Real-Time Face Tracking for Context-Aware Computing

Bogdan Raducanu [1] and Jordi Vitrià
Centre de Visió per Computador
Edifici O - Campus UAB
08193 Bellaterra, Barcelona, Spain

Abstract. The use of context can be a very relevant cue for computer vision-based systems, in order to eliminate a lot of ambiguity and uncertainty, otherwise inherent, in the human-computer interaction. Despite of the fact that its obvious importance is widely acknowledged, the great majority of the systems nowdays still lack of this capability. In this paper, we propose faces as a primary contextual information for person detection and present face tracking as a basic procedure for context-driven focus of attention applications. Our proposal was implemented in a combined system, by integrating a motion-based approach (Particle Filter) and a model-based approach (Ada-Boost).

Keywords. Computer vision, face detection and tracking, context-awareness

1. Introduction

Currently, many computer vision systems are still very limited, in the sense that they have no knowledge about themselves or about the environment where they are supposed to function. The decision of the presence or absence of an object in the image (scene) is taken mainly by thresholding a likelihood function. Thus, if the image is of low resolution or low quality, or the object is partially occluded, these approaches might fail in giving a correct answer and the number of false positives could increase, which finally will result in a wrong interpretation.

In real world, there is a strong relationship between an object and its surroundings. The decision about the presence or absence of an object can be greatly influenced by it, because the presence of different types of objects can be correlated (if you see a table, you can also expect to see a chair) [12]. If we aim to build intelligent systems that might be able to show awareness through perception, then we have to take into account the context. There are many definitionsfor context, but the one, which has been, most accepted is the one given in [3]: "...any information that can be used to characterize the current state of the environment and can be considered relevant for the interaction with the system".

[1] Correspondence to: Bogdan Raducanu, Centre de Visió per Computador, Edifici O - Campus UAB, 08193 Bellaterra, Barcelona, Spain. Tel.: +34 93 581 2440; Fax: +34 93 581 1670; Email: bogdan@cvc.uab.es.

Recently, many efforts have been made in the direction of enhancing the machines with perception ability and developing a certain level of awareness. By perception we refer to the whole class of sensing and pattern recognition techniques that are used to interpret and classify a certain situation. A pivotal aspect of this problem is represented by the human presence detection. From the context-awareness perspective, machine perception can help to disambiguate input, for instance by knowing which person is speaking at a given moment, which display is attended by a user, etc. There is psychological evidence in support of this [2]: the users almost universally are directing their attention towards the object they intend to interact with, before the actual command (speech, gesture) is effectively issued. If user's attention is acknowledged by the system, then the space of possible actions that describe the desired interaction with the system is largely simplified, eliminating a lot ambiguity and making in consequence the system to respond more accurately to user's demands.

Machine perception can be achieved through a variety of technologies: pressure sensors, video-cameras, radio-frequency tags, infra-red- or ultrasound-based transmitters/receivers, fingerprint readers, etc. Each of these technologies can present advantages and disadvantages concerning robustness, complexity and costs [7].

In the current paper we present a perceptual computer vision-based system for person presence detection through face detection and tracking. The tracking is implemented using a particle filter, while the face evidence is confirmed using Haar-like features trained using the Ada-Boost algorithm. The paper is structured as follows: section 2 is dedicated to discuss some aspects of face detection/tracking, sections 3 and 4 recalls briefly the Particle Filter and Ada-Boost, respectively. In section 5 we present and discuss our experimental results, meanwhile in section 6 we draw our conclusion to use in future work.

2. Face-Based Person Detection and Tracking

Human presence detection is a central problem in any human-computer interaction application. Solutions to address this issue depends on the scale of resolution we are interested on. Some applications require the whole body to be present in the image (surveillance), meanwhile others require only parts of it, likehands or faces.

The reason why faces can represent the main contextual information for assessing person presence for machine vision has very solid roots in biological vision. In [5], the authors argue that the new-born children come to the world pre-wired to be attracted by faces. It seems that, in general, they prefere to look at moving stimuli that resemble face-like patterns. In the same reference, it is showed that the humans are able to remember faces easier than other objects when presented in an upright orientation.

From a pattern recognition point of view, faces can be considered as geometric templates with a rough feature configuration, arranged in an identical spatial relationship, but showing slight deformations. For this reason, the results obtained in the study of face detection could be used for the study of other classes of objects that share similar properties, like handwritten characters for instance. An overview of the existing face detection techniques is out of the scope of this paper, but a very recent and comprehensive survey can be found in [15].

Face detection, although considered as a minor problem by many researchers, should be seen always as the primary step for any complex facial information processing system, dedicated to face recognition, facial feature extraction or facial expression recognition. Despite of the fact that for humans face detection is a natural task even in complex scenes, for machine vision systems is still a very challenging problem. Face detection asks for robustness against changes in illumination conditions, changes in scale, rotation and translation transformations, deformations due to pose variations, emotional expressions and natural or artificial partial occlusions (beards, glasses, etc.).

Besides these difficulties, another degree of complexity should be considered when we require temporal consistency. From HCI perspective, machine vision systems are supposed to work in unconstrained environments and thus should be able to perform a continous task.

There are two main approaches for representation and tracking of moving objects: motion-based and model-based. These shouldn't be seen as alternative solutions, but rather complementary ones, because they can increase the robustness of the resulting system [6]. Through their integration, we obtain a system in which the motion-based approach is used for prediction and the model-based approach is used to assess and improve the quality of the prediction. In our implementation, we chose Particle Filter as the motion-based approach and Ada-Boost as the model-based one. In the next two sections we will briefly recall them.

3. The Motion-Based Approach

The tracking problem can be formulated in the framework of partially observable Markov chains [4] that consider the evolution of a state vector sequence x_k over time. This implies the estimation of the *a posteriori* density over the state x_k from all available sensor measurements $z_{0:k} = z_0 ... z_k$. A solution to this problem is given by Bayes filters [9], which compute this *a posteriori* density recursively (we make here a first-order Markov assumption, i.e. the state x_k depends only of x_{k-1}):

$$p\left(x_k|z_{0:k}\right) = K \cdot p\left(z_k|x_k\right) \cdot \int p\left(x_k|x_{k-1}\right) \cdot p\left(x_{k-1}|z_{0:k-1}\right) dx_{k-1} \tag{1}$$

If the a posteriori distribution at any moment is Gaussian and the distributions $p\left(x_k|x_{k-1}\right)$ and $p\left(z_k|x_k\right)$ are linear in its arguments, than the equation (1) reduces to the Kalman filter [14]. If one of these conditions cannot be met, than a possible alternative is to use some approximation methods in order to linearize the actuation and measurements models. There are two extensions of the Kalman filter that can be used in these situations, known as Extended Kalman Filter [11] and Unscented Kalman Filter [10].

However, most of the real-world processes are non-linear and these approximations don't hold. The solution to handle a general case is offered by Particle filter, which has been introduced in [8].

The basic idea of the Particle filter is to maintain a multiple hypothesis (*particles*) about the object being tracked. This means that the distribution $p\left(x_k|z_{0:k}\right)$ is represented using a set $\{s_k^n, w_k^n\}$, where $n = 1..N$ (N is the number of particles). The weights should be normalised such that $\sum\limits_n w_n = 1$. In other words, each particle has associated a weight that reflects its relevance in representing the object. The initial set of particles can be randomly generated or using the output of a detector. The *a posteriori* probability is updated in a recursive way, according to the following steps:

- from the previous set of particles $\left\{s_{k-1}^{n}, w_{k-1}^{n}\right\}_{n=1..N}$, draw one particle through importance sampling, so s_k^m will correspond to some s_{k-1}^{j}:

$$s_k^n \sim w_{k-1}^n \tag{2}$$

- the chosen particle is propagated according to the model's dynamics:

$$s_k^n \sim p\left(x_k | x_{k-1} = s_k^m\right) \tag{3}$$

- assign a new weight to the recent generated particle equal to the likelihood of the observation, i.e.:

$$w_k^n \sim p\left(z_k | x_k = s_k^n\right) \tag{4}$$

then normalize again the resulting weights and store the new obtained set of particles.

Once the N particles have been constructed, an estimate for the current object state can be obtained from the following formula:

$$\xi\left[f\left(x_k\right)\right] = \sum_{n=1}^{N} w_k^n f\left(s_k^n\right) \tag{5}$$

From time to time, it is necessary to resample the particles, otherwise could appear degeneracy problems, which is the concentration of the whole weight on a single particle. The resampling procedure duplicates the particles with higher weights, while discards those with lower weights. In consequence, without resampling the performance of the algorithm will be considerable degraded. A more complete discussion about the degeneracy problem can be found in [1].

In regard with the dynamical model, we chose a very simple one, described by a linear stochastic differential equation:

$$x_k = Ax_{k-1} + Bu_k \tag{6}$$

where A is the state transition matrix, u_k is a vector of standard normal random variates and BB^T is the process noise covariance.

In our problem about face tracking, a particle is represented by a combination of spatial and color information: $s_k = \{px, py, p\sigma, p\Phi\}$. The first three parameters represent the state vector $\{x_k = px, py, p\sigma\}$ where (px, py) is the center of the particle and $p\sigma$ - the size of the particle. The last term $p\Phi$ is the iconic representation of the image '*under*' the particle.

The tracker is initialized using the output of a face detector that has been implemented using the Ada-Boost algorithm (see next section). The likelihood of each particle with respect to the output of the face detector (i.e., particles' weight) $F = \{fx, fy, f\sigma, f\Phi\}$ is given by:

$$w^n = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{d_n^2}{2\sigma^2}} \tag{7}$$

where d_n^2 represents the combination between difference in term of position, size and image similarity:

$$d_n^2 = (px - fx)^2 + (py - fy)^2 + (p\sigma - f\sigma)^2 + \|p\Phi - f\Phi\|_{L2}^2 \tag{8}$$

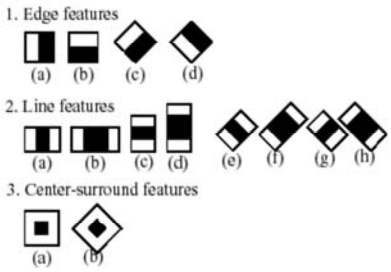
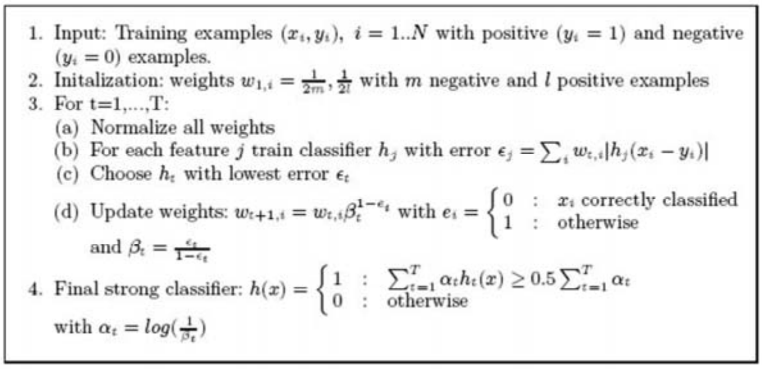


Figure 1. Haar-like wavelet features used to train the weak classifiers.



```
1. Input: Training examples $(x_i, y_i)$, $i = 1..N$ with positive $(y_i = 1)$ and negative
   $(y_i = 0)$ examples.
2. Initalization: weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ with $m$ negative and $l$ positive examples
3. For t=1,...,T:
   (a) Normalize all weights
   (b) For each feature $j$ train classifier $h_j$ with error $\epsilon_j = \sum_i w_{t,i} |h_j(x_i - y_i)|$
   (c) Choose $h_t$ with lowest error $\epsilon_t$
   (d) Update weights: $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$ with $e_i = \begin{cases} 0 & : \; x_i \text{ correctly classified} \\ 1 & : \; \text{otherwise} \end{cases}$
       and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$
4. Final strong classifier: $h(x) = \begin{cases} 1 & : \; \sum_{t=1}^{T} \alpha_t h_t(x) \geq 0.5 \sum_{t=1}^{T} \alpha_t \\ 0 & : \; \text{otherwise} \end{cases}$
   with $\alpha_t = log(\frac{1}{\beta_t})$
```

Figure 2. The Ada-Boost algorithm.

4. The Model-Based Approach

Ideally, techniques for face detection are desired to show robustness against changes in illumination, scale and head poses. They should be able to detect faces despite variation in facial expression and the presence or absence of natural or artificial accessories like beards and glasses. The face detection techniques can be divided in two main classes: holistic and local ones. The weakness of the holistic approaches is that they represent the images as raster vectors without any information about the 2D topology of facial features. In change, face detectors based on local information have the advantage of providing more accurate information about the presence of face parts, by classifying image patches in ‘face’ or ‘non-face’ vectors.

Following this second path, we implemented a boosted-based face detection algorithm similar to those proposed in [13]. The idea is to use a cascade of weak classifiers in order to produce a stronger one. Each classifier is trained with a few hundreds samples of a particular object (a face), using Haar-like wavelet features, depicted in figure 1.

After each training epoch, the wrong classified features are retrained. Each classifier outputs “1” if the corresponding sample corresponds to a face and “0” otherwise. The final strong classifier is obtained as a linear combination of weak classifiers, followed by a threshold.

The classifier is designed in such a way that allows detection of faces at different scales. This is a more convenient technique than downsampling the image itself. A very

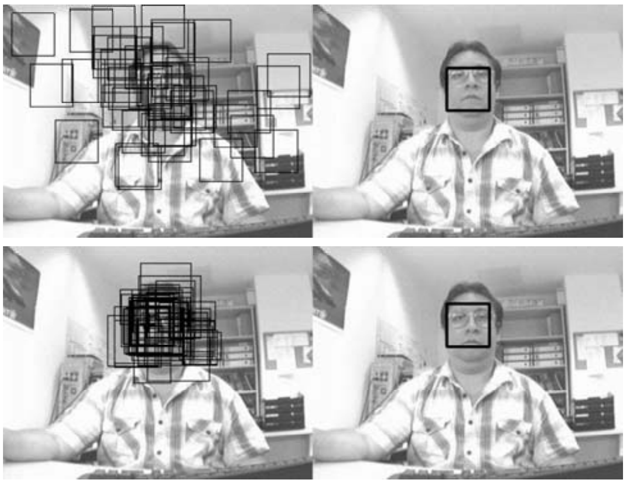


Figure 3. The behavior of our tracker. The top image represents the initialization of the particles, the bottom image shows the convergence after a few frames

popular algorithm to train the weak classifiers in order to obtain the strong one is represented by Ada-Boost [13]. The algorithm is summarized in figure 2.

5. Experimental Results

We tested our system on several video sequences. Figure 3 shows the initialization of the tracker, using as reference the output from the face detector, and the future evolution of the algorithm until it convergences. Although we tested our tracker with a number of particles varying from 200 to 800, in figure 3 we depicted, for simplicity, only 50 most relevant ones.

The purpose of our tests was to illustrate the robustness of our approach against changes in scale, small pan/tilt rotations of the head and occlusions (see figures 4 and 5). In our experiments, the distance of the face to the camera is varying between 40 cm and 2 meters. The limitation of distance to 2 meters was imposed because of the wide-angle lens mounted on the camera. Beyond this distance, the object's projection on the image plane will become to small for the detector.

In order to deal with occlusions, the tracker is 'clamped' on the last detected face position. Once the face becomes again visible, the tracking is resumed (see figure 5). The tracker maintains the current position for a few frames. If, after this timeframe, the answer from detector is still negative, then the particle filter shuts down and reinitialiaze when the next face detection is reported.

For the current implementation of the tracker, we treated also as occlusion situations in which the face detector fails due to some unexpected head movements or poses. In the future, we want to modify this behavior, in order to introduce an inertia factor, i.e. the tracker will remain 'active' and follows the 'expected' trajectory of the face despite the absence of the object in the scene. This makes sense because for the close-range tracking (aimed for an 'eye-to-eye' interaction), the most frequent case when the detector fails is in certain head poses.

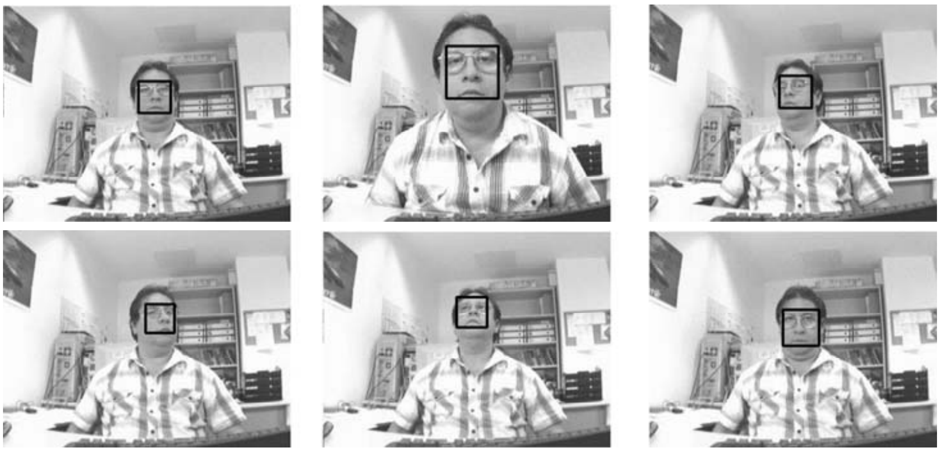
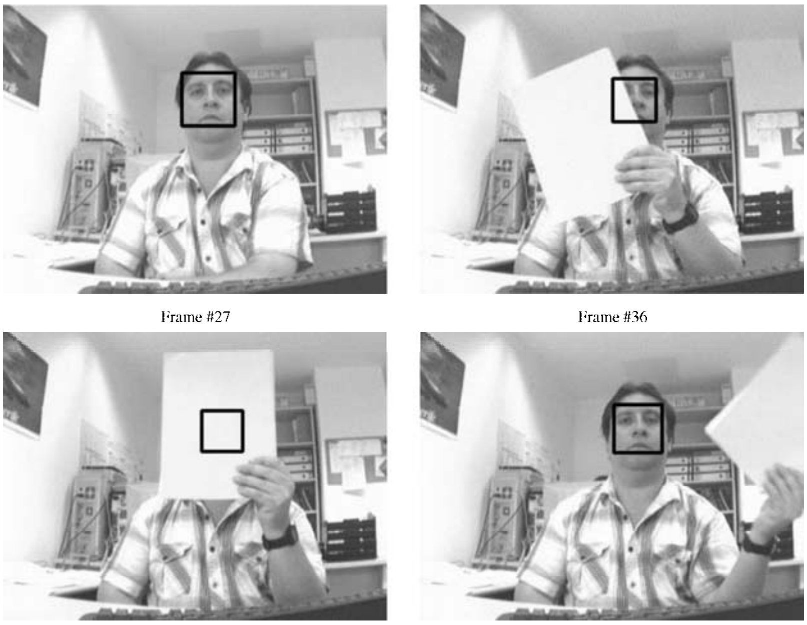


Figure 4. Some instances from a video sequence to illustrate the robutness of our tracker against some factors like scale and head pose.



Frame #27

Frame #36

Frame #41

Frame #49

Figure 5. Some instances from a video sequence in which we wanted to illustrate the robustness of our tracker against occlusions.

Both the detector and tracker are able to perform with multiple persons. The reason why we showed in our paper results involving only one person is motivated by the purpose of our system. If we see the things from the context-aware computing perspective, then the physical constraints of the system restrict it to perform interaction only with one person at a time. This is the same as having a multiple person detector/tracker, but the system focuses at the user closest to the camera.

6. Conclusions and Further Work

In this paper we presented a vision-based perceptive sytem, that is aimed to become aware at the human presence in the scene. The human presence is claimed by the detection of a face in the image. The system proposed consists of an integrated architecture, combining a motion-based approach and a model-based approach. The results presented in this paper show the robustness of our system against several factors like scale, small face rotations and occlussions.

Even at an early stage of our work, we look forward to give our system a real utility. Domains like Aware Environments and social robotics are only a few example where potential applications can be developped and tested.

7. Acknowledgements

Bogdan Raducanu is supported by the Ramon y Cajal research program, funded by the Ministry of Education and Science of the Spanish Government.

References

[1] M.S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, *IEEE Transactions on Signal Processing*, **50** (2002), 174–188.

[2] B. Brumitt and J.J. Cadiz, Let there be light: comparing interfaces for homes of the future, *Proceedings of Interact'01*, Japan (2001), 375-382.

[3] A.K. Dey, Understanding and using context, *Personal and Ubiquitous Computing Journal*, **5** (2001), 4-7.

[4] A. Doucet, J.F.D. de Freitas, and N.J. Gordons (eds.), *Sequential Monte Carlo methods in practice*, Springer, 2001.

[5] M.A. Fischler and R.A. Elschlager, The representation and matching of pictorial strucures, *IEEE Transactions on Computers*, **COM-22** (1973), 67-92.

[6] S. Gil, R. Milanese and T. Pun, Combining multiple motion estimates for vehicle tracking, *Proceedings of European Conference on Computer Vision*, United Kingdom, 1996, 307-320.

[7] E. Hoffman and J. Scott, Location of mobile devices using networked surfaces, *Proceedings of Ubicomp 2002*, Sweden, 2002, 281-298.

[8] M. Isard and A. Blake, CONDENSATION - Conditional Density Propagation for visual tracking, *International Journal of Computer Vision*, **29** (1998), 5-28.

[9] A.M. Jazwinsky, *Stochastic processes and filtering theory*, Academic Press, 1970.

[10] S.J. Julier and J.K. Uhlmann, A new extension of the Kalman filter to nonlinear systems, *Proceedings of SPIE on Signal Processing, Sensor Fusion and Target Recognition VI*, **3068** (1997), 182-193.

[11] P. Maybeck, *Stochastic models, estimation and control (I)*, Academic Press, 1979.

[12] A. Torralba, Contextual priming for object detection, *International Journal of Computer Vision*, **53** (2003), 169-191.

[13] P. Viola and M.J. Jones, Robust real-time face detection, *International Journal of Computer Vision*, **57** (2004), 137-154

[14] G. Welch and G. Bishop, An introduction to Kalman filter, *Tech. Report TR95-041*, Dept. of Comp. Science, Univ. of North Carolina, 2002

[15] M.-H. Yang, D. Kriegman and N. Ahuja, Detecting faces in images: a survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24** (2002), 34-58