Detection of Small Moving Objects Using a Moving Camera*

Moein Shakeri[1] and Hong Zhang[2]

Abstract— **In recent years, various background subtraction methods have been proposed and used in vision systems for moving object detection and tracking from moving cameras; however, most of them have difficulty in handling small and distant objects in complicated non-flat scenes. This paper presents a robust method to effectively segment moving objects from videos, captured by a camera on a moving platform. In our approach, a two-level registration is applied to estimate the effect of camera motion for motion compensation. After motion estimation and extraction of potential foreground pixels by Gaussian mixture model, noisy result is refined using component based and pixel based methods the latter of which uses the hidden markov model (HMM) for classifying pixels. Finally, foreground objects are tracked by a particle filter to exploit the temporal coherence of foreground motion and improve the detection accuracy through time. Experimental results show that our method outperforms competing methods for detecting moving objects in complex environments.**

I. INTRODUCTION

Detecting and tracking objects are very important topics of research in surveillance and monitoring applications. Many methods have been proposed for moving object detection with a stationary camera (e.g. [1], [2], [3], [4]). These methods are stable and real-time for indoor and outdoor applications; however, they cannot detect a moving object while the camera moves. Nowadays, a camera on a moving platform is pervasive, especially in robotics applications in which a mobile robot is required to detect moving objects while in motion. Therefore, more and more researchers are working on detecting and tracking objects using moving cameras. Existing methods have other applications in different domains such aerial surveillance, video segmentation, smart cars and driver assistance.

In this paper, we introduce a robust method to extract and track moving objects from a non-stationary platform. The main contributions of this paper are that we (a) estimate a motion model using wavelet transform to update the GMM background model (b) provide a learning framework based on HMM that classifies foreground and background pixels by integrating previous of pixel labels, and (c) propose to use a particle filter as the backend of the detection algorithm to track the foreground pixels.

The remainder of this paper is organized as follows. In Section II, we discuss related works to extract moving objects from a moving camera. In Section III, we introduce an approach to detect moving objects from a moving platform. Comparative experimental results based on existing datasets are described in Section IV and finally Section V summarizes our approach and concludes the paper.

*This work was supported by NCFRN.

[1]Moein Shakeri with the Department of Computing Science, University of Alberta, Edmonton, Canada. shakeri@ualberta.ca

[2]Hong Zhang is with faculty of Computing Science, University of Alberta, Edmonton, Canada. hzhang@ualberta.ca

II. RELATED WORKS

A common approach to detecting moving regions relies on the assumption of a planar scene as in the application of airborne image sequences analysis for moving object detection [5]. In [5] an adaptive particle filter and the EM algorithm are used to detect moving objects. This method can only be applied for simple motion and scene, whereas in practice scenes can be complicated and have objects with different depths. As an alternative, Wang et. al. [6] used optical flow between every two adjacent frames to obtain the motion information for each pixel. Based on the motion information, they transferred the background model in the previous frame in order to extract the foreground in the current frame. The above-mentioned methods [5], [6] assume a scene of a simple plane. This assumption is valid only for aerial videos captured by an air-borne vehicle. [7] presented a method utilizes multiple 2D affine transformations to describe the background motion by a multi-classes RANSAC method to solve the above limitation; however, this method just uses two successive frames. As a result, when the objects move slowly, only the edges can be detected.

To overcome the limitation of the planarity assumption, Yamaguchi et al. [8] proposed a moving object detection method by computing the ego-motion of the vehicle with a monocular camera and reconstructing the 3D scene around the vehicle. The method was successfully applied for the detection of moving vehicles or pedestrians on the road. [9] used multiview geometric constraints to detect objects. The approach is non-casual since future information is required which cannot be known at the current time. These methods depend on an analysis of a reconstructed depth map, which is a difficult problem by itself, and they are not expected to work well for distant, small object detection, given the fundamental limitation of structure-from-motion algorithms.

Without explicit 3D reconstruction, [10] presented an interesting idea of performing moving object detection by clustering the motion of the pixel trajectories, embedded in a low dimensional space. These trajectories provide a sparse labeling of the video and are used to build the background and the foreground models. The models are used to estimate the maximum posteriori pixel-wise labeling of the video. This method has two limitations: (a) due to the dependence on long term trajectories only, regions with no trajectories may be discarded as background altogether, and (b) their use of orthographic camera model means that

the motion information can be incomplete and the method may fail to capture object boundaries. To overcome the limitations of [10], Elqursh *et. al* [11] introduced a method that accurately models appearance and motion to achieve robust moving camera background subtraction. This method merges the long term trajectories to accurately model long term motion dependencies using a Gaussian mixture model, and a Bayesian filtering framework for pixel-level appearance models to estimate the regions as foreground or background. The Bayesian tracking framework is an effective step, which we adopt in our algorithm, although the idea of pixel trajectory clustering cannot deal with a small population size when moving objects are small.

Barnich *et. al* [12] proposed a novel method for background subtraction called ViBe which speeds up the parameter estimation step in modeling the background appearance by that traditionally depends on past images instead, only observed pixel values in the current image are used to approximate the temporal history of a pixel and as a result it provides fast responses to changing events in the background when for example the focal length of the camera changes.

In this paper, we propose a method to detect and track moving objects without requiring an explicit 3D reconstruction of the scene. Instead, the proposed method performs motion compensation by two kinds of registration methods, popularly used in medical image analysis, on wavelet components in two levels, with a combination of a component-based technique, and a pixel-based learning framework based on HMM. Finally, we optimize the performance of our algorithm by employing a pixel-wise particle filter as the backend to the entire detection process.

III. PROPOSED METHOD

In this section we present a summary of our approach which contains four major parts: motion recovery, background modeling, refinement of candidate detections, and tracking. The block diagram of our proposed method is shown in Fig. 1. First of all, using rigid and non-rigid registration methods on wavelet coefficients in the motion compensation step, a transformation T for warping the background model in the previous frame to that in the current frame can be estimated. Next, T is used to update the background model. Then based on observations from the previous frames, we can compute the probability of being foreground/background for each pixel in the current frame and then update the background model. Finally, to finalize the detection of foreground pixels that belong to the moving objects, a particle filter is applied. In the rest of this section we explain each part of the proposed method in details.

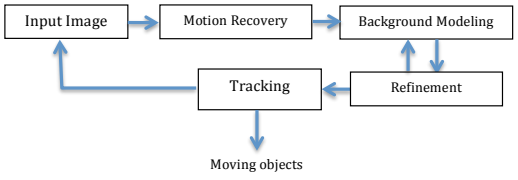

Fig. 1. The process flow of our proposed method.

A. Motion Recovery

Many methods have been proposed to compute motion between two images. Some of them work by finding features from images in the spatial domain [13], [14], [15] and the other group of algorithms work in the frequency domain [16], [17], [18], [19]. Images are typically registered by maximizing a similarity function, either locally or globally.

Displacement of each pixel between the current frame and the previous frame using wavelet coefficients begins with global motion estimation by a rigid registration method and in the second step, we use a non-rigid method to register pixels locally to perform local motion compensation. For global image registration, we make use of the wavelet transform, which decomposes an image into various sub images based on local frequency contents. Two dimensional discrete wavelet transform (DWT) can be obtained by applying DWT across rows and columns of an image. We use wavelet transform because of two main reasons. First, multi-resolution registration methods are faster in terms of generating results and this is mainly due to the low resolution of images at multi resolution level. Second, wavelet makes better results in comparison with other multi-resolution approaches such as Gaussian pyramid. Wavelet does not blur the images through the hierarchical pyramid as much as Gaussian pyramid [20]. We perform wavelet transform at two levels. We start the registration process from the coarsest level, i.e. level 2. The input image at each level is split into 4 bands ($Lo\_Lo = y_{00}, Lo\_Hi = y_{01}, Hi\_Lo = y_{10}$, and $Hi\_Hi = y_{11}$) using the lowpass and highpass wavelet filters on the rows and columns in turn. The Lo_Lo band subimage y_{00} is then used as the input image to the next level (Fig. 2). Subimages $y_{00,00}$ in Fig. 2 from the current and the previous frames are used to estimate the global motion model.

1) Global Motion Model: The global motion describes the overall motion of the current frame with respect to the previous frame due to camera motion. We compute the global motion based on affine transformation on subimages $y_{00,00}$ between the current and the previous frames [13]. Then, we update $y_{00,00}$ and other three wavelet coefficients $y_{00,01}, y_{00,10}$, and $y_{00,11}$ by the estimated affine transformation matrix, T_{global}. Now, we can reconstruct an approximated

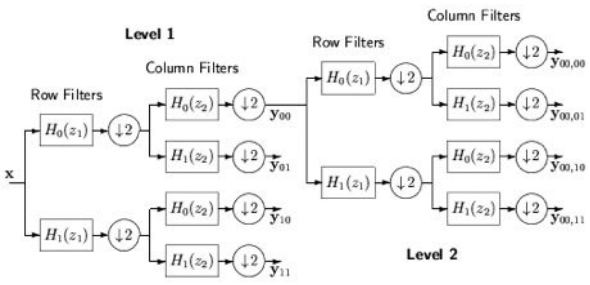

Fig. 2. Wavelet decomposition in 2 levels. We use $y_{00,00}$ for rigid registration to compute the global motion model

image in level 1. Also, we rescale the transformation matrix by a factor of 2, because we need this transformation matrix as the global motion model for updating the background model parameters in Section III-B and the size of the background model parameters should be the same as size of transformation matrix T_{global}.

2) Local Motion Model: The affine transformation captures only the global motion of the scene. In order to improve the registration accuracy and account for a non-plane scene, an additional transformation is required, which models the local deformation of the scene. The nature of the local deformation of the scene can vary significantly across the depth of the objects in the scene. To define the local deformation we use a free-form deformation (FFD) model, based on B-splines [21], [22], which has been previously applied to the tracking and motion analysis in cardiac images [23]. The resulting deformation controls the shape of the scene and produces a smooth transformation.

To deform a spline-based FFD, we denote the image size y_{00} with $0 \leq x \leq X$ and $0 \leq y \leq Y$, where X and Y are the number of rows and columns, respectively. Let Ψ denote an $n_x \times n_y$ mesh of control points $\psi_{i,j}$ with uniform spacing ρ. So, the FFD can be written as follows:

$$T_1(x,y) = \sum_{m=0}^{3}\sum_{l=0}^{3} B_m(u)B_l(v)\psi_{i+m,j+l} \quad (1)$$

where $i = \lfloor x/n_x \rfloor - 1, j = \lfloor y/n_y \rfloor - 1, u = x/n_x - \lfloor x/n_x \rfloor, v = y/n_y - \lfloor y/n_y \rfloor$ and where B_m represents the mth basis function of the B-spline [21], [22] defined by the following equations.

$$B_0(u) = (1-u)^3/6 \quad , \quad B_1(u) = (3u^3 - 6u^2 + 4)/6$$
$$B_2(u) = (-3u^3 + 3u^2 + 3u + 1)/6 \quad , \quad B_3(u) = u^3/6 \quad (2)$$

In contrast to thin-plate splines [24] or elastic-body splines [25], B-splines are locally controlled which makes them computationally efficient even for a large number of control points. In particular, the basis functions of B-splines have a limited support, i.e., changing control point $\psi_{i,j}$ affects the transformation only in the local neighborhood of that control point [13]. To compute the best motion model, both affine and FFD algorithm need to maximize the similarity functions.

After computing local motion model T_{local}, the motion model T in level one is computed by the sum of the local and global motion model as follows [13]:

$$T(x,y) = T_{global}(x,y) + T_{local}(x,y) \quad (3)$$

The motion model T is rescaled to generate a motion model with the same size as the input frame.

3) Optimization: to find the optimal transformation, we minimize a cost function associated with the global transformation parameters, as well as the local transformation parameters. The cost function for global transformation is the image similarity ζ as (4) where [13], [26] suggested the use of normalized mutual information (NMI) as a measure of image alignment.

$$\zeta(A,B) = \frac{H(A) + H(B)}{H(A,B)} \quad (4)$$

where $H(A)$, and $H(B)$ denote the entropies of images A, B, and $H(A,B)$ denotes their joint entropy, which is calculated from the joint histogram of A and B.

Also, the cost function for local transformation is computed by:

$$\zeta_{smooth} = \frac{1}{V}\int_0^X \int_0^Y [(\frac{\partial^2 T_1}{\partial x^2})^2 + (\frac{\partial^2 T_1}{\partial y^2})^2 + 2(\frac{\partial^2 T_1}{\partial xy})^2] dxdy \quad (5)$$

where V denotes the image size in pixels. ζ_{smooth} defines a cost function which is associated with the smoothness of the transformation. This quantity introduces a penalty term which regularizes the transformation. Non-rigid transformation iterates till $\|\nabla\zeta_{smooth}\| < \varepsilon_1$. After each iteration the control points Ψ are recalculated by $\Psi = \Psi + \lambda \frac{\nabla\zeta}{\|\nabla\zeta\|}$ [13].

B. Background Modeling

We use the standard adaptive Gaussian mixture model for background model, and it is summarized in this section for the completeness of our algorithm description. In general, we can estimate the background model by $\hat{p}(\vec{x}^{(t)}|x_T, BG) \sim \sum_{m=1}^{M} \hat{\pi}_m \eta(\vec{x}; \hat{\mu}_m, \hat{\sigma}_m^2 I)$. where $\hat{\mu}_m$ and $\hat{\sigma}_m^2, m = \{1,...,M\}$ are the estimates of the means and variances of the Gaussian components, respectively. The identity matrix I has proper dimensions, and η is a Gaussian probability density function.

After motion estimation, transformation matrix T is used to move the position of the GMM model parameters from the previous frame to the current frame. After this step of motion compensation, we update the GMM model by [4]. GMM provides the initial background model through time. This initial background model should be updated for the next frame before being used to extract moving objects in the current frame.

C. Refinement

After background subtraction by GMM, we have a binary image with some noisy detected moving objects. To reduce false-positive detections, we use component based and pixel based methods to further process the detected foreground pixels. First a component based method is used for each foreground connected component and then a pixel based method computes the probability of being foreground for each pixel by hidden markov model (HMM) trained over previous observations.

1) Component based refinement: In this step, we compare the similarity between the background model and the current frame where the connected components are considered as potential foreground objects. Let us denote connected components by CC_i, $i = 1,...,N$, where N is the number of connected components. Also, $L^t(CC_i, A)$ is the area of the connected component i at time t in image A. We filter the connected components by the following rule:

$$If\ corr(L^t(CC_i, CF), L^t(CC_i, BGM)) > Th_s$$
$$L^t(CC_i, BGM) = L^t(CC_i, CF),\ and\ \ CC_i = 0 \quad (6)$$

where $corr(.,.)$ computes correlation between two areas and Th_s is a chosen threshold. CF and BGM also are the current

frame and the background model, respectively. As the result of evaluating (6), some foreground pixels will be changed to background and, as a result, we must also reflect this change when updating the background model by:

$$\begin{cases} \hat{\mu}_m(x,y) = I(x,y), & for\ all\ m\ components \\ \hat{\sigma}_m^2 = c \end{cases} \quad (7)$$

where $I(x,y)$ is the intensity value of pixel (x,y) and c is the initial constant value. We set $c = 10$ as in [4].

2) Pixel based refinement: After coarse refinement of the connected components, we finalize our background model and detect moving objects to track. In order to decide the label of each pixel for being foreground or background at time t, the hidden markov model (HMM) is employed. With HMM, the state of a pixel at time t encapsulates all we need to know about the history of the pixel in order to predict its future label. Let us denote the observation at time t by the variable O_t. Also, the variable S_t refers to the state of each pixel at time t, which is a hidden variable. Each observation O_t may take values in $\{0,255\}$ and each hidden variable S_t may take state values in $\{b,f\}$. The joint probability distribution $p(O,S_T)$ of the whole set of variables $\{o,s\}$ is parameterized by three parameters: the transition matrix A, containing the transition probabilities $a_{kl} = p(S_t = k|S_{t-1} = l)$, the observation emission matrix B, containing the symbol emission probabilities $b_{kr} = p(O_t = r|S_t = k)$, and the initial state probabilities $\pi_k = p(S_0 = k)$. The probability is computed by:

$$p(O,S_T) = \pi_{s_0} \prod_{t=1}^{T-1} a_{s_t s_{t-1}} \prod_{t=0}^{T-1} b_{s_t o_t} \quad (8)$$

In our method, the initial state probabilities are $\pi_k = p(S_0 = k) = 0.5, k = \{b,f\}$ and the transition matrix A is $[\beta_1,\ \beta_2; \beta_3,\ \beta_4]$ in Matlab format, which means $p(S_t = b|S_{t-1} = b) = \beta_1, p(S_t = f|S_{t-1} = f) = \beta_4,\ p(S_t = b|S_{t-1} = f) = \beta_2$ and $p(S_t = f|S_{t-1} = b) = \beta_3$. Also, we assume the emission probabilities are $p(O_t = r|S_t = b) = p(O_t = r|S_t = f) = 0.5$.

To find the most likely state for any pixel in time, the forward algorithm can be used. Forward probability is computed by:

$$\hat{f}_{1:T}(s_i) = P(S_T = s_i|O_{1:T}) = \frac{P(o_1,o_2,...,o_T,S_T = s_i)}{P(o_1,o_2,...,o_T)} \quad (9)$$

where $\hat{f}_{1:T}$ are forward probabilities over T observations, respectively. Also, $\hat{f}_{0:0} = P(S_0 = k) = [0.5, 0.5]$. We use Eq. (10) to classify each pixel as background or foreground at time t.

$$\phi = log \frac{P(S_T = f|O_{1:T})}{P(S_T = b|O_{1:T})} \quad (10)$$

where for $\phi > 0$ the pixel is considered as foreground and it is otherwise background. Also, the background models of those pixels whose states are changed should be revised by Eq. (7). Now the background model can be used to extract moving objects from next frame.

D. Tracking

After the refinement step, there remain some false positive foreground pixels, especially in complicated scenes with noisy objects like leaves on trees. In order to further improve the performance of our method, we resort to tracking to exploit the temporal coherence constraint of foreground objects. Specifically, we apply a particle filter as a real-time tracker to find and track the moving objects from the pixels that are classified as foreground by HMM. Different implementations for particle filters exist in the literature. In this paper we use a generic particle filter [27], as summarized in Algorithm 1. In the algorithm, let $\{X_{0:k}^i, W_k^i\}_{i=1}^{N_s}$ denote the posterior pdf $p(X_{0:k}|Z_{1:k})$, where $\{X_{0:k}^i, i = 0,...,N_s\}$ is a set of support points with associated weights W_k^i and $X_{0:k}$ is the set of all states up to time k. Also, $p(.)$ and $q(.)$ denote the posterior and importance density functions. Z_k and N_{eff} show the observation at time k and the effective sample size, respectively. In this paper we maintain a particle filter for each foreground pixel where X_k and Z_k are the state and the observation of the pixel at time k.

IV. Experimental Results

The experimental results in this section demonstrates the utility of our proposed method in indoor and outdoor applications. All of these experiments are done in Centre for Intelligent Mining Systems (CIMS) at the University of Alberta and all implementations are done by MATLAB and C++. $\beta_1, \beta_2, \beta_3$ and β_4 in matrix A are set at 0.8, 0.2, 0.2 and 0.8, respectively, and Th_s in component-based refinement is set to 0.98 in our experiments. In the FDD algorithm we set the maximum number of iterations to 10. Furthermore, to keep FFD from deviating significantly from the background model, we set the number of control points to the one half of the number of rows times one half of the number of columns.

We tested our method on different video sequences taken in our campus, in the ViBe dataset from http://www2.ulg.ac.be/telecom/research/vibe, in the PETS dataset, and in the Freiburg-Berkeley dataset from http://lmb.informatik.uni-freiburg.de/resources/datasets.

Fig. 3 shows the results from one experiment in the indoor environment where a person moves across the scene and is detected by a moving camera. Rows (a) and (b) illustrate

Algorithm 1 Generic Particle Filter

1: $[\{X_k^i, W_k^i\}_{i=1}^{N_s}] = PF[\{X_{k-1}^i, W_{k-1}^i\}_{i=1}^{N_s}, Z_k]$
2: **for** $i = 1$ To N_s **do**
3: $\quad X_k^i \sim q(X_k|X_{k-1}^i, Z_k)$ and $W_k^i \propto W_{k-1}^i \frac{p(Z_k|X_k^i)p(X_k^i|X_{k-1}^i)}{q(X_k^i|X_{k-1}^i,Z_k)}$
4: **end for**
5: **for** $i = 1$ To N_s **do** Normalize W_k^i
6: **end for**
7: Calculate $\hat{N_{eff}} = \frac{1}{\sum_{i=1}^{N_s}(W_k^i)^2}$
8: **if** $\hat{N_{eff}} < N_T$ **then**
9: $\quad [\{X_k^i, W_k^i\}_{i=1}^{N_s}] = Resample[\{X_k^i, W_k^i\}_{i=1}^{N_s}]$
10: **end if**

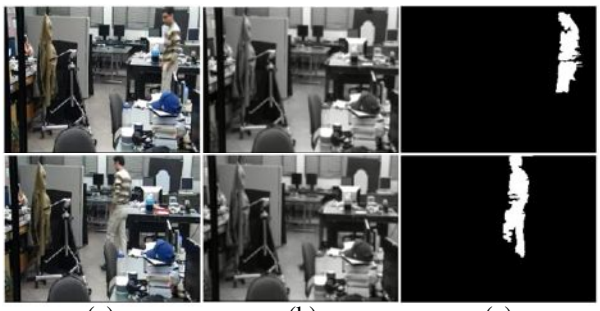

Fig. 3. Results of detecting a moving object by moving camera in indoor environment

selected two frames from video sequences. First column of Fig. 3 shows input frames in two different times and second column shows their background model computed by our method. The final column illustrates the results of the extracted moving object from the scene. In this experiment, the background is complicated with lots of edges. Most of methods have some difficulty with these kinds of backgrounds, when the camera moves, but our method can detect the moving object accurately. In this experiment, the color of the moving object is similar to parts of the background, and successful detection of the object in such a background shows the robustness of our method in difficult situations.

We have also conducted experiments in a video surveillance application in the outdoor environment and compared the results with [7] on a video in the PETS dataset. Fig. 4 shows some example results on outdoor videos to extract and track small, fast and distant objects. Second and third columns of Fig. 4 display foreground pixels by simple affine and multiple 2D affine transformation from [7], respectively. Column (d) shows the results of our proposed method based on two-level wavelet registration by combination of the affine and the FFD algorithm. This experiment shows the robustness of the registration technique in our method in comparison with two other methods.

Also, we compared results of our method with the ViBe method [12]. Fig. 5 shows the result on their dataset where the first column displays two sample frames in a video sequence, captured by a camera with a variable focal length.

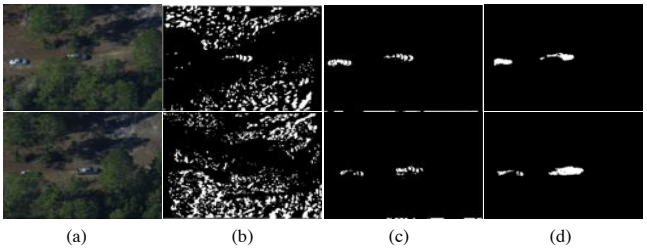

Fig. 4. Each column shows two selected frames from input video. Column (a) shows the input images. Column (b) illustrates the foreground pixels by affine transformation. Columns (c) and (d) show the result of multiple 2D affine transformation from [7], and our proposed method, respectively.

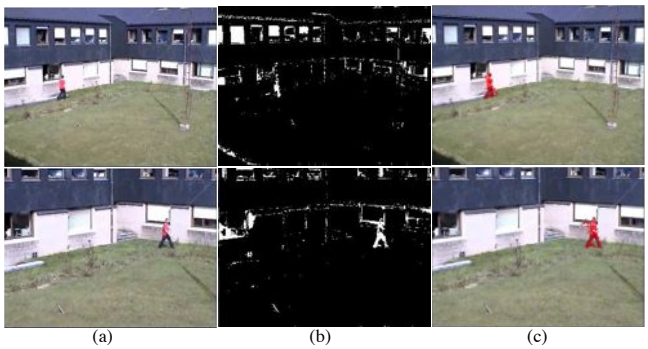

Fig. 5. Left column (a) shows input selected frames with variable focal length. Middle column (b) and right column (c) show detection results of the ViBe technique and our proposed method, respectively.

The middle and the last columns of Fig.5 are results of the ViBe and our method, respectively. The detected foreground objects in the last column are marked by red particles. Clearly, our method is able to handle edges much better than [12], which mistook many background pixels as foreground. We attribute this largely to the use of a particle filter that is less affected by momentary mistakes than ViBe is.

Also, we applied the proposed method on a challenging image sequence from the Freiburg-Berkeley dataset. Fig. 6 shows the results of our method in comparison with methods from [10] and [11]. First and second columns of Fig. 6 display the results from their methods, respectively and the last column illustrates the result of our proposed method. In this case, columns (a) and (b) show significant false positive pixels whereas our method is able to remove those pixels from the foreground. In addition to the qualitative results shown in figures 5 and 6, we also provide the qantitative comparison of the aforementioned methods including [12], [10], [11], with our proposed method in table I, on two benchmark video sequences. Three common metrics; namely, Percision, Recall and F1-score [11] are used for evaluation. We can infer that our method is superior to all the evaluated

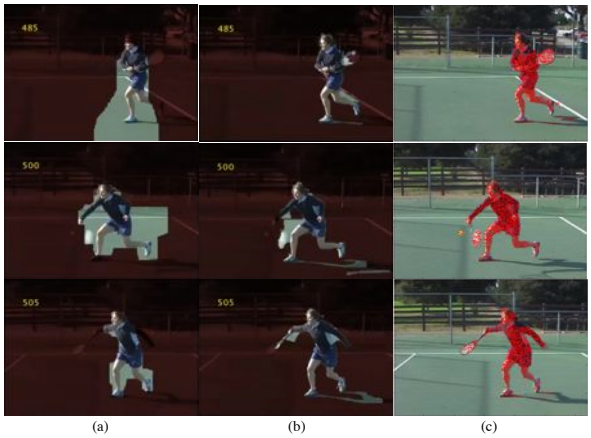

Fig. 6. Columns (a) and (b) show the detection results of [10] and [11], respectively. Last column displays the results of our proposed method in comparison with two other methods.

TABLE I

QUANTITATIVE COMPARISON OF DIFFERENT METHODS

Dataset	Methods	Prec	Rec	F1
ViBe dataset	Our method	0.95	0.98	0.97
(Fig. 5)	Barnich et al. [12]	0.13	0.71	0.22
Freiburg-Berkeley	Our method	0.89	0.90	0.89
Tennis video sequences	Elqursh et al. [11]	0.86	0.92	0.89
(Fig. 6)	Sheikh et al. [10]	0.27	0.83	0.40

methods. Finally, we examine video sequences acquired by an aerial vehicle and by a hand held moving camera looking downwards from a building. First row of Fig. 7 shows the detection of moving objects in an aerial video with fast zoom variation. Second row of Fig. 7 illustrates results on videos from a hand-held camera that is moving freely. Column (c) in the second row of Fig. 7 shows three moving objects in which third object is hard to see or detect even by a stationary camera; however, our proposed method can detect the third object when the camera is moving.

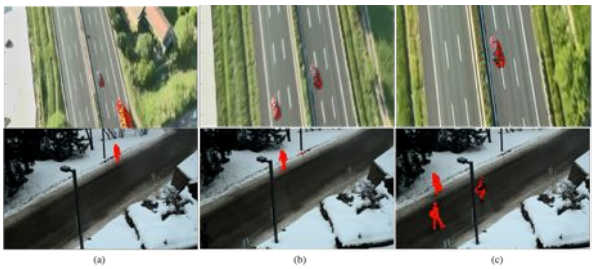

Fig. 7. First row shows results of the fast, small and distant object on video sequences captured by a fast moving cameras with variable focal length through time. Second row illustrates results of distant object on a video captured by freely moving handy-held camera.

V. CONCLUSION

We have presented a method to extract moving objects from a moving platform in which the motion model is estimated by a two-level registration method with discrete wavelet transform and that model is then used to align images and update the GMM background model for extracting foreground pixels. We propose to suppress false positive foreground pixels effectively with component-level and pixel-level techniques. Finally, a particle filter is used to detect and track moving objects in order to further improve the detection accuracy. Our experiments on different challenging video sequences validate our proposed method for indoor and outdoor applications. In particular, our registration method for motion compensation combines global and local techniques and generates higher quality foreground pixel labels. As well, our particle filter backend is shown to be effective to eliminate false-positive detections by integrating temporal history of pixel labels.

REFERENCES

[1] C. Stauffer, W.E.L. Grimson, Adaptive background mixture models for real-time tracking in IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1999, vol.2, 1999

[2] L. Li, W. Huang, I. Gu, and Q. Tian, Foreground object detection from videos containing complex background, in Proceedings of the ACM international conference on Multimedia, 2003, pp. 210.

[3] A. Elgammal, R. Duraiswami, D. Harwood, L. S. Davis,"Background and foreground modeling using nonparametric kernel density estimation for visual surveillance", IEEE Transaction, 90(7),1151-1163, 2002

[4] Z. Zivkovic "Improved adaptive Gaussian mixture model for background subtraction", Proc. Int. Conf. Pattern Recognit.,pp.28 -31 2004.

[5] B. Jung and G. Sukhatme, Detecting moving objects using a single camera on a mobile robot in an outdoor environment, in International Conference on Intelligent Autonomous Systems, 2004, pp. 980987.

[6] Y. Wang, Z. Zhang, Y. H. Wang,"Moving object detection in aerial video", In Machine Learning and Applications (ICMLA), 11th International Conference on (Vol. 2, pp. 446-450). IEEE, 2012

[7] X. Zhang, S. Wang, X. Ding, "Beyond dominant plane assumption: Moving objects detection in severe dynamic scenes with Multi-Classes RANSAC", ICALIP, IEEE, 2012.

[8] K. Yamaguchi, T. Kato, and Y. Ninomiya. Vehicle egomotion estimation and moving object detection using a monocular camera. In IEEE International Conference on Pattern Recognition, volume 4, 2006.

[9] J. Kang, I. Cohen, G. Medioni, and C. Yuan. Detection and tracking of moving objects from a moving platform in presence of strong parallax. In IEEE International Conference on Computer Vision, 2005.

[10] Y. Sheikh, O. Javed, and T. Kanade, "Background subtraction for freely moving cameras", In ICCV, pages 1219-1225. IEEE, 2009.

[11] A. Elqursh, A. Elgammal, "Online moving camera background subtraction", In Computer VisionECCV, (pp. 228-241). Springer Berlin Heidelberg, 2012.

[12] O. Barnich, V. Droogenbroeck,ViBe: A universal background subtraction algorithm for video sequences, IEEE Transaction on Image processing, vol.20(6), pp:1709-1724, June 2011.

[13] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, D. J. Hawkes, , "Nonrigid registration using free-form deformations: application to breast MR images.", Medical Imaging, IEEE Transactions on, 18(8), 712-721., 1999

[14] H. Wang, L. Dong, J. O'Daniel, R. Mohan, A. S. Garden, K. Ang, R. Cheung, R., "Validation of an accelerated'demons' algorithm for deformable image registration in radiation therapy.", Physics in Medicine and Biology, 50(12), 2887, 2005

[15] Y. Zheng, C. Zhiguo, X. Yang "Multi-spectral remote image registration based on SIFT" Electronics Letters 44, no. 2, pp:107-108, 2008

[16] D. Sasikala, R. Neelaveni, R.,"Image registration using modified adaptive polar transform". Proc. Computer Science, 2, 321-329, 2010

[17] R. K. Brown, L. Roger, "Image registration using redundant wavelet transforms" (Master's thesis, Air Force Institute of Technology), 2001

[18] L. Mainardi, K. M. Passera, A. Lucesoli, D. Vergnaghi, G. Trecate, E. Setti, S. Cerutti, S. , "A nonrigid registration of MR breast images using complex-valued wavelet transform.", Journal of Digital Imaging, 21(1), 27-36,2008

[19] B. S. Reddy, B. N. Chatterji, " An FFT-based technique for translation, rotation, and scale-invariant image registration", Image Processing, IEEE Transactions on, 5(8), 1266-1271, 1996

[20] J. Wu, A. C. Chung, "Multimodal brain image registration based on wavelet transform using SAD and MI". In Medical Imaging and Augmented Reality (pp. 270-277). Springer Berlin Heidelberg,2004.

[21] S. Lee, G. Wolberg, K.-Y. Chwa, and S. Y. Shin, Image metamorphosis with scattered feature constraints, IEEE Trans. Visualization Comput. Graph., vol. 2, pp. 337354, Oct. 1996.

[22] S. Lee, G. Wolberg, and S. Y. Shin, Scattered data interpolation with multilevel B-splines, IEEE Trans. Visualization Comput. Graph., vol.3, pp. 228244, July 1997.

[23] E. Bardinet, L. D. Cohen, and N. Ayache, Tracking and motion analysis of the left ventricle with deformable superquadrics, Med. Image Anal., vol. 1, no. 2, pp. 129149, 1996.

[24] F. L. Bookstein, Principal warps: Thin-plate splines and the decomposition of deformations, IEEE Trans. Pattern Anal. Machine Intell., vol.11, pp. 567585, June 1989.

[25] M. H. Davis, A. Khotanzad, D. P. Flamig, and S. E. Harms, A physicsbased coordinate transformation for 3-D image matching, IEEE Trans. Med. Imag., vol. 16, no. 3, pp. 317328, May 1997.

[26] C. Studholme, D. L. G. Hill, and D. J. Hawkes, An overlap invariant entropy measure of 3D medical image alignment,Pattern Recognit., vol. 32, no. 1, pp. 7186, 1998.

[27] M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, T., " A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking". Signal Processing, IEEE Transactions on, 50(2), 174-188, 2002