# Real-Time Tracking of Multiple Persons

*Javier Ruiz-del-Solar*, *Alon Shats*, *Rodrigo Verschae*

Department of Electrical Engineering, Universidad de Chile, Santiago, CHILE.
Email: {jruizd}@cec.uchile.cl

## Abstract

Robust tracking of persons in real-world environments and in real-time is a common goal in many video applications. In this paper a computational system for the real-time tracking of multiple persons in natural environments is presented. The system integrates state-of-the-art methodologies for the analysis of movement and color, as well as for the detection of faces. Face detection is complemented by a face tracking module based on heuristics developed by the authors. Exemplary results of the integrated system working in real-world video sequences are shown.

## 1. Introduction

In the last years we have seen an increasing interest on video processing applications. Technological reasons like cheaper and more powerful computers, extended use of surveillance cameras, new intelligent processing methods, as well as security requirements (remember 11 Sept. 2001!) make multiple-person tracking a hot area. Video surveillance, teleconferencing, video retrieval, multimedia and virtual reality are some of the most popular applications. A common goal to these applications is robust tracking in real-world environments and in real-time.

In this context, the aim of this article is to present a computational system for real-time tracking of multiple persons in real-world environments, which integrates state-of-the-art methodologies for the analysis of movement and color, as well as for the analysis of faces.

The movement analysis subsystem is based in the system proposed in [1, 2], which uses background subtraction and selectively to exclude from the background model moving visual objects and their shadows, while retaining ghosts. The color analysis subsystem uses a standard skin detector algorithm (see for example [3]), which increases the performance of the whole system by reducing the searching region for the faces. Finally, the face analysis subsystem is based mainly on the face detection system proposed in [5]. This system uses simple, rectangular feature face detectors (a kind of Haar wavelets), the integral image for fast computation of these feature detectors, asymmetrical Adaboost as a boosting strategy for the training of the classifiers, and a cascade structure for combining successively more complex classifiers. Face detection is complemented by a face tracking module based on heuristics developed by the authors.

The article is structured as follows. Some related work is presented in section 2. The proposed real-time tracking system is presented in section 3. In section 4 exemplary results of the system applied to real video sequences are shown. Finally, in section 5 some conclusions of this work are given.

## 2. Related work

A large literature exists concerning movement analysis in video streams. As an example, last year was held the PETS 2002 event, in which several state-of-the-art tracking and surveillance systems were presented and tested [4]. Different approaches have been proposed for moving object segmentation; including frame difference, double frame difference, and background suppression. In the absence of any a priori knowledge about target and environment the most widely adopted approach is background suppression [2]. Among other approaches, the work proposed in [1] has shown good performance in the analysis of real-world video sequences [4]. It includes selective background update [2], a verification step for including ghosts into the background model and the use of the HSV color space for dealing with shadows.

Regarding face detection, different approaches have been proposed to solve this task. A very comprehensive review can be found in [6][7]. Main approaches can be classified as feature-based (low-level analysis, feature analysis and active shape models) and image-based (linear subspace methods, neural networks and statistical approaches) [6]. Image-based approaches have shown a better performance than feature based. Among them, systems like the ones developed by “Sung&Poggio”, “Rowley”, “Schneiderman” and SNoW have shown very good results (for references

Proceedings of the 12th International Conference on Image Analysis and Processing (ICIAP’03)
0-7695-1948-2/03 $17.00 © 2003 **IEEE**

and exact performance information please see [6] or [7]). However, the system proposed by Viola and Jones [5] outperforms previous systems in terms of processing speed. This system uses simple, rectangular features (a kind of Haar wavelets), a cascade of filters that discard non-face images, the integral image for fast computation of these filters and asymmetrical Adaboost as a boosting strategy for the training of the detectors.

As mentioned, the here-proposed system is based in the previously outlined state-of-the-art methods for motion analysis [1] and face detection [5]. Some modifications performed on the original algorithms are described in 3.2.5, 3.4.2, and 3.4.3. These modifications correspond to heuristics developed for the background update, the merging of overlapping face detections and for the tracking of faces.

Furthermore, a module for color analysis, based on a standard skin detector algorithm increases the system performance when color images are used.

We believe that the proposed system outperformed similar systems in the task of robust, real-world, real-time tracking of multiple-persons. For instance, a related multiple-person tracking was proposed in [3]. This system is composed by an *Interesting Region Extractor* module and a *Face Detector* Module. The region extractor is based on the integration of skin-color, motion and silhouette features, while the face detector uses a simple, rule-based face detection algorithm and SVM. Although this is a real-time system (20 frames/second using 320x240 pixels on a Pentium III, 500 MHz), it seems that it is not robust enough to work on real-world environments.

## 3. The proposed system

### 3.1 System Overview

In figure 1 is shown a block diagram of the proposed system. The system is composed by 3 main subsystems: *Movement Analysis*, *Color Analysis* and *Face Analysis*.

### 3.2. Movement Analysis Subsystem

*3.2.1. Background Suppression & Noise filtering*

The background suppression module selects foreground points at each time *t* by computing the distance, in the RGB color space, between the current frame $I_t$ and the current background model $B_t$, obtaining the difference image $DI_t$, defined for each image point (x,y) as:

$$DI_t(x,y) = \begin{cases} 1 \ if \ \ max_c\left(\left|I_t(x,y).c - B_t(x,y).c\right|\right) > Tp \\ \qquad c = R,G,B \\ 0 \ \ otherwise \end{cases} \quad (1)$$

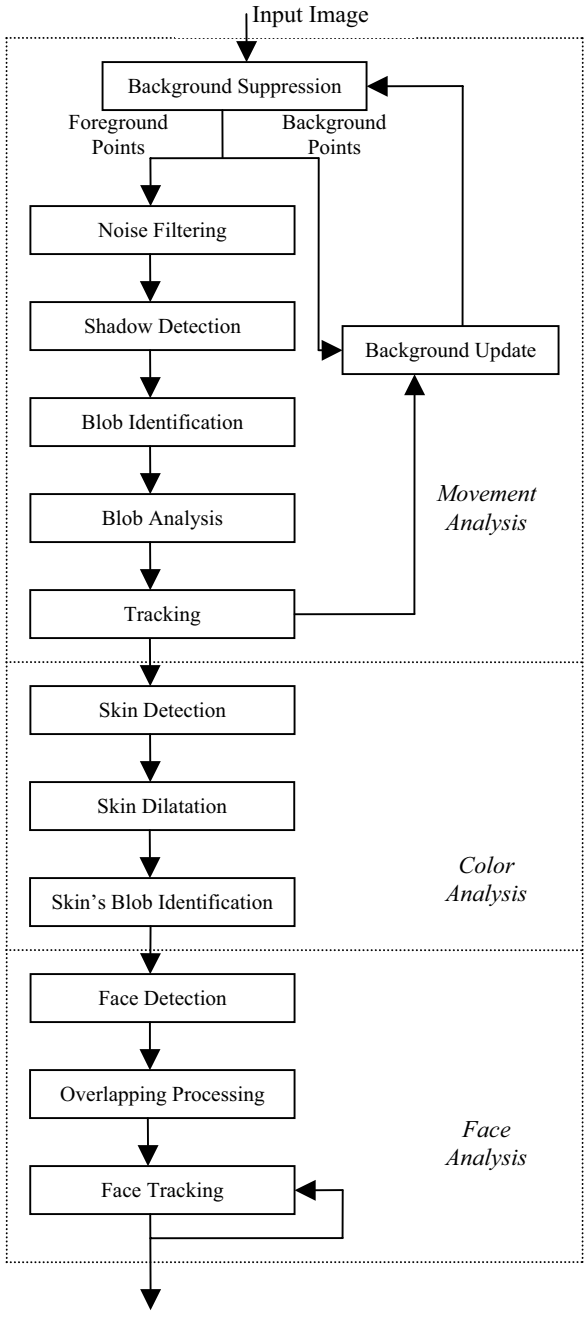



Figure 1. Block diagram of the proposed system.

$DI_t$ contains the initial set of foreground points that are candidate to belong to the MVOs (Moving Visual Object). Among the selected points, some are discarded as noise applying a 5x5 morphological opening.

*3.2.2. Shadow Detection*

The HSV color space is used to classify points as shadow. The following mask is employed:

$$SP_t(x,y) = \begin{cases} 1 & if\ \alpha \le \dfrac{I_t(x,y).V}{B_t(x,y).V} \le \beta \\ & \wedge I_t(x,y).S - B_t(x,y).S \le Ts \\ & \wedge \left| I_t(x,y).H - B_t(x,y).H \right| \le Th \\ 0 & otherwise \end{cases} \quad (2)$$

The following assumptions are taken into account:

i) A shadow darkens the background, while an object not necessary will do so. That is why the lightness ratio between a point in the current frame and in the background is bounded by two thresholds $\alpha$ and $\beta$. The first one depends in the illumination of the scene. The second value is chosen empirically (between $\alpha$ and 1), and is use to discard noise and label it as a shadow.

ii) A shadow will not result in a great modification in the color information. That is why the differences of both H and S components (i.e. the chromaticity information) are limited. The values of the two thresholds *Ts* and *Th* are both chosen empirically, and depend on the illumination conditions.

*3.2.3. Blob Identification*

By means of 8-connectivity, we detect all the blobs of connected candidate moving points. Blobs with small area are discarded as noise (blobs should have an area greater than a *Tb* threshold depending on the scene), while the rest are validated as actual MVOs.

*3.2.4. Blob Analysis and Tracking*

For each MVO we compute its average speed *AS* by means of frame-difference. By using a threshold on *AS* we separate the MVO as a moving MVO and stopped MVO. Then we make a so-called history list *DBVOt* of all the MVO at time *t* as:

$$DBVO_t = \{movingMVO_t + stoppedMVO_t\} \quad (3)$$

A $stoppedMVO_t$ could be due a ghost or a stopped object. In both case a tracking is needed for its correct analysis and for a correct background update. The tracking is achieved by comparing the visual features of a MVO (moving or stopped) with $DBVO_{t-1}$ using a comparison function. This function verifies whether the MVO is similar to a MVO detected at the previous frame or not, in terms of position and area ratio. In this way, the system keeps track of the detected MVO. Thus, if an object is detected as stopped for a time greater than a *Maximum Stopped Time* (MST) threshold, it is included in the background.

*3.2.5. Background Update*

The background model is computed as a statistical combination of a sequence of previous frames and the previously computed background (adaptability). The statistical function used is the median. In order to improve the background update, we use *selectivity*, so the background is updated as follows:

$$B_{t+1}(x,y) = \begin{cases} B_t(x,y) & if\ (x,y) \in MVO \\ I_t(x,y) & if\ (x,y) \in \{stoppedMVO \wedge MST\} \\ \text{median}(I_t(x,y), I_{t-1}(x,y), ..., & \\ \qquad I_{t-n}(x,y), W_b B_t(x,y)) & otherwise \end{cases} \quad (4)$$

A novel approach implemented in this work is that the selectivity is not only used in the current frame, but also in the previous sequence of frames. That means that points belonging to MVOs in each frame of to the most current frame's sequence are discarded from the median function. We call this proposal *historical selectivity*. In this way, the statistical function is using only the information of points not belonging to an MVO, and for that the adaptive background update is improved. Then, the median function is implemented as follows:

$$\text{median}(S_t(x,y), S_{t-1}(x,y), ..., S_{t-n}(x,y), W_b B_t(x,y)) \quad (5)$$

where:

$$S_t(x,y) = \begin{cases} I_t(x,y) & if\ (x,y) \notin MVO \\ S_{t-1}(x,y) \wedge \mathrm{S_{t-i}} = \mathrm{S_{t-(i+1)}}, & \\ \qquad i = 1 \ldots n & otherwise \end{cases} \quad (6)$$

## 3.3. Color Analysis Subsystem

*3.3.1. Skin Detection*

To reduce the search area for face detection (increasing the processing speed and decreasing the false detection rate), we used simple rules to verify if a point belonging to MVO has a skin color or not, using the pixels' normalized RG color space information. Normalized RG color space is compute as:

$$Rn = \frac{R}{R+G+B}, \qquad Gn = \frac{G}{R+G+B} \quad (7)$$

In this color space, skin colors create a cluster that can be delimitated by five straight lines (figure 2), whose parameters ($m_i$, $c_i$) were chosen by inspection. The skin pixels of the cluster were collected from several images in which skin pixels were manually selected. We select a point (x,y) as a candidate of skin as:

$$SO(x,y) = \begin{cases} 1 & if\ (x,y) \in BDVO_t \\ & \wedge m_i \times (x,y).Gn + c_i \le p_i \times (x,y).Rn \\ & \forall i \in \{1,2,3,4,5\} \\ & p_i \in \{-1,1\} \\ o & otherwise \end{cases} \quad (8)$$

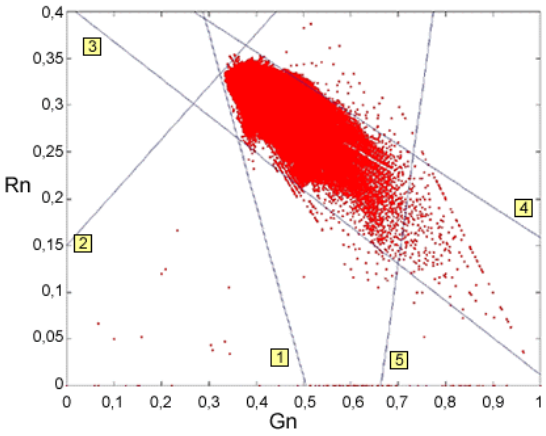


Figure 2. Skin cluster en the RG color space.

*3.3.2. Skin Dilation*

Among the selected points, some are discarded as noise applying a 5x5 morphological opening, and afterward a dilatation using a 3x3 square structuring element is performed.

*3.3.3. Skin´s Blob Identification*

A region-based labeling is preformed to compute the connected skin's blobs of skin pixels (by means of 8-connectivity). Blobs with small area are discarded as noise (blobs have to have an area greater than a *Td* threshold), while the rest are validated as a *Potential Face Area* (PFA).

## 3.4. Face Analysis Subsystem

*3.4.1. Face Detection*

The implemented detection subsystem detects frontal faces with small in-plane rotations and it is based mainly on [5]. This face detector corresponds to a cascade of filters that discard non-faces and let faces to pass to the next stage of the cascade. This architecture seeks to have a fast face detector, considering the fact that only a few faces are to be found an image, while almost all the image area correspond to non-faces. The fast detection is achieved in two ways: (i) having a small complexity in the first stages of the cascade (filters composed by few detectors, 2 to 5) and greater complexity in the later stages of the cascade (filters composed by many detectors, 100 to 400), and (ii) using simple features called rectangular features (the detectors), which are quickly evaluated using a representation of the image called integral image.

Each of the filters of the cascade is trained using an asymmetric version of Adaboost (see explanation in [5]), which gives more importance to errors occurring during the training process when classifying faces as non-faces than non-faces as faces. Adaboost sequentially trains and selects a small number of rectangular features. The main problem of this face detector is training time, which can extend for weeks or even months when a single computer is used.

The final cascade Adaboost detector implemented in this work has 21 layers and was trained in about one month. To train each layer 1338 face images were used, and non-faces images were collected from 4000 images that did not contain any face. All these training images were obtained mainly from Internet, especially from the google image searcher, and from personal images. For training the first 2 filters, 4000 and 3000 non-face images were randomly chosen from our dataset. For training the remaining filters of the cascade, 1500 non-face images wrongly classified by the already trained cascade were collected (a kind of boostraping). For reducing the training time a randomly chosen subset of the set of rectangular features was used in each iteration of the Adaboost algorithm. Each time that a decision rule was trained, only 50% of the training examples (faces and non-faces) were employed. As a results of the training, the final number of detectors used at each of the 21 stages of the cascade was 2, 5, 20, 20, 50, 50, 100, 100, 100, 100, 100, 200, …, 200 and 400, respectively. The detector was trained using 24x24 pixel windows. For detecting faces at different scales, the detector was scaled multiple times using a scale factor of 1.2. The detector was only applied to areas were movement and skin were located in the present frame.

*3.4.2 Overlapping Detections Processing*

Face windows obtained in the face detection module are processed and fused for determining the size and position of the final detected faces. Overlapping detections were processed for filtering false detections and for merging correct ones. All detections (detected face region) were separated in disjoint sets using the following heuristic.

Considering the inscribed circumference of each square face region, two detections belonged to the same set if the sum of their circumference radius is smaller than 0.4 times the distance between their centers, and if each radius is not larger than twice the other. If a set contained only one element, this detection is discarded. Detections belonging to each set are merged by averaging the coordinates of the corners of all square face regions. Figure 3 shows an example of applying this heuristic on an image that has two close faces.

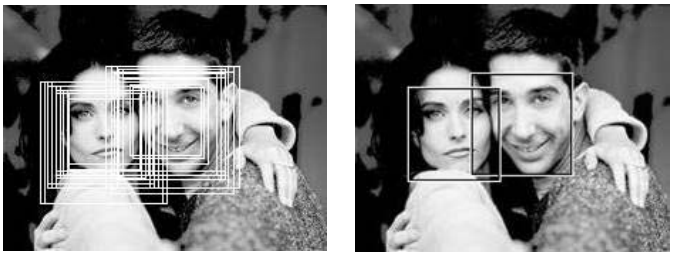


**Fig. 3**. Results from the use of the heuristic that separates and merges overlapping detections. (a) Overlapping detection, (b) Final detections

*3.4.3. Face Tracking*

This module was used to filter false detections. This filtering corresponds to an inter-frame operation, while the filtering applied in the *Overlapping Detections Processing* module to an intra-frame operation.

Face detections belonging to consecutive frames were considered to be the same face, by applying the same heuristic used to process overlapping detections (previously described in section 3.4.2), but with a threshold of 0.6 instead of 0.4. To consider consecutive detections as a single face, detections at 3 consecutive frames should occur. A living time of 2 frames was assigned to each detection. Each time a face was not detected this value was diminishing in one. If the face was detected again, this value was set to 2.

## 4. Exemplary results

In figure 4 and 5 are shown exemplary results of the application of the proposed multiple person tracking system using real-world video sequences taken at our Faculty. Video sequences were 320x240 pixels size. In figure 4, frames 96, 102, 108 and 114 of a first video (of 200 frames) sequence are shown. Face detection windows are indicated in green. In figure 5, frames 84, 88, 92 and 96 of a second video (of 169 frames) sequence are shown. Every person, whose face appears frontal, was detected at least once in these sequences. In figures 4 and 5 face detection windows are indicated in green. The videos were captured with a digital video camera with a frame rate of 29.74 frames per second, and later compressed in avi format, obtaining low quality images. Both videos contain 4 or more faces that appear at least on 30 frames each face. The computer used was an Athlon 1.2 GHz. The frame rate obtained was about 8.5 frames per second. At the moment we are performing extensive tests of our system, which are going to be documented in future publications.

## 5. Conclusions

We believe that the proposed system outperformed similar systems in the task of robust, real-world, real-time tracking of multiple-persons. Work has to be done to improve the speed of the system, mainly in the face detector (to achieve a performance similar to the original version [5]) and in the blob identification block (color and movement) which are the slowest parts of the system. Further work can be done in terms of skin segmentation (automatic illumination adjustment, skin model and skin clustering), object tracking (overlapped object tracking and future position estimation) and human body detection


## Acknowledgement

This research was funded by the FONDECYT Project 1030500 (CONICYT, Chile).


## 6. References


[1] R. Cucchiara, C. Grana, M. Piccardi and A. Prati, "Detecting Objects, Shadows and Ghosts in Video Streams by Exploiting Color and Motion Information", *Proc. of the 11th Int. Conf. On Image Analysis and Processing – ICIAP*, 360 – 365, Palermo, Sept. 2001.

[2] R. Cucchiara, C. Grana and A. Prati, "Detecting Moving Objects and their Shadows - An evaluation with the PETS2002 Dataset", *Proc. of the 3rd IEEE Int. Workshop on PETS*, 18-25, Copenhagen, June 2002.

[3] J.-W. Hsieh, Y.-S. Huang, "Multiple-Person Tracking System for Content Analysis", *Int. Journal of Pattern Recog. and Artificial Intelligence - IJPRAI*, Vol. 16, No. 4 (2002) 447 – 462.

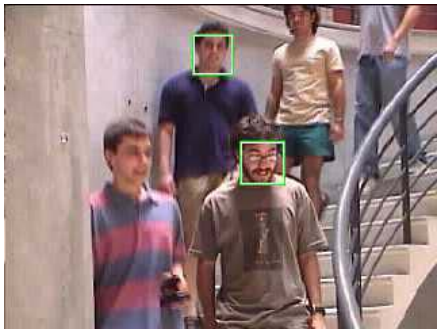
[4] J. Ferryman (Ed.), *Proc. of the 3rd IEEE Int. Workshop on PETS*, Copenhagen, June 2002.

[5] P. Viola and M. Jones, "Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade", Advances in Neural Information Processing System 14, MIT Press, Cambridge, MA, 2002.

[6] E. Hjelmås, B. K. Low, "Face detection: A survey", *Computer Vision and Image Understanding* 83, 236-274, 2001.

[7] M. Yang, D. Kriegman, N. Ahuja, "Detecting Faces in Images: A Survey", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 24, No 1, pp. 34-58, Jan. 2002.

Frame 96

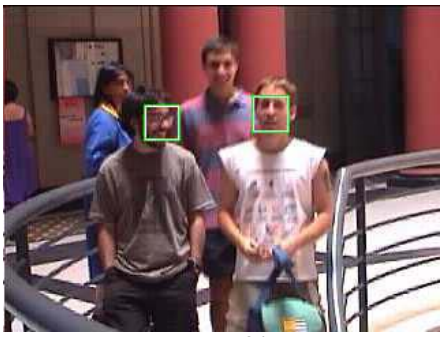


Frame 102

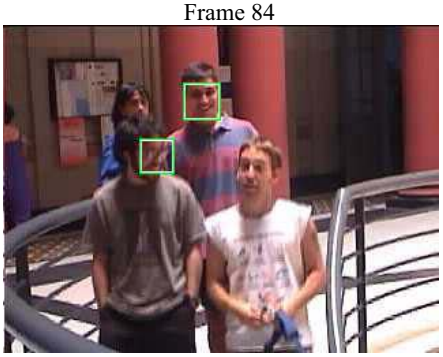
Frame 108

Frame 114

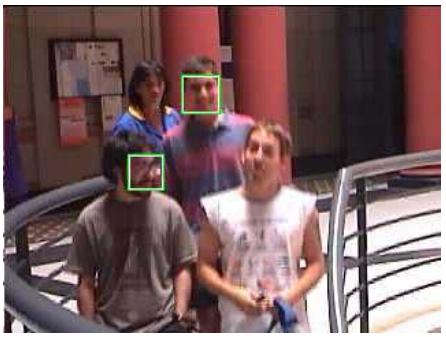
**Fig. 4**. Exemplary results of the proposed system on the video 1. Face detection windows are shown. All faces were detected at least once. One frontal face was not detected in the frame 96. The remaining frontal faces were always detected.



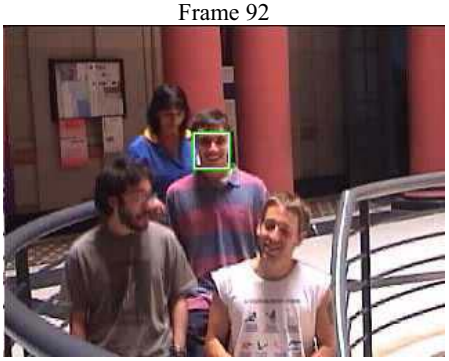
Frame 84



Frame 88



Frame 92



Frame 96

**Fig. 5**. Exemplary results of the proposed system on de video 2. Face detection windows are shown. One frontal faces was not detected in each frame.