)}{2\sigma^2}\right) \quad (17)$$

where (x,y) is a point on the template, (x_c, y_c) is the center point of the function template. After obtaining $g(x,y)$, minimum value of the function is normalized to unity as in (18).

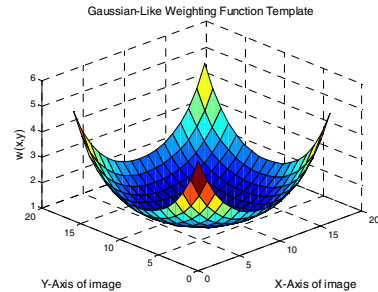


Fig. 7. Surface graph of the weighting function template

$$n(x,y) = g(x,y) / m(g) \quad (18)$$

In above, n is the weighting function template for d of (14), $m(g)$ is the minimum value in g . Fig. 7 shows the surface graph for n with size 17×17 adopted here. Weighted distances denoted by w are calculated using n as follows:

$$w(x, y) = d(x, y)n(x, y) \quad (19)$$

The best matching template is acquired by looking for the minimum of w . Therefore, the center point of the best matching template on an image frame with minimum weighted SSD value is marked to be the tracked point on the object.

III. EXPERIMENTAL RESULTS

In the tracking application described here, a point on object to be followed is designated before the proposed algorithm runs. Base template, which will be used in matching on following frames, is selected according to the chosen pivot point coinciding with the center of the template. The boundaries of the first base template do not exceed the contours of the predefined object in this work. Proposed algorithm is implemented on a test video used several times in the literature, [8, 12].

The mask size of the GW is chosen as 5×5 , which is determined experimentally among those having sizes 3×3 , 5×5 , 7×7 , and 9×9 . The criteria to choose 5×5 are minimum computation time and best tracking result. Size of the image to be represented using GW is 279×339 . Increasing the mask size may result in better representation of the image, yet this would introduce extra computational burden that is not preferable for real-time video processing applications. Higher mask size can result in better tracking but in return the frames per second (fps) of the system will decrease due to the computational requirements. Proposed method is successfully implemented on the system by selecting the maximum order $n=3$ and maximum repetition $m=3$ for the aforementioned pseudo-Zernike moments based feature extraction algorithm.

The philosophy behind the scene is to find the most similar template to the base template among possible various templates on the following frame. The possible templates are selected with the aid of the previous best matching template's center point. Assuming object motion is not bigger than 8 pixels on the image, the proposed method uses 289 center points to create 289 possible templates to find the most similar template with proposed algorithm. The most similar template to the base template among 289 possibilities is marked as the tracked point on the object. In addition, templates are used with double window selection for feature extraction as mentioned in section 2.2. The size of the first window is the template size which is 17×17 with a range of 8 pixels, and second window has the size as 9×9 with a range of 4 pixels from the template center point.

For a comparison, NCC algorithm [13, 14] and two stage template matching (TS-TM) algorithm discussed in [15] are

applied on the same test video with same template size and search area size. The work in [16] improves TS-TM by proposing an adaptive fuzzy weighted template matching (AF) algorithm with improved feature space. The experimental results of [16] were demonstrated with three different parameter settings in fuzzy module so we also included AF comparison with all presented settings. In NCC, TS-TM, AF, and proposed algorithms, next search point is determined by the previous selected point of the best matching template. Therefore, in all these four methods, a failure on tracking cannot be tolerated afterwards and the algorithms continue incorrect tracking on the subsequent frames. NCC is a fast method and it is invariant to illumination changes; however, invariance against rotation is not well preserved. We choose NCC to emphasize the importance of the rotation invariance on a tracking system.

In addition to the NCC, TS-TM, and AF, which can be grouped as the template matching algorithms, we also selected two different region trackers based on color histograms in our comparison to demonstrate that texture and shape descriptors of template can be as good as color tracker, even they can have more stable tracking performance under the high illumination variations. For this purpose, an improved mean-shift tracker [17] is selected, which is scale and orientation invariant. And, particle filter based color tracker [18, 3] is used on the test video as another region tracker. Results of the algorithms are examined based on the approximate center of the tracked object. Distance to the center point on each frame is defined as the error measure for the accuracy of the system. Comparison is made using the distance value from tracked points to approximate center of the object (20) where $e(t)$ is the distance between approximate center point and tracking point at frame t , $C(x, y, t)$ is the approximate center point of the object at frame t , and $T(x, y, t)$ is the track point obtained by tracking algorithms at frame t .

$$e(t) = \|C(x, y, t) - T(x, y, t)\| \quad (20)$$

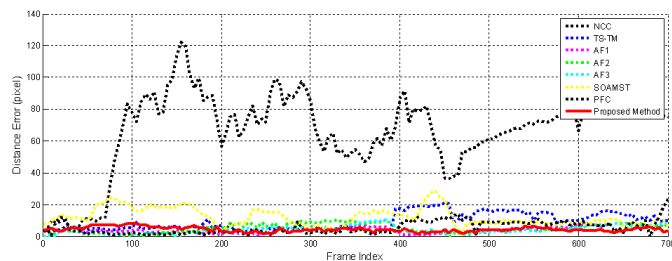


Fig. 8 Error based on distance measure for each method

Fig. 8 shows the error for each method calculated in the first frame and every 5 frames with respect to determined center points. This graph also shows that NCC fails, and proposed algorithm has a better tracking result than TS-TM algorithm used in this paper. On the other hand, it is obvious that TS-TM algorithm is faster than proposed algorithm. However, the tracking results show that it is not as stable as the proposed algorithm while tracking the object as shown

in Table 1. AF improves TS-TM but parameter selection is a difficulty to adjust membership functions. Template matching algorithms either suffer from the illumination changes or rotations, or long processing times. The latter is useful especially in applications where there are no processing time restrictions imposed by the real time task. In addition, color region trackers are successfully tracks the vehicle but centroid of the tracked region may vary a lot. In real time applications, the practical challenge is to attain a good performance with tractable computational intensity. Our system works at a rate of 10 to 11 fps, which is an acceptable performance value for time complexity of the proposed on real-time video processing application. In Fig. 9, the results obtained with the proposed method are shown. A van is monitored and the system marks the vehicle as shown in the snapshots of the journey. It is seen that the vehicle is marked accurately as there is a continuous update of base template making the surveillance system robust against orientation changes. Further to computational advantages, Table 1 also emphasizes the usefulness of the proposed method due to the low mean and low variance values on the error measure used as a basis for the comparison.

Table 1. Error Mean & Variance Comparison for Models

Compared Models	Mean	Variance
NCC [13, 14]	67.3538	665.9475
TS-TM [15]	9.1281	27.9134
AF1 [16]	4.0641	4.0284
AF2 [16]	4.9323	8.1342
AF3 [16]	4.2106	5.6246
SOAMST [17]	11.9946	30.2560
PFCT [18, 3]	5.5740	17.5628
Proposed Method	4.3690	2.0516

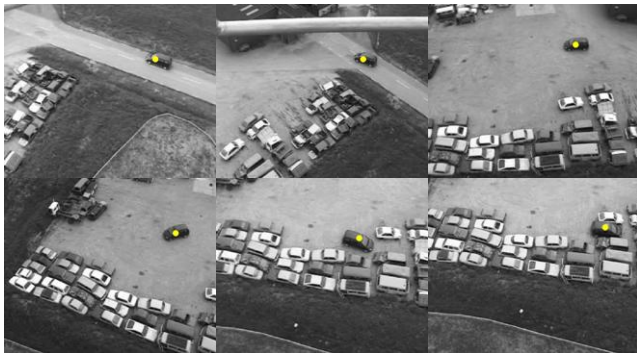


Fig. 9. Tracking results obtained with the proposed method

IV. CONCLUSIONS

This work presents the tracking a pre-defined object within the video frames. The system proves that the tracking task can be accomplished under the presence of illumination changes, noise and motion of both the object and the observer. A comparison with NCC, TS-TM, AF, SOAMST and PFCT algorithms is also provided to demonstrate the prominent performance obtained with the proposed

technique. According to the obtained results, gradient-based Gabor wavelet represented double-window template matching using pseudo-Zernike moment features is found a reliable method for real-time tracking applications. The contribution of this paper is the extraction and use of a set of texture features obtained from the GW representation of images to track a moving object in a moving scene. Future work of the authors has the goal of improving the performance by considering the information via modeling the motion and information describing the object.

V. REFERENCES

- [1] R. Cheng, W. Heinzelman, M. Sturge-Apple, and Z. Ignjatovic, "A motion-tracking ultrasonic sensor array for behavioral monitoring," *IEEE Sensors Journal*, vol.12, no.5, pp.707-712, March 2012.
- [2] S. Sarkka, V. V. Viikari, M. Huusko, and K. Jaakkola, "Phase-based UHF RFID tracking with nonlinear kalman filtering and smoothing," *IEEE Sensors Journal*, vol.12, no.5, pp.904-910, May 2012.
- [3] M. Talha and R. Stolkin, "Adaptive fusion of infra-red and visible spectra camera data for particle filter tracking of moving targets," *IEEE Sensors Conference*, Taipei, Taiwan, 2012, pp.1-4.
- [4] H. Liu, M. Yuan, F. Sun, J. Zhang, "Spatial neighborhood-constrained linear coding for visual object tracking," to appear in *IEEE Trans. on Industrial Informatics*, 2013.
- [5] H. Liu, S. Chen, N. Kubota, "Intelligent video systems and analytics: A survey," to appear in *IEEE Trans. on Industrial Informatics*, 2013.
- [6] C. Liu, "Gabor-based Kernel PCA with Fractional Power Polynomial Models for Face Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.26, no.5, pp.572-581, 2004.
- [7] C. Liu, "Gabor-based Kernel PCA with Fractional Power Polynomial Models for Face Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.26, no.5, pp.572-581, 2004.
- [8] Kaàniche, K., Champion, B., and Pégard, C., and Vasseur, P., *A Vision Algorithm for Dynamic Detection of Moving Vehicles with a UAV*, In *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.1878-1883, 2005.
- [9] R. C. Gonzales and R. E. Woods, *Digital Image Processing*, Prentice Hall, Upper Saddle River, New Jersey, pp.165-168, 2008.
- [10] Chong, C. W., Mukundan, R., and Raveendran, P., *An Efficient Algorithm for Fast Computation of pseudo-Zernike Moments*, *International Journal of Pattern Recognition and Artificial Intelligence*, v.17, no.6, pp.1011-1023, 2003.
- [11] A. Khotanzad and Y. H. Hong "Invariant Image Recognition by Zernike Moments," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.12, pp.489-497, 1990.
- [12] <http://www.comets-uavs.org/results/perception.shtml>
- [13] J. Wu, H.-J. Yue, Y.-Y. Cao, Z.M. Cui, "Video object tracking method based on normalized cross-correlation matching," *Ninth Int. Symp. on Distributed Computing and Applications to Business, Engineering and Science*, 2010, pp.523-527.
- [14] M. Asgarizadeh, H. Purgassem, G. Shahgholian, "Robust object tracking using regional mutual information and normalized cross correlation," *Fourth Int. Conf. on Computational Intelligence and Communication Networks*, 2012, pp.411-415.
- [15] Choi, M.-S. and Yui, W.-Y., *A Novel Two Stage Template Matching Method for Rotation and Illumination Invariance*, *Pattern Recognition*, v.35, pp.119-129, 2002.
- [16] N. Imamoglu, A. Eresen, M. O. Efe, "Adaptive fuzzy weighted template matching using invariant features for a tracking application," *10th IFAC Int. Workshop on Adaptation and Learning in Control and Signal Processing (ALCOSP)*, 2010, pp.259-264.
- [17] J. Ning, L. Zhang, D. Zhang, C. Wu, "Scale and Orientation Adaptive Mean Shift Tracking", *IET Computer Vision*, vol.6, no.1, pp.52-61, 2012
- [18] Nummiaro, k., Koller-Meier, E., Van Gool, L., "An adaptive color-based particle filter," *Image and Vision Computing*, pp.99-110, 2003.