Intelligent Video Monitoring for Anomalous Event Detection

Iván Gómez Conde, David Olivieri Cecchi, Xosé Antón Vila Sobrino,
and Ángel Orosa Rodríguez

Abstract. Behavior determination and multiple object tracking for video surveillance are two of the most active fields of computer vision. The reason for this activity is largely due to the fact that there are many application areas. This paper describes work in developing software algorithms for the tele-assistance for the elderly, which could be used as early warning monitor for anomalous events. We treat algorithms for both the multiple object tracking problem as well simple behavior detectors based on human body positions. There are several original contributions proposed by this paper. First, a method for comparing foreground - background segmention is proposed. Second a feature vector based tracking algorithm is developed for discriminating multiple objects. Finally, a simple real-time histogram based algorithm is described for discriminating movements and body positions.

Keywords: computer vision, foreground segmentation, object detection and tracking, behavior detection, tele-assistance, telecare.

1 Introduction

Life expectancy worldwide has risen sharply in recent years. In 2050 the number of people aged 65 and over will exceed the number of youth under 15 years, according to recent demographic studies [7]. Combined with sociologic factors, there is thus a growing number of elderly people that live alone or with their partners. While people may need constant care, there are two problems: not enough people to care for elderly population and the government can not cope with this enormous social spending. Thus, Computer Vision can provide a strong economic savings by eliminating the need for 24 hour in-house assistance.

Computer vision has entered an exciting phase of development and use in recent years. Present applications go far beyond the simple security camera of a decade

Iván Gómez Conde · David Olivieri Cecchi ·
Xosé Antón Vila Sobrino · Ángel Orosa Rodríguez
University of Vigo (Department of Computer Science)
e-mail: {ivangconde,olivieri,anton,aorosa}@uvigo.es

P. Novais et al. (Eds.): Ambient Intelligence - Software and Applications, AISC 92, pp. 101–108.
springerlink.com

ago and now include such fields as assembly line monitoring, robotics, and medical tele-assistance. Indeed, developing a system that accomplishes these complex tasks requires coordinated techniques of image analysis, statistical classification, segmentation and inference algorithms.

The motivation for this paper is the development of a tele-assistance application, which represents a useful and very relevant problem domain. First we must detect what we consider to be the foreground objects [4, 5], we must then track these objects in time (over serveral video frames) [8] and discerning something about what these objects are doing. There is a large body of literature in the area of human action recognition. For segmentation and tracking, for example, the review by Hu [3] provides a useful review and taxonomy of algorithms used in multi-object detection. For human body behavior determination from a video sequences, the recent reviews by Poppe [6] and Forsyth [2] provide a whirlwind tour of algorithms and techniques.

In this paper, we describe the architecture of our software system, as well as details of motion detection, segmentation of objects, and the methods we have developed for detecting anomalous events. Finally, we show the performance results and conclusions of this work.

2 The Software System

Our software application has been written in C++ and uses the OpenCV library [1], which is an open-source and cross-platform library, for developing a wide range of real-time computer vision applications. OpenCV implements low level image processing as well as high level machine learning algorithms. For the graphical interface, the QT library is used since it provides excellent cross-platform performance.

This software is an experimental application, the graphical interface is designed to provide maximum information about feature vector parameters. Thus, the system is not meant for end-users at the moment. Instead, the architecture of the system provides a plugin-framework for including new ideas. A high level schema of our software system is shown in Figure 1 with the component diagram.

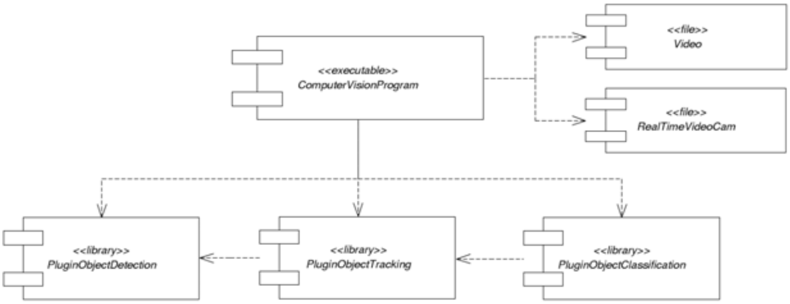

Fig. 1 Computer Vision system for video surveillance

2.1 Foreground Segmentation

The first phase of extracting information from videos consists of performing basic image processing: loading the video, capturing individual frames, and applying various smoothing filters. Next, blobs are identified based upon movement between frames. There are several background subtraction methods: Running Average and Gaussian Mixture Model (Figure 2).

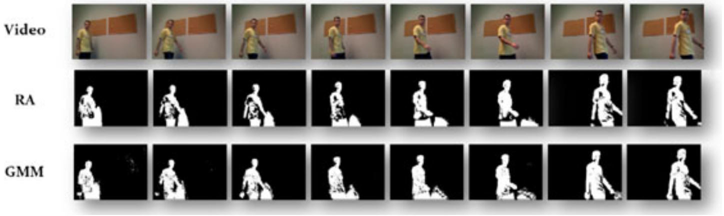

Fig. 2 Execution of “Running Average (RA)” and “Gaussian mixture model (GMM)”

The **Running Average** [1] is by far easiest to comprehend. Each point of the background is calculated as by taking the mean of accumulated points over some pre-specified time interval, Δt. In order to control the influence of previous frames, a weighting parameter α is used as a multiplying constant in the following way:

$$A_t(x,y) = (1-\alpha)A_{t-1}(x,y) + \alpha I_t(x,y) \tag{1}$$

where the matrix A as the accumulated pixel matrix, $I(x,y)$ the image, and α is the weighting parameter. We have tested 8 executions with values of α betweeen 0 and 0.8.

The **Gaussian Mixture Model** [4] is able to eliminate many of the artefacts that the running average method is unable to treat. This method models each background pixel as a mixture of K Gaussian distributions (where K is typically a small number from 3 to 5). The probability that a certain pixel has a value of x_N at time N can be written as:

$$p(x_N) = \sum_{j=1}^{K} w_j \eta(x_N; \mu_j, \sigma_j^2) \tag{2}$$

where w_j is the weight parameter of the j_{th} Gaussian component, and $\eta(x_N; \mu_j, \sigma_j^2)$ is the Normal distribution of j_{th} component.

The K distributions are ordered based on the *fitness value* $\frac{w_j}{\sigma_j}$ and the first B distributions are used as a model of the background of the scene where B is estimated as:

$$B = \arg\min \left(\sum_{j=1}^{b} w_j > T \right) \tag{3}$$

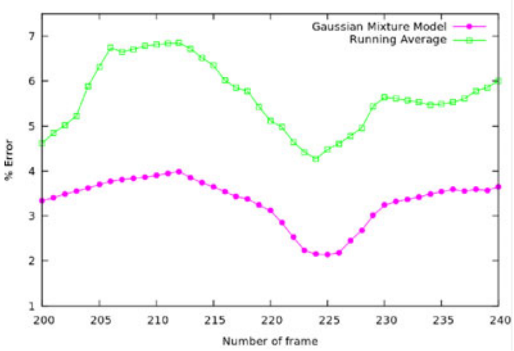

Fig. 3 % error ($\frac{FN+FP}{640 \cdot 480}$) of the best configuration with the *Running Average Model* and with the *Gaussian Mixture Model*

The threshold T is the minimum fraction of the background model. In other words, it is the minimum prior probability that the background is in the scene. In this paper we have tested 8 executions with different values of variance between 1 and 5.

The algorithms *Running Average* and *Gaussian Mixture Model* have been tested with our computer vision system. The data consists of 1 video sequence of resolution 640x480 pixels, 22 seconds of duration and 25 frames per second. We select the frames between 200 and 240. For each frame, it was necessary to manually segment foreground objects in order to have a *ground truth* quantitative comparison. We calculate the number of foreground pixels labeled as background (false negatives - FN), and the number of background pixels labeled as foreground (false positives - FP), and the total percentage of wrongly labeled pixels $\frac{FN+FP}{640 \cdot 480}$. The figure 3 shows the best results for the *Running average* with small values for alpha ($\alpha = 0.05$) and for the *Gaussian mixture model* with ($\sigma = 2.5$).

2.2 Finding and Tracking Individual Blobs

Foreground objects are identified in each frame as rectangular blobs, which internally are separate images that can be manipulated and analyzed. In order to classify each blob uniquely, we define the following feature vector parameters: (a) the size of the blob, (b) the Gaussian fitted values of RGB components, (c) the coordinates of the blob center, and (d) the motion vector. The size of blobs is simply the total number of pixels. Histograms are obtained by considering bin sizes of 10 pixels. We also normalize the feature vectors by the number of pixels.

In order to match blobs from frame to frame, we perform a clustering. Since this can be expensive to calculate for each frame, we only recalculate the full clustering algorithm when blobs intersect. Figure 4 shows excellent discrimination by using the norm histogram differences between blobs for each color space. The x-axis is the norm difference of red, while the y-axis is the norm difference histogram for green.

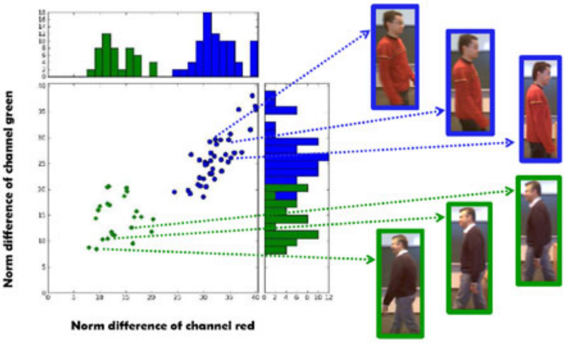

Fig. 4 Discrimination of the histogram of color space between blobs in taken from different frames

The tracking algorithm used is similar to other systems described in the literature. Once segmented foreground objects have been separated, we characterize the blob by its feature vector.

2.3 *Detecting Events and Behavior for Telecare*

For our initial work, we have considered a limited domain of events that we should detect, namely: arm gestures, and body positions upright or horizontal, to detect falls. These two cases are used to address anomalous behavior or simple help signals for elderly in their home environments. Our analysis is based upon comparing histogram moment distributions through the normalized difference of the histograms as well as the normalized difference of the moments of each histogram.

$$Hist(H_i, H_j) = \sum_{i>j} |H_i - H_j| \tag{4}$$

$$MHist(H_i, H_j) = \sum_{i>j} |M_i - M_j| \tag{5}$$

In order to test our histogram discriminatory technique, video events were recorded with very simple arm gestures, as shown in Figure 5. The foreground object was subtracted from the background by the methods previously described. For each of the histograms obtained in Figure 5, and for the histograms of Figure 6, statistical moments are calculated and then normalized histograms (normalized both by bins and total number of points) are obtained. Clustering can then be performed (similar to that of figure 4), by calculating $MHist(H_i, H_j)$, the normed difference. The histograms are obtained by summing all the pixels in the vertical direction.

The discrimination of the different body positions is possible comparing the moments of the histograms obtained (H_y - the vertical histogram). Figure 7 shows the results of different moments for frames shown in Figures 5 and 6. For example, the figure 5b (the central histogram) demonstrates a highly peaked third moment. For Figure 6, the difference in the distributions in the first and third moments is highly pronounced, and thus discrimination of the two cases is easily obtained from the simple moment analysis.

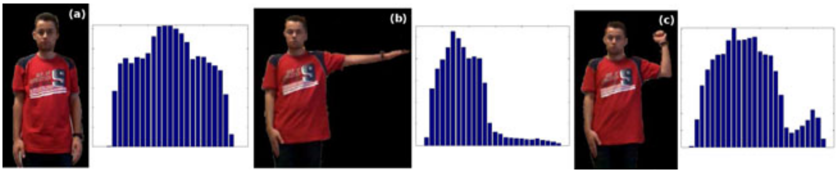

Fig. 5 Simple histogram results for obtaining moments for detecting arm gestures

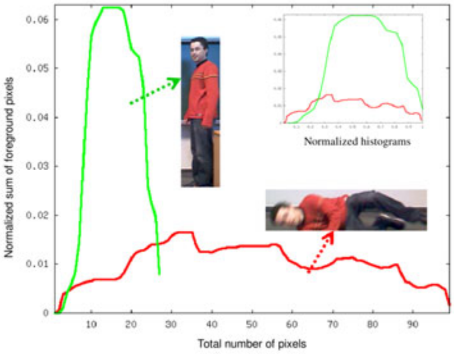

Fig. 6 Basic histogram technique used for discrimination body position. The inset image demonstrates the color space normalized to unity.

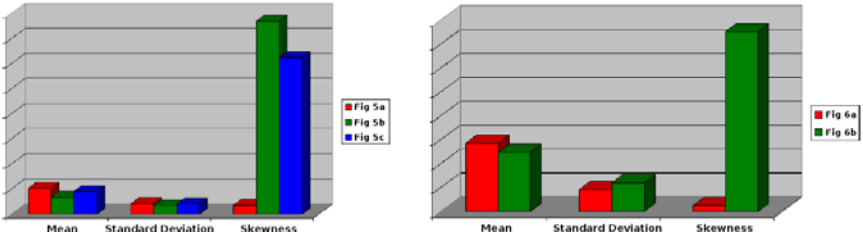

Fig. 7 Comparison of different histogram moments obtained from the video frames studied in Figure 5 and Figure 6

3 Experimental Results and Discussion

All the experimental tests and development were performed on a standard PC, consisting of an Intel Pentium D CPU 2.80GHz, with 2G of RAM and using the Ubuntu 9.10 Linux operating system. Videos and images were obtained from webcam with 2MPixel resolution.

The Figure 8 shows on a logarithmic scale the time results of the performance of the algorithms with a video of 30fps and 12 seconds of duration. The blue line represents the normal video reproduction, the magenta line is the video playing with our system without processing, the red color represents the foreground segmentation and the green line adds the time for processing blob clustering between each frame.

As shown in the previous section, the results of Figure 7 demonstrate that we can use statistical moment comparisons of histograms in order to discriminate between simple body positions. Thus, we have found that although our simple histogram techniques for human body position works well for some cases of interest and is easy to implement, it is not sufficiently robust. Because of its simplicity, however, we are presently improving the technique while at the same time investigating other algorithms.

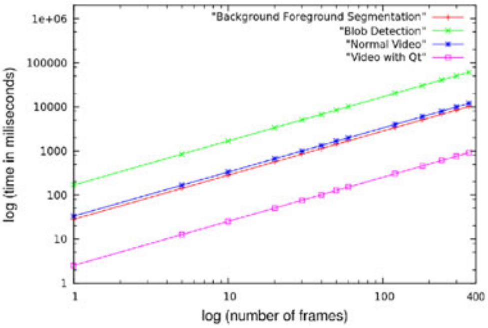

Fig. 8 Time of the different video reproductions

4 Conclusion

In this paper we have described preliminary work and algorithms on a software system which shall allow us to automatically track people and discriminate basic human motion events. This system is actually part of a more complete tele-monitoring system under development by our research group. The complete system shall include additional information from sensors, providing a complete information about a patient in their home. In this paper, however, we have restricted the study to video algorithms that shall allow us to identify body positions, in order to translate this information from a low level signal to a higher semantic level.

The paper provides encouraging result and opens many possibilities for future study. In particular, in the field of segmentation, the quantative comparison we described is an effective methodology which can be used to *optimize* parameters in each model. While the feature based tracking that used in this paper is rudimentary, a future study could combine this information with modern sequential Monte Carlo methods in order to obtain a more robust tracking. Finally, while the histogram model developed in this paper provides detection for a limited set of actions and events, it is a fast *real-time* method, that should have utility in real systems.

Acknowledgements. This work was supported by the Xunta de Galicia under the grant 08SIN002206PR.

References

1. Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly, Cambridge (2008)
2. Forsyth, D.A., Arikan, O., Ikemoto, L., O'Brien, J., Ramanan, D.: Computational studies of human motion: part 1, tracking and motion synthesis. Found. Trends. Comput. Graph. Vis. 1(2-3), 77–254 (2005)
3. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 34(3), 334–352 (2004)
4. Kaewtrakulpong, P., Bowden, R.: An improved adaptive background mixture model for realtime tracking with shadow detection. In: Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, AVBS01, VIDEO BASED SURVEILLANCE SYSTEMS: Computer Vision and Distributed Processing (September 2001)
5. Meeds, E.W., Ross, D.A., Zemel, R.S., Roweis, S.T.: Learning stick-figure models using nonparametric bayesian priors over trees. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8 (June 2008)
6. Poppe, R.: A survey on vision-based human action recognition. Image and Vision Computing 28(6), 976–990 (2010)
7. Department of Economic United Nations and Social Affairs Population Division. World population ageing 2009. Technical report (2010), `http://www.un.org/esa/population/publications/WPA2009/WPA2009-report.pdf`
8. Wei, Z., Bi, D., Gao, S., Xu, J.: Contour tracking based on online feature selection and dynamic neighbor region fast level set. In: Fifth International Conference on Image and Graphics, ICIG 2009, pp. 238–243 (September 2009)