Fast Moving Object Detection from Overlapping Cameras

Mikaël A. Mousse[1,2], Cina Motamed[1] and Eugène C. Ezin[2]
[1]*Laboratoire d'Informatique Signal et Image de la Côte d'Opale, Université du Littoral Côte d'Opale, Calais, France*
[2]*Unité de Recherche en Informatique et Sciences Appliquées, Institut de Mathématiques et de Sciences Physiques, Université d'Abomey-Calavi, Abomey Calavi, Bénin*

Keywords: Motion Detection, Codebook, Homography, Overlapping Camera, Information Fusion.

Abstract: In this work, we address the problem of moving object detection from overlapping cameras. We based on homographic transformation of the foreground information from multiple cameras to reference image. We introduce a new algorithm based on Codebook to get each single views foreground information. This method integrates a region based information into the original codebook algorithm and uses CIE L*a*b* color space information. Once the foreground pixels are detected in each view, we approximate their contours with polygons and project them into the ground plane (or into the reference plane). After this, we fuse polygons in order to obtain foreground area. This fusion is based on geometric properties of the scene and on the quality of each camera detection. Assessment of experiments using public datasets proposed for the evaluation of single camera object detection demonstrate the performance of our codebook based method for moving object detection in single view. Results using multi-camera open dataset also prove the efficiency of our multi-view detection approach.

1 INTRODUCTION

In computer vision community, the use of multi-camera takes a lot of scope. Indeed, motivations are multiple and concern various fields as monitoring and surveillance of significant protected sites, control and estimation of flows (car parks, airports, ports, and motorways). Because of the fast growing of data processing, communications and instrumentation, such applications become possible. These kind of systems require more cameras to cover overall field-of-view. They reduce the effects of objects dynamic occlusion and improve foreground zone estimation accuracy. The performance of multi-camera surveillance systems depends on the quality of each single view object detection algorithm and on the quality of the fusion of the single views decision.

In single camera system, many algorithms about object detection exist with different purposes. These algorithms are subdivide in three categories : without background modeling, with background modeling and combined approach. Algorithms based on background modeling are recommended in case of dynamic background observed by a static camera. Many research works used this approach.One of these algorithms is the codebook based algorithm proposed in (Kim et al., 2005).

1.1 Object Detection using Codebook

The method proposed by Kim et al. detects in real-time object in dynamic background. In this method, each pixel p_t is represented by a codebook $\mathcal{C} = \{c_1, c_2, ..., c_L\}$ and each codeword c_i, $i = 1, ..., L$ by a RGB vector v_i and a 6-tuples $aux_i = \{\check{I}_i, \hat{I}_i, f_i, p_i, \lambda_i, q_i\}$ where $\check{I}$ and $\hat{I}$ are the minimum and maximum brightness of all pixels assigned to this codeword c_i, f_i is the frequency at which the codeword has occurred, λ_i is the maximum negative run length defined as the longest interval during the training period that the codeword has not recurred, p_i and q_i are the first and last access times, respectively, that the codeword has occurred. The codebook model is created or updated using two criteria. The first criterion is based on color distortion (1) whereas the second is based on brightness distortion (2).

$$\sqrt{||p_t||^2 - C_p^2} <= \varepsilon_1 \tag{1}$$

$$I_{low} \leq I \leq I_{hi} \tag{2}$$

In (1), the autocorrelation value C_p^2 is given by equation (3) and $||p_t||^2$ is given by equation (4).

$$C_p^2 = \frac{(R_i R + G_i G + B_i B)^2}{R_i^2 + G_i^2 + B_i^2} \tag{3}$$

Mousse M., Motamed C. and Ezin E..
Fast Moving Object Detection from Overlapping Cameras.
DOI: 10.5220/0005541402960303
In *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2015), pages 296-303
ISBN: 978-989-758-123-6

$$||p_t||^2 = R^2 + G^2 + B^2 \quad (4)$$

In relation (2), $I_{low} = \alpha \hat{I}_i$, $I_{hi} = min\{\beta \hat{I}, \frac{\check{I}}{\alpha}\}$ and $I = \sqrt{R^2 + G^2 + B^2}$.

After the training period, if an incoming pixel matches with a codeword in the codebook, then this codeword will be updated and this pixel will be treated as a background pixel. If the pixel doesn't match, its information will be put in cache word and this pixel will be treated as a foreground pixel. Due to the performance of the codebook based method, several researchers continue by exploring further (Li et al., 2006; Cheng et al., 2010; Fang et al., 2013; Mousse et al., 2014).

In this work, instead using HSV color space (Doshi and Trivedi, 2006), HSL color space (Fang et al., 2013) or YUV color space (Cheng et al., 2010), we investigate a new color space. Then, we propose a new approach by converting pixels to CIE L*a*b* color space. We also use the Improved Simple Linear Iterative Clustering (Schick et al., 2012) to cluster pixels. Finally, we build the codebook model on every cluster. With each single view foreground information, multi-camera object detection can be performed.

1.2 Multi-camera Object Detection

According to Xu et al., existing multi-camera surveillance systems may be classified into three categories (Xu et al., 2011).

- The system in the first category fuses low-level information. In this category, multi camera surveillance systems detect and/or track in a single camera view. They switch to another camera when the systems predict that the current camera will not have a good view of the scene (Cai and Aggarwal, 1998; Khan and Shah, 2003).
- In the second one, system extracts features and/or even tracks targets in each individual camera. After this, we integrate all features and tracks in order to obtain a global estimate.These systems are of intermediate-level information fusion (Kang et al., 2003; Xu et al., 2005; Hu et al., 2006).
- The system in the third category fuses high-level information. In these systems, individual cameras don't extract features but provide foreground bitmap information to the fusion center. Detection and/or tracking are performed by a fusion center (Yang et al., 2003; Khan and Shah, 2006; Eshel and Moses, 2008; Khan and Shah, 2009; Xu et al., 2011).

This paper points out on the approaches in the third category. In this category some algorithms have been proposed. (Khan and Shah, 2006) proposed to use a planar homographic occupancy constraint to combine foreground likelihood images from different views. It resolves occlusions and determines regions on the ground plane that are occupied by people. (Khan and Shah, 2009) extended the ground plane to a set of planes parallel to it, but at some heights off the ground plane to reduce false positives and missing detections. The foreground intensity bitmaps from each individual camera are warped to the reference image by (Eshel and Moses, 2008). The set of scene planes are at the height of people heads. The head tops are detected by applying intensity correlation to align frames from different cameras. This work is able to handle highly crowded scenes. (Yang et al., 2003) detect objects by finding visual hulls of the binary foreground images from multiple cameras. These methods use the visual cues from multiple cameras and are robust in coping with occlusion. However the pixel-wise homographic transformation at image level slows down the processing speed. To overcome this drawback, (Xu et al., 2011) proposed an object detection approach via homography mapping of foreground polygons from multiple camera. They approximate the contour of each foreground region with a polygon and only transmit and project the vertices of the polygons. The foreground regions are detected by using Gaussian mixture model. These polygons are then rebuilt and fused in the reference image. This method provides good results, but fails if a moving object is occluded by a static object.

In this work, we propose a new strategy based on polygons projection. For each object, our strategy is to detect the camera which has the best view of the object and we only project the corresponding polygon in the reference plane. A foreground polygon is obtained by finding the convex hull of each single view foreground region. This selection incorporates geometric properties of the scene and the quality of each single view detection in order to make better decisions. This paper is subdivided in four sections. The second section presents our approach. The experimental results and performance evaluation are presented in the third section. In the fourth section, we conclude this work.

2 OVERVIEW OF OUR METHOD

In this section we present our multi-view foreground regions detection algorithm. Firstly we detect in each single view the foreground maps and fuse them to obtain a multi-view foreground object.

This section is subdivided in four parts : the first subsection presents the foreground pixels identification algorithm, the second shows how foreground pixels are grouped, the planar homography mapping is presented in the third subsection and the fourth subsection presents our multi-view fusion approach.

2.1 Foreground Pixels Segmentation

In this part, we present our single view foreground pixel extraction based on codebook. We convert all pixel from RGB to CIE Lab color space and integrate superpixel segmentation algorithm in background modeling step. The use of superpixels becomes increasingly popular for computer vision applications. In this work, we adopt the algorithm proposed by (Schick et al., 2012) because they proved the efficacy of this superpixels segmentation algorithm in image segmentation. After the extraction of superpixels, we build a codebook background model. Let $P = \{s_1, s_2, ..., s_k\}$ represents the K superpixels obtained after superpixels segmentation. Each superpixel $s_j, j \in \{1,2,...,k\}$ is composed by m pixels. With each superpixel, we buid a codebook $C = \{c_1, c_2, ..., c_L\}$ which contains L codewords $c_i, i \in \{1,2,...,L\}$. Each codewords c_i consists on an vector $v_i = (\bar{a}_i, \bar{b}_i)$ and 6-tuples $aux_i = \{\check{L}_i, \hat{L}_i, f_i, p_i, \lambda_i, q_i\}$ in which $\check{L}_i, \hat{L}_i$ are the minimum and maximum of luminance value, f_i is the frequency at which the codeword has occurred, λ_i is the maximum negative run length defined as the longest interval during the training period that the codeword has not recurred, p_i and q_i are the first and last access times, respectively, that the codeword has occurred. $\bar{L}$, $\bar{a}$, $\bar{b}$ are respectively the average value of component L*, a* and b* in a superpixel. We compute the color distortion by replacing (5) and (6) into (1). For the brightness distortion degree we use $\bar{L}$ value as the intensity of the superpixel. Therefore in expression (2) the value of I is given by $\bar{L}$.

$$p_t = \bar{a}^2 + \bar{b}^2 \tag{5}$$

$$C_p^2 = \frac{(\bar{a}_i\bar{a} + \bar{b}_i\bar{b})^2}{\bar{a}_i^2 + \bar{b}_i^2} \tag{6}$$

After the learning phase, we detect foreground pixel by subtracting the current image from the background model. The substraction method is based on superpixels. The proposed algorithm is detailed in Algorithm 2.

[1] $BGS(p_k)$ is a procedure which subtracts an incoming pixel p_k from the background. It's defined as Algorithm 3.

Algorithm 1: Foreground objects segmentation.

$l \leftarrow 0, t \leftarrow 1$
for *each frame* F_t *of input sequence* S **do**
  Segment frame F_t into k-superpixels
  for *each superpixels* Su_k *of frame* F_t **do**
    $p_t(\bar{L}, \bar{a}, \bar{b})$
    for $i = 1$ *to* l **do**
      if *(colordist* (p_t, v_i)*) and (brightness (*$\bar{L}$, $\check{L}_i$, $\hat{L}_i$*))* **then**
        Select a matched codeword c_i
        Break
    if *there is no match* **then**
      $l \leftarrow l + 1$
      create codeword c_L by setting parameter $v_L \leftarrow (\bar{a}, \bar{b})$ and $aux_L \leftarrow \{\bar{L}, \bar{L}, 1, t-1, t, t\}$
    else
      update codeword c_i by setting $v_i \leftarrow (\frac{f_i\bar{a}_i + \bar{a}}{f_i+1}, \frac{f_i\bar{b}_i + \bar{b}}{f_i+1})$ and $aux_i \leftarrow \{min(\bar{L}, \check{L}_i), max(\bar{L}, \hat{L}_i), f_i + 1, max(\lambda_i, t - q_i), p_i, t\}$
  for *each codeword* c_i **do**
    $\lambda_i \leftarrow max\{\lambda_i, ((m \times n \times t) - q_i + p_i - 1)\}$
  if $t > N$ **then**
    for *each pixels* p_k *of frame* F_t **do**
      $BGS(p_k)$ [1]
  $t \leftarrow t + 1$

Algorithm 2: procedure $BGS(p_k)$

for *all codewords* **do**
  find the codeword c_m matching to p_k based on color and brightness distortions.

$$BGS(p_k) = \begin{cases} \textbf{foreground} & \text{if there is no match} \\ \textbf{background} & \text{otherwise} \end{cases}$$

2.2 Foreground Pixels Grouping

After the foreground pixels in each view are detected, these pixels need to be grouped into foreground region. The region is obtained by finding the convex hull of all contours detected in threshold image. Then, all region can be approximated by a polygon and each polygon is convex. To use the polygon vertices in the information fusion module, we have decided to assign an unique identifier *id* to each polygon

and each polygon is characterized by a vector $\mathcal{V}_{\text{id}} = (v_{1\_id}, v_{2\_id}, ..., v_{k\_id})$ in which each $v_{j\_id}, j = 1, ..., k$ represents a vertex of the polygon.

2.3 Planar Homography Mapping

Homographies are usually estimated between a pair of images by finding feature correspondence in these images. The most commonly used feature is corresponded points in different images, though other features such as lines or conics in the individual images may be used. These features are selected and matched manually or automatically from 2D images to compute the homography between two camera views or the homography between one camera view and the top view. The homography transformation is a special variation of the projective transformation. Let us consider the point $x = (x_s, y_s, 1)$ in the image without distortion and the point $X = (X_w, Y_w, Z_w, 1)$ in the 3D world. The projection transformed from X to x is given by equation (7).

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} f/s_x & s & C_x & 0 \\ 0 & f/s_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix} \quad (7)$$

If X is limited on the ground plane, therefore Z_w will be 0 and the projection transformed from X to x becomes:

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} f/s_x & s & C_x \\ 0 & f/s_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \quad (9)$$

Planar homography mapping consists in finding 3×3 matrix which correspond a point $pt(x, y)$ to an other point $pt'(x', y')$ on the ground plane in two different views.

2.4 Fusion Approach

After the detection of each foreground map, we need to fuse these regions to get a multi-view information. The aim of our approach is to detect the camera which provides the best view of each object in the scene. The union of the selected objects views gives an overview of the scene. This selection is based on geometric properties of the scene and on the quality of each camera detection. Let us consider a scene being observed by cameras with overlapping views as shown in Figure 1. In Figure 1, the scene is observed by two cameras.

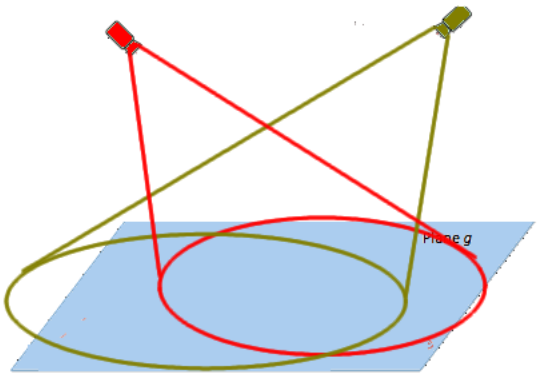

Figure 1: Illustration of scene observed by two cameras.

Each camera observes the scene differently and it is necessary to choose the camera which provides the best view of an object.

Based on each single foreground polygons information about the coverage area, we identify three cases :

1. a polygon from the view of a camera is not associated with any polygon from the view of another camera;
2. a polygon from the view of a camera is associated with only one polygon from the view of another camera;
3. a polygon from the view of a camera is associated with more than one polygon from the view of another camera;

A polygon $P1$ will be considered associated with a polygon $P2$ from another camera if the projection of one of the $P1$ vertex into the plane of the second camera view belongs to $P2$. The projection of the vertex is performed by using the principles of planar homography mapping. Thus, a calibration of the stage must be carried out for obtaining the projection matrix. The ray casting algorithm proposed in (Sutherland et al., 1974) has been used in order to resolve point-in-polygon problem.

In the first case, we remark that only one camera detect the object and we assume that this camera has the best possible view of the object. Then the vertices of the corresponding polygon are projected in the ground plane (or the reference plane). The corresponding multi-view foreground map is then obtained by filling in the projected polygon.

In the second case, the object is detected by more than one camera. For the selection of the best camera, firstly we prioritize the cameras that detect the largest

number of the lowest points of each polygon associated to the object as foreground pixels. For each polygon the lowest point is the point nearest of the ground. It is the vertex which has the largest value on the ordinate component in the camera coordinate system. If at least two associated polygons satisfy this criterion, then the selection is made with respect to the position of the object relative to the camera. Indeed, this position has an influence on the rendering in the homographic plan. For performing this selection, for each camera we calculate the distance between the projection of the highest vertex of the polygon in the ground plane (or the reference plane) and the projection of the lowest vertex of the polygon in the ground plane (or the reference plane). The best camera is the camera which has the smallest distance. Then the vertices of the corresponding polygon are projected in the ground plane (or the reference plane). The corresponding multi-view foreground map is then obtained by filling in the projected polygon.

In the third case,we conclude that this is dynamic n occlusion between objects. According to this, the best camera is the camera in which more than one polygon is associated. Then the vertices of the corresponding polygons are projected in the ground plane (or the reference plane). The corresponding multi-view foreground map is then obtained by filling in the projected polygons.

3 EXPERIMENTAL RESULTS AND PERFORMANCE

In this section, we present the performance of the proposed approach by comparing it with other state of the art method. Firstly we present the experimental environment and results. After that we present and analyze the performance of our system.

3.1 Experimental Results

We present experimental results at two levels. For the validation of our single view algorithm, We have selected two public benchmarking datasets (available from http://www.changedetection.net) which are covered under the work done in (Goyette et al., 2012). They are “fall” and “boats” datasets. In order to test our multi-view object localization algorithm, we used video sequence which contains significant lighting variation, dynamic occlusion and scene activity. We selected “ Dataset 1” (available from http://ftp.pets.rdg.ac.uk/pub/PETS2001/DATASET1/ sequence which has been used for testing several multi view detection/tracking algorithms. These datasets are made available to researchers for the evaluation of moving object detection algorithms.

The experiment environment is Intel Core i7 CPU L 640 @ $2.13GHz \times 4$ processor with 4GB memory and the programming language is C++. The parameters of superpixels segmentation algorithm is given by (Schick et al., 2012). In our implementation if the size of input frame is $(M \times N)$ then we construct $\frac{M \times N}{50}$ superpixels. The single view segmentation results are presented in Figure 2 and Figure 3 whereas the multi-view segmentation results are presented in Figure 4.

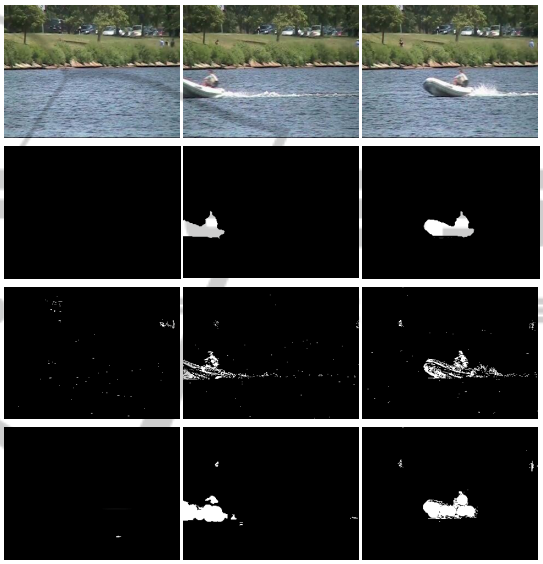

Figure 2: “boats” dataset segmentation results. The first row shows the original images. The second row shows the ground truth. The third row shows the detected results by (Kim et al., 2005), and the last row shows the detected results by our proposed algorithm.

3.2 Performance Evaluation and Discussion

Firstly, we evaluate the performance of our proposed single view object detection and compare it to other research works which extend the original codebook by exploiting other color space (RGB, HSV, HSL, YUV) information. In order to perform this comparison, we use an evaluation based on ground truth. The ground truth has been obtained by manually labeling foreground objects in the original frame. The ground truth based metrics are : true negative (TN), true positive (TP), false negative (FN) and false positive (FP). We use these metrics to compute other parameters for the evaluation of the algorithm. These parameters are false positive rate (FPR), true positive rate (TPR), precision (PR) and F-measure (FM).These parameters are respectively computed using expressions

(10), (11), (12) and (13).

$$FPR = 1 - \frac{TN}{TN+FP} \quad (10)$$

$$PR = \frac{TP}{TP+FP} \quad (11)$$

$$TPR = \frac{TP}{TP+FN} \quad (12)$$

$$FM = \frac{2 \times PR \times TPR}{PR+TPR} \quad (13)$$

Results are presented in Table 1 and Table 2. In these tables, CB refers to the method proposed by (Kim et al., 2005), CB_HSV refers to the method suggested by (Doshi and Trivedi, 2006), CB_HSL refers to the algorithm proposed by (Fang et al., 2013), CB_YUV refers to the algorithm suggested by (Cheng et al., 2010) and CB_LAB refers to our proposed algorithm.

Table 1: Different metrics according to experiments with "boats" dataset.

Metrics	CB	CB_HSV	CB_HSL	CB_YUV	CB_LAB
FPR	0.23	0.25	0.21	0.27	**0.21**
PR	0.87	0.86	0.89	0.80	**0.91**
TPR	0.46	0.48	0.50	**0.55**	0.53
FM	0.60	0.62	0.64	0.65	**0.66**

According to results presented in Table 1 and Table 2, we can conclude that our modified codebook works better than standard codebook. The proposed method reduces the percentage of false alarms and increases the precision of object detection. According

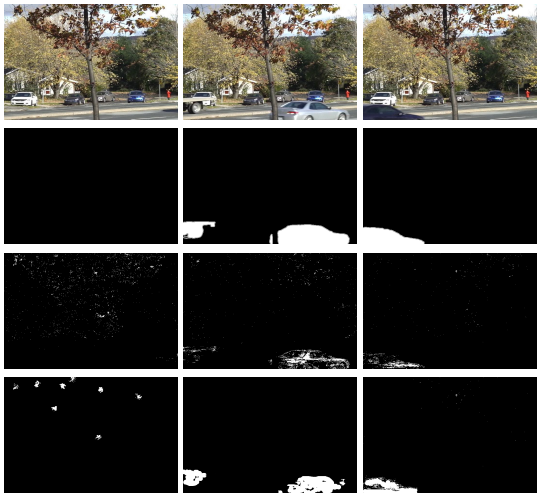

Figure 3: "fall" dataset segmentation results. The first row shows the original images. The second row shows the ground truth. The third row shows the detected results by (Kim et al., 2005), and the last row shows the detected results by our proposed algorithm.

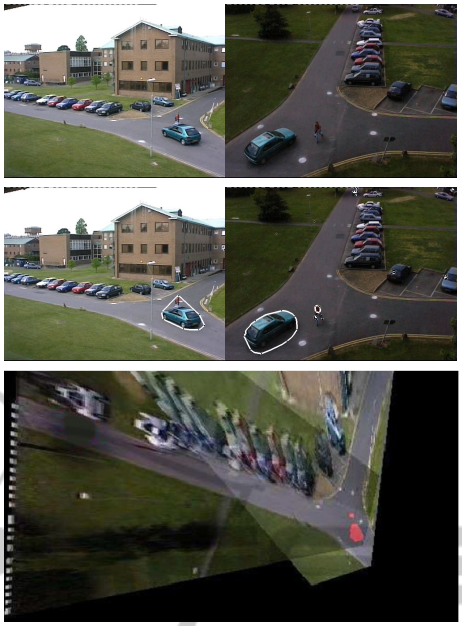

Figure 4: The first row shows the original 2 single views. The second rows shows the foreground region detected in single view. The third row shows the segmentation result using a multi view informations.

Table 2: Different metrics according to experiments with "fall" dataset.

Metrics	CB	CB_HSV	CB_HSL	CB_YUV	CB_LAB
FPR	0.31	0.33	0.25	0.38	**0.23**
PR	0.56	0.60	0.63	0.41	**0.67**
TPR	0.32	0.39	0.43	0.45	**0.45**
FM	0.41	0.47	0.51	0.43	**0.54**

to F-measure values, we can claimed that the accuracy of object detection by using static camera is also improved by using our proposed model. When we compare our approach to other enhancement of codebook based algorithm, we show that our method has the best accuracy of object detection. With "boats" dataset, the method proposed in (Cheng et al., 2010) detects more true positive pixels than our proposed method. But it also emits more false positive alarm than our approach. The use of superpixel segmentation reduces the complexity of the modeling process using codebook model. Secondly, we evaluate the performance of our multi view detection approach. The results high lightly that our approach provides a good moving object location and provide good accuracy in a dynamic occlusion case. It has similar performance in moving object detection to other approaches of the state of the art such as (Xu et al., 2011). However our approach detects less false pos-

itive than the approach proposed by Xu et al. mainly due to the fact that our single view detection algorithm emits less false positive alarm than MoG which is used in (Xu et al., 2011). Our proposed algorithm is also faster than state of the art multi-view approach. Indeed, comparing our method to Xu et al. algorithm which is known as been 40 times faster than the existing algorithms, we note that our method improves the execution time by 22%.

4 CONCLUSIONS

In this paper, we have proposed a fast algorithm for object detection by using overlapping cameras based on homography. In each camera, we propose an improvement of codebook based algorithm to get foreground pixels. The multi-view object detection that we proposed in this work algorithm is based on the fusion of multi camera foreground informations. We approximate the contour of each foreground region with a polygon and only project the vertices of the relevant polygons. The experiments have shown that this method can run in real time and generate results similar to those warping foreground images.

ACKNOWLEDGEMENTS

This work is partially funded by the Association AS2V and Fondation Jacques De Rette, France. We also appreciate the help of Mr Léonide Sinsin and Mr Patrick Aïnamon for proof-reading this article. Mikaël A. Mousse is grateful to Service de Coopération et d'Action Culturelle de Ambassade de France au Bénin.

REFERENCES

Cai, Q. and Aggarwal, J. (1998). Automatic tracking of human motion in indoor scenes across multiple synchronized video streams. In *Proc. of IEEE International Conference on Computer Vision*.

Cheng, X., Zheng, T., and Renfa, L. (2010). A fast motion detection method based on improved codebook model. *Journal of Computer and Development*, 47:2149–2156.

Doshi, A. and Trivedi, M. M. (2006). Hybrid cone-cylinder codebook model for foreground detection with shadow and highlight suppression. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 121–133.

Eshel, R. and Moses, Y. (2008). Homography based multiple camera detection and tracking of people in a dense crowd. In *Proc. of 18th IEEE International Conference on Computer Vision and Pattern Recognition*.

Fang, X., Liu, C., Gong, S., and Ji, Y. (2013). Object detection in dynamic scenes based on codebook with superpixels. In *Proceedings of the Asian Conference on Pattern Recognition*, pages 430–434.

Goyette, N., Jodoin, P. M., Porikli, F., Konrad, J., and Ishwar, P. (2012). Changedetection.net: A new change detection benchmark dataset. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*.

Hu, W., Hu, M., Zhou, X., Tan, T., Lou, J., and Maybank, S. (2006). Principal axis-based correspondence between multiple cameras for people tracking. In *IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 29*.

Kang, J., Cohen, I., and Medioni, G. (2003). Continuous tracking within and across camera streams. In *Proc. of International Conference on Pattern Recognition*.

Khan, S. and Shah, M. (2003). Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. In *IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23*.

Khan, S. M. and Shah, M. (2006). A multi-view approach to tracking people in crowded scenes using a planar homography constraint. In *Proc. of 9th European Conference on Computer Vision*.

Khan, S. M. and Shah, M. (2009). Tracking multiple occluding people by localizing on multiple scene planes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31*.

Kim, K., Chalidabhonse, T. H., Harwood, D., and Davis, L. (2005). Real-time foreground-background segmentation using codebook model. In *Elsevier Real-Time Imaging, 11(3) : 167-256*.

Li, Y., Chen, F., Xu, W., and Du, Y. (2006). Gaussian-based codebook model for video background subtraction. In *Lecture Notes in Computer Science*.

Mousse, M. A., Ezin, E. C., and Motamed, C. (2014). Foreground-background segmentation based on codebook and edge detector. In *Proc. of International Conference on Signal Image Technology & Internet Based Systems*.

Schick, A., Fischer, M., and Stiefelhagen, R. (2012). Measuring and evaluating the compactness of superpixels. In *Proc. of International Conference on Pattern Recognition*, pages 930–934.

Sutherland, I. E., Sproull, R. F., and Schumacker, R. A. (1974). A characterization of ten hidden surface algorithms. In *ACM Computing Surveys (CSUR)*.

Xu, M., Orwell, J., Lowey, L., and Thirde, D. (2005). Architecture and algorithms for tracking football players with multiple cameras. In *IEE Proc. of Vision, Image and Signal Processing*.

Xu, M., Ren, J., Chen, D., Smith, J., and Wang, G. (2011). Real-time detection via homography mapping of foreground polygons from multiple. In *Proc. of 18th IEEE International Conference on Image Processing*.

Yang, D. B., Gonzalez-Banos, H. H., and Guibas, L. J. (2003). Counting people in crowds with a real-time network of simple image sensors. In *Proc. 9th IEEE International Conference on Computer Vision*.