A MODEL FOR DETECTING AND TRACKING HUMANS USING APPEARANCE, SHAPE, AND MOTION

C. Pertuz, L. Mayron, D. Socek, and O. Marques.
Florida Atlantic University
777 Glades Rd., Boca Raton, FL 33428 USA
cpertuz@fau.edu, lmayron@fau.edu, dsocek@brain.math.fau.edu, omarques@fau.edu

ABSTRACT
The field of automated video surveillance has experienced increased research interest due to falling costs of video sensors, increasing security concerns, and the need for improved algorithm for extracting high-level information from video sequences. The analysis of human activities and their environment within the context of security provides information enabling the proactive identification of anomalous behavior. This makes human detection a prerequisite for the automatic extraction of higher level information, such as the recognition of the activities of individual humans. In this paper, we approach the challenge of detecting humans within video sequences as a classification task; moving objects in the foreground are either human or non-human. The classification approach presented in this work is based on motion (periodic motion detection), appearance (skin color detection), and shape (MPEG-7 shape descriptors). A modular infrastructure for data collection, object instantiation, and tracking was also implemented.

KEYWORDS
Video segmentation, human detection, object tracking

1 Introduction

The challenge of automatically extracting high-level information from video sequences has been a subject of research interest for over three decades. Even though the advances have been significant, each solution is fallible in certain circumstances. There is tremendous room to improve previous research as well as to propose new solutions.

Currently, lower prices of remotely-accessible surveillance cameras, compounded by ever-increasing security considerations have stimulated the development of surveillance centers that collect and observe the visual information. However, security personnel face the risk of being overwhelmed with massive amounts of information, potentially exceeding their monitoring capabilities. This has propelled the efforts to enhance automated video surveillance systems which have to be able to operate with little or no human intervention.

In a such an environment simpy the manifestation of an object of interest is not enough to warrant extra attention or sound an alarm. It is the interrelation between humans and objects, or just between humans, that can give hints which can be used to proactively identify an anomaly. Hence, human detection must be the foundation of systems where higher-level information, such as recognizing the activities of humans, is necessary

In this paper we propose an approach to human detection and tracking based on motion, appearance, and shape. Furthermore, we establish a framework based on a state machine for automated object extraction, instantiation, and tracking.

2 Background

2.1 Periodic motion detection

People in motion (e.g. walking or running) exhibit periodic characteristics. This was exploited by Cutler et al. [1], among others ([2],[3],[4],[5]). Similarly, Lipton,uses residual flow to analyze periodicity and rigid motion [6]. There is evidence that animals and humans can recognize the motion of other biological entities according to its periodic characteristics [6].

The technique proposed in [1] requires the isolation of an object across N consecutive frames. Once we collect that information, we resize the different images according to the median values. Then we calculate its correlation matrix, according to the following equation:

$$St_1t_2 = \sum_{(x,y)\in Bt_1} | Ot_1(x,y) - Ot_2(x,y) | \qquad (1)$$

Where B is the bounding box of object O, and t_i makes reference to the different resized instances of the object ($0 \leq i \leq N$). The next step is the computation of the correlation matrix Discrete Fourier Transform (DFT). The object's motion will be considered as periodic if there are values that meet the condition below:

$$P > \mu_p + K\sigma_p \qquad (2)$$

Where P is the DFT of the correlation matrix, K is a threshold value (typically 3, according to the authors), and

μ_P and σ_P are the mean and standard deviation of P respectively.

2.2 Skin color-based detection

Motion-based classification techniques are useful when there are humans walking (exhibiting detectable motion) in the scene, but what if the present subjects are standing still (no movement can be detected)? Using appearance-based classification we can point to objects in the scene that contain human characteristics (e.g. color). Skin color is determined by a single pigment (melanin), and only its density differs between different ethnic groups [8]. This appearance feature can be used to detect humans.

In this scheme, we take advantage of the independence between luminance and the chroma components of the *YCrCb* color space. After performing background subtraction, the chroma components of the foreground pixels are compared against predefined thresholds of the skin color. If the values Cr and Cb fall within the range [133, 173] and [77, 127], the pixel is labeled as "skin color" [7].

2.3 Shape-based detection

If we use motion and appearance techniques in a scene where our actors are walking perpendicularly to the plane of the camera or are hiding their faces, our hypothetical system, most likely, wouldn't be able to recognize them as humans. To circumvent these limitations, we can use the shape characteristics of the human silhouette in order to detect humans in the scene. In this work we use two such techniques: dispersedness and the MPEG-7 region-based shape descriptor.

2.3.1 Dispersedness

The mathematical definition of this metric is [8]:

$$Dispersedness = \frac{Perimeter^2}{Area} \quad \textbf{(3)}$$

where *Perimeter* is the number of pixels that belong to the contour of a shape, and *Area* the number of pixels contained inside the contour.

Humans tend to have more complex shapes than vehicles. If we compare a vehicle and a human in an image where both silhouettes are clearly defined the human will exhibit greater dispersedness due to the greater complexity of its shape. The computational cost of this metric is low.

2.3.2 MPEG-7 region-based shape descriptor

MPEG-7 is a standard specifically designed for multimedia content description. In this section we refer to the region-based shape descriptor, which in order to describe a shape uses the Angular Radial Transform (ART) [9] to extract the set of coefficients [10] that define the descriptor.

The distance (or dissimilarity) between two shapes described by the ART descriptor is calculated using the function:

$$Dissimilarity = \sum_i \left\| M_d[i] - M_q[i] \right\| \quad \textbf{(4)}$$

where M is the array of ART descriptor values of images d and q respectively;

The descriptor is characterized by its small size, fast extraction time, and matching [11].

3 The Proposed Approach

3.1 Block Diagram

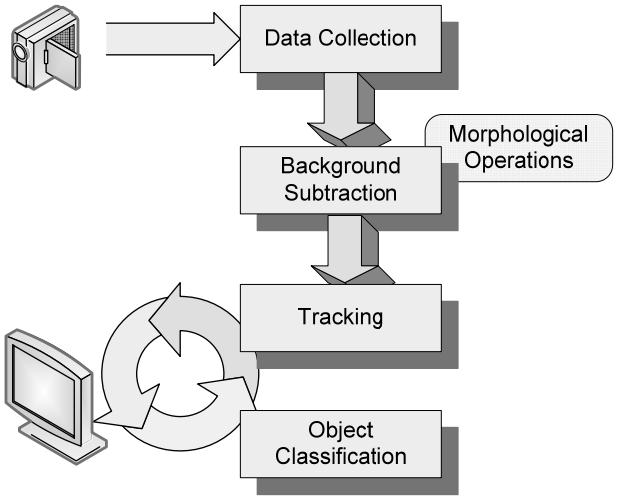


Figure 1. Block diagram of the proposed solution

The proposed solution arranges processing blocks following the diagram in Figure 1. The flow of data in the block diagram is explained in the following subsections.

3.2 Data collection

In this process we capture, edit, resample, and store the video sequences.

3.3 Background subtraction

This block extracts the foreground objects present in the video sequence. The proposed solution uses the prototype described in [12]. The output of this block is a set of binary images (each pixel is expressed with a single bit) with the same dimensions as the original image. This processing block is fundamental; performance of subsequent blocks depend on it. Its output must be as noise-free as possible. In this implementation, a *Morphological Operations* block is applied to the output of the *Background Subtraction* block in order to filter noise.

3.4 Tracking

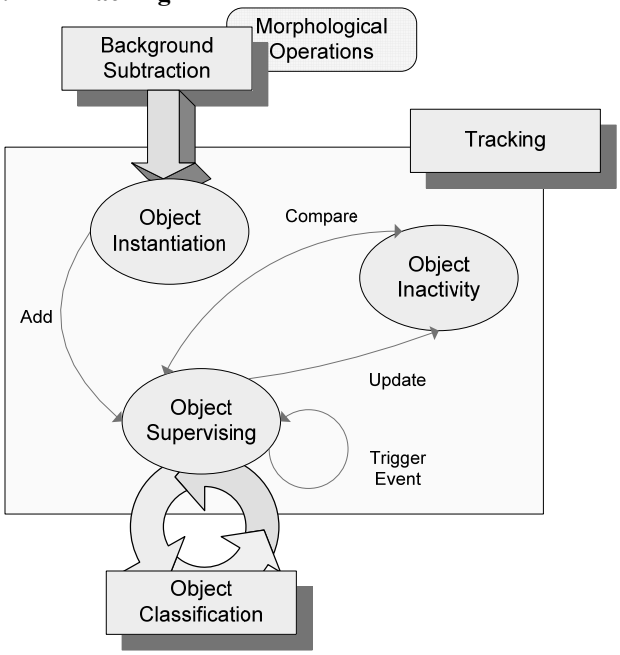


Figure 2. Monitoring State Machine

At this point we are ready to apply the Tracking block. This block follows the displacements of foreground objects present in the scene. The solution is based on the work developed by [13]. The implemented tracking algorithm consists of the following parameters [14]:

- *Object representation:* object's bounding box.
- *Feature selection for tracking:* manual, foreground object's size.
- *Object detection:* background subtraction based.
- *Object Tracking:* point tracking.

We chose to model the tracking problem using the finite state machine (FSM) shown in Figure 2.

This machine maintains the individuality of each foreground object. The transition of a foreground object from one state to another can be modeled also as the flow of information between data structures. Figure 3 shows the associated data structures for the different states of the machine:

The *3-frame buffer* belongs to the *Object Instantiation* state. The *Objects* and *Active Tracking* lists belong to the *Object Supervising* state. The *Inactive Objects* list belongs to the *Inactivity* state. The *Priority Queue* belongs to the *Trigger State* action.

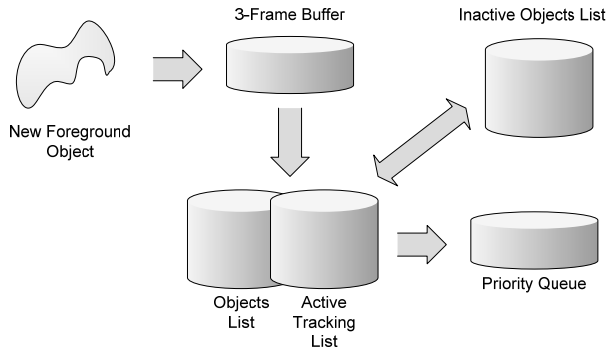


Figure 3. Associated data structures for the monitoring state machine

3.4.1 Object instantiation

In this state we label the unconnected objects in the scene. Selective information about each object is obtained. The algorithm extracts the centroid of the object, an indexed list of the pixels contained in the object, and the appropriate data needed to locate its bounding box. When a new object is discovered it is placed in a 3-frame buffer (see Figure 3), i.e., if the object is successfully tracked over three frames, it is then removed from the buffer and is ready to enter the *Object Supervising* state.

3.4.2 Object supervising

In this state an object's trajectory is calculated according to its size, presence, and previous bounding box. Also, for each object, the system keeps track of different features such as size, position, velocity, brightness level, etc.

The last function performed in this state is the comparison against the objects in the Inactivity state. If an object resembles one of the objects there, we have reasons to believe it is the same object, which had disappeared and now appears again. In that case we reuse a previous object's accumulated data.

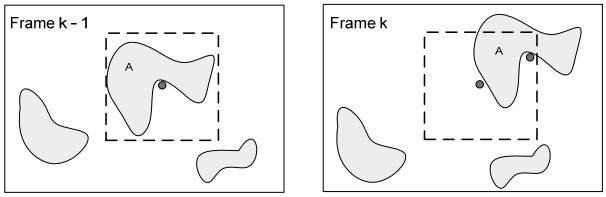


Figure 4. A foreground object's displacement

Trigger Event is the action performed by the *Object Supervising* state. For this action, different events are placed in a priority queue in which each event's priority is determined by the results of the Object Classification block (see Figure 1). Events with higher priority will be those where our object's motion, appearance and shape are human-like, whereas other events with lower priority are those where an object moves from one state to the other. In this solution the event handler places the events in two different log files. The events with the highest

priority are placed in the main log file and graphical alarms are generated for them. The other events are stored in the secondary log file.

3.4.3 Object inactivity

An object will reach this state when it is not visible anymore. For this state we implement an *Inactive Objects* list, where each element has a specific time to live (ttl). In this state flags and logical comparisons are computed in order to handle occlusion or possible feature inheritance. In the case of Figure 5, Object *B* has been occluded by object *A,* therefore becomes inactive and its *insider* flag is raised because disappeared inside the dotted region of the frame.

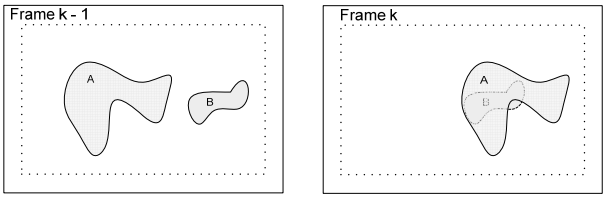


Figure 5. Occlusion example

When the object *B* reappears inside the dotted region (see Figure 5), it is labeled as an insider and it is compared only with the inactive insider objects and inherits the properties of the object whose size is the most similar to. In cases where the new object appeared outside the dotted region, it is considered as outsider and inherits the characteristics of the first inactive object that meets predetermined Size and position conditions.

3.5 Object Classification

The techniques chosen in this block for human detection are periodic motion detection [1], skin color detection [7] and the MPEG-7 shape descriptors.

The periodic motion detection sub-module requires the accumulated data from a specific number of frames. Once this information has been collected the algorithm is applied and different flags in the Active Tracking List and Objects list are raised or cleared. The *skin color detection* and *shape descriptors* sub-modules outputs only require information from the actual frame (frame *k*), However, since the system has to wait for the *periodic motion detection* sub-module output, temporal averages from these outputs are also stored. In the future a voting strategy could be applied and the object will be labeled as human if a majority of sub-modules voted in concurrence. If the results don't show anything "interesting" we continue supervising. Otherwise the system will trigger an event. Naturally, at a certain point in time a tracked object could disappear from view. In this case the object's state will change to *inactive*.

4 Results

Figure 6 shows the tracked trajectory of a human subject and its bounding box. In the right side of the figure we find the *trackingData* structure where we accumulate data for the tracked objects in the video sequence.

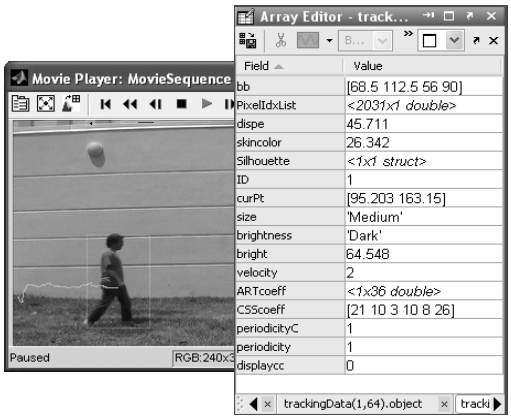


Figure 6. Trajectory, bounding box and data structure for tracked human subjects

To determine if an object has reentered the scene, i.e. was inactive in the frame (*k-1*) and is active in the frame *k*, a parameter or set of parameters must be compared. In the current implementation the objects' positions and sizes, are used to perform the comparison. In short, if the position and size conditions are met the new object inherits an inactive object's data.

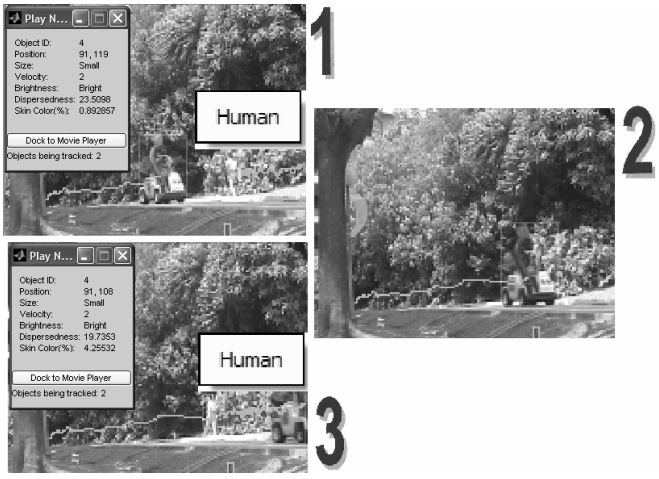
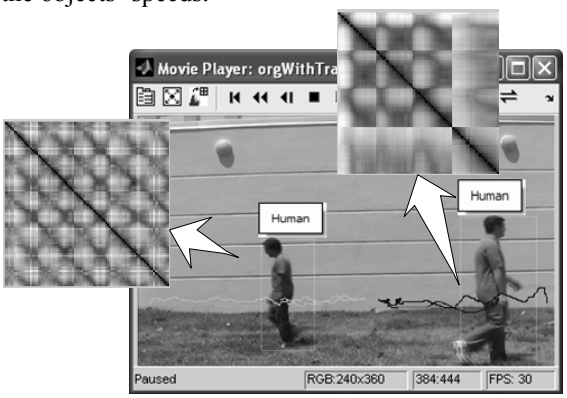


Figure 7. Object inactivity and occlusion handling

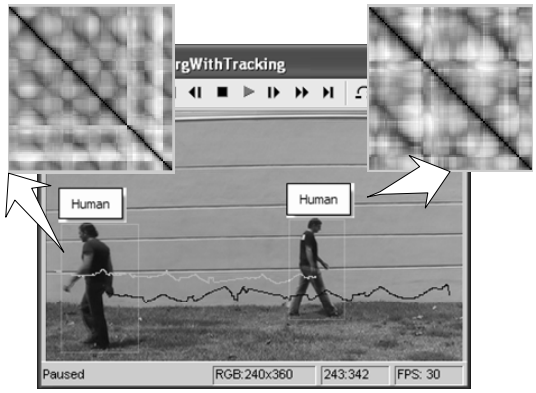
Figure 7 illustrates a scene where the previous conditions proved to be effective. In the scene a walking human identified with ID 4 (frame 1) is occluded by a vehicle (frame 2), becoming inactive. When the object reappears, first it is considered as a new object. A short while later the *compareInactive* function determines if it is similar to an inactive object. Since this is true in this case, it inherits its accumulated data (frame 3), e.g. the inactive objects' ID, the "Human" label, and the "trace" of previous tracking (trace in Figure 7)

Figure 8 shows two different scenes where periodic motion was used to detect human subjects. In Figure 8 (b) the subject with the trajectory walks at a lower pace in

comparison with the other subject present in the scene. If we chose the same column of the correlation matrices the fast Fourier Transform is used to confirm that the main frequency components are different. Accordingly, better correlation matrices could be obtained if the number of frames used to create the matrices are calculated based on the objects' speeds.



(a)



(b)

Figure 8. Human subjects and their respective correlation matrices: (a) scene one (b) scene two

In Figure 8 (a) the correlation matrix of the subject on the right side was calculated after changing the subject changed direction. This means that subject started walking from right to left and then turned around and walked from left to right. Again, better results can be obtained if the direction of motion is taken into account when calculating the correlation matrix,

Figure 9 shows a rigid object and its correlation matrix. In comparison to the matrices shown in Figure 8 the lack of symmetry to the main diagonal is perceptible and, thus, the object is labeled as not-human.

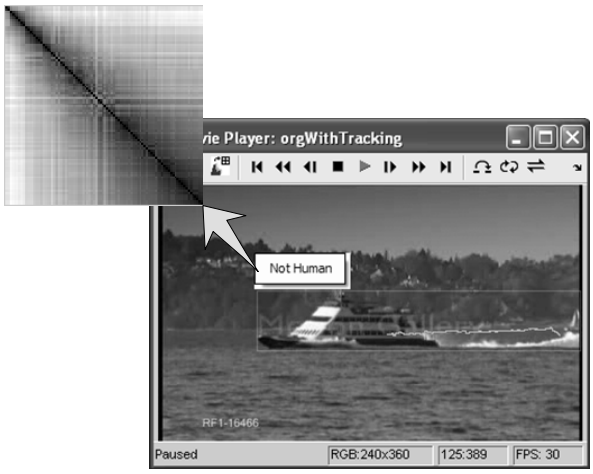


Figure 9. A rigid subject and its correlation matrix

Figure 10. Skin color detection on a sample scene

Figure 10 shows the output of the skin detection function. From a qualitative point of view the algorithm is successful labeling skin pixels in foreground objects. However, the frame in the lower right side shows some false positives. The explanation to this: shadows are elected as part of the foreground object in the background subtraction process, also parts of the floor where the shadows occur are skin colored, and therefore labeled as skin pixels.

5 Conclusions and future work

In this work, different background subtraction techniques where implemented and compared. The one that showed better qualitative and quantitative results was the one implemented in [**12**], although the tuning of the alpha parameter consumed more resources than anticipated. The measure of computation time is not critical in this implementation, but must be improved if a real-time implementation is required.

It is important to highlight that the performance of the tracking block is strongly dependent on the outputs of the background subtraction block, Also, the comparison features used in this work (centroid and bounding box) proved to work in scenes with a limited number of subjects (maximum two). Use of new features and their probabilistic properties could improve this block's

accuracy and computational performance when several objects are present in the scene at the same time.

Data collection from the MPEG-7 shape descriptors was implemented, although further testing and new metrics are required to add these integrate with the human detection algorithm. Regarding appearance, the skin detection component proved to detect "hot" spots under ideal conditions; however is not reliable as a stand alone human detector in cases where there skin is not visible or distractors (such as like-colored clothes) are present. For motion, only one descriptor was implemented. An object's velocity and direction values could be used to improve the performance of this sub-module. Empirical evaluation determined that this motion information was able to distinguish humans from non-humans.

Directions for future work include the implementation of a module for activity recognition, in order to complete the automated surveillance system. Furthermore, implementation and evaluation of a larger number of motion, appearance, and shape detection algorithms is required in order to improve the human detection module. A wider variety of test video sequences could also provide further insight. Finally, a real-time implementation must also be considered, along with the requirements such a system presents.

References

[1] R. Cutler and L. S. Davis, "Robust real-time periodic motion detection, analysis, and applications," IEEE Trans. Pattern Anal. Machine Intell., vol. 22, Aug. 2000, pp. 781–796.

[2] M. Allmen, "Image sequence description using spatiotemporal flow curves: toward motion-based recognition," PhD thesis, University of Wisconsin-Madison, 1991.

[3] Polana, R and Nelson, R C, "Detecting activities", Proc. Conf. on Computer Vision and Pattern Recognition. New York, NY, June 15-17 1993, pp 2-7

[4] Tsai, P-S, Shah, M, Keiter, K and Kasparis, T, "Cyclic motion detection for Motion Based Recognition," Pattern. Recognition, vol. 27, 1994.

[5] Allmen, M C and Dyer, C R 'Cyclic motion detection using spatiotemporal surfaces and curves', Proc. 10th Int. Conf. Pattern Recognition, Atlantic City, NJ (1990) pp 365-370

[6] G. Johansson, "Visual Perception of Biological Motion and a Model for its Analysis," Perception and Psychophysics, vol. 14, 1973, pp. 210-2 11.

[7] Chai, D.; Ngan, K.N., "Face segmentation using skin-color map in videophone applications ," Circuits and Systems for Video Technology, IEEE Transactions on , vol.9, no.4, Jun 1999, pp.551-564.

[8] A.J. Lipton, H. Fujiyoshi, R.S. Patil, "Moving target classification and tracking from real-time video," Proceedings of the IEEE Workshop on Applications of Computer Vision, 1998, pp. 8–14.

[9] Miroslaw Bober, "MPEG-7 Visual Shape Descriptors," IEEE Transactions On Circuits And Systems For Video Technology, vol. 11, No. 6, Jun. 2001, pp 716-719.

[10] Miroslaw Bober, "MPEG-7 Visual Shape Descriptors," IEEE Transactions On Circuits And Systems For Video Technology, vol. 11, No. 6, Jun. 2001, pp 716-719.

[11] http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm#E12E30

[12] Dubravko Culibrk, Oge Marques, Daniel Socek, Hari Kalva, Borko Furht, "A neural network approach to bayesian background modeling for video object segmentation," VISAPP, 2006, pp. 474-479.

[13] Jeremy Jacob. "Motion Tracking In The Presence Of Dynamic Background Movement". Final report for the Video Processing course, FAU, 2005.

[14] A. Yilmaz, O. Javed, M. Shah, "Object Tracking: A Survey," ACM Computing Survey, vol. 38, 2006.