MULTIPLE OBJECT TRACKING UNDER HEAVY OCCLUSIONS BY USING KALMAN FILTERS BASED ON SHAPE MATCHING

L.Marcenaro, M.Ferrari, L. Marchesotti and C.S.Regazzoni

DIBE – University of Genoa, Via Opera Pia 11a 16145 Genoa ITALY
e-mail: carlo@dibe.unige.it

ABSTRACT

This paper describes a technique for tracking single objects moving within the guarded scene during dynamic occlusion situations. The processing modules used for objects detection and tracking will be shown in details and the performances of the algorithm will be discussed. The proposed approach uses an empty reference image for object extraction through image difference; the reference frame is updated continuously by a background updating module taking into account the detected objects. The tracking module is responsible for objects labeling being able to preserve objects identity even when an overlapping occurs on the image plane between different objects. A shape matching technique is used that is based on a linear Kalman filter. The system has been tested on several outdoor sequences showing dynamic occlusions among objects in order to show the validity of the approach.

1. INTRODUCTION

During the last few years, many algorithms have been studied for automatic object tracking from video sequences for video surveillance or scene understanding purposes [1,2]. Images sequences acquired from a real outdoor scenario are characterized by a high complexity; this is typically due to different factors such as illumination changes, background motion (i.e., moving trees), complex objects interactions (i.e., structural or dynamic occlusions) and cluttered scenes. The increase of the complexity of the scene can cause several errors especially in the tracking stage. Object tracking is the process of coherently assigning identifiers to each single object in the scene during successive time instants. The tracking is lost when a certain object is mislabeled (i.e., its identifier changes during the time) or when the object is no more detected in an image of the sequence. The output of the tracking module is the basis of several higher-level algorithms that can be able, for example, to classify a dangerous behavior or situation, count the number of people in a certain area, etc. The robustness of the tracking module is then extremely important for improving the performances of the overall surveillance system. Several tracking algorithms such as the one presented in [4] are able to track individual objects whenever they are well-isolated on the image plane, but the tracking is lost when an occlusion situation is verified.

Methods to solve the occlusion problem have been previously presented [1, 6, 7, 8]. Chang et al. [6] and Dockstader et al. [7] use a multiple camera system to overcome occlusion in multi-object tracking. Khan and Shah [8] presented a system to track people in the presence of occlusion. The system segments a person into classes of similar color using the Expectation Minimization Algorithm and then uses a maximum a posteriori probability approach to track these classes from frame to frame.

This paper proposes and evaluates a novel method for object tracking under static or dynamic occlusions: by using this method, it is possible to preserve object identities when they are partially occluded by a static or moving object in the scene; the main innovation of the method consists in the joint use of the Kalman filtering with a correlation based shape matching algorithms: this allows the tracker to preserve objects identities during occlusions.

The paper is organized as follows: section 2 contains a description of the system with particular regard to the logical modules used for image analysis and scene understanding; section 3 focuses on the dynamic occlusion problem describing the limitations of the standard tracking algorithm. Sections 4 and 5 respectively explain Kalman filter and shape matching methods for dynamic occlusion handling. Main results are given in section 6 while conclusions drawn in section 7.

2. DETECTING AND TRACKING OBJECTS FROM VIDEO SEQUENCES

The processing algorithms performed by a object tracking system can be divided into separated logical modules each performing a well-defined processing task. In figure 1 a logical architecture of the proposed system is shown.

In the early stages, the system performs several operations directly on the signal coming from the frame grabber. If the sensor is very noisy, a linear noise or a median filter can be applied in order to reduce the noise in the acquired images.

The next step in the logical modules chain is the change detection module. This module can be considered as the basis of a typical automatic video-surveillance systems [3]; its purpose is to localize the objects in the scene thus reducing the amount of data to be processed by the following tasks. In a fixed-camera surveillance system a reference image (background) is often available: this can be considered as the image of the guarded area with no additional object in it. By subtracting and thresholding the current processed frame from the background frame, it is possible to produce a binary change detection image that has white pixels in correspondence of the changed areas with respect to the reference frame. Typically the background cannot be considered as a fixed image. In particular, in outdoor

0-7803-7622-6/02/$17.00 ©2002 IEEE

environments there are often heavy illumination changes that are mainly due to the variability of the sun illumination. This causes relevant errors in a video-surveillance system because, if the illumination of the acquired image is different from the reference image, the resulting change detection has a potentially high number of false alarms. The background updating stage is not strictly necessary in video-surveillance systems operating in indoor environment, but it is extremely important in outdoor scenes where the lighting condition are potentially widely variable [4].

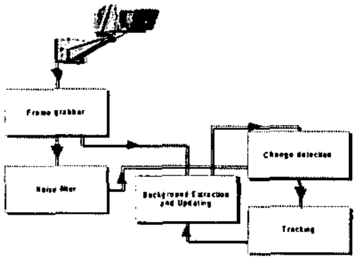

Figure 1 Logical architecture of the proposed system

The classical background updating algorithm [5] involves a simple weighting operation for generating the new background: a new background image is generated starting from the current image and the old background by using the following equation:

$$B_{k+1}(x,y)=I_k(x,y)+\alpha\left[B_k(x,y)-I_k(x,y)\right] \qquad (1)$$

being $B_k(x,y)$ and $I_k(x,y)$ respectively the background and the current image at the time k. The parameter $\alpha \in [0,1]$ is the so-called background updating coefficient. It can be seen that if α is near to 0, the new background is similar to the current image while if α is close to 1, the background updating is extremely slow. However the main problem of this procedure is that if the updating coefficient is high, the updating can be not fast enough to absorb changes due to illumination changes while if the coefficient is low, the background image tends to absorb the objects and the system becomes almost blind and unable to track the moving objects. A possible solution to this problem could be to adopt two different updating strategies according to the results from the higher level modules, by using the following rule:

$$B_{k+1}(x,y)=\begin{cases} I_k(x,y)+\alpha\left[B_k(x,y)-I_k(x,y)\right], \alpha \approx 1 \\ \quad \text{if } I_k(x,y) \in O_j, j=1,\dots,N \\ I_k(x,y)+\beta\left[B_k(x,y)-I_k(x,y)\right], \beta << 1 \\ \quad \text{if } I_k(x,y) \notin O_j, j=1,\dots,N \end{cases} \qquad (2)$$

where O_j is the *j-th* objects detected in the scene, while *N* is the total number of detected objects in the *k-th* frame. By using this technique, the background is kept updated, by absorbing variations due to the illumination changes, without integrating in the reference frame objects that are stopped in the guarded area.

The obtained binary change detection images $C_k(x,y)$ can present isolated spots due to the noise in the image; in order to avoid this kind of problem, a morphological filter can be used: in particular, statistical erosion followed by a dilatation operation with a squared structural element is performed on the image. The binary image obtained from the change detection algorithm is processed for finding 4-connected changes regions: the focus of attention module basically uses a recursive region growing algorithm. After this task, the system provides a list of regions of interest (ROI) bounded by a set of related minimum bounding rectangles:

$$S_k=\left\{R_i^k, i=1,\dots,N\right\} \qquad (3)$$

where R_i^k is the *i-th* region of interest at the frame *k* and *N* is the total number of ROIs detected in the image. Each ROI corresponds to one or more moving objects present in the scene and is defined with a four dimensional vector:

$$R_i=\left(x_i \quad y_i \quad w_i \quad h_i\right) \qquad (4)$$

where *x, y, w* and *h* are the coordinates of the upper left vertex of the *i-th* ROI and its width and height respectively. The algorithm merges regions that are partially overlapped or near: a set of close bounding boxes can be due to the splitting of a previously connected region caused by some noise in the scene. From each detected region, several features can be extracted, such as object histogram, baricenter, corners or contour points for a spline approximation of the object silhouette.

A simple object tracking procedure is based on the spatial relations between blobs in subsequent frames obtained by comparing each extracted region of interest R_i. Two detected regions R_i and R_j are overlapped if $R_i \cap R_j \neq 0$, while they are disjointed if $R_i \cap R_j = 0$: this simple test can be done on the basis of the coordinates of each extracted ROI. Each region extracted in the frame *k* is compared with each region extracted in the frame *k+1* searching for overlapping correspondences.

$$\begin{aligned} &\forall i \in \left(1,\dots,N_k\right), \forall j \in \left(1,\dots,N_{k+1}\right), \\ &I_{ij}^{k,k+1} = Area\left(R_i^k \cap R_j^{k+1}\right) \end{aligned} \qquad (5)$$

The function *Area(T)* simply computes the area of the region *T*: the area of a binary region can be extracted by counting the number of white pixels in the image. *I* is a $N_k \times N_{k+1}$ matrix capturing the information about the overlapping of the regions in consecutive frames. For assigning the correct labels to the ROIs detected in the frame *k+1*, the matrix *I* is scanned along its columns. The following three cases can be found by looking at the column *h*:

1) each entry of the column *h* is 0: the blob *h* in the frame *k+1* is marked as NEW and a new identifier is given;
2) just the element in the *s*-th row is non-zero: the blob *h* in the frame *k+1* is marked as OLD and it inherits the identifier of the region *s*-th in the frame *k*;
3) more than one element of the column *h* is different from zero: this means that the detected ROI in the frame *k+1* is obtained because of a merging of blobs in the frame *k*. The blob *h* in the frame *k+1* is marked as MERGED and it inherits each identifier of the overlapped regions.

3. OCCLUSION HANDLING

Third case represents a warning that a dynamic occlusion has occurred between two or more moving objects in the scene: by labeling the ROI as MERGED the information about the position of each single object in the ROI is lost while histogram matching techniques can be used in order to re-assign correct identifiers after the occlusion. In the following sections a technique for tracking single objects during occlusion situations based on shape matching is proposed.

The illustrated tracking algorithm does not take into account the objects movements: in particular if the acquisition frame rate is not fast enough to ensure that a certain object is overlapped in consecutive frames, the objects identifiers can be lost and the blob can be labeled as NEW in each frame of the sequence. Correct object tracking is ensured only if the acquisition frame rate is higher than the dimension of each ROI divided by its speed in the image plane:

$$t_c < \frac{w_i}{v_{x_i}} \text{ and } t_c < \frac{h_i}{v_{y_i}} \text{ with } \forall i = 1, \ldots, N \qquad (6)$$

In general this condition is not satisfied in outdoor video-surveillance systems tracking high-speed moving vehicles far from camera. The problem can be solved by using an estimation technique for predicting the position and the dimension of the ROI in the next frame: this can be done by using a Kalman filter.

4. LINEAR KALMAN FILTER

The well-known Kalman filter equations are given by [5]:

$$\begin{aligned}\mathbf{x}(k) &= \mathbf{A}\mathbf{x}(k-1) + \mathbf{w}(k-1)\\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k)\end{aligned} \qquad (7)$$

where $x(k)$, $x(k\text{-}1)$ are the state vectors at time k and $k\text{-}1$, $y(k)$ is the observation vector at time k, w and v are the noises on the state and the observation respectively. A and C are matrices specifying the state and measurement models for the considered system. The Kalman theory gives the equations for the optimal estimate $\hat{x}(k+1|k)$ given the statistics of the system and observation noises.

In the case of object tracking, the following state vector can be chose $x = [x \quad y \quad w \quad h \quad v_x \quad v_x \quad v_w \quad v_h]^T$ and $y = [x \quad y \quad w \quad h]^T$.

The covariances of the state and measurement noises are estimated directly from data. In particular the noise in the state equation should simulate the second order derivates of the observed variables.

During blob tracking a new Kalman filter is instantiated for each blob labeled as NEW: the filter is used in order to predict the position of the blob in the next frame. At the time step $k+1$, the extracted list of ROIs R_j^{k+1} is compared with the predicted list of blobs from the previous frame $\hat{R}_i^{k+1}$, where $\hat{x}^{k+1} = \left\lfloor \hat{R}_i^{k+1} \quad v_x \quad v_y \quad v_w \quad v_h \right\rfloor$. Kalman estimation is then used in order to release the condition (6). If a MERGED blob is detected, the system is not able to retrieve a new observation vector and the Kalman filter is updated only by using the previous state vector. This approach is correct if the motion of the objects in the scene is uniform, i.e. the speed is constant. If the acceleration of the considered object is not zero, the prediction error of the Kalman filter increases and it can cause a substantial tracking failure. In order to handle this kind of situations, a strategy for retrieving objects features even during a dynamic occlusion should be considered. In the following section a method based on a shape matching algorithm is proposed.

5. SHAPE MATCHING

The shape of a isolated object i in the frame k can be defined as the subpart of the change detection image within the associated bounding box, i.e.:

$$S_i^k(x, y) = \left\{C_k(x, y), (x, y) \in R_i^k\right\} \qquad (8)$$

The shape image is then a binary image and it is stored for each moving object labeled as OLD. For explaining the shape matching procedure the following notation will be adopted:

- $W_\Delta^{k+1} = \left\lfloor \hat{R}_i^{k+1} + \Delta \right\rfloor$ is a window centered on the prediction $\hat{R}_i^{k+1}$ and depending on the vector $\Delta = [a \quad b \quad c \quad d]^T$
- $\hat{S}_i^{W_\Delta^{k+1}}$ corresponds to the shape S_i^k translated and rescaled accordingly with W_Δ^{k+1};
- $\Phi(A, B) = \Phi_{A,B} = \sum_s \sum_t |A(s,t) - B(s,t)|$ is the used correlation function with A and B binary images.

Whenever an occlusion is detected in the scene (i.e., a ROI is labeled as MERGED), the following procedure is performed:

- □ $\hat{R}_i^{k+1}$ is computed by using the Kalman estimator for each blob involved with the merging event;
- □ the following correlation function is minimized with respect to the vector parameter Δ:

$$\min_{\Delta \in \Sigma}(f(\Delta)) = \min_{\Delta \in \Sigma}\left(\Phi\left(\hat{S}_i^{W_\Delta^{k+1}}, C_{k+1}\right)\right) \qquad (9)$$

being Σ the search range for the shape matching.

For each blob i in the merged ROI, the output of the procedure is a vector $\overline{\Delta}_i = \left[\overline{x} \quad \overline{y} \quad \overline{w} \quad \overline{h}\right]$ that maximizes the correlation between the stored shape and the merged change detection image. Vector $\overline{\Delta}_i$ is passed to the Kalman filter as the new measurement vector for i-th blob in the frame $k+1$.

6. RESULTS

In this section the results on several tracking experiments with various complexity will be shown. In figure 2 the output of a system without any occlusion tracking module is shown. It can be seen that during the occlusion the objects are considered as a single region of interest, while the identities can be correctly retrieved after the occlusion by using a color matching algorithm.

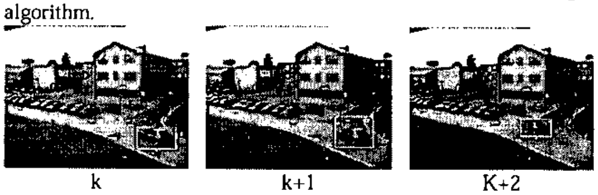

Figure 2 The output of a tracking system that is not able to individually track objects during occlusions

Figure 3 shows the result of the system using only linear kalman filtering without any shape matching procedure. In this case the Kalman estimator is updated only on the basis of the state vector during occlusions.

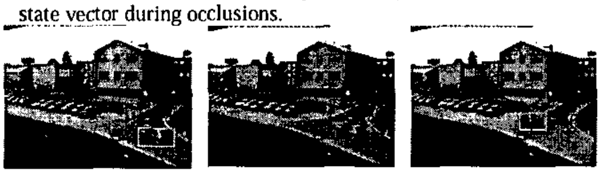

Figure 3 System using linear Kalman filtering for tracking is able to correctly solve simple occlusions where objects speed is constant during the occlusion

Figure 4 shows the tracking module output of the same system on a more complex sequence, where the speed of the tracked object sensibly changes during occlusion: in this example the tracked car stops and inverts its speed while it is merged with another vehicle. It can be seen that the tracking modules based only on the Kalman filer completely fails in this case.

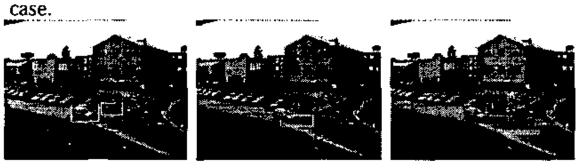

Figure 4 Kalman filter without any shape matching strategy fails when the speed of the objects changes during the occlusion

By adding the proposed shape matching algorithm, the vehicles are successfully tracked through the occlusion situation. Results are shown in figure 5.

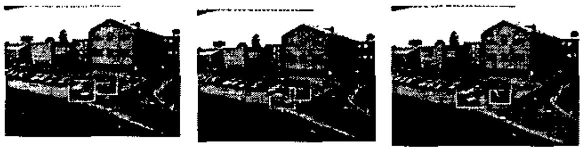

Figure 5 Tracking module based on Kalman filter and shape matching is able to track individual objects with high accuracy during occlusions

Depending on the dimension of the search range Σ in functional (9), the computational complexity of the proposed tracking module can increase. It can be experimentally found that the performances of the system can be improved by minimizing functional (9) in a two-dimensional space by considering w and h in vector Δ as fixed. It has been experimentally shown that a correct pose estimation for the occluded blob is mostly affected by the positional parameters in vector Δ, while one can suppose blob dimensions to be slowly varying during the occlusion.

The system was tested on a PC based on a 1.7 GHz processor with a Linux operating system. The image resolution used for the test was 768x576 pixels, 24 bits per pixel.

The following table summarize the average frame rate for the different systems.

Considered system	N° of frames	Time to process	Frame rate
Standard tracking module	3000	250 sec	12 fps
Kalman filtering	3000	361 sec	8.3 fps
Shape matching/ Kalman filter	3000	389 sec	7.7 fps

Table 1 Average frame rates for considered tracking algorithms

The average misdetection rate on the tested sequence is about 2% by using the shape matching technique for solving occlusion situations. In the final paper more detailed and statistically significant results will be presented over a larger set of situations.

7. CONCLUSIONS

The paper described a tracking algorithm able to solve occlusion situations among moving objects in the scene. The technique is based on a shape matching algorithm that is initialized by using a linear Kalman filter.

It has been shown that this approach is able to track moving objects within the guarded environment during a dynamic occlusion.

The shape matching algorithm is based on the maximization of a correlation function varying the shape pose parameters.

The system has been tested on outdoor sequences characterized by dynamic occlusions with different complexities; tracking results showed the validity of the approach.

8. REFERENCES

[1] I. Haritaoglu, D. Harwood, and L.S. Davis. "W^4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):809-830, August 2000.

[2] C.R. Wren, A. Azarbayejani, T. Darrel, and A.P. Pentland. "Pfinder: real-time tracking of the human body" *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):780-785, August 1997.

[3] G.L.Foresti and C.S.Regazzoni, "A change-detection method for multiple object localization in real scenes", *Proceedings of the IECON 1994*, Bologna Italy, pp. 984-987, 1994.

[4] L.Marcenaro, F.Oberti and C.S.Regazzoni, "Multiple objects color-based tracking using multiple cameras in complex time-varying outdoor scenes", *Proceedings of the PETS2001 workshop*, Kauai, Hawaii, USA, Dec. 2001.

[5] R.E. Kalman, "A new approach to linear filtering and prediction problems", *Trans. Of the ASME Journal of Basic Engineering*, pp. 35-45, 1960.

[6] T.H. Chang, S. Gong and E.J. Ong. "Tracking multiple people under occlusion using multiple cameras" *In Proc. 11th British Machine Vision Conference*, 2000

[7] S.L. Dockstader and A.M. Tekalp. "Multiple camera fusion for multi-object tracking". *In Proc. IEEE Workshop on Multi-Object Tracking*, pp 95-102, 2001

[8] S. Khan and M. Shah. "Tracking people in presence of occlusion". *In Asian Conference on Computer Vision*, 1998.