Title	Real-time eye gaze tracking for gaming design and consumer electronics systems
Author(s)	Corcoran, Peter
Publication Date	2012
Publication Information	Corcoran, P.M.,Nanu, F.,Petrescu, S.,Bigioi, P. (2012) 'Real-time eye gaze tracking for gaming design and consumer electronics systems'. Consumer Electronics, IEEE Transactions on, 58 (2):347-355.
Publisher	IEEE
Item record	http://hdl.handle.net/10379/3625
DOI	http://dx.doi.org/http://dx.doi.org/10.1109/TCE.2012.6227433

Real-Time Eye Gaze Tracking for Gaming Design and Consumer Electronics Systems

Peter M. Corcoran, *Fellow*, IEEE, Florin Nanu, Stefan Petrescu, and Petronel Bigioi, *Senior Member*, IEEE.

Abstract — *Real time face detection combined with eye-gaze tracking can provide a means of user input into a gaming environment. Game and CE system designers can use facial and eye-gaze information in various ways to enhance UI design providing smarter modes of gameplay interaction and UI modalities that are sensitive to a users behaviors and mood. Here we review earlier approaches, using wearable sensors, or enhanced IR illumination. Our technique only requires video feed from a low-resolution user-facing camera. The algorithm is described and some comparative results on a range of embedded hardware are provided. The potential for using eye-gaze as a means of direct user input and improving the accuracy of estimation accordingly is also discussed.[1]*

Index Terms — Face Tracking, Eye Tracking, Eye-Gaze, HCI, User Interface, Embedded Imaging.

I. INTRODUCTION

Gaming is one of the fastest developing IT industries in recent years. Providing new means of man-machine interaction can change the way gaming is perceived and enhance the underlying user experience. And with the proliferation of portable entertainment devices in various form factors, it becomes more important to monitor the user to determine their engagement and reaction to the gameplay.

Here we present solutions to enable real time human face detection and tracking combined with real time eyeball tracking. Additionally, these solutions can be complemented with techniques for the determination of human emotions to provide feedback into the gaming user interface and gameplay logic [1]. Portions of the underlying work are already deployed in a number of commercial devices in either software and/or hardware accelerated versions.

II. BACKGROUND & RELATED WORKS

A. HCI with Eye-Gaze

The use of eye-gaze as an input mechanism is not a particularly new idea. Early research focused mainly on computer interfaces for quadriplegic users [1], [2]. Indeed Levine [2] can probably lay first claim to a computer with an eye-gaze based user interface back in 1981.

Jacob [3] is probably one of the earliest researchers to consider the use of eye-gaze as a means to provide a real-time

dynamic interface means for practical use by normal users. Interestingly he comments on the "Midas Touch" effect:

"At first, it is empowering to be able simply to look at what you want and have it happen, rather than having to look at it (as you would anyway) and then point and click it with the mouse or otherwise issue a command. Before long, though, it becomes like the Midas Touch. Everywhere you look, another command is activated; you cannot look anywhere without issuing a command." We will return later to this insightful commentary.

More recently Duchowski [4] has taken a broad look at the use of eye-tracking and provides similar observations to those of Jacob. He cautions with regard to overloading the perceptual system of the eye with the motor-task of a pointing device, but indicates that the use of gaze is more promising as an indicator of intent and to assist with interaction in increasingly complex contextual situations.

B. Measuring Eye-Gaze

A great deal of research has been undertaken using various head-mounted systems to accurately measure eye-gaze. However such systems are not of interest or relevance in the context of this work as consumers are unlikely to find awkward headwear an acceptable solution for daily usage.
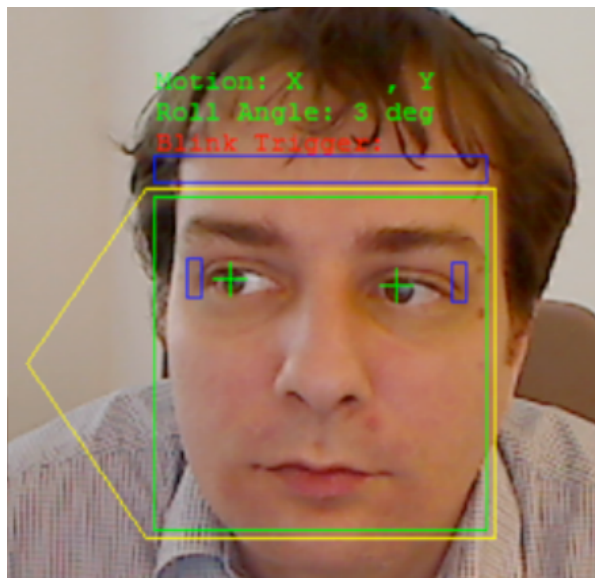

Figure 1: Real-time output data with indicators from our algorithm, including face pose and eye-events; note the absence of "headgear".

Recent improvements in the capabilities of embedded imaging both from the optical and computing perspectives suggest that remote monitoring of eye-gaze offers an acceptable consumer-oriented solution. The authors of [5]

comment on the combined problems of determining head pose and eye-gaze, which is partly dependent on the head pose and orientation. To improve accuracy they have used a zoom camera system to enable remote eye-tracking at several meters distance. This is certainly a problem for many CE applications where users will be located at a distance of several meters from the main display device when viewing TV or playing games on a console which is interfaced with the TV.

Other researchers have employed stereo imaging [6], combinations of wide-angle and zoom cameras [7], near-Infrared (NIR) illumination [1] and combinations of these in order to allow larger head movements [8]. While some of the techniques are interesting we remark that a mechanical pan & zoom camera is unlikely to be suitable for consumer applications. Fortunately there are developments in consumer imaging technology that may provide an electronic alternative, as we shall discuss shortly.

More recently the authors of [9] have described a gaze-tracking system to control a large-screen TV at a distance. A remote gaze-tracking camera captures the user's eye and so the user does not wear any device. The gaze position on the TV screen is obtained by using a simple 2D method based on a geometric transform with pupil center and four cornea specular reflections. This does require a near-infrared (NIR) passing filter on the camera and NIR illuminators, but has the advantage that the pupil region is easily distinguished in the input image regardless of the change in the environmental visible light.

C. Face Detection & Tracking

Face detection and tracking can be considered an enabling technology for a range of image enhancements technologies, authentications and advanced UIs in handheld devices. Having the ability to detect human faces in a scene, during the preview, image quality can be greatly enhanced. Facial location can be used to improve both the efficiency and/or quality of auto-focus, auto-exposure, and auto-white balance algorithms. The faster that face detection is performed the more time available for additional image processing based on the face data.

The first practical real-time algorithm to provide real time object detection was described by Viola & Jones in 2001 [10], [11]. A range of improvements has led to increased performance and today very acceptable open-source software is available to implement real-time frontal face tracking on AndroidTM handheld devices. However to achieve practical and robust real-time face tracking it is necessary to use the specialized hardware capabilities of embedded imaging devices [12]. And to fully realize the potential of face tracking it is important to adopt a hybrid software/hardware architecture [13] that enables substantial improvements in detection speed, accuracy and robustness.

D. Advanced Face and Face Feature Models

Much of the past literature suggests that the most appropriate and effective uses of eye-tracking and advanced facial analysis will be directed towards improving the context and environment for the user to ensure an improved experience. In addition to the direct measurement and analysis of eye-gaze there are a range of additional techniques that can be combined to improve our analysis capabilities.

1) Using non-Face Information

One very effective technique to gather additional information about a person is to take observations on their clothing and hair. These can be analyzed by considering peripheral regions in the vicinity of the face [14], [15]. Often a distinctive item of clothing can be easily identified and can be strongly indicative of a person's situation or mood. Similarly the appearance of our hair can indicate a formal or informal context.

2) Advanced Face Models

Active Appearance Models (AAM) [16], [17], [18] can be used effectively to model facial structure and features once some underlying information about the face is obtained. In particular it is possible to use truncated shape models to good effect to determine facial expressions [16], [18] and where it is practical to employ a person-specific model performance can be much improved [18]. While these models can be employed for eye-tracking [17] this can best play the role of refining a preliminary estimate of eye-gaze as the computation requirements for real-time tracking would require a dedicated processor.

3) Stereoscopic Imaging Methods

The use of stereoscopic imaging has potential to solve many long-standing problems in face analysis. For example AAM models can be improved through stereo capture, offering a practical means to realize person-specific face models for an individual [19]; accurate and real-time determination of image depth [20] can improve many algorithms, including the accuracy of eye-tracking. For practical deployment of such techniques, however, it is necessary to realize a low-cost consumer stereo imaging system [21].

III. THE NEED FOR REAL-TIME EYE TRACKING

The presence of a user-facing camera in modern portable devices (mobile phones, gaming portable devices, etc..) is not a new concept and has been installed there by various manufacturers for various communication use-cases. Another use for this camera is related to various scenarios where the user captures self-pictures and/or movies. The presence of this user-facing camera opens up a number of additional possibilities and usage scenarios as described in this section.

A. Power Management

An important aspect for battery-operated single user devices, such as smart-phones, gaming consoles and laptops is optimization of power usage. When the user is not directly viewing the monitor, then the device should implement progressive levels of power management – beginning with a dimming of the display backlight and culminating with a complete shut-down of the device. Assuming the eye gaze can be detected successfully and the capability of the display to have the backlight controlled on specific areas, the operating system could also implement selective ROI highlighting,

while keeping the remainder of the screen dimmed.

B. Auto Stereoscopic 3D Displays

3D displays are beginning to appear in single-user consumer devices. However, one of the main barriers to success seems to be the need for special glasses, an inconvenient requirement for many users. One solution to this problem is a single user auto stereoscopic display where the 3D effect is optimized for a certain distance to the display (e.g. 35 or 40 cm depending on the display size). Such displays rely on the use of a specialized coating on the display known as a parallax barrier. This is a coating placed in front of a display, to allow it to show a stereoscopic image. For an LCD, it consists of a layer of material with a series of precision slits, allowing each eye to see a different set of pixels, thus creating a sense of depth through parallax. The viewer must be positioned centrally, and at a specific distance in order to correctly experience the 3D effect.

Active research is ongoing into displays where the point of parallax for the display can be adapted using software control. A key requirement for such an approach to function properly is the capability to accurately measure real time face and eye movement, location and tracking data at frames rates of 30 fps or more. Without such real-time face and eye data it is not possible to control the display in a real time feedback loop.

C. Automatic Authentication and Profile Switching

Using face detection in combination with face recognition we can continuously authenticate users. This enables a range of new user-specific services and applications and also opens the door to micropayment based business models. For certain shared devices such as gaming consoles, the same information can be used for automatic user profile switching to allow personal customizations and settings to be automatically applied when a known user is detected.

Device security is also enhanced if a device requires authentication from a master user to active new services or enable certain device features.

D. Enhanced User Interface

Knowing with high degree of precision where the eyes of the player are focused and having the ability to track facial features and emotions, game designers can provide enhanced character representation, e.g. animated avatar, as well as enhanced content presentation, e.g. changing perspective of the gameplay window. User emotions can also be used as feedback to the gaming engine to dynamically adjust difficulty levels and/or decide the content presented to the user.

E. Facial Gesture Control

Various applications and games may accept user input based on facial event monitoring. In this respect, the presented technology can detect with high degree of precision various eye related events - single eye blink, double eyes blink and other blink combinations; one or both eyes shut. And these can be extended to mouth area (e.g. open/close mouth detector). Those events can be associated quite nicely with common actions in a game that otherwise require keyboard interaction

(e.g. one eye shut in a shooting game can immediately bring up the scope mode).

The use of such gestures in a gaming environment requires more study and practical testing but now that the underlying technology is sufficiently mature it is only a matter of time before we see such studies.

IV. IMPLEMENTATION

A. Foundation Technologies

The origins of our algorithm can be found in earlier work on face tracking [12]. In broad terms this employed a modified face detection algorithm based on the ideas of Viola and Jones [10], [11] but also leveraging elements of the internal hardware of the digital imaging pipeline. These ideas have found more recent realization in hybrid hardware/software architectures for face tracking [13]. What is interesting is that the underlying templates used for detecting face regions can be equally applied to individual facial features, thus the scope of the resulting hardware engine can be easily broadened to detect features within faces. And leveraging the hybrid nature of this architecture any additional logic can be implemented within the top-level software layer.

Figure 2: Face detection scans each image frame; in practice the entire image frame does not need to be scanned if a-priori data can be used to predict where the face will be found in the current frame.

Naturally we still employ a real time face detector/tracker to restrict the search for the eyes. In the embodiment described in this paper the initial face detection is performed on a low-resolution image (typically WQVGA resolution) to improve performance. A minimum face size of 14x14 pixels can be detected. On a hybrid platform there is not need to use a lo-res image as the heavy lifting is performed by the built-in hardware. (Expect to see chipsets with these capabilities available from 2013).

Once the face is detected, the principle face coordinates are up-sampled to match the full image resolution used for a precise face redetection step (typically at WVGA resolution). At this stage for the algorithm the minimum face size is 64x64 and is selected so that the eye size is large enough to ensure proper detection and tracking. Remember that the target is for CE devices and thus sensor costs should be minimized insofar

as is practical. Real-time performance can be achieved on larger image sizes, thus the limiting factor is the costs of hardware, including imaging sensor, rather than the algorithm performance.

B. Algorithm Workflow

The main steps of eye and pupil detection and tracking are shown in Figure 3. Note that eye-state, particularly eye-blink is also determined and recorded [17]. The last step is to predict the position of the face in the next frame and eventually refractor and provide the adjusted current face rectangle. During this prediction step a decision is also made whether a half face re-detection should be employed or not [22]. The whole cycle will be repeated for the next frame.
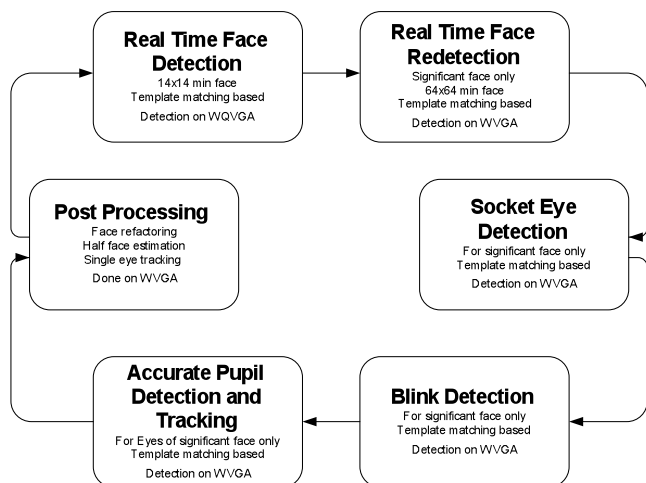

Fig 3: Eyeball Detection and Tracking Algorithm Flow

1) High-Quality Face Re-Detection

To ensure optimal quality and performance, the face is re-detected in a main, hi-res image. Ideally this is of at least WVGA and the minimum size of face region that can be reliably processed is 128x128 pixels in size. The minimum size of face region will, in turn, determine (i) the size of eye region classifiers; (ii) the minimum angle of eye-gaze that can be resolved, and (iii) the processing power required by the system.

The first step in re-detection is to up-sample the face co-ordinates from the lo-res tracking module to match the larger hi-res image. If this is a new detection then a small neighbor search is then performed with face sizes at 90%, 100% and 110% of the expected face size across the full image frame. This ensures we get a good lock on at least one of these face sizes. If the face region was already detected and is thus being tracked from frame-to-frame then information about its previous size and position can be used to limit the search area and scan size.

The template-matching engine can also be relaxed based on prior frame matches and can thus compensate for poor lighting conditions or extremes of facial pose or expression. Skin texture and color tracking can also be helpful to retain a face-lock during such extremes. These additional tracking algorithms are activated when the facial template matching

fails.

The face-tracker logic can also be used to signal a reliability metric to the eye-tracking module. Thus when extremes of pose, lighting, illumination, or other non-standard situations are encountered then eye-tracking is disabled.

It is surprising how important, and effective this simple approach can be. We recall the comments of Jacob [3] with regard to the "Midas Touch" of eye-tracking interfaces. Here we see a simple solution where the eye-tracking disengages once the user moves their head to an extreme position. This technique can be fine tuned so that only certain face poses or positions are active. Again much depends on the end-application.

In the final stage of the face-tracking process a chain of filters is applied to minimize various false-positives (FPs) and inform the end application of the confidence level, history and current shape and size of a detected/tracked face-region.

2) Socket-Eye Detection

At this point we have determined the presence of a hi-res facial region. The next task is to locate two eye-patches within said region. This is outlined in Figure 4 below.

Some global data is available about the enclosing face region, including size, shape, tracking history and detection-confidence. The area of interest is the top region of the face (green rectangle, Figure 4), excluding the nose region (red circle, Figure 4). This approach minimizes both the false positives and execution time.

The original detected hi-res face region (blue rectangle, Figure 4) is used to determine a matching base. Eye searching is carried out in a radial pattern, starting from the geometrically estimated eye-center position. This in turn is estimated based on the boundary of the face region (or from previous/predicted eye positions which are available for previously tracked faces). Once a predetermined level of confidence is exceeded the detection of a good eye is signaled; the searching process is then stopped for that eye.
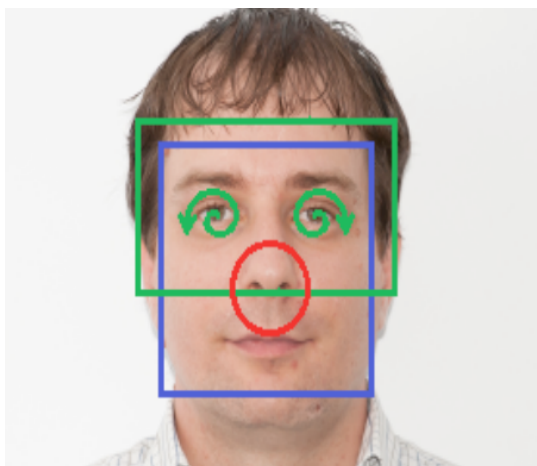

Figure 4: Eyeball Detection and Tracking Algorithm Flow

The process continues on 3-4 eye scales that are generated proportional to the face size determined during the hi-res re-detection step. This may lie between 90% and 110%o of the face size expected from the initial lo-res face-tracking algorithm. The selected eye positions are processed with a few

algorithms in order to keep a good consistency over a frame sequence (smoothing, inertia algorithms, etc).

3) Pupil and Eye-Socket - Refined Detection

An example eye-region is shown in Figure 5 below. The key points of interest are the edge corners of the eye region, defined by the blue crosses; the center of the eye-socket (small red cross) and the center of the eye-pupil (large green cross). Together these points delineate the eye region and can provide an accurate measure of the direction of a person's eye-gaze.

Figure 5: Key points within the eye-socket region; the eye corners (blue crosses); the pupil center (large green cross); and the estimated center of the eye-socket region (red cross).

Eye corners first need to be detected and this can be achieved efficiently by delineating an elliptical region containing the entire eye-socket region. Convolution kernels are next applied to accurately detect the left-hand and right-hand eye corners:

$$
K\_L = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad K\_R = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}
$$

The center of the socket-eye is located by these two corners and detected using a different convolution kernel on the region between them:

$$
K = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & -1 & 1 & 1 \\ 1 & -1 & -11 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}
$$

The pupil center is detected by convolution of the haar-classifier, shown in Figure 6 below, with the eye socket region. The maximum value indicates the pupil center, as long as it lies centrally on the eye-socket. Edge detection and analysis is then employed to further refine the location of the pupil center.

A number of geometric filters are also applied as sanity checks. For example the distance between left-eye, outer corner and the right-eye, inner corner should be within 25% of each other; several similar sanity checks are employed.

Figure 6: Classifiers, calculated using *Integral Image* techniques are used to determine the pupil center by scanning across the eye-socket region (yellow circle) as delineated by the eye-corners (blue crosses).

4) Gaze Estimation

To understand how a reasonable estimate of the gaze direction can be achieved with a relatively low-resolution face region - typically 128x128 pixels - several refinements need to be explained.

Firstly it is important to ensure that the face region is normalized for in-plane rotation, given that the face pose is substantially frontal. The correction for in-plane rotation is helped greatly by the face detection algorithm that uses multiple classifier cascades to detect potential face regions. These cascades are rotation sensitive and provide an initial indication of the angular orientation of the face region [23]. If the face pose is not frontal then additional pose detection and compensation may be employed, but this depends very much on the capabilities of embedded CPU and it is more practical to simply disable gaze estimation for non-frontal poses.

A fine-tuning of the initial orientation is achieved, firstly by locating both eye regions and determining the relative locations of each outer and inner pair of eye sockets; due to facial symmetry each eye should have a similar slant angle and if there is a mismatch then both eyes are rotated slightly within the face region to obtain an exact horizontal alignment of the eye-pair as shown in Figure 7 below.

Figure 7: Normalized and aligned eye-pair; note cross-eyed direct-gaze.

5) Sub-Pixel Estimation

As can be seen from Figure 7 there is a relatively coarse refinement of the location of socket center and pupil center; given the limited resolution of 128x128 used for the initial normalization. Fortunately we have several tricks on our side:

(i) in most practical realizations the original, higher resolution face region is available; (ii) as we are operating in real-time at 30 fps a constant stream of low-resolution images are also available; and, (iii) it is possible to further refine the accuracy of detection using bi-cubic interpolation techniques as explained, in the context of mapping applications, by the authors of [24].

Initial experiments indicate that a gaze resolution accuracy of 3-5 cm depending on the distance to monitor screen (40cm - 70cm) is possible without enhancement. This would be sufficient for many purposes, including simple context adjustment within an adaptive gaming system. Achieving pixel level accuracy on a full-HD screen is significantly more challenging and would require approximately two orders of magnitude improvements in accuracy, but might be feasible through a combination of techniques (i), (ii) and (iii) as described above. However it is not clear that this would actually yield a useful method to interface with a CE device due to the eye-gaze "Midas Touch" effect [3].

6) Pre and Post-Processing Filters

Several additional filtering operations are provided to support the main algorithm workflow. Firstly, a pre-processing blink filter is used to determine if an eye is sufficiently open to continue with the remainder of the workflow. This employs a classifier cascade specific to the eye-region of the detected face region and is amenable to hardware embodiments. Where one eye is missing a post-processing filter can estimate its position, although with reduced accuracy. Additional algorithms can provide detailed information on in-plane face rotation - typically to within 2-degree accuracy and also on overall face pose, although with lower resolution.

V. ALGORITHM TESTING & PERFORMANCE

The library size is dependent upon the enabled features, minimum eye-pair size support (which depends on the lens and image resolution), target platform and configuration. The eye-tracker library is very flexible and can be compiled with many different options, based on application requirements and the use case. While typical code size for a configuration optimized for quality is around 860KB, the size range can vary down to 430KB depending upon the target platform constraints and library requirements. For very "low-resource" platforms and with certain trade-offs code size can be further reduced to less than 300KB.

The library has been tested in a variety of different configurations and use-cases varying from simple still images, playback video and real-time video streams. As the real-time video stream use-case is the most challenging we restrict ourselves in this paper to documenting this particular use-case.

A. Video Stream Use-Case (Tracking Mode)

The Eye Tracking mode is specifically designed for video streams. The recommended input image size for 3D use-case is VGA (640x480)/WVGA (800x480). Such a resolution will provide a good distance to subject (over 2m) while keeping the required processing power low.

In a real time video sequence, eye detection needs to be run

as a non-blocking process and so operates in parallel with the video stream and thus does not impact upon the frame rate. It is recommended to have a high priority assigned to the library for good performance. In practice the library needs to be calibrated for each particular platform: CPU, lens, sensor and FOV can all affect performance and accuracy. Tables 1&2 below show sample initialization code for two extremes of use case: one for an embedded 3D imaging application where eye tracker is used to manage a single-user autostereoscopic display requiring high accuracy and performance; and secondly a conventional embedded imaging system with low system resources and less critical performance requirements[2].

TABLE 1
CONFIGURATION FOR 3D EMBEDDED USE CASE

Typical 3D embedded configuration

```
ET_DETECTOR_CFG
 memset(&cfg, 0, sizeof(ET_DETECTOR_CFG));
 cfg.focalLength = camera focal length;
 cfg.structSize = sizeof(ET_DETECTOR_CFG);
 cfg.flags = ET_TRACKING_MODE |
ET_COLOR_FILTER |
ET_FAST_LOCK |
ET_ENABLE_SEMIPROFILE_DETECTION |
ET_ENABLE_PROFILE_DETECTION |
ET_ENABLE_COLOR_TRACKING |
ET_ENABLE_HIGHER_DETAIL_PROCESSING |
ET_ENABLE_HALF_FACE_DETECTION_ONLY_ON_BORDER;
 cfg.framesPerLock = 2;
 cfg.lockPercentageW = 100;
 cfg.lockPercentageH = 100;
 cfg.maxEyePairCount = 1;
 cfg.minEyeDist = 30;/* @ VGA */
 cfg.boostETvsFP = 30;
 cfg.boostETvsSPEED = 100;
 {
 static INT16 lockAngles[] = {0, 30, 0, -30,  ET_ANGLE_END};
 cfg.lockAngles = lockAngles;
}
```

TABLE 2
CONFIGURATION FOR "LOW RESOURCE" EMBEDDED USE CASE

Slow platform embedded configuration

```
ET_DETECTOR_CFG
 memset(&cfg, 0, sizeof(ET_DETECTOR_CFG));
 cfg.focalLength = camera focal length;
 cfg.structSize = sizeof(ET_DETECTOR_CFG);
 cfg.flags = ET_TRACKING_MODE |
ET_COLOR_FILTER |
ET_ENABLE_COLOR_TRACKING;
 cfg.framesPerLock = 5;
 cfg.lockPercentageW = 85;
 cfg.lockPercentageH = 80;
 cfg.maxEyePairCount = 1;
 cfg.minEyeDist = 30;/* @ VGA */
 cfg.boostETvsFP = 30;
 cfg.boostETvsSPEED = -40;
 {
 static INT16 lockAngles[] = {0, 30, 0, -30,   ET_ANGLE_END};
 cfg.lockAngles = lockAngles;
}
```

Measurements have been performed on a range of CPUs, as detailed in tables 3 & 4. These range from low power/low resource configurations operating at 200 MHz to high speed

---

[2] Detailed datasheets and API docs are available from DOC under NDA.

operation at 1-2 GHz. In some tests the slower clock rates can be competitive due to single instruction multiple data (SIMD) enhanced cores in the latest generation of embedded CPU architectures.

Note that additional processing time is needed when no face is detected as the algorithm much continue to search the full image across all possible face-scales. After detection much of the image can be searched in a progressive and "smarter" manner depending on the underlying system application.

B. Test Results for Video Stream

The initial face-lock is performed directly on hi-res image frame and takes up to 22ms for the first detection. This initial face-lock is not included in our measurements as it represents a once-off initialization cost for the algorithm. Further, in a real-world application it would be provided from a false-lock on a lo-res preview image frame with lower time & CPU costs. Once the face is detected, the track time per frame and the overall time (including re-locking) are detailed in Tables 3 & 4, below. It can be seen that only the slowest hardware configuration has difficulty achieving real time 30 frames per second processing rates. For the more challenging 3D use-case there is a clear need for higher-performance CPU. The longer time requirements for "no-face" scenario are due to the algorithm constantly "hunting" for a face.

For testing 10 movies with 0 faces and 40 movies with a single face are used; library has been compiled with full optimizations for ARM CPU, where relevant.

TABLE 3
PERFORMANCE METRICS FOR 3D EMBEDDED USE CASE

Platform	Res.	No. faces	Avg. time (ms)	Avg. track (ms)
ARM ™ Cortex A9 (PandaBoard ™ Rev A3) @ 1 GHz, 32K/32K cache, 1MB L2 cache	VGA	0 faces	30	-
		1 face	5	5
Intel ™ i5-2520M @ 2.5 GHz, 32K/32K cache, 2 x 256 KB L2 cache	VGA	0 faces	5	-
		1 face	1	1
ARM ™ Cortex-A8 (BeagleBoard ™ Rev.4) @ 720 MHz, 16K/16K L1 cache, 256 KB L2 cache	VGA	0 faces	72	-
		1 face	11	11
ARM ™ 1176 @ 1.2 GHz, 32K/32K cache, 32 bit DDR400 (CL3)	VGA	0 faces	53	-
		1 face	10	9
ARM ™ 1176 @ 208 MHz, 16K/16K cache, 16 bit DDR100 (CL3)	VGA	0 faces	276	-
		1 face	48	43

TABLE 4
PERFORMANCE METRICS FOR "LOW RESOURCE" USE CASE

Platform	Res.	No. faces	Avg. time (ms)	Avg. track (ms)
ARM ™ Cortex A9 (PandaBoard ™ Rev A3) @ 1 GHz, 32K/32K cache, 1MB L2 cache	VGA	0 face	1	-
		1 face	4	4
Intel ™ i5-2520M [9] @ 2.5 GHz, 32K/32K cache, 2 x 256 KB L2 cache	VGA	0 faces	1	-
		1 face	1	1
ARM ™ Cortex A8 (BeagleBoard ™ Rev.4) @ 720 MHz, 16K/16K L1, 256 KB L2 cache	VGA	0 face	4	-
		1 face	9	9
ARM ™ 1176 @ 1.2 GHz, 32K/32K cache, 32 bit DDR400 (CL3)	VGA	0 faces	4	-
		1 face	10	7
ARM ™ 1176 @ 208 MHz, 16K/16K cache, 16 bit DDR100 (CL3)	VGA	0 face	19	-
		1 face	47	34

TABLE 5
SPECS & REQUIREMENTS FOR 3D EMBEDDED CONFIGURATION

Item	Specifications & Requirements	Library capability in VGA @ 30fps
Min luminance	20 lux	Depends strongly on image quality in face area (mean luminance/contrast/sharpness/ noise).
Skin and image color	All	The library works on skin tone. Please note that the library is calibrated for best performance (speed, FP%, DR%) for good white balance.
Head roll	[-30º, 30º] locking [-45º, 45º] tracking	For good faces, library locks eyes on [-30, 30], and tracks them on any roll [-179, 180]. If the platform has an orientation sensor and the library is set to work with it, then the locking is [0, 360].
Head pitch	[-15º, 30º] locking [-30º, 65º] tracking	Library locks on the required range. For good tracking performance the library requires acceptable face details and skin color.
Head yaw	[-15º, 15º] locking [-30º, 30º] tracking	Library locks on semi-profile and profile faces [-90, 90].
Min face size (pixels)	30 pixels between eyes	Library locks on required range.
Max face occlusions allowed (%)	< 40%	Library has half detection support and the faces can be covered 40% (left/right/bottom) during tracking (as long as one of the eyes and its eyebrow are not occluded).
Frame rate	>=30 fps	Typically the library works at 30fps, depending on platform. For more details please see results in tables 4 and 5.
Face number	1 face only	The library can work with any face number, but it is calibrated in this scenario for one face only.
Environ-mental lighting	Interior light, lowlight, backlight	The library works on faces illuminated at minimum 20-30 lux. It depends strongly on the image quality in the face area (mean luminance / contrast / sharpness / noise) [7].
Triggering minimum movement (in mm)	2/3 mm (for distances to the subject up to 60/100 cm)	The library should trigger the detection on this variation. Please note that motion blur, image quality and compression can affect variation accuracy. Such precision could only be measured in a controlled robotized setup.
Supported image formats	YUV_420P_FLAT YUV_420SP_FLAT	YUV_420P_FLAT or YUV_420SP_FLAT (for higher speeds), GRAY8, BGR24, RGB24, BGR32, YUV_444, YUYV_422, UYVY_422, YVYU_422, YUV_422P, YUV_422SP.
CPU Loading	Intel-I5 <10% ARM-A9 <20% Only single CPU core is used on eye-tracking.	For more details please see PandaBoard and Intel-I5 results on tables 4 and 5.

VI. SYSTEM OUTPUTS

Here were provide some example output from the system showing its real-time capabilities in tracking the eyes of the user of a CE device or gaming system.
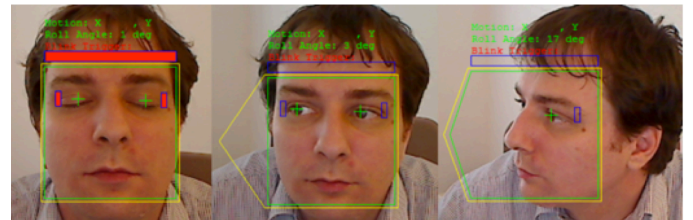

Figure 8: Examples of the system implementing, from left, (i) blink trigger; (ii) demonstrate gaze orientation; (iii) face pose and gaze orientation.
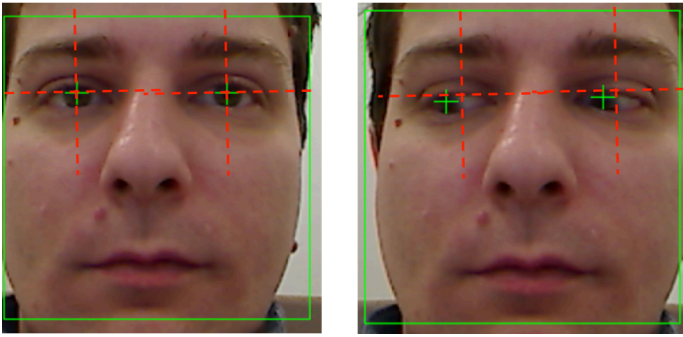
Figure 9: Eye and Pupil tracking - left-hand shows frontal gaze; right-hand shows gaze direction to lower right-hand corner of the screen.
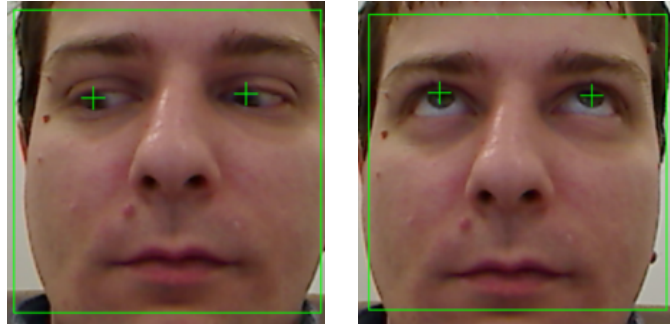

Figure 10: Pupil tracking - note the rectangular bounding box for the face region.

VII. CONCLUSIONS

An advanced real-time eye & gaze-tracking algorithm as outlined in this paper has a wide range of potential applications in gaming systems and CE device applications. The most unique aspect of our system is that it does not require any wearable attachments, supplementary lighting, nor rely on the use of eye-glint phenomena but only employs a single user-facing camera.

For many applications a low-end VGA or WVGA camera can yield very useful results. In one example configuration it is more than capable to determine eye-positions in real-time for use in managing 3D autostereoscopic displays. Even in relatively low-end embedded configurations the algorithm can be tuned to run in real-time at 30 fps although this does imply removing many of the more sophisticated features and filters. As more dedicated imaging hardware becomes available in embedded chipsets we expect that the CPU loading of this algorithm will shortly become negligible. Many aspects of this algorithm draw on well known cascaded classifier techniques and such algorithms are already used widely in CE devices and dedicated hardware IP cores are available. Modifying such a core to also implement eye and gaze tracking is relatively straightforward.

While our initial studies have focused on face regions which are of relatively low resolution at 128x128 pixels, it is clear from related work in [25], [26] coupled with the possibility to use bi-cubic interpolation techniques [24] that it should be feasible to extend these techniques to provide non-intrusive yet accurate determination of eye-gaze on a wide-screen monitor. Extending the system for distances < 1 meter as in [9] will be challenging, but should also be feasible.

Several challenges remain. Perhaps the most significant of these is to find a practical use-case for a gaze-tracking user interface. Past researchers have commented on the "Midas Touch" effect [3] and it is clear from this and other early work in this field that eye-gaze does not lend itself to use as a substitute "mouse" in everyday applications. However a new generation of smart CE devices, particularly mobile, handhelds and tablet devices may inspire designers and engineers to innovate with new approaches to man-machine interactivity.

Other challenges include a need to refine the tracking algorithm to handle occasional saccades of the eye; extending the algorithm to be operable in low-lighting environments - important for many CE applications - and compensating for different locations of the camera relative to the display screen. This last point is particularly challenging for new tablet devices where the user may switch from portrait to landscape modes at a whim.

In the end the further adoption of this system will be driven by new compelling applications and use cases. There is a great potential to enhance user experience through observation of user behaviors and modification of the game environment or CE device function accordingly. It is hoped that this paper will serve to inspire further research and seed new designs and workflow ideas, for both games designers and CE device engineers.

REFERENCES

[1] T E Hutchinson, K P White, W N Martin, K C Reichert, L A Frey, "Human-computer interaction using eye-gaze input." IEEE Transactions On Systems Man And Cybernetics, vol. 19, no. 6, pp1527-1534, 1989.

[2] J. L. Levine, "An eye-controlled computer." IBM Thomas J, Watson Research Center, Research Report RC-8857, Yorktown Heights, N. Y., 1981.

[3] R.J.K., Jacob, "The use of eye movements in human-computer interaction techniques: what you look at is what you get." ACM Transactions on Information Systems, vol. 9, no. 2, pp.152-169, April 1991.

[4] A. T. Duchowski , "A breath-first survey of eye-tracking applications." Behavior Research Methods, Instruments, & Computers, vol. 34, no. 4, pp. 455-470, 2002.

[5] J. G. Wang, and E. Sung, "Study on eye gaze estimation," IEEE Trans. Syst., Man, and Cybernetics - Part B, vol. 32, no. 3, pp. 332-350, Jun. 2002.

[6] S. W. Shih, and J. Liu, "A novel approach to 3-D gaze tracking using stereo cameras," IEEE Trans. Syst., Man, and Cybernetics - Part B, vol.34, no. 1, pp. 234-245, Feb. 2004.

[7] D. H. Yoo, and M. J. Chung, "A novel non-intrusive eye gaze estimation using cross-ratio under large head motion," Computer Vision and Image Understanding, vol. 98, no. 1, pp. 25-51, Apr. 2005.

[8] A. Sugioka, Y. Ebisawa, and M. Ohtani, "Noncontant video-based eye-gaze detection method allowing large head displacements," International Conference on Medicine and Biology Society, pp. 526-528, 1996.

[9] H. C. Lee, D. T. Luong, C. W. Cho, E. C. Lee, and K. R. Park, "Gaze Tracking System at a Distance for Controlling IPTV." IEEE Transactions on Consumer Electronics, vol.56, no.4, pp.2577-2583, November 2010.

[10] Viola P, Jones M.: Robust Real-time object detection. IEEE ICCV Workshop on Statistical and Computational Theories of Vision 2001.

[11] Viola P, Jones M.: Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2001

[12] P. M. Corcoran, E. Steinberg, S. Petrescu, A. Drimbarean, F. Nanu, A. Pososin, P. Bigioi, "Real-time face tracking in a digital image acquisition device", US patent 7,469,055, December 2008.

[13] P. Bigioi, C. Zaharia, P. M. Corcoran, "Advanced hardware real time face detector," Consumer Electronics (ICCE), 2012 IEEE International Conference on , vol., no., pp.528-529, 13-16 Jan. 2012.

[14] P. Corcoran, G. Costache, "Automated sorting of consumer image collections using face and peripheral region image classifiers", IEEE Transactions on Consumer Electronics, vol.51, no. 3, pp. 747- 754, Aug. 2005.

[15] G. Costache, Advances in Automated Image Categorization: Sorting Images using Person Recognition Techniques PhD thesis, NUI, Galway, Jan 2006.

[16] I. Bacivarov, P.M. Corcoran, "Facial expression modeling using component AAM models — Gaming applications," Games Innovations Conference, 2009. ICE-GIC 2009. International IEEE Consumer Electronics Society's , vol., no., pp.1-16, 25-28 Aug. 2009.

[17] I. Bacivarov, M. Ionita, P.M. Corcoran, "Statistical models of appearance for eye tracking and eye-blink detection and measurement," Consumer Electronics, IEEE Transactions on , vol.54, no.3, pp.1312-1320, August 2008.

[18] I. Bacivarov, P.M. Corcoran, M. Ionita "Smart cameras: 2D affine models for determining subject facial expressions," Consumer Electronics, IEEE Transactions on , vol.56, no.2, pp.289-297, May 2010.

[19] Ionita, M.C.; Andorko, I.; Corcoran, P.; , "Enhanced real-time face models from stereo imaging for gaming applications," Games Innovations Conference, 2009. ICE-GIC 2009. International IEEE Consumer Electronics Society's , vol., no., pp.17-24, 25-28 Aug. 2009

[20] Andorko, I.; Corcoran, P.; Bigioi, P.; , "Depth map generation and depth layer separation for information customization in computer gaming," Games Innovations Conference (ICE-GIC), 2010 International IEEE Consumer Electronics Society's , vol., no., pp.1-4, 21-23 Dec. 2010

[21] Andorko, I.; Corcoran, P.; Bigioi, P.; , "A dual image processing pipeline camera with CE applications," Consumer Electronics (ICCE), 2011 IEEE International Conference on , vol., no., pp.737-738, 9-12 Jan. 2011.

[22] W. Chen, T. Sun, X. Yang, L. Wang, "Face detection based on half face-template," Electronic Measurement & Instruments, 2009. ICEMI '09. 9th International Conference on , vol., no., pp.4-54-4-58, 16-19 Aug. 2009

[23] E Steinberg, Y Prilutsky, P Corcoran, P Bigioi, L Blonk, M Gangea, C Vertan' "Detecting orientation of digital images using face detection information," US Patent 7,565,030, July 2009.

[24] R. C. Hardie, K. J. Barnard, E. E. Armstrong, "Joint MAP registration and high-resolution image estimation using a sequence of undersampled images," Image Processing, IEEE Transactions on , vol.6, no.12, pp.1621-1633, Dec 1997.

[25] P. Bigioi, P. M. Corcoran, E. Steinberg, "Image Processing Method and Apparatus", WO/2008/150285, December 2008.

[26] P. Bigioi, P. M. Corcoran, E. Steinberg, "Image Processing Method and Apparatus", US Patent 7,916,971, March 2011.

BIOGRAPHIES

Peter Corcoran received the BAI (Electronic Engineering) and BA (Math's) degrees from Trinity College Dublin in 1984. He continued his studies at TCD and was awarded a Ph.D. in Electronic Engineering for research work in the field of Dielectric Liquids. In 1986 he was appointed to a lectureship in Electronic Engineering at NUI, Galway. He is currently vice-dean in the College of Engineering & Informatics at NUI, Galway. His research interests include embedded systems, home networking, digital imaging, pattern recognition, biometrics, smart-grid and wired and wireless networking technologies.

M.Sc. Eng. **Florin Nanu** is Research Manager at DigitalOptics Corporation. He has a B.Sc. degree in electronic engineering and M.Sc. in Biomedical Engineering from Polytechnic University Bucharest, Romania. Florin has over 9 years of experience in image processing for digital still camera industry. He has over 60 granted and published US patents to date.

M.Sc. Eng. **Stefan Petrescu** is Project Manager at DigitalOptics Corporation. He has a B.Sc. degree in electronic engineering and M.Sc. in Biomedical Engineering from Polytechnic University Bucharest, Romania. Stefan has over 9 years of experience in image processing for digital still camera industry. He has over 40 granted and published US patents to date.

Petronel Bigioi received his B.S. degree in Electronic Engineering from the "Transilvania" University of Brasov, Romania, in 1997. At the same university he received in 1998 M.S. degree in Electronic Design Automation. He received a M.Eng.Sc. degree in Electronic Engineering at National University of Ireland, Galway in 2000. Currently he is General Manager and VP Engineering at DigitalOptics Corporation Europe Ltd.. His research interests include VLSI design, communication, embedded hardware design, smart-imaging and non-linear optical systems.