

# ROBUST OBJECT TRACKING USING CORRESPONDENCE VOTING FOR SMART SURVEILLANCE VISUAL SENSING NODES

*Mayssaa Al Najjar, Soumik Ghosh, Magdy Bayoumi*

The Center for Advanced Computer Studies  
University of Louisiana at Lafayette, Lafayette, USA  
{mea5645, sxx5317, mab}@cacs.louisiana.edu

## ABSTRACT

This paper presents a bottom-up tracking algorithm for surveillance applications where speed and reliability in the case of multiple matches and occlusions are major concerns. The algorithm is divided into four steps. First, moving objects are detected using an accurate hybrid scheme with selective Gaussian modeling. Simple object features balancing speed, reliability, and complexity are then extracted. Objects are matched based on their spatial proximity and feature similarity. Finally, correspondence voting solves multiple match conflicts, segmentation errors, and occlusion cases. This approach is very simple, which makes it suitable for implementation at smart surveillance visual sensing nodes. Moreover, the simulation results demonstrate its robustness in detecting occlusions and correcting segmentation errors without any prior knowledge about the objects models or constraints on the direction of their motion.

*Index Terms*— Object tracking, similarity matching

## 1. INTRODUCTION

Over the last decade, surveillance systems have gained a lot of attention for security and safety concerns. Surveillance systems started as monitoring systems for military bases, but soon found their way to every aspect of our daily life whether in production monitoring, traffic reporting, or even ensuring personal safety at home. The purpose is to monitor a scene, detect suspicious object motion, and signal an alarm. Object detection and tracking are the first steps in such systems and are particularly important as their accuracy affects later analysis. They are usually studied together, where segmentation is required at least for initializing the tracking and tracking may correct segmentation errors over time [1].

One of the major challenges in current surveillance systems is to implement these algorithms on smart video sensor cameras. Instead of transmitting the entire image and exhausting the communication bandwidth, only the information about detected objects, if any, is sent. However, these camera nodes have limited resources, which impose constraints on the complexity and memory requirements of the running algorithms.

This paper presents a tracking technique that is suitable for implementation at the sensor camera node. Simple adaptive background subtraction (BS) and 3 Frame Difference (FD) are used for detecting objects in indoor scenes. A hybrid BS technique based on selective Gaussian modeling is used for monitoring outdoor scenes with gradual illumination changes and clutter motion in the background. This technique reduces the computations required in traditional Mixture of Gaussian (MoG) and improves its accuracy by focusing the attention on the most probable foreground pixels. Tracking is done using a bottom-up similarity matching scheme and non linear feature voting based on shape, color, and texture. This technique is very fast, simple, and

yet able to track multiple objects, handle object merges/splits, and correct segmentation errors without any prior knowledge about the object models or constraints on their direction of motion.

The rest of the paper is organized as follows. Section 2 reviews the main detection and tracking schemes as well as those used in real surveillance systems. Section 3 presents the proposed detection and tracking technique. Section 4 discusses the experimental results, demonstrating the efficiency of the proposed scheme. Section 5 contains the conclusion.

## 2. RELATED WORK

There are two major approaches for object tracking: bottom-up also known as target representation and localization, and top-down also known as filtering and data association [2].

Top-down approaches rely on prior information about the scene or object, dealing with object dynamics, and evaluation of different hypotheses [2]. The idea is to predict the position of the objects in the current frame, given their positions in the previous frame, and then update these estimates based on the current observations. Common filters include the Kalman Filter (KF) and the Particle Filter (PF). Additional data validation and association is required for multiple tracking. Even though these techniques are more reliable than bottom-up schemes, they are by far more computationally expensive and fail if there is not enough information about the scene/target or in case of model drifts [3].

Bottom-up approaches consider the changes in the targets' appearances. The idea is to detect the moving objects first, represent each object according to a certain model, and associate it with the matching object in the next frame [2]. Background Subtraction techniques are the most common detection schemes used in surveillance systems with stationary cameras [4]. There are several approaches for background modeling ranging from single background estimates like Running average Filter (RAF) and providing acceptable accuracy for simple indoor applications, all the way to complicated techniques with full density estimates like MoG [5] for outdoor scenes with gradual illumination changes and swinging tree branches. Once objects are detected, features are extracted. Nearest Neighbors matching techniques and Data Association Filters based on spatial proximity or appearance similarity are then used [1]. In general, bottom-up methods have low computational complexity, but are more sensitive to errors due to inaccurate foreground detection. This may be solved using additional processing steps. For instance, Amer's work [6] uses object voting based on distance, shape, and motion to handle multiple occlusions without prior knowledge about object models. However, it is not robust to object changes due to global illumination changes. It also assumes motion is smooth and objects do not suddenly change direction, which may not be very practical. To overcome these limitations, other descriptors, robust to illumination changes, should be considered.

Table 1 summarizes the current detection and tracking techniques used in the well established surveillance systems.

**Table 1 Detection and Tracking in current surveillance systems**

System	Detection	Tracking
VSAM [7]	3FD + RAF	Extend KF to support multiple hypothesis
W <sup>4</sup> [8]	Bimodal distribution	Combine motion estimation and correspondence matching
Vigilant [9]	MoG	Combine bayesian classification based on velocity, box ratio and Hidden Markov Models
Knight [10]	Pixel/region level subtraction	Voting based on spatial, temporal, appearance (shape, color, motion)
IBM S3 [11]	MoG	Use appearance models, resolve depth ordering of occluded objects

### 3. PROPOSED TECHNIQUE

The proposed scheme is a robust bottom-up correspondence-based tracking technique for smart camera nodes. The algorithm is divided into four steps as shown in Fig. 1: detect moving objects, extract their representing features that will be used next for matching these objects from frame to frame, and finally handle occlusion/segmentation errors. Thus, the major surveillance challenges, related to the algorithm speed, simplicity, and ability to handle occlusions are addressed at the different stages:

- Faster, better detection accuracy than MoG by using our hybrid selective detection method. Since motion areas are much smaller than the image, pixel matching, parameter updating, and sorting are significantly reduced. By focusing the attention on most probable foreground pixels, selective MoG decreases the probability of misclassifying a background pixel, while hysteresis thresholding improves the recall for grayscale images
- Simple shape, color, and texture feature extraction as a compromise between descriptor accuracy and heavy computations. These features provide robust and sufficient description for surveillance purposes where the speed of execution is a higher concern than accurate object features
- Real time tracking by narrowing the search area, matching based on spatial proximity and feature similarity, and using non linear voting to resolve multiple matching conflicts
- Reliability in detecting occlusions, correcting segmentation errors, and updating trajectory based on feedback from the last stage. This is done without any prior knowledge, assumption about object models, or constraints on the direction of motion.

#### 3.1. Object Detection

The first step involves detecting moving objects (foreground masks). For indoor monitoring, 3FD and adaptive BS can be used. For outdoor monitoring, statistical background models are needed. The hybrid technique proposed in our previous work [12] combines simple FD, adaptive BS, and accurate Gaussian modeling to benefit from MoG's high accuracy in such outdoor scenes, while reducing its computations. This is done in two steps. First, we divide the frame into static and non static region also known as region of motion. A pixel  $I_t(x,y)$  at frame  $t$  and location  $(x,y)$  is classified as part of the motion region if:

$$|I_t(x,y) - I_{t-1}(x,y)| > I_{th} \text{ or } |I_t(x,y) - B_t(x,y)| > I_{th} \quad (1)$$

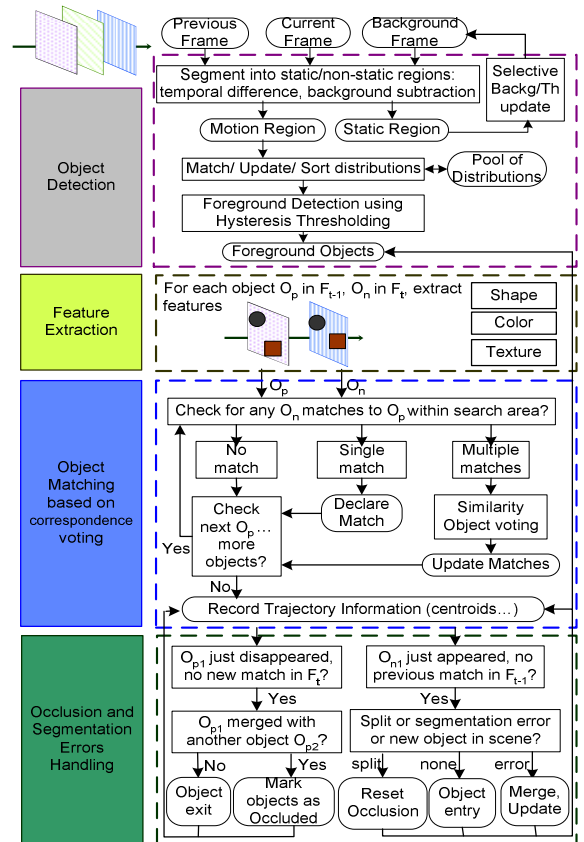
The background  $B_t(x,y)$ , and detection thresholds  $I_{th}(x,y)$  for static pixels are updated at each frame using some kind of RAF to adapt to scene changes. Now, the region of motion contains interesting

objects as well as swinging tree branches. So, the second step is to keep only interesting moving objects using selective MoG. Given a pixel  $I_t$  in the motion area and its  $K$  distributions, we look for  $I_t$ 's best match. If there is a match, the mean  $\mu_i$  and deviation  $\sigma_i^2$  of this match distribution, and the weights  $w$  for all distributions are updated as in (2), where  $\alpha$  is the adaptation rate that defines the speed at which the parameters change, and thus affects how long it will take for a static foreground to be integrated in the background.

$$\begin{aligned} \mu_{i,t} &= (1 - \rho)\mu_{i,t-1} + \rho I_t \\ \sigma_{i,t}^2 &= (1 - \rho)\sigma_{i,t-1}^2 + \rho(I_t - \mu_{i,t})^2 \\ w_{i,t} &= (1 - \alpha)w_{i,t-1} + \alpha M_{i,t} \end{aligned} \quad (2)$$

$$\rho = \alpha / w_{i,t} \text{ and } M_{i,t} = 1 \text{ if matched, } 0 \text{ elsewhere}$$

Otherwise, the distribution with the least  $w$  is replaced by a new one with mean  $I_t$ , large  $\sigma_i^2$ , and small  $w_i$ . All other weights are lowered. The distributions are then sorted by their  $w/\sigma_i$  and higher rank distributions are chosen as background.  $I_t$  is compared to these background distributions using hysteresis thresholding and is classified as background, foreground, or candidate pixel [12]. For a candidate pixel, additional connected component check is required to classify it as foreground/background. Morphological operators enhance the quality of the final object, which is now ready for feature extraction.



**Figure 1 Proposed technique block diagram**

#### 3.2. Feature Extraction

After detecting moving objects, their corresponding features are extracted. These features will be used to match objects from frame to frame. Simple shape, color, and texture features are extracted as a tradeoff between matching quality and computational complexity [13]. Shape features include information about the object and its

bounding box (BB), width (W), height (H), object size (S), centroid ( $C_m = \{x_{C_m}, y_{C_m}\}$ ), compactness (CO), and extent measure (E) as shown in (3), which is rotation and scale invariant [6].

$$x_{c_m} = \frac{\sum_{i=1}^{i=S} x_i}{S}, y_{c_m} = \frac{\sum_{i=1}^{i=S} y_i}{S}, CO = \frac{S}{H \times W}, E = \begin{cases} H/W & \text{if } H < W \\ W/H & \text{otherwise} \end{cases} \quad (3)$$

Since shape features alone are not reliable enough especially for deformable objects and during occlusions, the normalized quantized color histogram ( $H_c$ ) is computed with  $N=21$  bins. Statistical moments of the histogram, such as relative smoothness in (4), give a better description of the object texture that is robust to illumination changes.

$$smooth = \sum_{i=1}^N (i - \text{mean}(H_c))^2 H_c(i) = \sum_{i=1}^N i^2 H_c(i) - (\text{mean}(H_c))^2 \quad (4)$$

### 3.3. Object Matching based on Correspondence Voting

The purpose here is to establish a correspondence between objects in consecutive frames based on spatial proximity and feature similarity. This is done in two steps: matching objects  $O_p$  in the previous frame  $F_{t-1}$  to objects  $O_n$  in the new frame  $F_t$ , and solving conflicts in case of multiple correspondences to the same object.

Ideally,  $O_p$  is matched to the nearest object  $O_n$  with similar features. Thus, for each object  $O_p$ , we consider objects  $O_n$ s lying within  $O_p$ 's search area. If the distance between the two centroids  $dis(C_p, C_n)$  is relatively small, their size ratio  $RS = S_p/S_n$ , extent ratio  $RE = E_p/E_n$ , and compactness ratio  $RC = CO_p/CO_n$  have not changed much between consecutive frames as shown in (5), then most probably these objects correspond to the same one and should be declared as a match:

$$dis(C_p, C_n) < Th \text{ and } RS < Th_1 \text{ and } RE < Th_2 \text{ and } RC < Th_3 \quad (5)$$

Where  $Th$ ,  $Th_1$ ,  $Th_2$ ,  $Th_3$  are predefined thresholds related to the frame rate, frame size [6], object size, and camera location. For instance, if the camera is placed on top of the building, objects are not going to shrink/grow a lot between consecutive frames.

A problem arises when multiple objects in the previous frame are matched to the same object in the new frame or vice-versa. For instance, if  $O_p$  has two possible matches  $O_{n1}$  and  $O_{n2}$  in  $F_t$ , the conflict must be resolved using similarity voting as in [6] but based on distance, shape, color, and texture, which is more robust to object deformations and illumination changes. Two voting variables  $v_1$  and  $v_2$  are initialized to zero. Assume  $v_1$  represents the case where  $\{O_p \rightarrow O_{n1}\}$  and  $\{O_p \rightarrow O_{n2}\}$  and  $v_2$  the opposite case. Each voting variable is incremented every time the match it represents has a higher rank. Let  $d_1(a,b)$  be the distance between the color histograms of objects  $a$  and  $b$ ,

$$d_1(a,b) = \sum_{i=1}^N |H_{c_a}(i) - H_{c_b}(i)| \quad (6)$$

The minimum distance leads to a higher rank. This is done for all features. The variable with the highest vote dictates how the match is done. By the end of this stage, each object in  $F_{t-1}$  is matched at most to one object in  $F_t$  and vice-versa.

### 3.4. Occlusion and Segmentation Errors Handling

Two cases still need to be considered: objects merge and objects split. If an old object  $O_{p1}$  is not matched to any new object, it may have disappeared (object exit) or it may be occluded by another object. Thus, if  $O_{p1}$  disappears in the new frame, we check if there is a BB for another object  $O_n$  in  $F_t$  that overlaps with  $O_{p1}$ 's BB. If yes, we check if  $O_n$  has already a match  $O_{p2}$  in  $F_{t-1}$ ; this means that  $O_{p1}$  was occluded by  $O_{p2}$ . So we do not delete the occluded object

$O_{p1}$ , we mark  $O_{p1}$ ,  $O_{p2}$  and  $O_n$  as occluded and save them in an occlusion group, along with their features.

Another case occurs when a new object  $O_{n1}$  is not matched to any old object: it may be a new object entering the scene, the result of improper segmentation, or an object that was previously occluded and is no longer occluded. If a new object  $O_{n1}$  is detected, check whether there is a previous object  $O_p$ , whose BB overlaps with  $O_{n1}$ 's BB, and who has a new match  $O_{n2}$ . If its occlusion bit is set with valid occlusion ID, then split  $O_p$ . Now, there are two objects  $O_{p1}$  and  $O_{p2}$  in a previous frame, that were occluded in  $O_p$ . So  $\{O_{p1}$  and  $O_{p2}\}$  should be matched to  $\{O_{n1}$  and  $O_{n2}\}$ . A similar voting based on the features stored at the time of merging helps in recognizing the matching. The corresponding trajectories are then updated and occlusion bits reset. If  $O_p$  was not previously occluded, but  $O_{n1}$  is close enough to another object  $O_{n2}$  in  $F_t$ , and together form a better match to  $O_p$ , then  $O_{n1}$  and  $O_{n2}$  are the result of improper segmentation, and should be merged. This should be fed back to the detection stage and features for the merged object are extracted. Otherwise,  $O_{n1}$  is a new object that just entered the scene.

## 4. SIMULATION RESULTS

To verify the functionality and reliability of the proposed technique at different levels, video sequences from the wallflower paper [14], PETS 2006 [15], and other sequences generated on campus were used. Fig. 2 shows a comparison of the hybrid detection scheme with 3FD and MoG using the "walking man" sequence where ground truth objects are provided. Table 2 shows the quantitative comparison using evaluation metrics from [16]:

$$\text{Recall} = \frac{\text{Foreground Pixels correctly identified by algorithm}}{\text{Total Foreground Pixels in Ground Truth}} \quad (7)$$

$$\text{Precision} = \frac{\text{Foreground Pixels correctly identified by algorithm}}{\text{Total Foreground Pixels detected by algorithm}}$$

Even though the recall value is high in the case of 3FD, the precision value is unacceptable because both interesting and non-interesting objects are detected as foreground. Gaussian models, on the other hand, provide statistical descriptions of background models and can thus distinguish swinging trees from moving objects. The proposed technique reduces MoG's computations and provides better detection results as explained earlier. The tracking technique was then tested on several indoor and outdoor video sequences. Fig. 3 and 4 show selected frames with each object enclosed in its BB. Objects enter the scene and exit at different times, and their trajectories are plotted as a function of the frame number. Fig. 5 shows how the algorithm is able to recover from segmentation errors based on the feedback information from the last stage: A person's head and body are first detected as two different objects. But when matching them over time, the algorithm correctly identifies them as belonging to one object and merges

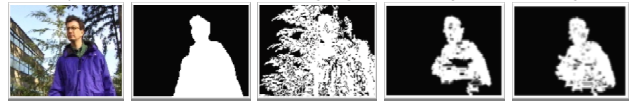


Figure 2 From left to right: Man walking in outdoor scene [14], Ground truth, FD result, MoG result, Proposed scheme result

Table 2 Quantitative Analysis

	Recall	Precision
3 FD	0.8270	0.4481
MoG	0.6101	0.9658
Proposed	0.7414	0.9561

them together. Fig. 6 illustrates how the algorithm handles occlusion: two persons walking towards each other, colliding, and then continuing in their separate ways. When occlusion occurs, the algorithm saves the information for both objects before merging them together. This helps, when the splits occur, in recognizing that these are the same objects previously occluded not some new objects entering the scene. Note that when objects occlude, they are not deleted and their trajectories are not disconnected. The trajectory of each object is updated based on similarity voting. Although object 2 was occluded by object 1 for several frames, once it reappears, the algorithm successfully recognizes this object as the same object that disappeared earlier and not as a new object.

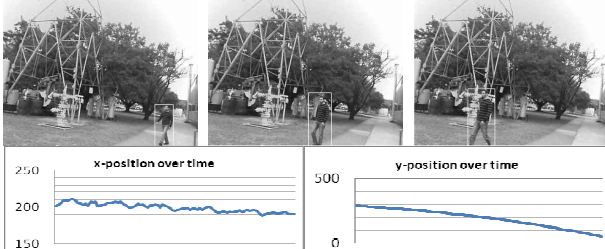


Figure 3 From left to right, top to down: Tracking outdoor at frames 20, 50, 70, x and y positions as function of frame nb

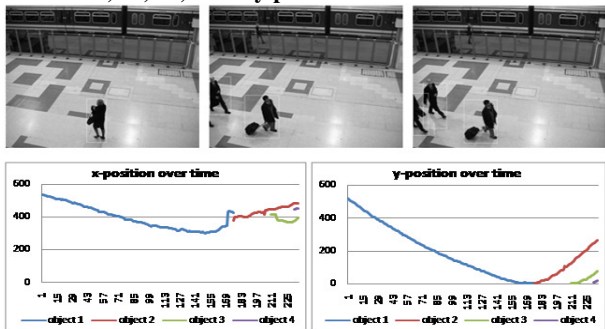


Figure 4 From left to right, top to down: Tracking indoor [15] at frames 99, 215, 230, x and y positions as function of frame nb



Figure 5 From left to right: Frame 150 [15], Segmentation before feedback, Segmentation after feedback

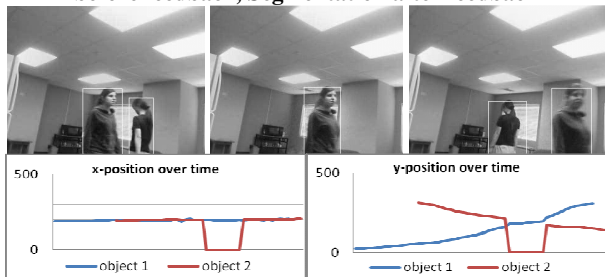


Figure 6 From left to right, top to down: frame before occlusion, objects merge, objects split, x and y positions as function of frame nb

## 5. CONCLUSION

We proposed a robust tracking technique based on similarity matching and correspondence voting for surveillance applications.

The algorithm proceeds with object detection using a hybrid selective Gaussian scheme. Simple features, comprising speed and complexity, are extracted. Then, a fast correspondence matching using non linear voting handles multiple matches and occlusions. This approach is very simple which makes it suitable for implementation at smart surveillance camera nodes. Yet, the simulation results proved its robustness in tracking multiple objects, handling occlusions, and correcting segmentation errors, without having any assumption about object models or constraint on the direction of their motion.

## 6. REFERENCES

- [1] D. Rowe, *Towards robust multiple-tracking in unconstrained human-populated environments*, Ph.D. Thesis, Universitat Autonoma de Barcelona, Spain, 2008.
- [2] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE T. Pattern Anal. and Machine Intel.*, vol. 25, no. 5, May 2003.
- [3] O. King, and D. Forsyth, "How does Condensation behave with a finite number of samples?" in *ECCV*, 2000, pp.695-709.
- [4] M. Piccardi, "Background subtraction techniques: a review," in *IEEE Conf. Systems, Man and Cybernetics*, Oct. 2004, pp. 3099-3104.
- [5] C. Stauffer and W. E. Grimson, "Adaptive background mixture models for real time tracking," in *IEEE Conf. CVPR*, Jun. 1999, pp.246-252.
- [6] A. Amer, "Voting-based simultaneous tracking of multiple video objects," *IEEE T. Circuits and Systems for Video Technology*, vol. 15, no. 11, pp. 1448-1462, 2005.
- [7] R. T. Collins, A. J. Lipton, and T. Kanade, "A system for video surveillance and monitoring," *Tech. Rep. CMU-RI-TR-00-12*, Robotics Institute, Carnegie Mellon University, May 2000.
- [8] I. Haritaoglu, D. Harwood, L. S. Davis, "W4: real-time surveillance of people and their activities," *IEEE T. Pattern Anal. and Machine Intel.*, vol. 22, no. 8, pp.809-830, Aug. 2000.
- [9] P. Remagnino and G. A. Jones, "Classifying surveillance events from attributes and behaviour," *Br. Machine Vision Conf.*, 2001.
- [10] M. Shah and O. Javed, K. Shafique, "Automated visual surveillance in realistic scenarios," *IEEE Multimedia*, vol. 14, no. 1, pp. 30-39, 2007.
- [11] Y. L. Tian, M. Lu and A. Hampapur, "Robust and efficient foreground analysis for real-time video surveillance," in *IEEE Conf. CVPR*, Jun. 2005, pp. 1182-1187.
- [12] M. A. Najjar, S. Ghosh, M. Bayoumi, "A hybrid adaptive scheme based on selective gaussian modeling for real-time object detection," in *IEEE Symp. Circuits and Systems*, May 2009, pp. 936-939.
- [13] M. Peura and J. Iivarinen, "Efficiency of simple shape descriptors," in *Int. Workshop Visual Form*, pp. 443-451, May 1997.
- [14] K. Toyama, J. Krumm, B. Brumitt, B. Meyers, "Wallflower: principles and practice of background maintenance," in *IEEE Conf CVPR*, 1999, pp. 255-261.
- [15] PETS 2006 Benchmark Data, <http://pets2006.net/>.
- [16] S. S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," *EURASIP J. Applied Signal Processing*, vol. 5, pp. 2330-2340, 2005.