



# Real-time face tracking for attention aware adaptive games

Matthieu Perreira Da Silva, Vincent Courboulay, Armelle Prigent, and Pascal Estraillier

Université de La Rochelle,  
Laboratoire Informatique Image Interaction,  
17071 La Rochelle cedex 9, France.  
mperreir@univ-lr.fr

**Abstract.** This paper presents a real time face tracking and head pose estimation system which is included in an attention aware game framework. This fast tracking system enables the detection of the player's attentional state using a simple attention model. This state is then used to adapt the game unfolding in order to enhance user's experience (in the case of adventure game) and improve the game attentional attractiveness (in the case of pedagogical game).

## 1 Introduction

Usually, computer software has a very low understanding of what the user is actually doing in front of the screen. It can only collect information from its mouse and keyboard inputs. It has no idea of whether the user is focused on what it is displaying, it doesn't even know if the user is in front of the screen. One of the first steps to making computer software more *context aware* is to make it *attention aware*. In [1] attention aware systems are defined as

Systems capable of supporting human attentional processes.

From a functional point of view, it means that these systems are thought as being able to provide information to the user according to his estimated attentional state. This definition shares some objectives with the definition of adaptive systems [2] which are system that are able to adapt their *behaviour* according to the *context*. An attention aware system must be an adaptive system.

According to [1], one of the best ways to estimate user's attention is using gaze direction. Consequently, building good adaptive attention aware system requires estimating reliably user's gaze, but in an unconstrained environment this is a very difficult task. To overcome this problem we propose, in a first step, to use head pose as an estimator for gaze direction. Head pose provides less accurate but more robust gaze estimation.

In this paper, we present a real time vision based face tracking and head pose estimation system coupled with a simple inattention model that will allow an adaptive game to detect attentional shifts and adapt its unfolding accordingly. This model is implemented in two different types of systems (see 1):

- A pedagogical game in which information about the user's attention is used to adapt the game unfolding in order to refocus the player's attention. It is used as a tool for pedo-psychiatrists working with children with autism in the pedo-psychiatric hospital of La Rochelle in order to improve children's attention.
- An adventure game in which user's attentional state helps modifying the game scenario in order to make the game more immersive and fun to play.

In the next section we introduce the attention aware adaptive game framework which is using our head pose based gaze estimation algorithm.



**Fig. 1.** Left: A screen capture of the adventure game prototype. Right: a screen capture of the pedagogical game together with a preview of the head tracking system.

## 2 An attention aware adaptive game framework

Adaptive game is a subcategory of adaptive systems that can take good advantage of being attention aware. They are designed to react and adapt their unfolding according to the players (explicit) actions. Thus, being able to *see* if the player is facing the screen and is attentive allows the game to adapt its actions to the situation.

### 2.1 Attention

Attention is historically defined as follows [3]:

Everyone knows what attention is. It is the taking possession by the mind in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought...It implies withdrawal from some things in order to deal effectively with others.

Thus, attention is the cognitive process of selectively concentrating on one thing while ignoring other things. In spite of this single definition, it exists several types of attention [4]: awakening, selective attention, maintained attention, shared attention, internal or external absent-mindedness and vigilance. For an interactive

task, we are mainly interested in *selective and maintained attention*. The analysis of the first one allows knowing whether people are involved in the activity. The second one enables us to assess the success of the application.

It has been proven that the same functional brain area were activated for attention and eye movements [5]. Consequently, the best attention marker we can measure is undoubtedly eyes and gaze behaviour. A major indicator concerning another type of attention, *vigilance*, named PERCLOS [6] is also using such markers. In continuity of such studies, we based our markers on gaze behaviour, approximated by head pose, to determine selective and maintained attention. A weak hypothesis is that a person involved in an interesting task focuses his/her eyes on the salient aspect of the application (screen, avatar, car, enemy, text...) and directs his/her face to the output device of the interactive application (screen). Nevertheless, if a person does not watch the screen, it does not necessarily mean that he/she is inattentive; he/she can be speaking with someone else about the content of the screen [7].

In order to treat these different aspects, we have decided to adopt the following solutions: if the user does not watch the screen during a time  $t$ , we conclude to inattention. In the following subsection, we present how  $t$  is determined. If inattention is detected, we inform the application.

**A simple model of human inattention.** The goal of this model is to define the delay after which the application tries to refocus the user on the activity. Actually, we easily understand that in this case, an interactive application does not have to *react* the same way if people play chess, role player game or a car race. Until now, this aspect of the game was only directed by the time during which nothing was done on the paddle or the keyboard.

We based our model of what could be named *inattention* on two parameters:

1. the type of application;
2. the time spent using the application.

The last parameter depends itself on two factors:

1. a natural tiredness after a long time
2. a disinterest more frequent during the very first moments spent using the application than once attention is focused, this time corresponds to the delay of *immersion*.

Once the parameters are defined, we propose the following model in order to define the time after which the application try to refocus the player who does not look at the screen.

*Potential of attention.* As we mentioned, potential of attention depends mainly on two parameters, tiredness and involvement. We have decided to model arousal (the opposite of tiredness), or potential of attention, by a sigmoid curve parameterized by a couple of real number  $\beta_1$  and  $\beta_2$ .  $\beta_2$  represents the delay after

which the first signs of fatigue will appear and  $\beta_1$  is correlated to the speed of apparition of tiredness (Figure 2).

$$P_{arousal} = \frac{\exp^{-\beta_1 t + \beta_2}}{1 + \exp^{-\beta_1 t + \beta_2}}, \quad (1)$$

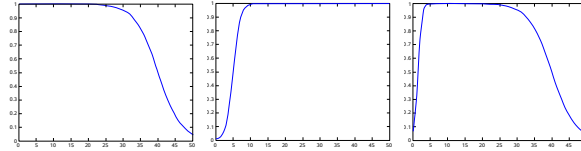
where  $\beta_1$  and  $\beta_2$  are two real parameters.

For the second parameter, we have once again modelled involvement, or interest probability, by a sigmoid. We started from the fact that activity is *a priori* fairly interesting, but if the person is involved after a certain time ruled by  $\alpha_2$ , we can consider that interest is appearing at a speed correlated to  $\alpha_1$  (Figure 2).

$$P_{interest} = \frac{1}{1 + \exp^{-\alpha_1 t + \alpha_2}}. \quad (2)$$

For our global model of potential of attention, we couple both previous models in order to obtain:

$$P_{attention} = P_{interest} * P_{arousal}, \quad (3)$$



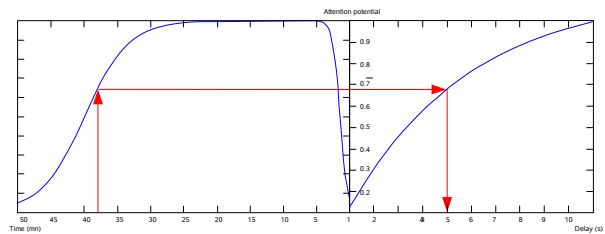
**Fig. 2.** Left: model of tiredness evolution. Middle: model of interest probability. Right: model of potential of attention. Abscissa represents time in minutes. Each curve is designed for a single person involved in a unique activity. ( $\alpha_1 = 1$ ,  $\alpha_2 = 3$ ,  $\beta_1 = 0.3$ ,  $\beta_2 = 12$ )

*Delay of inattention.* Once this model is defined, we are able to determine the time after which the software has to react (if the person still does not look at the screen). Here, it is an arbitrary threshold  $\gamma$  guided by experience, which characterizes each application. The more the application requires attention, the higher this coefficient is. The model we have adopted is an exponential function.

$$D_{game} = \exp^{\gamma(t) * P_{attention}}, \quad (4)$$

$\gamma$  is a function of time because we have estimated that it can exist several *tempo* in an application (intensive, stress, reflection, action ...). As a conclusion, we can summarize our model of inattention by the two following steps (Figure 3):

- depending on the time elapsed from the beginning of the application, we estimate the potential of attention  $P_{attention}(t)$ ;



**Fig. 3.** Curves used to determine the delay of software interaction.

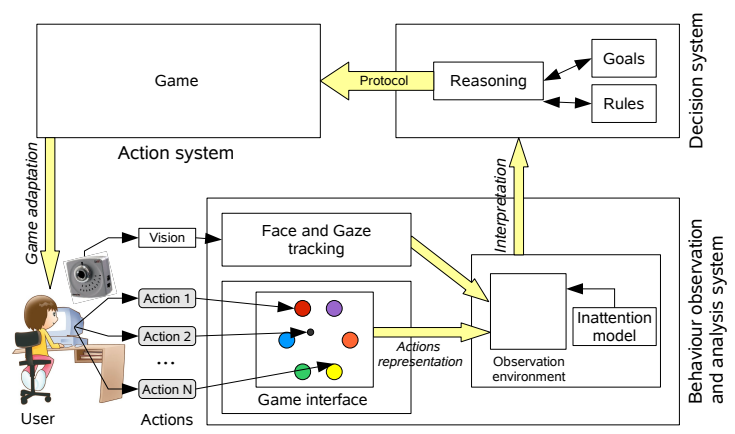
- depending on this potential and the application, we estimate the delay  $D_{\gamma(t)}(P_{attention}(t))$  after which the software has to refocus the inattentive player.

Please note that this model was validated on only five children and still need to be validated on more people. Nevertheless, the first tests are quite promising.

### 2.2 An adaptive framework

As we already mentioned, our objective is to implement a system that interacts in a smart way with the person using the application. It consists in establishing a multi-mode and multimedia dialogue between the person and the system.

Consequently, the conception of the platform, represented Figure 4, was guided



**Fig. 4.** General architecture.

by the following constraints:

- it must track gaze behaviour, determined as the best markers of attention;

- it must allow to focus user attention and be able to refocus it if necessary;
- it must not perturb the user by inopportune interaction.

The system can be divided into three sub-parts:

- *the system of observation and behaviour analysis*: monitors the player's attention;
- *the decision system*: adapts the execution of games;
- *the action system*: runs the game.

This architecture needs a fast and robust gaze estimation system in order to respond to its environment in an adaptive and dynamic manner. To achieve this goal, we have built a fast tracking system dedicated to adaptive systems. This system is described in the following section.

### 3 Face tracking for adaptive games

#### 3.1 Constraints

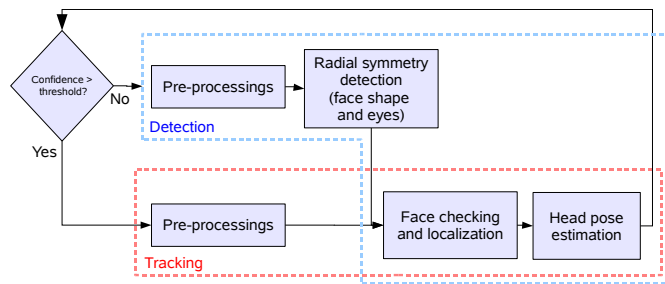
As the system is designed to be used by a wide range of applications and users (from educational games for children with autism to adventure games for "common gamers"), some constraints have emerged:

- non invasive material;
- low cost;
- single user;
- recordable information;
- standard computer;
- unconstrained environment.

Our system is based on a low cost iee-1394 camera connected to a standard computer. Despite its low cost, this camera captures video frames of size 640x480 at 30 frames per second which are suitable characteristics for both accurate face features localization and efficient face features tracking. The choice of a grayscale camera instead of a more common colour camera is driven by the fact that most of the aimed applications are performed in an indoor environment. In such an environment, the amount of light available is often quite low, as grayscale cameras usually have more sensitivity and have a better image quality (as they don't use Bayer filters), they are the best choice. Another advantage of grayscale cameras is that they can be used with infra-red light and optics that don't have infra-red coating in order to improve the tracking performance by the use of a non invasive more frontal and uniform lightning.

#### 3.2 Architecture

The tracking algorithm we have developed is built upon four modules which interoperate together in order to provide a fast and robust face tracking system (see Figure 5). The algorithm contains two branches: one for face detection and a second for face tracking. At run time, the choice between the two branches is made according to a confidence threshold, evaluated in the *face checking and localization* module.



**Fig. 5.** Architecture of the face tracking and pose estimation system.

*Pre-processing.* Before face or radial symmetry detection, the input image must be pre-processed in order to improve face and face feature detection performance. The main pre-processing steps are image rescaling and lightning correction, also called contrast normalization. The input image is rescaled so that low resolution data is available for radial symmetry detection algorithms. Contrast normalisation consists in adapting each pixel intensity according to the local mean intensity of its surrounding pixel. As this task is performed by humans retina, several complex models have been developed to mimic this processing [8]. As our system needs to be real-time, we have chosen to approximate this retinal processing by a very simple model which consists in the following steps:

1. For each image pixel, build a weighted mean  $M_{x,y}$  of its surrounding pixels. In our implementation we used first order integral filtering (Bartlett Filter) in order to achieve fast filtering. Note that first order integral filters are an extension of the commonly used zero order integral filters (box filters). For more information about generalized integral images see [9]
2. Calculate the normalized pixel intensity:

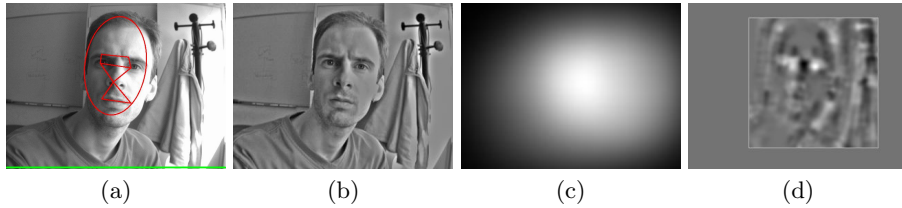
$$I_{x,y} = \frac{S_{x,y}}{(M_{x,y} + A)}$$

with  $S$  the source image,  $I$  the normalized image, and  $A$  a normalization factor.

Figure 6 shows the result of our simple contrast normalization algorithm on a side lit scene.

*Radial symmetry detection for face shape and eye detection.* Once the image is pre-processed, we use a set of radial symmetry detector in order to localize a face region candidate that will be further checked by the *face checking and localization* module. Once again our real-time constraint guided the choice of the algorithms we used.

Face ovoid shape is detected in low resolution version of the input image (typically 160x120) using an optimized version of the Hough transform (Fig. 6.c)



**Fig. 6.** a) Tracking result of a side lit scene. b) Source image after lightning correction. c) Result of face ovoid detection (Hough Transform). d) Result of eyes detection (Loy and Zelinsky Transform).

whereas eyes are detected using an optimized version of the Loy and Zelinsky transform [10](Fig. 6.d). In order to speed up both transforms, the following improvements have been made: only pixels with a gradient magnitude above a pre-defined threshold are processed, the algorithms vote in only one accumulator for all radius and accumulators are smoothed only once at the end of the processing. Using these two symmetry map and a set of face geometry based rules we define the face candidate area.

*Face checking and localization.* This module serves two purposes:

- When called from the face detection branch, it checks if the face candidate area really contains a face and outputs the precise localization of this face.
- In the case of face tracking, it only finds the new position of the face.

In both cases, the module outputs a confidence value which reflects the similarity between the interface face model and the real face image.

Face checking and localization is based on the segmentation of the face candidate image into several blobs. The source frame is first filtered by a DoG filter; the result image is then adaptively thresholded. The resulting connected components (blobs) are then matched on a simple 2D face model in order to check and localize the face.

*Head pose estimation.* Similarly to previous research we use triangle geometry for modelling a generic face model. For example, [11] resolves equations derived from triangle geometry to compute head pose. Since we favour speed against accuracy, we use a faster and simpler direct approximation method based on side length ratios.

### 3.3 Performance and robustness

*Processing time.* We measured the processing time of the algorithm on a laptop PC equipped with a 1,83GHz *Intel Core Duo* processor. We obtained the following mean processing times:

- Face detection: 30 milliseconds (first detection or detection after tracking failure).

- Face tracking: 16 milliseconds

Processing time include all processing steps, from pre-processing to head pose estimation. Since image capture is done at 30fps and the algorithm is using only one of the two processor cores, our system uses 50% of processor time for face detection and 25% of processor time for face tracking. Consequently, the algorithm is fast enough to enable running a game in parallel on a standard middle-end computer.

*Robustness.* As can be seen from the tracking example shown on figure 7, the algorithm can handle a broad range of head orientation and distance. The contrast normalization step also allows the algorithm to run under different illumination conditions. However, this algorithm is designed to be fast, as a consequence tracking performances still need to be improved under some lightning conditions (backlit scene, directional lighting casting hard shadows, etc.). A future version of the algorithm may use a colour camera and skin colour detection algorithms (as in [12] or [13]) to improve face detection and tracking robustness. This modification would however prevent us from using infra-red light to improve the algorithm performances under poor lightning conditions. Another possibility would be to add an infra-red lightning and adapt the current algorithm to this new lightning in order improve the robustness of the system.



**Fig. 7.** Left: Result of the tracking algorithm for different face distances and orientation.

## 4 Discussion and future work

We described a complete framework for attention aware adaptive games. This framework uses a fast face tracking and head pose estimation algorithm coupled with a simple model of human inattention in order to generate feedback information about the attentional state of the player. This system is currently implemented in an adventure and a pedagogical game.

In order to improve this framework, we are currently working at building a more complex and realistic attentional model which would not only react to attentional shifts but could also predict them. We are also working at using colour information to enhance the accuracy and robustness of our face tracking system as it can provide performance improvements without impacting too much processing time.

## 5 Acknowledgement

This research is partly funded by the French Poitou-Charentes county and the Orange Foundation.

The authors would also like to thank doctor Mr. D. Lambert Head of Department of Child Psychiatry of La Rochelle hospital (France) and his team, in particular: Mr. V. Gabet for their useful advices regarding the rehabilitation methods dedicated to children with autism.

## References

1. Roda, C., Thomas, J.: Attention Aware Systems: Theories, Applications, and Research Agenda. Volume 22 of Computers in Human Behavior. Elsevier (2006)
2. Weibelzahl, S.: Evaluation of Adaptive Systems. PhD thesis, University of Education, Freiburg (October 2002)
3. James, J.: The Principles of Psychology. (1890, (1983))
4. Ould Mohamed, A., Courboulay, V., Sehaba, K., Menard, M.: Attention analysis in interactive software for children with autism. In: Assets '06: Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility, New York, NY, USA, ACM Press (2006) 133–140
5. Corbetta, M., Akbudak, E., Conturo, T., Snyder, A., Ollinger, J., Drury, H., Linenweber, M., Petersen, S., Raichle, M., Van Essen, D., Shulman, G.: A common network of functional areas for attention and eye movements. *Neuron* **21**(4) (1998) 761–773
6. Dinges, D., Mallis, M., Maislin, G., Powell, J.: Evaluation of techniques for ocular measurement as an index of fatigue and the basis for alertness management. Technical Report Report No. DOT HS 808 762, Final report for the USDOT, National Highway Traffic Safety Administration(NHTSA) (1998)
7. Kaplan, F., Hafner, V.: The challenges of joint attention. *Interaction Studies* **7**(2) (2006) 129–134
8. Beaudot, W.: The neural information in the vertebrate retina: a melting pot of ideas for artificial vision. PhD thesis, TIRF Laboratory, Grenoble, France (1994)
9. Derpanis, K., Leung, E., Sizintsev, M.: Fast scale-space feature representation by generalized integral images. Technical Report CSE-2007-01, York University (January 2007)
10. Loy, G., Zelinsky, A.: Fast radial symmetry for detecting points of interest. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(8) (2003) 959–973
11. Kaminski, J., Teicher, M., Knaan, D., Shavit, A.: Head orientation and gaze detection from a single image. In: International Conference Of Computer Vision Theory And Applications. (2006)
12. Schwerdt, K., Crowley, J.L.: Robust face tracking using color. In: FG '00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000, Washington, DC, USA, IEEE Computer Society (2000) 90
13. Ségurier, R.: A very fast adaptive face detection system. In: International Conference on Visualization, Imaging, and Image Processing (VIIP). (2004)