Computing and Informatics, Vol. 33, 2014, 735–756

DISTRIBUTED REAL-TIME COMPUTATION OF THE POINT OF GAZE

Antonio García Dopico*, José L. Pedraza, M. Luisa Córdoba
Francisco M. Sánchez, Antonio Pérez

Departamento de Arquitectura y Tecnología de Sistemas Informáticos
Facultad de Informática, Universidad Politécnica de Madrid
Campus de Montegancedo s/n, E-28660, Boadilla del Monte, Madrid, Spain
e-mail: {dopico, pedraza, mcordoba, fsanchez, aperez}@fi.upm.es

Abstract. This paper presents a minimally intrusive real-time gaze-tracking prototype to be used in several scenarios, including a laboratory stall and an in-vehicle system. The system requires specific infrared illumination to allow it to work with variable light conditions. However, it is minimally intrusive due to the use of a carefully configured switched infrared LED array. Although the perceived level of illumination generated by this array is high, it is achieved using low-emision infrared light beams. Accuracy is achieved through a precise estimate of the center of the user's pupil. To overcome inherent time restrictions while using low-cost processors, its main image-processing algorithm has been distributed over four main computing tasks. This structure not only enables good performance, but also simplifies the task of experimenting with alternative computationally-complex algorithms and with alternative tracking models based on locating both user eyes and several cameras to improve user mobility.

Keywords: Eye-tracking, cluster computing, real time distributed applications, parallel image processing, MPI, embedded systems

1 INTRODUCTION

Automatic eye gaze direction tracking by means of non-intrusive methods is considered a major interest research area since it started to become feasible in the

* Corresponding author

nineties [1, 2]. Gaze tracking is being proposed as the supporting technology for real applications in multiple fields, ranging from monitoring systems for analyzing human behavior [3] to different kinds of human-computer interfaces, both for conventional computers [4], and for helping persons with motor disabilities [5]. With current commercial of-the-shelf computers, gaze tracking can be implemented in a cost effective manner.

In the past years and due to real-time constraints and the computational load inherent to tracking the considerable freedom and speed of movement of the human eye, the need for developing parallel systems allowing the execution of real-time image processing algorithms arose [5, 6, 7]. On the other hand, parallel systems are easier to configure, build and program than ever, and only require very low power as well as being sufficiently small to be incorporated into a conventional vehicle dashboard, or just below a TFT monitor in front of a computer user. These observations led the autors to work on the design and development of a parallel low-power and minimally invasive real-time eye gaze tracking system, which is the main objective of the system described in this paper. The system is based on performing a real-time analysis of images taken by conventional video cameras and real-time control of a pan-tilt unit to track the user's head movements.

There are multiple potential uses for the proposed system, the first being a laboratory stall for performing experiments on point of gaze fixations, and including its use as the basis for an in-vehicle system for analyzing car drivers' behavior under different stress conditions. The laboratory stall can be used as an experimental prototype for building gaze-based interfaces or for other similar functions. The authors are also considering other applications such as those described in [8, 9], e.g. guiding a TV camera based on the point of gaze of the operator while watching a football or tennis match.

In this paper we describe a minimally intrusive or remote [6] low cost eye tracking system aimed at solving certain problems not often described in the literature, mainly:

- Ensuring the trial subject is not exposed to unnecesary levels of infrared light. Potential harm could be caused by infrared eye trackers, due to excesive exposure to non-visible light. Some proposals use natural light to avoid this problem. However, their use is restricted to stable light conditions. The proposed eye-tracker is aimed at working inside a conventional vehicle where highly variable light conditions are usual.
- Allowing the system to be scalable to be supported by low-cost parallel hardware. Although one of the components used in the initial version, a highly luminous pupil camera, cannot strictly be considered a low-cost device, it was used to allow working with extremely low-power infrared pulses.

The rest of this paper presents a brief overview of other eye tracker proposals, followed by a section giving a general overview of the proposed system. This is followed by three sections about the system architecture and design considerations for process decomposition and temporal requirements. The next section looks at implementation characteristics and the integration of the system. Finally there is a section devoted to explaining results obtained from field trials, including a brief introduction to the basis of the calibration process.

2 STATE OF THE ART

Eye tracking systems are based on mathematical models that connect eye images being captured in real time with the gaze direction at which the subject is looking at the corresponding instant [10, 11]. Eye trackers can be based on natural light conditions [12] so they are actually non-intrusive systems. However, these systems do not behave well under variable light conditions. The usual alternative to solve these problems is the use of infrared illumination. Infrared eye trackers are usually considered non-intrusive. However, it is more precise to classify them as low or minimally intrusive systems, because of the direct infrared light projection on the trial subject's eyes. In this sense, a measure of the intrusiveness factor is the infrared light energy reaching the trial subject's eyes.

Several minimally intrusive eye trackers have been described in the literature, based on the use of single or dual cameras and developed as academic or commercial systems. For example, [13] and [14] describe very low-cost systems with the infrared emitter integrated in the camera or located very close to it. None of these systems take into account the infrared power emitted to the person's eyes.

As regards algorithms, some proposals are analyzed in [15] and [16]. A comparison of five pupil center detection algorithms, including a preliminary version of the present proposal [17], is shown in [18]. Challenging lighting conditions are considered in [13], where a cascaded classifier for dark and bright pupil images is presented.

Finally, two of the more thorough descriptions of eye gaze tracking experimentation systems are found in [19] and [20]. In both cases, infrared illuminators are placed in the four corners of a rectangle facing the user. The paper by Yoo et al. [19] includes a complete study of the theoretical foundations and describes a system whose setup is somewhat similar to the prototype shown in the present authors' previous work [17]. However, it uses only a single LED for each corner while in the work described here it was decided to use a switched LED array in order to take advantage of the fact that although the perceived intensity generated by this array is higher, it is in fact achieved using low emission infrared light beams. The other in-depth description by Noureddin et al. [20] includes a simple but interesting observation relating to the three tasks necessary to support a complete video-oculographic system: camera orientation control to compensate for natural head movements, detection of the features of each image for parameter extraction, and mapping the derived parameters to a problem space.

3 GENERAL OVERVIEW

The proposed gaze tracking system is designed for use inside a conventional car to track the eye gaze of the driver. The system consists of the following main components, which are briefly described below: two cameras with their corresponding frame-grabbers, four infra-red illuminators, a pan-tilt unit, and a four-PC cluster.

The main camera captures an image of the user eye from which the point of gaze is obtained. In order to improve point of gaze accuracy, a narrow field of view (NFV) camera pointing directly at one of the user's eyes is used. This camera has been mounted on a pan-tilt unit conveniently driven to track the eye for compensating head movements. When the eye leaves the field of view of the camera due to a sudden head movement, a tracking error is produced. In order to quickly recover from this error, a second wide-field-of-view (WFV) camera is used. The WFV camera gathers images including the face of the user. Using the WFV camera images, the eye is located and the NFV camera is repositioned to center the eye in its field of view. Two main approaches are reported to be used in face identification algorithms: local feature-based methods and global methods [21]. Some authors detail how to locate precisely both user eyes from a face image using local methods [22, 23] and establish it as a crucial step for many applications [24, 25]. However, the approach taken while designing the system described in this paper does not require such a high precission but instead it requires high speed. The method used in our prototype for locating the eyes only acts as a helping-aid to the estimation of the eye gaze direction and is commonly classified as a local feature-based method in face based identification algorithms.

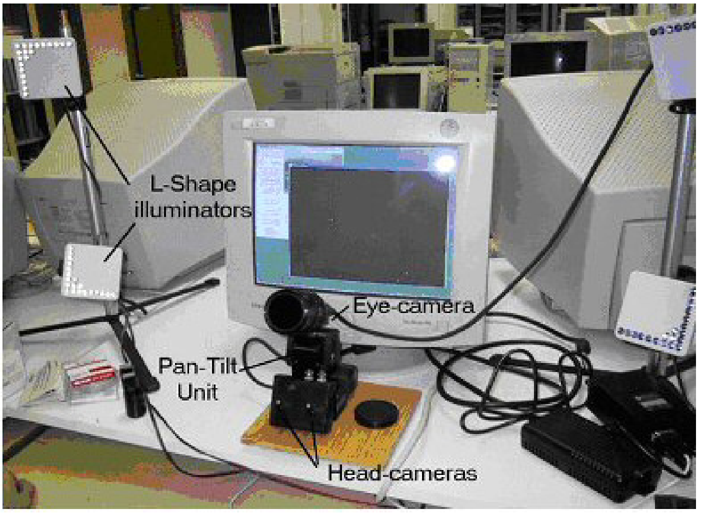


Figure 1. Prototype installation

Figure 1 shows a general view of the system prototype including the four illuminators, the NFV video camera, and the WFV camera.

The NFV camera is an infrared sensitive 1/2" Sony CCD-IR XC-EI50 with a 75 mm lens and a 2X focal extender, mounted on a Direct Perception PTU-46-17.5 pan-tilt unit. Tracking is controlled from one of the PCs via RS-232. To have sufficient depth of field to maintain focus with small movements of the head the F-stop has to be set over 5.6. Due to this restriction, together with the long focal length (150 mm) a very sensitive sensor is needed, so a camera with a CCD sensor (instead of CMOS) has been chosen.

The WFV camera, used to obtain the image of the face, is a low cost 1/3" board camera (S130312), also infrared sensitive, with automatic electronic shutter.

A configuration with four L-shaped infrared illuminators has been defined for the system to obtain an eye image containing at least one valid glint. The specific shape of the illuminators facilitates glint identification. The iris center and radius have to be determined in order to reject the glints located near its border due to distortion. Figure 2 shows two images with glint distortion.

In order to minimize the influence of external illumination variations, which are unavoidable in a conventional driving session, the NFV camera electronic shutter, working at the speed of 1/2 000 s, has been synchronized with the illuminators working in pulsed mode (500 microseconds illumination flashes per frame). The system is suitable for working outdoors because the pulsed infrared illumination can be much higher than the ambient light (even with the sun shining above), maintaining the average illumination power at a level which avoids damaging the eye, following ophthalmologic recommendations [30, 31] and working well below dangerous levels.

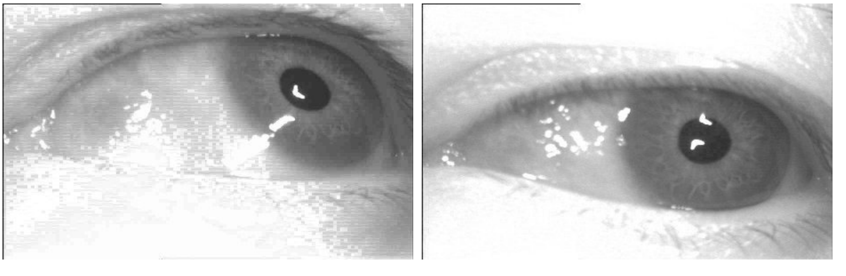


Figure 2. Eye images with glint distortion

4 SYSTEM ARCHITECTURE

From the point of view of system architecture, as the initial main application for this system was the analysis of driver behavior inside a conventional but sensorized vehicle, the mini-ITX format was chosen for the processing hardware. However, as a single processor is incapable of supplying the required processing power, the system has been based on a small cluster. As an added benefit, this also allows to develop an upgradeable system for experimenting with more complex images (larger, with more bits per pixel) and more complex image processing algorithms. Therefore, a PC cluster based on four low-power 1 GHZ C7-nanoBGA2 processors, running Linux Fedora Core was chosen to run the application. The cluster is connected by a Gigabit-Ethernet and MPI is employed as communication development support [26, 27, 28]. This architecture is flexible, thus allowing to extend the scope of the research, e.g. tracking the point of gaze using the information from both eyes.

5 PROCESSES

The gaze tracking system developed is based on three conceptually independent processes, which can be associated to the three specific devices used: the NFV camera pointing at one of the user's eyes, the WFV camera taking images of an area including the face of the user, and the orientating device that allows the NFV camera to follow the eye (pan-tilt unit). The associated processes are as follows:

Point of gaze detection. This consists of the following conceptually sequential phases:

- PG 1. Determining the pupil center and the iris radius in the NFV camera image.
- PG 2. Identifying the representative point of each glint produced on the eye by the infrared illuminators.
- PG 3. Estimating the point of gaze, based on the pupil glint vector determined during previous phases.

The last phase, in which the point of gaze is estimated, is based on a set of parameters previously obtained by means of a calibration process (see Section 9). It has to be performed at the beginning of the session and every time the system user changes.

Eye tracking. The eye of the user has to remain centered in the image taken by the NFV camera. Eye tracking consists of the following phases:

- ET 1. Identification of the pupil center in the NFV camera image. If the pupil can be located, the system goes directly to phase ET3.
- ET 2. If the pupil could not be located in ET1, this process returns to the main program and the control signal for the pan-tilt unit is generated from the information given by the eye position detection process.
- ET 3. Computation of the deviation of the pupil center with respect to the image center.
- ET 4. Generation of the signal to control the pan-tilt unit as a function of the deviation computed in ET3. This deviation is used as the error signal for a PID control system.

Eye position detection. This process locates the user's eye previously used for tracking in the WFV camera image. It consists of the following phases:

- ED 1. Delimitating the user's face and identification of both user's eyes in the WFV camera image.
- ED 2. Calculating the coordinates of both eyes.
- ED 3. Selecting the eye previously tracked.
- ED 4. Determining the coordinates to feed the tracking system, in order to place the user's eye in the center of the NFV camera image.

The last phase, in which the coordinates are determined, is based on a set of parameters previously obtained by means of an additional calibration process (see Section 9).

6 TEMPORAL REQUIREMENTS

The three above-mentioned processes could be analyzed independently, although from this brief description it transpires that a very clear connection exists between them. Moreover, it is convenient to delimit the most computationally demanding aspects, in order to properly dimension the system and determine if a parallel solution is needed [17].

Firstly, it can be observed that both the point of gaze detection and the eye tracking control processes include the pupil center identification phase (using the NFV camera image). A priori, sharing this phase between both processes might seem useful; but taking into account that the eye tracking process chiefly needs to be as fast as possible without requiring high accuracy for pupil center identification, the authors opted for separate implementations for the gaze detection and the eye tracking processes.

Eye tracking uses a downsized image to reduce computing time and speed up recovery from tracking errors.

Point of gaze detection requires working with high resolution images to obtain high accuracy, making it an expensive task. However, determining the exact point of gaze is the final objective of the whole system and in this sense, it is the most important process.

Focusing the analysis on the point of gaze identification process, two NFV camera working modes can be established:

Frame mode. In this mode a complete frame (two interlaced fields) is captured by the NFV camera. Once such frame has been processed, the calculated point of gaze is considered to be the point of gaze for the instant in which the frame was captured. As CCIR European standard cameras and acquisition systems are being used (interlaced images, 2 fields, 25 Hz frames), tasks associated with processing each frame must be performed in less than 40 ms. Using conventional CCIR frame grabbers and capturing with 256 gray levels, the information to be processed every 40 ms corresponds to an image of 768x576 bytes.

Field mode. In this mode, just a single image field has to be acquired before performing its processing and obtaining the point of gaze associated to that field. As a new field is obtained every 20 ms, this is the maximum time available for performing the whole processing needed to obtain the point of gaze. Using conventional CCIR frame grabbers and capturing with 256 gray levels, 768×288 byte images are generated.

From the above-mentioned data, it follows that computational requirements are similar in both cases, because although in field mode there is only half the time available for processing, this is offset by the fact that only half the amount of pixels have to be processed. On the other hand, field mode gives a better temporal resolution, because the point of gaze can be determined at twice the frequency of frame mode. Moreover, focusing the attention on the eye tracking process, this double frequency in the results considerably enhances control, avoiding a high percentage of the potential tracking losses when working in frame mode. Despite these considerations, the current system works in frame mode, due to limitations of the frame grabber driver being used.

7 IMPLEMENTATION

In the next sections the system implementation is described focusing on the three processes described in Section 5.

7.1 Point of Gaze Detection

The most complex phase in the point of gaze detection process in terms of the amount of processing power required corresponds to the identification of the pupil center. That is because different filters have to be applied in order to obtain acceptable results when locating the pupil and identifying its center. Some of these are applied over the whole image, while others are only applied to a subset of its pixels. The objective of applying these filters is to minimize both the FAR ("false acceptance rate", i.e. the percentage of frames in which the pupil has been erroneously identified) and the FRR ("false rejection rate", i.e. the percentage of frames in which the image is discarded because a pupil has not been detected despite being present). The current system (low power 1GHZ C7-nanoBGA2) requires nearly 85 ms for this detection phase, including the accurate location of the pupil center once it has been identified.

In this process, the pupil center is estimated by determining the pupil mass center (without the glints) and the iris radius is estimated by determining the edge of the iris. Next, a blob analysis inside the iris is performed, in order to locate all the visible glints fulfilling certain criteria. The glints are then sifted to identify genuine glints.

Once the pupil center and the position of the glints have been determined, the point of gaze is estimated using the pupil-glint vector and the parameters of the

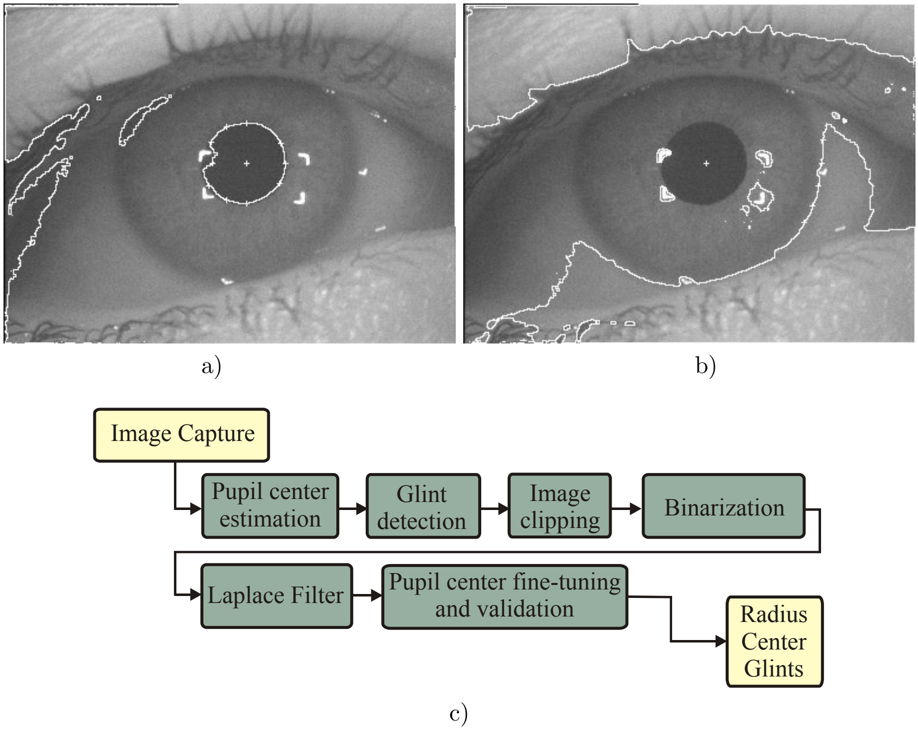


Figure 3. Processing stages associated to point of gaze detection: a) Determining pupil contour and center, b) Determining the radius of the iris, c) Summary of processing stages

model, obtained by means of a previous calibration. The pupil-glint vector is defined by the pupil center and the location of a single glint. Using four infrared illuminators assures to have at least one valid glint even if some glints are distorted (see Figure 2), but only one vector at each time is used to determine the point of gaze. If the calibration process has been done correctly there should be very small differences when using different glints. The calibration procedure is explained in Section 9 although it is explained in more detail in [17]. The time spent in this phase to determine the point of gaze is almost negligible, a few microseconds.

To determine the point of gaze, its coordinates (X_G, Y_G) and the components of the pupil-glint vector (d_x, d_y) have to be correlated. Their relationship depends on the eye geometry and on the relative position of the eye with respect to the glint and the point of gaze. This correlation can be determined (if enough pairs $[(d_x, d_y), (X_G, Y_G)]$ are available) by means of a polynomial regression, using a function that generates the regression matrix R and the coefficients. As the eye and head parameters differ for different users, a set of parameters have to be determined in each case. Therefore a calibration procedure has been established to obtain the necessary pairs (d_x, d_y), (X_G, Y_G) that allows the determination of the parameters (Section 9).

Figure 3 shows the different processing phases applied to each image in order to detect the point of gaze. Note that the required accuracy in the identification of the characteristic points (pupil center and glints) requires the use of sub-pixel resolution.

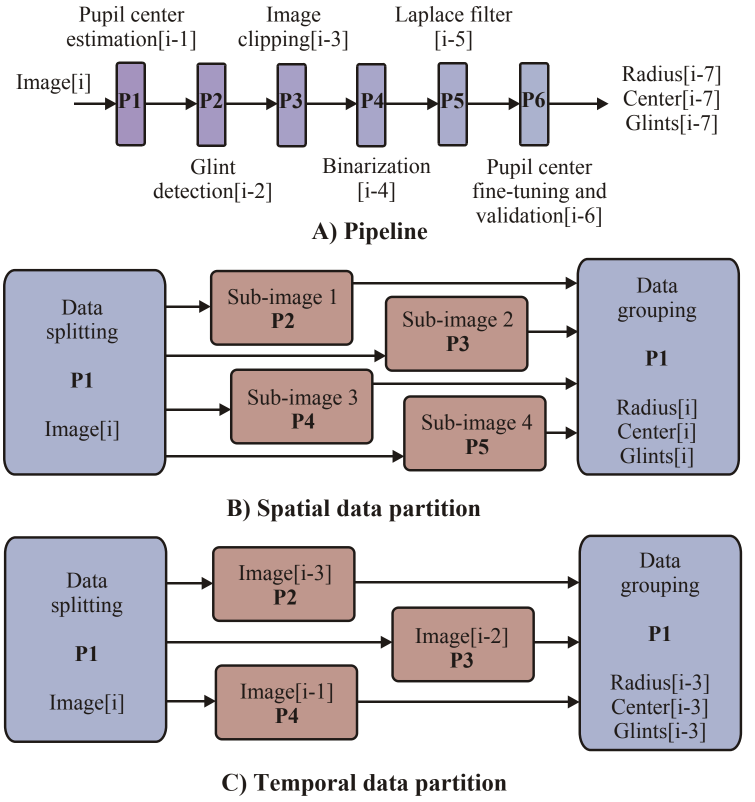


Figure 4. Parallel processing models considered

7.1.1 Parallelization

The process associated to the NFV camera images needs more than the 40 ms available. Although the required processing power could be obtained by using a conventional high performance computer, the embedded systems usually used on board of mobile platforms, such as vehicles, have a reduced performance and low power consumption (see Section 4). Therefore, it was decided to parallelize the processes associated to determining the point of gaze. Besides, a parallel system with a higher processing power than required is useful for experimenting and pursuing more ambitious goals.

Considering the characteristics of the processing involved (basically image filtering) and the difference between the time available and the time required by the process (a factor in the order of 2 to 4), the parallel approach could either involve a pipeline model or spatial data partition (for example, with 4 sub-images) or even temporal data partition (each processor works with the data corresponding to a given time, that is, to a full image). Figure 4 shows the models considered. The temporal data partition model has been chosen for the proposed system because it is simpler than spatial data partition. Temporal partition avoids splitting images and joining results. Splitting images creates additional challenges because the pupil and iris shapes are lost and the glints may not be together. Each slave has to analyze an unknown part of the eye. Joining the results also adds complexity as several pupil centers may be identified by the different slaves and the master would have to discard the wrong ones.

There is a master and three slave processors. The master distributes complete eye images among the slaves using MPI (actually OpenMPI 1.3). This means that each slave computer performs the complete algorithm with an independent full-sized image, which has been captured at a different time (every 120 ms).

The task distribution is as follows [29]:

- Every 40 ms the master performs three tasks: first, it delivers the image through the gigabit network to a slave PC, which lasts 5 ms, then it processes the eye-tracking algorithm, which lasts 10 ms, and finally it receives the pupil center coordinates.
- The three slave PCs carry out the point of gaze task, which lasts 85 ms.

A point of gaze estimate is obtained every 40 ms but with a global delay of 120 ms (three full-size images are processed every 120 ms).

When the eye-tracking process loses the eye, the master starts an additional task before receiving the pupil center coordinates: the eye detection algorithm. This process adds 5 ms (Table 1). However, this fact does not produce any delay in the tasks of the next cycle, and if the eye can be found in the WFV camera image, the system will recover from that tracking error.

The processing model which the system is based on implying that the master process has to maintain information about as many images as slave processors that have been defined plus one (all the images that are being processed by slave nodes and the one being acquired by the WFV camera). The characteristics of that information depend on the final use of the system. For example, if the system is used for studying the behavior of a vehicle driver in a conventional driving session, it would be enough to store the timestamp of each image capture. Later, once the point of gaze coordinates have been obtained, they will be associated to their timestamp. On the other hand, for a laboratory installation where the tracking results have to be

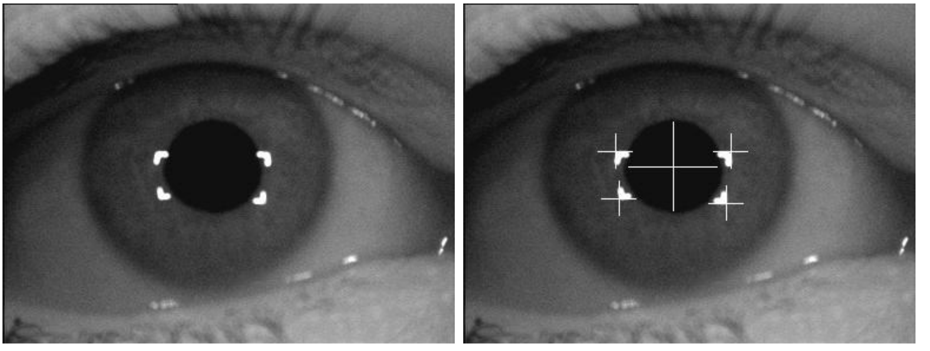


Figure 5. Pupil center and glint identification

displayed, storage of the whole image may be needed (this is the worst case scenario in the current implementation), in which case the image is overlayed with the marks corresponding to the pupil center and glints obtained (Figure 5). Thus, the results are displayed with a delay of three frames with respect to the image capture time. From a practical point of view, for example, in the case of implementing a user interface based on this system, a delay of up to five frames (200 ms) can be considered valid.

7.2 Eye Tracking

Eye tracking is based on a very simple concept: maintaining the pupil at the center of the NFV image, i.e. controlling the camera orientation in such a way that its axis is aligned with the pupil center.

The main objective of this phase is to determine the pupil center (because the image center is already known), so as to identify the tracking error and use it to feed the position control system. A critical aspect to obtain an adequate position control is the sampling frequency, i.e., the frequency of generation of the error signals feeding the control algorithm (a modified PID). Therefore, the previously described procedure for computing the point of gaze is inadequate, as this would imply obtaining the pupil center with a delay of several frames with respect to the actual time in which the measure was taken.

Moreover, the eye tracking process does not require high accuracy in determining the pupil center, and it is preferable to focus efforts on obtaining this value at full frame rate, albeit with a small margin of error.

The solution adopted saves time by processing the image locally. Eye image resolution has been downsized to 192×144 pixels, a value allowing the whole processing required to be completed by a single processor in less than the frame capturing time.

This time limitation is actually harder to tackle, because the same processor has to execute other functions in the same time interval. Also, the situation could become more complex in the future if fields are used instead of frames, because, although image resolution is halved, the processing time available is also halved (see Section 6).

Figure 6 shows the processing phases performed with each image to control eye camera orientation (pan-tilt) through the PID error signal.

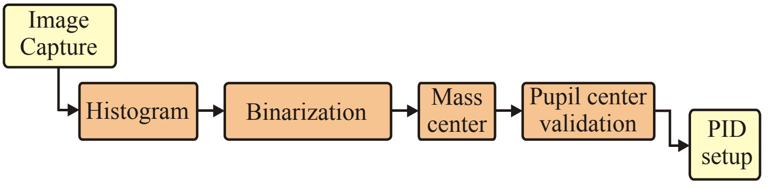


Figure 6. Processing stages of the eye tracking process

7.3 Eye Position Detection

The process for detecting the eye position in the image provided by the WFV face camera is initiated whenever a tracking error is produced, i.e. the pupil disappears from the NFV eye camera image. It must be considered that in the current configuration a cycle corresponds to the time taken to capture an image frame, but if fields were used, it would correspond to the time taken to capture an image field.

On the other hand, the system should respond properly when an error occurs. In this system, the response to a tracking error consists in repositioning the pupil in the center of the NFV camera image. It must be taken into account that tracking errors can be relatively frequent, due to sudden head and eye movements and to slow blinking. These errors are inevitable and not indicative of the global system quality.

Two implementation alternatives were considered for the eye position detection process: the first one is based on parallelizing the process of detecting the eye in the face image. The second one consists in executing it on the master processor once the size of the problem has been reduced. As the tracking process does not require high accuracy, the second alternative was chosen.

Figure 7 shows the processing phases performed to estimate the position of both eyes starting from the WFV camera image. An example with results obtained from this process is also shown in Figure 7.

8 SYSTEM INTEGRATION

The three processes on which the gaze tracking system has been based must cooperate, in the sense that they share input information (eye images are shared between the point of gaze identification and eye tracking processes) and output information

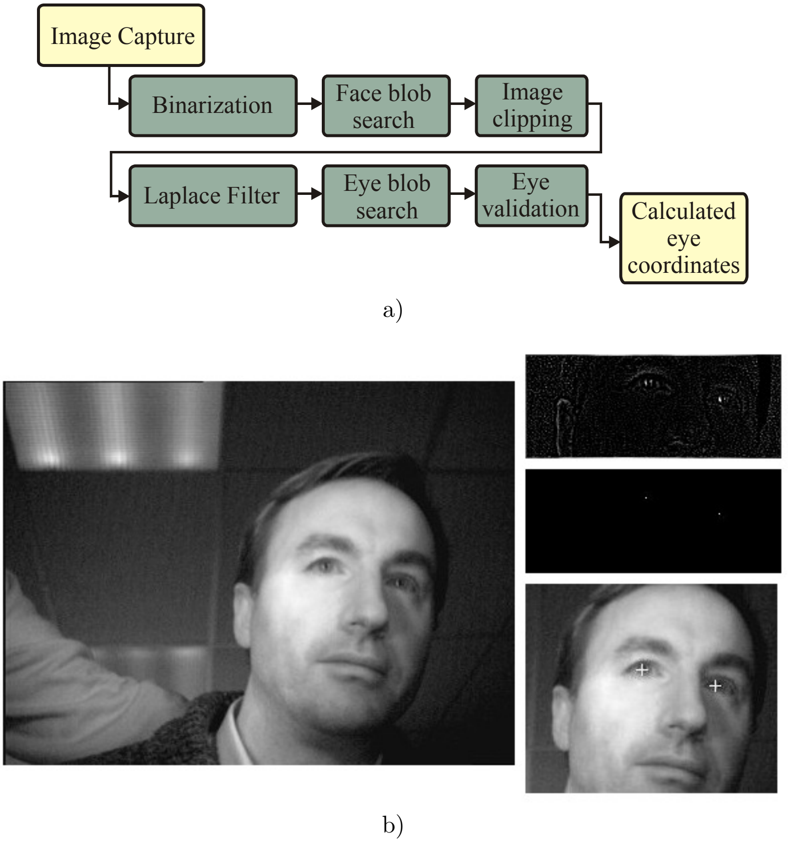


Figure 7. Face analysis for eye location: a) Eye location processing stages, b) Application example

(correct detection of the pupil determines if the tracking system is fed with the information generated by the tracking process or by the eye detection process).

An initial prototype of the system has been implemented and a simple application, whose only purpose is the functional validation of the system, has been employed as a workbench.

Implementation has been based on temporal data partition. Figure 8 shows a general view of the system with the different processing and communication tasks.

Table 1 shows some of the time measures taken for different phases of the system, obtained by instrumenting the program code with system calls to measure the elapsed time with microsecond resolution.

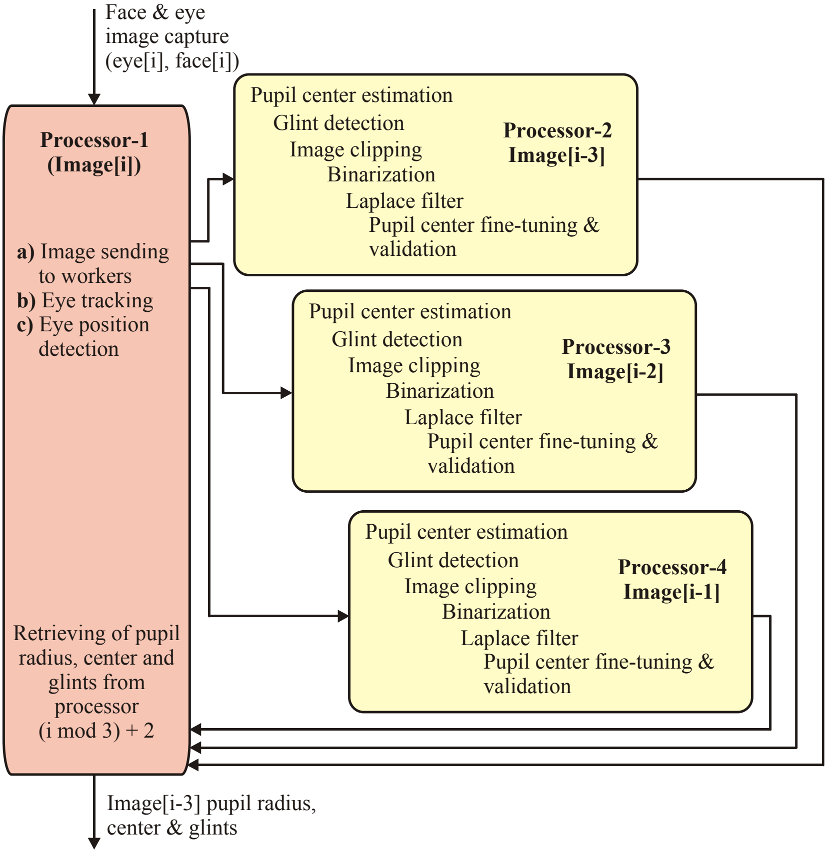


Figure 8. Temporal data partition of processing tasks

FUNCTION	time (μs)
Available between frames	40 000
Master tasks	< 20 000
– Image transmission (1 Gb/s)	5 000
– Eye tracking	10 000
– Eye position detection	5 000
Response reception (1 Gb/s)	< 40
Slave tasks	< 90 000
– Image reception (1 Gb/s)	5 000
– Point of gaze detection	85 000
Response transmission (1 Gb/s)	< 40
Total processing of each image	< 120 000 (3 frames)

Table 1. Processing and communication times in the parallel system

9 CALIBRATION

The application tries to identify the point of gaze over a uniformly colored panel which is presented to the user carrying out the experiment. First of all, a calibration process is performed using the interface, and later the accuracy of the system is evaluated.

The calibration process consists in repeating a process in which the user moves the pointer to an arbitrary point of the screen which will be considered the calibration point; next, fixing the gaze on that point, she/he clicks a mouse button. The system will identify the position of the point on the panel, as well as the pupil center and the position of the glints in the eye image.

Once this process has been repeated for a predetermined number of uniformly distributed points a polynomial regression is performed to obtain the parameters of the model. At least nine points must be used but defining a higher number of points would provide a more precise calibration. Once the system has been installed in the car, the callibration points will be located on the windshield and the dashboard, and a scene camera will be used to establish the correlation between the user gaze and the position of the callibration points.

10 RESULTS

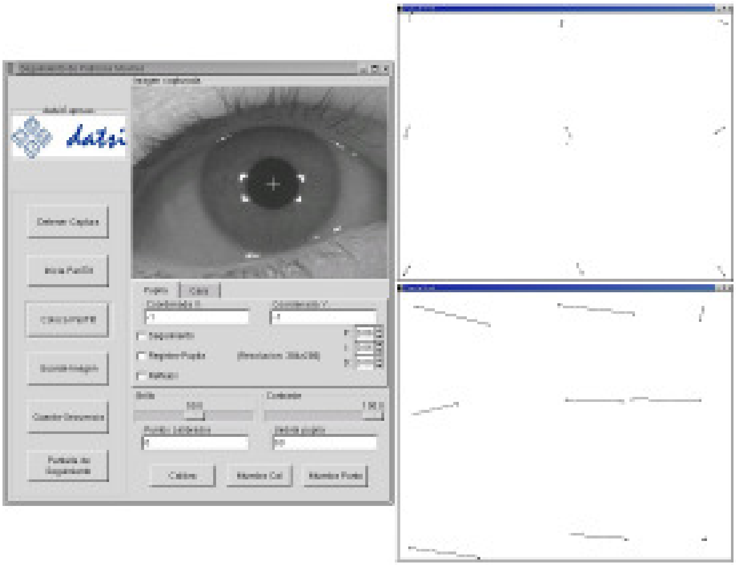
Once the system has been calibrated, it is ready to be used for trials. In order to get useful feedback about experimental results, a graphical application has been developed to show the images and their analysis. Using this application, the user can place the pointer on any panel position, fix his/her gaze on that point direction, and finally click a mouse button. The system identifies the estimated point of gaze based on the model parameters and on the pupil-glint vector. Both the actual and the estimated point of gaze are represented in the same window, thus allowing the user to see the estimated error.

Figure 9 shows the application interface and some characteristic results obtained with the current system. Each line in the result images represents the error, i.e. the distance between the estimated point of gaze and what the user has determined as her/his actual point of gaze.

The dispersion observed in the results, specifically near the lower end of the figure, is due to a calibration problem: this phase is critical for achieving accuracy in the global process of identification of the point of gaze. If any of the calibration points is stored with an error (due to a lack of concentration on behalf of the user) it will affect all of the system estimates, especially those in the vicinity of the erroneous calibration point.

The results of experiments carried out with this system and with different people acting as subjects are summarized in Table 2.

The calibration method developed for this system helps the user to discard those calibration points classified as unacceptable by the system. Once properly



a) b)

Figure 9. Interface and results obtained with the test application: a) application interface, b) results for incorrectly and well calibrated experiments

Calibration quality	Abs ERR	Rel ERR
Good X-axis Good Y-axis	20 pixels	2.5 %
Good X-axis Bad Y-axis	180 pixels	22 %
Bad X-axis Good Y-axis	220 pixels	27 %
Bad X-axis Bad Y-axis	285 pixels	35 %

Table 2. Experimental results

calibrated, the system can be used as an interface for other applications providing consistently reliable results.

11 CONCLUSIONS

A complete prototype of a distributed, real-time, and non-intrusive gaze tracking system has been developed. The chosen hardware, four low-power C7-nanoBGA2 processors, is cheap, standard and can be easily installed inside a vehicle, while the distributed computing solution provides enough processing power to obtain good quality results. The system determines the point of gaze from the pupil-glint vector, using sub-pixel estimates of both the pupil center and the glint position. Four infrared L-shaped illuminators, synchronized with the NFV camera shutter, allow the eye gaze to be determined, even at extreme eye or head angles. The prototype has been tested for many users, showing that it works properly, and it has a high potential for use in other applications.

Several aspects of the system are being modified to improve its robustness in adverse conditions. Among others, the following improvements are being carried out or are under consideration:

1. The calibration process is being modified to improve the accuracy of the system. Also the calibration process for both cameras will be unified by making adequate use of the information obtained with them.
2. As the point of gaze is determined from a single pupil-glint vector, the user's head movements towards and away from the camera can introduce some error. The four infrared illuminators can be used to implement a pupil center-multiple glints method to eliminate the influence of such head movements.
3. Substitution of the tracking mechanism by a mobile mirror placed on a gimbal and reprogramming of the control system. The mirror is much lighter than the camera with its lens, which should lead to a significant improvement in tracking precision.
4. The proposed system currently works in frame mode due to limitations of the frame grabber driver being used. The frame grabber driver is going to be replaced and the software is going to be adapted to use field mode instead of frame mode (double tracking frequency).
5. Refinement of the algorithms that determine the pupil center, improving accuracy.
6. In the near future more complex processing functions will be incorporated to make the system more robust. Identifying the point of gaze based on both eyes is another interesting approach.

REFERENCES

[1] Arne, J.—Engell-Nielsen, T.: Eye Controlled Media: Present and Future State. DIKU, University of Copenhagen, Tech. Rep., 1995, [online], available: `http://wscg.zcu.cz/wscg2003/papers_2003/a83.pdf`.

[2] Böhme, M.—Meyer, A.—Martinetz, T.—Barth, E.: Remote Eye Tracking: State of the art and Directions for Future Development. In COGAIN 2006, The 2nd Conference on Communication by Gaze Interaction, 2006, pp. 12–16.

[3] Funahashi, T.—Hoshino, T.—Tokuda, N.—Fujiwara, T.—Koshimizu, H.: Complete Vision-Based Modeling of Human Behavior in Visual Inspection. Conference on Human System Interactions, May 2008, pp. 356–359, [online], available: `http://www.inb.uni-luebeck.de/publikationen/pdfs/BoMeMaBa06.pdf`.

[4] Kim, K.—Ramakrishna, R. S.: Vision-Based Eye-Gaze Tracking for Human Computer Interface. In Proceedings of the IEEE International Conference on Systems,

Man and Cybernetics 1999, Vol. 2, pp. 324–329, [online], available: `http://www.worldscinet.com/ijprai/23/2303/free-access/S0218001409007259.pdf`.

[5] Sesin, A.—Adjouadi, M.—Ayala, M.—Barreto, A.—Rishe, N.: A Real-Time Vision Based Human Computer Interface as an Assistive Technology for Persons with Motor Disability. NWSEAS Transactions on Computer Research, Vol. 2, 2007, p. 115.

[6] Meyer, A.—Böhme, M.—Martinetz, T.—Barth, E.: A Single-Camera Remote Eye Tracker. In PIT 2006, LNAI 4021. Springer 2006, pp. 208–211.

[7] Lee, C. H.—Kim, J. S.—Jung, Y. K.—Park, K. H.: A Real-Time, Parallel Face Recognition on Task-Level Pipelined Multicomputer. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics 1995, pp. 2289–2294, [online], available: `http://www.cogain.org/cogain2008/COGAIN2008-Proceedings.pdf`.

[8] Liu, J.—Beldie, I. P.—Wöpking, W.: A Computational Approach to Establish Eye-Contact in Videocommunication. In Int. Workshop on Stereoscopic and Three Dimensional Imaging 1995, pp. 229–234.

[9] Merchant, S.—Schnell, T.: Applying Eye Tracking as an Alternative Approach for Activation of Controls and Functions in Aircraft. In Proceedings of the Digital Avionics Systems Conferences 2000, Vol. 2, pp. 5A5/1–5A5/9.

[10] Villanueva, A.—Cabeza, R.—Porta, S.: Gaze Tracking System Model Based on Physical Parameters. IJPRAI, Vol. 21, 2007, No. 5, pp. 855–877.

[11] Villanueva, A.—Cabeza, R.: Models for Gaze Tracking Systems. J. Image Video Process., Vol. 2007, 2007, No. 4, pp. 1–16, [online], available: `http://dx.doi.org/10.1155/2007/23570`.

[12] Su, M.-C.: An Eye Tracking System and Its Application in Aids for People with Severe Disabilities. Biomedical Engineering: Applications, Basis and Communications, Vol. 18, 2006, No. 6, pp. 319–327.

[13] Hansen, D. W.—Hansen, J. P.: Eye Typing with Common Cameras. In ETRA '06: Proceedings of the 2006 Symposium on Eye Tracking Research & Applications, New York, NY, USA, ACM 2006, p. 55.

[14] Droege, D.—Geier, T.—Paulus, D.: Improved Low Cost Gaze Tracker. In Proceedings of the 3rd COGAIN Conference 2007, pp. 37–40.

[15] Li, D.—Winfield, D.—Parkhurst, D.: Starburst: A Hybrid Algorithm for Video-Based Eye Tracking Combining Feature-Based and Model-Based Approaches. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR Workshops), June 2005, p. 79.

[16] Daunys, G.—Ramanauskas, N.: The Accuracy of Eye Tracking Using Image Processing. In NordiCHI '04: Proceedings of The Third Nordic Conference On Human-Computer Interaction, New York 2004, pp. 377–380.

[17] Pérez, A.—Córdoba, M. L.—García, A.—Méndez, R.—Muñoz, M.—Pedraza, J. L.—Sánchez, F.: A Precise Eye-Gaze Detection and Tracking System. In WSCG '03 – 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, pp. 105–108, [online], available: `http://dx.doi.org/10.1016/j.cviu.2004.07.005`.

[18] Droege, D.—Schmidt, C.—Paulus, D.: A Comparison of Pupil Centre Estimation Algorithms. In Proceedings of the 4^{th} COGAIN Conference 2008, pp. 23–26.

[19] Yoo, D. H.—Chung, M. J.: A Novel Non-Intrusive Eye Gaze Estimation Using Cross-Ratio Under Large Head Motion. Comput. Vis. Image Underst., Vol. 98, 2005, No. 1, pp. 25–51, [online], available: http://www.springerlink.com/content/a776424687t41161/fulltext.pdf.

[20] Noureddin, B.—Lawrence, P.—Man, C.: A Non-Contact Device for Tracking Gaze in a Human Computer Interface. Comput. Vis. Image Underst., Vol. 98, 2005, No. 1, pp. 52–82, [online], available: http://www.sciencedirect.com/science/article/pii/S1077314204001122.

[21] Lu, Y.—Zhou, J.—Yu, S.: A Survey of Face Detection, Extraction and Recognition. Computing and Informatics, Vol. 22, 2003, No. 2, pp. 163–195.

[22] Mohammed, G. J.—Hong, B. R.—Jarjes, A. A.: Eyeball Localization Based on Angular Integral Projection Function. Informatica, Vol. 33, 2009, pp. 475–480.

[23] Mohammed, G. J.—Hong, B. R.—Jarjes, A. A.: Accurate Pupil Features Extraction Based on new Projection Function. Computing and Informatics, Vol. 29, 2010, No. 4, pp. 663–680.

[24] Jing, M.-Q.—Chen, L.-H.: A Novel Method for Horizontal Eye Line Detection Under Various Environments. IJPRAI, Vol. 24, 2010, No. 3, pp. 475–498.

[25] Campadelli, P.—Lanzarotti, R.—Lipori, G.: Precise Eye and Mouth Localization. IJPRAI, Vol. 23, 2009, No. 3, pp. 359–377, [online], available: http://dx.doi.org/10.1142/S0218001407005697.

[26] Gropp, W.—Lusk, E.—Skjellum, A.: Using MPI: Portable Parallel Programming with the Message-Passing Interface. 2^{nd} ed., MIT Press 1999.

[27] Gropp, W.—Lusk, E.—Skjellum, A.: Using MPI-2: Advanced Features of the Message-Passing Interface. MIT Press 1999.

[28] Pacheco, P.: Parallel Programming with MPI. Morgan Kaufmann 1997.

[29] García, A.—Sánchez, F. M.—Pérez, A.—Pedraza, J. L.—Méndez, R.—Córdoba, M. L.—Muñoz, M. L.: A Distributed Real Time Eye-Gaze Tracking System. In 9^{th} IEEE International Conference on Emerging Technologies and Factory Automation, 2003, pp. 545–551.

[30] Guidelines on Limits of Exposure to Broad-Band Incoherent Optical Radiation (0.38 to $3\,\mu$m). International Commission on Non-Ionizing Radiation Protection, Health Physics, Vol. 73, 1997, No. 3, pp. 539–554.

[31] ICNIRP Statement on Light-Emitting Diodes (LEDS) and Laser Diodes: Implications for Hazard Assessment. International Commission on Non-Ionizing Radiation Protection. Health Physics, Vol. 78, 2000, No. 6, pp. 744–752.

Antonio García Dopico received the M. Sc. degree in Computer Engineering and the Ph. D. degree in Computer Science from the Universidad Politécnica de Madrid (UPM), Spain, in 1993 and 2001, respectively. He is currently an Associate Professor in the Department of Computer Systems Architecture and Technology at UPM. His research interests include parallel and distributed computer systems.

José L. Pedraza is an Associate Professor in the Department of Computer System Architecture and Technology at the Technical University of Madrid (UPM) in Spain. He received his M. Sc. degree in Electronics in 1981 from the University of Salamanca (Spain) and his Ph. D. in 1987 from the same University. His research interests include real time, image processing and embedded systems.

M. Luisa Córdoba received the M. Sc. degree in Computer Engineering and the Ph. D. degree in Computer Science from the Universidad Politécnica de Madrid (UPM), Spain, in 1986 and 1996, respectively. She is currently an Associate Professor in the Department of Computer Systems Architecture and Technology at UPM. Her research interests include computer architecture and high performance computing.

Francisco M. Sánchez is Ph. D. in Electronics and Control. He is an Associate Professor in the Department of Computer System Architecture and Technology at the Technical University of Madrid (UPM) in Spain from 1998. His main research interest is developing of software for ubiquitous and web devices for giving Collective Intelligence to an unlimited number of people. He is currently applying his research in Citizen Science in the field of astronomy. He participates in several national and international projects of citizen science applied to astronomy. He is currently coordinator and PI of GLORIA project (GLObal Robotic telescopes Intelligent Array for e-Science (`http://gloria-project.eu`)).

Antonio Pérez received the M. Sc. degree in Telecommunication Engineering and the Ph. D. in Computer Science from the Technical University of Madrid (UPM), Spain, in 1979 and 1982, respectively. He is currently a Full Professor in the Department of Computer Systems Architecture and Technology at UPM. His research interests include computer architecture, fault tolerant computers, and microprocessor systems design.