FACE TRACKING BASED ON HAAR-LIKE FEATURES AND EIGENFACES

Paulo Menezes [*,**] **José Carlos Barreto** [*,***]
Jorge Dias [*]

[*] *ISR-University of Coimbra, Portugal*
[**] *LAAS-CNRS, Toulouse, France*
[***] *ESA, Villafranca, Spain*

Abstract: This paper describes an algorithm for human tracking using vision sensing, specially designed for a human machine interface of a mobile robotic platforms or autonomous vehicles. The solution presents a clear improvement on a tracking algorithm achieved by using a machine learning approach for visual object detection and recognition for data association. The system is capable of processing images rapidly and achieving high detection and recognition rates. This framework is demonstrated on the task of human-robot interaction. There are three key parts on this framework. The first is the person's face detection used as input for the second stage which is the recognition of the face of the person interacting with the robot, and the third one is the tracking of this face along the time. The detection technique is based on Haar-like features, whereas eigenimages and PCA are used in the recognition stage of the system. The tracking algorithm uses a Kalman filter to estimate position and scale of the person's face in the image. The data association is accelerated by using a subwindow whose dimensions are automatically defined from the covariance matrix of the estimate. Used in real-time human-robot interaction applications, the system is able to detect, recognise and track faces at about 16 frames per second in a conventional 1GHz PentiumIII laptop.

Keywords: Visual Tracking, Kalman Filter, Cascaded Classifier

1. INTRODUCTION

The development of new human-machine interface for autonomous vehicles and mobile platforms is a key feature to increase the number of applications of these technologies. The actual use of these devices is strongly dependent on new interfaces specially based on off-the-shelf technologies, such as video sensors. This paper describes a solution for human tracking using video signals which was specially designed for use in a human machine interface and action interpretation.

Towards this end it was constructed a Real-time face recognition system with a preprocessing stage based on a rapid frontal face detection system using Haar-like features introduced by Viola et al. (Viola and Jones, 2001) and improved by Lienhart et al. (Lienhart and Maydt, 2002; Rainer Lienhart and Pisarevsky, 2002).

The detection technique is based on the idea of the wavelet template (Oren *et al.*, 1997) that defines the shape of an object in terms of a subset of the wavelet coefficients of the image. Like Viola et al. (Viola and Jones, 2001) we use a set of features which are reminiscent of Haar Basis functions.

Any of these Haar-like features can be computed at any scale or location in constant time using the integral image representation for images. In spite of having face detection and false positive rates equivalent to the best published results (Rowley *et al.*, 1998; Schneiderman and Kanade, 2000; Sung and Poggio, 1998), this face detection system distinguishes from previous approaches (Yang, 2002) in its ability to detect faces extremely fast.

The face recognition system is based on the eigenfaces method introduced by Turk et al. (Turk and Pentland, 1991). Eigenvector-based methods are used to extract low-dimensional subspaces which tend to simplify tasks such as classification. The Karhunen-Loeve Transform (KLT) and Principal Components Analysis (PCA) are the eigenvector-based techniques we used for dimensionality reduction and feature extraction in automatic face recognition.

The built system, that will be used in a human-robot interaction application, is able to robustly detect and recognise faces at approximately 16 frames per second in a 1GHz PentiumIII laptop.

This article is structured as follows: Section I presents to the face detection mechanism that uses classifiers based on Haar-like features. Section II refers to the eigenimage based recognition of faces. Section III presents the tracker mechanism which is based on a Kalman filtering approach. Section IV presents the architecture of the on-line face recognition system whose results are presented on section V. In this latter section some real data results are presented where it can be seen that multiple faces are detected in images but only one is recognised as the interacting one. Section VI concludes this article.

2. USING FEATURES

Isolated pixel values do not give any information other than the luminance and/or the colour of the radiation received by the camera at a given point. So, a recognition process can be much more efficient it is based the detection of features that encode some information about the class to be detected. This is the case of Haar-like features that encode the existence of oriented contrasts between regions in the image. A set of these features can be used to encode the contrasts exhibited by a human face and their spacial relationships. One of the problems that these kind of approaches present is the computation effort that is required to compute each of the features as a window sweeps the whole image at various scales. Fortunately, each of the used features can be computed by peeking 8 values in a table (the integral image) independently of the position or scale.

Our feature pool was inspired by the over-complete Haar-like features used by Papageorgiou et al. in (Oren *et al.*, 1997; Mohan *et al.*, 2001) and their very fast computation scheme proposed by Viola et al. in (Viola and Jones, 2001) improved by Lienhart et al. in (Lienhart and Maydt, 2002). More specifically, we use 14 feature prototypes (Lienhart and Maydt, 2002) shown in Fig. 1 which include 4 edge features, 8 line features and 2 centre-surround features. These prototypes are

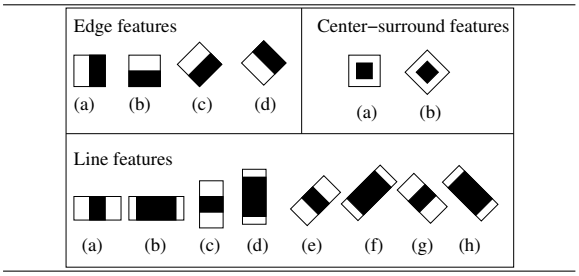

Fig. 1. Examples of the used Feature prototypes

scaled independently in vertical and horizontal direction in order to generate a rich, over-complete set of features. These features can be computed in a constant and short time irrespectively of the their position as shown in (Barreto *et al.*, 2004).

2.1 Learning Classification Functions

Given a feature set and a training set of positive and negative sample images, any number of machine learning approaches could be used to learn a classification function. A variant of AdaBoost is used both to select a small set of features and train the classifier. In its original form, the AdaBoost learning algorithm is used to boost the classification performance of a simple (also called weak) learning algorithm. Recall that there are over 117,000 rectangle features associated with each image 24×24 sub-window, a number far larger than the number of pixels. Even though each feature can be computed very efficiently, computing the complete set is prohibitively expensive. The main challenge is to find a very small number of these features that can be combined to form an effective classifier. In support of this goal, the weak learning algorithm is designed to select the single rectangle feature which best separates the positive and negative examples. For each feature, the weak learner determines the optimal threshold classification function, such that the minimum number of examples are misclassified. A weak classifier $h_j(x)$ thus consists of a feature f_j, a threshold θ_j and a parity p_j indicating the direction of the inequality sign:

$$h_j(x) = \begin{cases} 1 \; p_j f_j(x) < p_j \theta_j \\ 0 \; otherwise \end{cases} \qquad (1)$$

here x is a 24×24 pixel sub-window of an image. See (Freund and E.Schapire, 1996) for a summary of the boosting process.

2.2 Cascade of Classifiers

This section describes an algorithm for constructing a cascade of classifiers (Viola and Jones, 2001) which achieves increased detection performance while radically reducing computation time. The key insight is that smaller, and therefore more efficient, boosted classifiers can be constructed which reject many of the negative sub-windows while detecting almost all positive instances. Simpler classifiers are used to reject the majority of sub-windows before more complex classifiers are called upon to achieve low false positive rates.

A cascade of classifiers is degenerated decision tree where at each stage a classifier is trained to detect almost all objects of interest while rejecting a certain fraction of the non-object patterns (Viola and Jones, 2001) (see Fig. 2).

Each stage was trained using the Adaboost algorithm. At each round of boosting is added the feature-based classifier that best classifies the weighted training samples. With increasing stage number, the number of weak classifiers, which are needed to achieve the desired false alarm rate at the given hit rate, increases (for more detail see (Viola and Jones, 2001)).

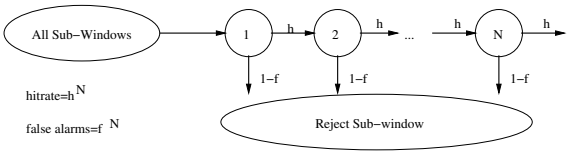

Fig. 2. Cascade of Classifiers with N stages.

3. FACE RECOGNITION USING EIGENFACES

The face recognition system is based on eigenspace decompositions for face representation and modelling. The learning method estimates the complete probability distribution of the face's appearance using an eigenvector decomposition of the image space. The face density is decomposed into two components: the density in the principal subspace (containing the traditionally-defined principal components) and its orthogonal complement (which is usually discarded in standard PCA) (Moghaddam and Pentland, 1995).

3.1 Principal Component Analysis (PCA)

Given a training set of $W \times H$ images, it is possible to form a training set of vectors $\mathbf{x}^T$, where $\mathbf{x} \in \mathbb{R}^{N=W*H}$. The basis functions for the Karhunen Loeve Transform (KLT) are obtained by solving the eigenvalue problem:

$$\Lambda = \Phi^T \Sigma \Phi \tag{2}$$

where Σ is the covariance matrix, Φ is the eigenvector matrix of Σ and Λ is the corresponding diagonal matrix of eigenvalues λ_i. In PCA, a partial KLT is performed to identify the largest eigenvalues eigenvectors and obtain a principal component feature vector $\mathbf{y} = \Phi_M^T \tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$ is the mean normalised image vector and Φ_M is a sub-matrix of Φ containing the principal eigenvectors. PCA can be seen as a linear transformation $\mathbf{y} = T(\mathbf{x})$: $\mathbb{R}^N \rightarrow \mathbb{R}^M$ which extracts a lower-dimensional subspace of the KL basis corresponding to the maximal eigenvalues. These principal components preserve the major linear correlations in the data and discard the minor ones.

Using the PCA it is possible to form an orthogonal decomposition of the vector space $\mathbb{R}^N$ into two mutually exclusive and complementary subspaces: the feature space $F = \{\phi_i\}_{i=1}^M$ containing the principal components and its orthogonal complement $\bar{F} = \{\phi_i\}_{i=M+1}^N$. The $\mathbf{x}$ component in the orthogonal subspace $\bar{F}$ is the *distance-from-feature-space* (DFFS) while the component which lies in the feature space F is referred to as the "distance-in-feature-space" (DIFS) (Moghaddam and Pentland, 1995). Fig. 3 presents a prototypical example of a distribution embedded entirely in F. In practice there is always a signal component in $\bar{F}$ due to the minor statistical variabilities in the data or simply due to the observation noise which affects every element of x.

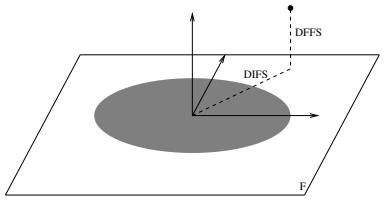

Fig. 3. Decomposition into the principal subspace F and its orthogonal complement $\bar{F}$ for a Gaussian density

The reconstruction error (or residual) of the eigenspace decomposition (referred to as DFFS in the context of the work with eigenfaces (Turk and Pentland, 1991)) is an effective indicator of similarity. This detection strategy is equivalent to matching with a linear combination of eigentemplates and allows for a greater range of distortions in the input signal (including lighting, and moderate rotation and scale).

The DFFS can be thought as an estimate of a marginal component of the probability density and a complete estimate must also incorporate a second marginal density based on a complementary DIFS. Using these estimates the problem of face recognition can be formulated as a maximum likelihood estimation problem. The likelihood estimate can be written as the product of two marginal and independent Gaussian densities corresponding to the principal subspace F and its orthogonal complement $\bar{F}$:

$$\hat{P}(\mathbf{x}) = P_F(\mathbf{x}) \cdot \hat{P}_{\bar{F}}(\mathbf{x}) \tag{3}$$

where $P_F(\mathbf{x})$ is the true marginal density in $F-$ *space* and $\hat{P}_{\bar{F}}(\mathbf{x})$ is the estimated marginal density in the orthogonal complement $\bar{F}-$ *space* (Moghaddam and Pentland, 1995).

4. TRACKING ALGORITHM

The inclusion of a Kalman filter serves two purposes: increase the quality of the tracking and increase the processing speed. The first purpose will help in producing estimates of the position of the tracked face when the face detector failed. Although the cascade classifier is quite robust it is trained to detect frontal faces only and when the user turns slightly his head to look at something else, the classifier might fail. The role of the tracker is to produce an estimate that is used as the best information when the classifier fails. A constant velocity model for the dynamics of the target in the image plane of the form

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \nu_{k-1}) \tag{4}$$

for the evolution of system state and

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mu_k) \tag{5}$$

for the measurements, where $\mathbf{x}_k = \begin{bmatrix} x & y & s & \dot{x} & \dot{y} & \dot{s} \end{bmatrix}_k^T$ is the state vector that contains the position in the image plane and a scale factor as well as their first derivatives. ν_k and μ_k are realisations of the process and measurement noise respectively. This model is used to construct a Kalman filter whose equations can be found in (Kalman, 1960).

The purpose of increasing the speed of the tracker is attained by reducing the image area where the classifier is going to search for faces. The search area is centred on the estimated position and its size depends on the values found on the diagonal of the covariance matrix. The effect of this is that when the estimate is good enough and the tracked face is found inside the search window the variance is small and so is the size of this window resulting in a higher frame processing rate. If the face is not found inside the search window the prediction is not corrected and the covariance grows. After a few iterations without detecting the tracked face the search window will occupy the area corresponding to the whole image what will reduce to the classical application of the classifier.

5. SYSTEM ARCHITECTURE

The system architecture is made of three main modules: learning, face detection and face recognition. The first one is the learning process in which the system builds the eigenspace of the person with whom the robot is going to interact. Once this eigenspace is calculated the system is able to recognise the face of the person during the tracking process. For each captured image the system detects and extracts the faces, and projects them in the eigenspace of the person the robot is interacting with in order to know if it is interacting with the right person and where is the person in the image (see figure 4.

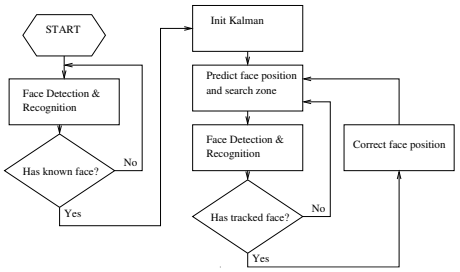

Fig. 4. System Architecture

5.1 Learning Process

The learning process starts with the acquisition of a face images sequence of the person the robot is going to interact with. The person should stay in front of the camera until face detector detects and extracts 40 face images.

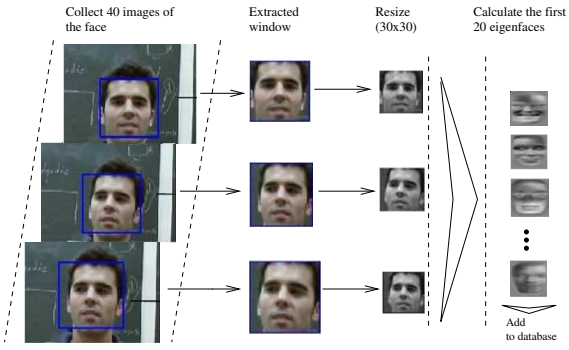

Fig. 5. Learning process

Every face image extracted is converted to grey level and scaled to 30×30 pixels. With this set of 40 grey level 30×30 face images the system

is able to build the eigenspace of the person by calculating his first 20 eigenfaces (PCA). Fig. 5 illustrates the complete learning process of a person. It takes about 15 seconds in a 450 Mhz Pentium II processor.

5.2 Recognition Process

As in the learning process, the first stage of the recognition process is the detection and extraction of faces from the input image. Once this images were extracted they are scaled to 30×30 pixels and projected in the eigenspace of the person the robot is interacting with. From the coefficients of projection the system is able to compute the probability of each detected person being the right one. The probability values are stored in a linked list in descendant order. Using a decision mechanism the system is able to know whether or not the robot is interacting with the right person and in the negative case the robot can recognise, among the people around, the person it should interact with.

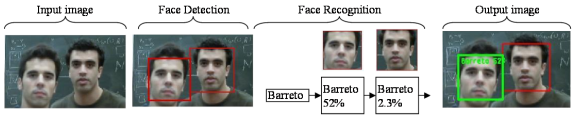

Fig. 6. Recognition process

In practice a very simple framework is used to produce an effective a highly efficient decision mechanism which is described elsewhere (Barreto *et al.*, 2004). This mechanism increases system's performance for the case where two or more people are detected in the image.

6. RESULTS

A 13 stage cascaded classifier was trained to detect frontal upright faces. Each stage was trained to eliminated 50% of the non-face patterns while falsely eliminating only 0.2% of the frontal face patterns. In the optimal case, we can expect a false alarm rate about $0.002^{13} = 8 \cdot 10^{-36}$ and a hit rate about $0.998^{13} = 0.97$ (see Fig. 2).

To train the detector, a set of face and non face training images were used. The face training set consisted of over 4,000 hand labelled faces scaled and aligned to a base resolution of 24×24 pixels. The non-face subwindows used to train the detector come from over 6,000 images which were manually inspected and found to not contain any faces. Each classifier in the cascade was trained with the 4,000 training faces and 6,000 non-face windows using Adaboost.

6.1 Speed of the Final Recognition System

The introduction of the Kalman filter to reduce the search region has demonstrated its value. Actually on a 1GHz PIII laptop, the detection and recognition runs at a 8.6 fps whereas with the Kalman improvement its processing rate depends on the area occupied by the face. Naturally the larger improvements are observed when the user's face occupies the least detectable area on the image. In this case processing speeds of 24 fps are obtained.

6.2 Experiments on Real-World Situations

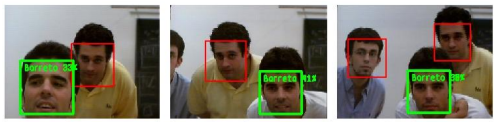

Fig. 7. Three frames from a Real-Time Face Recognition system output sequence.

The system was tested in some real-world situations and Fig. 7 presents a sequence of images captured by the robot's camera and processed by the real-time face recognition system. Figure 8

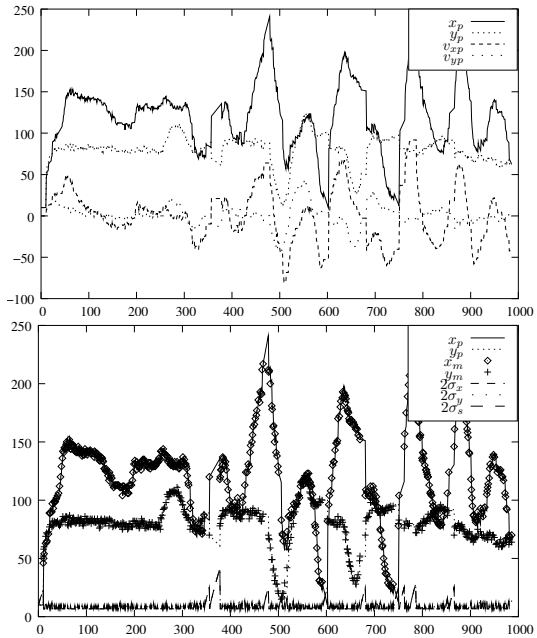

Fig. 8. Top: Estimated position and velocity. Bottom: Estimated position, measured position and prediction covariance

shows an example of the estimated parameters by the Kalman filter that can be compared to the measured ones. Figure 9 shows a sequence of tracking where it is visible that when recognition fails the search area grows, in fact its size is related to the prediction covariance of the filter.

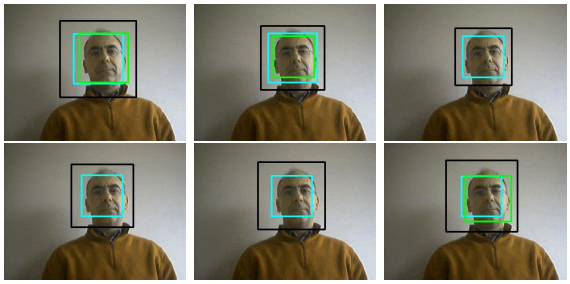

Fig. 9. Tracking Sequence where it is visible the search region (black), predicted face region (cyan) and detected face region (green).

7. CONCLUSIONS

This article gives a contribution for the development of new human-machine interfaces for mobile robots and autonomous systems, based on computer vision techniques. The article presented an approach for real-time face recognition and tracking which can be very useful for human-robot interaction systems. In a human robot interaction environment this system starts with a very fast real-time learning process and then allows the robot to follow the person and to be sure it is always interacting with the right one under a wide range of conditions including: illumination, scale, pose, and camera variation. The face tracking system works as a preprocessing stage to the face recognition system, which allows it to concentrate the face recognition task in a sub-window previously classified as face. This abruptly reduces the computation time. The introduction of a position predictive stage would also reduce the face search area driving to the creation of a robust automatic tracking and real-time recognition system.

This paper also presents a Pre-Learnt User Recognition System which works in almost real-time and that can be used by the robot to create a set of known people that can be recognised anytime. The robot has a certain number of people in the database and once a known face is found it can start following and interacting with it. Of course this system can also be used in security applications since it has the ability of tracking a set of known people.

ACKNOWLEDGMENTS

The authors thank the Portuguese Foundation for Science and Technology (FCT) by the support to accomplish the work presented in this article through the project DIVA and the scholarship for doctoral studies for the author Paulo Menezes.

REFERENCES

Barreto, José, Paulo Menezes and Jorge Dias (2004). Human-robot interaction based on Haar-like features and eigenfaces. In: *International Conference on Robotics and Automation.*

Freund, Yoav and Robert E.Schapire (1996). Experiments with a new boosting algorithm.

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Transactions of ASME - Journal of Basic Engineering* (82 series D), 35–45.

Lienhart, Rainer and Jochen Maydt (2002). An extended set of haar-like features for rapid object detection. In: *IEEE ICIP 2002, Vol. 1, pp 900-903.*

Moghaddam, B. and A.P. Pentland (1995). Probabilist visual learning for object representation. *Technical Report 326, Media Laboratory, Massachusetts Institute of Technology.*

Mohan, Anuj, Constantine Papageorgiou and Tomaso Poggio (2001). Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(4), 349–361.

Oren, M., C.Papageorgiou, P.Sinha, E.Osuna and T.Poggio (1997). Pedestrian detection using wavelet templates.

Rainer Lienhart, Alexander Kuranov and Vadim Pisarevsky (2002). Empirical analysis of detection cascades of boosted classifiers for rapid object detection. *MRL Technical Report, Intel Labs.*

Rowley, Henry A., Shumeet Baluja and Takeo Kanade (1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(1), 23–38.

Schneiderman, H. and T. Kanade (2000). A statistical method for 3D object detection applied to faces and cars. *In International Conference on Computer Vision.*

Sung, K. and T. Poggio (1998). Example-based learning for viewbased face detection. *In IEEE Patt. Anal. Mach. Intell.* **20**(1), 39–51.

Turk, M.A. and A.P. Pentland (1991). Face recognition using eigenfaces. *In Proc. of IEEE Conference on Computer Vision and Pattern Recognition* pp. 586 – 591.

Viola, Paul and Michael Jones (2001). Rapid object detection using boosted cascade of simple features. In: *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition 2001.*

Yang, Ming-Hsuan (2002). Detecting faces images: A survey. *IEEE Transations on Pattern Analysis and Machine Inteligence* **24**(1), 34–58.