Towards a Low-Cost Augmented Reality Head-Mounted Display with Real-Time Eye Center Location Capability

Thiago D'Angelo*, Saul Emanuel Delabrida*,
Ricardo A. Rabelo Oliveira* and Antonio A. F. Loureiro†
*Department of Computer Science - Federal University of Ouro Preto (UFOP)- Ouro Preto, Brazil
thiago.d.angelo@hotmail.com - {saul.delabrida,rabelo}@iceb.ufop.br
†Department of Computer Science - Federal University of Minas Gerais (UFMG) - Belo Horizonte, Brazil
loureiro@dcc.ufmg.br

***Abstract*—The real-time detection of eye center location provides valuable information to be used in a wide range of applications such as face alignment, face recognition, human-computer interaction, device control for people with disabilities and users attention detection. Gaze tracking systems is another type of application that uses the eye center location to infer the direction of the users gaze. These systems can be applied to Augmented Reality (AR) Head-Mounted Displays (HMDs) in order to improve the User Experience. This work presents a low-cost AR HMD prototype with real-time eye center location capability. This work also presents an overview of Augmented Reality Head-Mounted Displays and methods for eye center location found in the literature. To assess our AR HMD prototype, we choose a state-of-the-art method for eye center location found in the literature and evaluate its real-time performance in different development boards.**

I. Introduction

Head-Mounted Displays (HMDs) that provide Virtual Reality (VR) and Augmented Reality (AR) capabilities have been emerging in the last years. These technologies sound like the new hot topic for the next years. Users have had the opportunity to enjoy these technologies for entertainment, work tasks, retrieving health and body information, and other daily activities.

For years, engineers and researchers have been developing technologies to build the Head-Mounted Displays for many different purposes. Manufacturers are seeking to design hardware and software that improve the user experience and the user immersion in augmented and virtual worlds, in order to increase the user adoption and transform these devices into products of the mainstream market.

Despite the recent release of many AR and VR HMDs[1], two major problems are hindering the AR HMDs from reaching the mainstream market: the extremely high costs (from US$800 to US$3000 [1], [2]) and the user experience issues such as the vergence-accommodation conflict [3].

[1]Microsoft Hololens, Daqri Smart Helmet, Metavision Meta 2, and Magic Leap are some of the most prominent AR HMD's examples. Oculus Rift, Samsung Gear VR, and Google Cardboard are examples of VR HMDs.

In order to minimize these problems, we have developed a simple AR HMD prototype based on a smartphone and on other low-cost materials, such as a beam splitter, a webcam (pointed to the user's eye, in order to perform the eye center location task) and a development board capable of running digital image processing algorithms.

The smartphone generates the 3-D (stereoscopic) virtual objects images and displays these images into the beam splitter. The beam splitter combines the virtual objects, formed by the smartphone, with the real ones, which are in the real environment. Therefore, the smartphone and the beam splitter are responsible for providing the Augmented Reality visualization to the user.

The prototype is also capable of running an eye center location algorithm, which is used to improve the user experience. The eye center location information will be used to correct the right and left images' position of the virtual object in the stereoscopic view generated by the smartphone application. The correction of the image position using this approach guarantees the successful 3-D visualization of the virtual object independently of the users gaze, improving the user experience.

We implemented the state-of-the-art algorithm proposed by Valenti et al. [4], which performs the eye center location task in low resolution images. We had to adapt that algorithm to work with the HMD setup since the original version did not perform well in this scenario. A sequential version and a parallel version of the algorithm were developed. Both versions of the algorithm were evaluated in different embedded platforms. The results show that our implementation of the algorithm is in accordance with the system requirements, but it still needs some adjustments.

This work also represents the first step in the the development of a low-cost AR HMD with gaze tracking capability.

The main contributions of this paper are:

- To introduce a low-cost AR HMD prototype with eye center location capability.
- To perform an evaluation of different hardware platforms for the eye center location algorithm.

2324-7894/16 $31.00 © 2016 IEEE
DOI 10.1109/SBESC.2016.3

This work is organized as follows: Section II presents an overview of the HMD's classification, features, and user experience issues. Section III presents an overview of the eye center location problem and the methods for finding the eye center location. Section IV introduces the AR HMD prototype with the eye center location capability. Section V describes the experiments performed and the corresponding results. Finally, Section VI presents the conclusions and future work.

II. Augmented Reality Head-Mounted Display Overview

This section presents an overview about Augmented Reality Head-Mounted Displays (ARHMDs). It presents an HMD classification as well as its characteristics and limitations. Furthermore, it discusses user experience issues with ARHMDs systems.

Since the late 1960s, when the first Head-Mounted Display (HMD) was released, researchers and manufacturers have made many attempts to develop a variety of HMDs aimed for Virtual Reality (VR), Augmented Reality (AR) and Wearable Computing applications. HMDs have a wide range of applications in AR, including military, industrial, medical, educational, training, navigation and entertainment applications. The development of an ideal HMD to all situations is extremely hard. Therefore, the identification of the target application requirements and restrictions is crucial to define the technologies to be employed in the development of a specific HMD for a given target application. Some of the main characteristics and aspects that need to be observed when developing an HMD for an application are: type of see-through display and ocularity demanded by the application; optical design to be employed; resolution that the application requires; field of view amplitude; occlusion capability and depth of field requested by the application; latency, parallax effect, distortions and aberrations introduced by the chosen architecture; and matters related to the user experience and acceptance.

A. HMD Classification

Augmented Reality Head-Mounted Displays can be classified according to various parameters. Therefore, we choose to classify HMDs according to three main parameters, as suggested in [5], [6]: the type of see-through display, the type of ocularity and the optical design employed in the HMD development.

See-through Displays

In general, there are two main types of see-through displays in AR: optical see-through and video see-through displays.

1) Optical See-Through Displays: Through an optical system, the real and virtual images are combined using an optical device that is partially transmissive and reflective. The real-world image is fully seen through this optical combiner while the virtual image overlays the real one. The advantages of the optical system see-through include: natural and instantaneous view of the real world, and its structures are usually light and simple [5]. Most of HMDs use an optical see-through display to provide an augmented view for users. Some examples of optical see-through HMDs are Google Glass, Optinvent Ora, Epson Moverio and Microsoft HoloLens [5], [6], [1], [2].

2) Video See-Through Display: When using a video see-through display, the real-world image is first captured by a video camera, then, the captured image and the virtual one are digitally combined. Finally, the combination of the images is displayed to the user through a video display, as an LCD or LED screen. The advantages of the video system in relation to the optical system include a pictorial consistency (precise overlay of the virtual image on the real one) and the availability of countless image processing techniques [5]. In [7] and [8], the authors show HMDs that use video see-through systems to display an augmented view for users. Steve Manns EyeTap HMD [9] can also be considered as a video see-through HMD.

Ocularity

Another criterion used for categorizing HMDs is the ocularity, a measure of the number of eyes needed to see something. There are three types of ocularity: monocular, bi-ocular and binocular. A monocular HMD, as the name suggests, has a single viewing device and is recommended for applications in which stereoscopic view is not required, such as general purpose and daily usage HMDs. Google Glass, Optinvent Ora and EyeTap are examples of monocular HMDs [1], [6], [9], [10]. A biocular HMD provides a single image to both eyes while a binocular HMD has two separate displays with two input channels, one for each eye [5]. A binocular HMD can function as a stereoscopic HMD only when two different image sources are properly provided. For AR, binocular video see-through HMDs are highly recommendable due to their capability of generating stereoscopic images [5]. Epson Moverio and Microsoft HoloLens are examples of binocular HMDs [2], [6].

Optical Design

Regarding the optical designs, HMDs can be divided into two categories: pupil-forming and non-pupil-forming. The architecture that represents the pupil formation has frequently been used since the first HMDs allow a wide field of view, despite presenting greater size and weight. This architecture generates, at least, one intermediate image and the exit pupil is collimated by the eyepiece. In relation to the size of the device that creates the images, the existence of an intermediate image offers a flexible optical design [3], [5].

With the emergence of high resolution displays and small imaging devices, the architectures without pupil formation have become more common [5]. Besides, high resolution and small imaging devices allowed a moderate field of view in a light and compact structure [3], [5]. On the other hand, they have a less flexible optical design and do not generate any intermediate image [5]. Free-form prisms, holographic optical elements and optical waveguide are some of the technologies used in architectures without pupil formation.

Some recent HMDs like Google Glass, Optinvent Ora, Epson Moverio and Microsoft HoloLens use optical designs based on waveguides [1], [2], [5], [6].

B. HMD Characteristics and Limitations

The main characteristics of HMDs (such as image resolution, field of view amplitude, occlusion capability, depth of field and optical design) are intrinsically related to the current limitations of the technology (such as pictorial consistency, vergence-accommodation conflict, latency, parallax effect, distortions, and aberrations). After analyzing the constructive aspects of the HMDs, it is important to define the characteristics that are demanded by the target application and try to minimize the technology limitations related to it.

Resolution

The resolution of a see-through determines the integrity of the virtual image in relation to the real image. The resolution of the whole system is limited by the optical system, by the image generator device, and possibly, by the camera resolution (in the case of video see-through HMD). Regarding the resolution of the virtual image, an ideal HMD will need to have up to 12,000 $\times$ 7,200 pixels to compete with human view, which has an angular resolution of 60 pixels per degree (PPD), considering the human total field of view of 200° (horizontal) per 120° (vertical) [5]. As it is not possible to achieve this value of PPD with the current technologies, it is necessary to make a trade-off between the angular resolution and the amplitude of the field of view to achieve a viable solution. However, as the resolution of the screen tends to keep increasing, this trade-off between angular resolution and amplitude of the field of view must disappear in the future [5]. It is important to note that only the augmented view suffers from limited resolution issues in optical see-through HMDs system while in video see-through HMDs both, the real and augmented views, suffer from resolution limit [3].

Field of View, Depth of Field, Vergence-Accommodation Conflict and Occlusion Capability

In Augmented Reality Head-Mounted Displays, the Field of View (FOV) is an important parameter, which is typically measured in degrees and gives us an idea of how wide is the augmented view that the user sees. Generally, HMDs for AR applications, such as Microsoft HoloLens, require wide and stereoscopic FOV, while HMDs for smart glasses applications, such as Google Glass, can have narrower and monocular FOV [5], [6].

Meanwhile, the depth of field refers to the set of distances in relation to the eye (or to the camera) in which a given object remains focalized into the FOV. In real life, the accommodation of the eye is automatically adjusted to focus on an object according to the distance, and objects outside this depth of field seem to be distorted. On the other hand, the virtual image is usually observed from a fixed distance. This focal distance of virtual image represents a problem because the accommodation and the convergence of the human view system are intrinsically linked. This way, adjusting only one of these aspects and keeping the other fixed might cause ocular fatigue [3], [5], [6]. This problem is also known as vergence-accommodation conflict [3]. It can be minimized by using a new technology known as light-field display, but this technology demands high-cost hardware with high computational resources for rendering the light-field images [11], [12]. Therefore, it is not feasible to use this kind of display with regular embedded systems.

Another desirable characteristic for AR HMDs is the occlusion capability. An HMD with occlusion capability can introduce a virtual object between real objects providing important depth information about the augmented view [5], [13]. The occlusion occurs in such way that the real object in front occludes part of the virtual object, and the virtual object occludes part of the real object behind it. Occlusion capability is more easily achievable with video see-through HMDs than with optical see-through HMDs [5], [13], [14].

HMD Limitations

Some of the main limitations in Augmented Reality Head-Mounted Displays are latency, parallax effect, distortions, and aberrations. These restrictions are related to the chosen optical design, as well as to other hardware issues, and must be minimized. Some of these problems are harder to deal with in an optical see-through design than in a video see-through design, and vice-versa.

C. User Experience in AR HMDs Systems

Opposed to smartphones or smartwatches, an HMD can be a discomfort when the users need to wear it and take it off frequently. A future perspective for HMDs is that they will become light, small and comfortable, in a way that users will be able to use them continuously for a long period during the day for diverse purposes. Nevertheless, the HMD might be useless, or even harmful if the content it shows is irrelevant to the current context of the user. This issue is least noticeable in HMDs for AR, once it is expected to have a wide field of augmented view covering the central field of view of the user. In such situations, an AR system must be aware of the users environment contexts, so it must change its content and presentation style correctly and dynamically according to the context [5], [9].

HMDs used inappropriately might induce undesirable symptoms like headache, shoulder stiffness, nausea, or even more severe harm to the users health. From an ergonomic point of view, HMDs must be as light, small and comfortable as possible during usage. Besides, the look and appearance of HMDs must satisfy the applications requirements. The center of mass in the HMD must be placed as close as possible to the users head. A heavy and well balanced HMD might seem lighter to the user than a light and unbalanced HMD [5], [14].

The matters of safety have equal importance. Because of their nature, AR applications tend to distract the users attention from what happens around him/her due to virtual images overlaying the real environment. AR applications must present minimum information to avoid catastrophic results and at the same time, help during the realization of the target task satisfactorily. When the matter of security is considered a top priority, HMDs with optical see-through systems are recommended over HMDs with video see-through systems. This happens because video see-through HMDs restrict the user's peripheral view. Moreover, in case of failure, the central view of the user would be lost [5], [13], [14].

III. Overview of Eye Center Location Methods

This section presents an overview of the methods for eye center location and shows the importance of the eye center location for Augmented Reality Head-Mounted Displays systems. Furthermore, a classification of eye center location methods is provided. The key features, advantages and disadvantages of each type of method are briefly described.

A. Eye Center Location Methods for AR HMDs

The eye center location is a relevant information for several applications and researches related to computer vision. This information is usually used in applications such as face alignment, face recognition, human-computer interaction, device control for handicapped, user's gaze and attention detection [4]. Moreover, with the arising of several models of Augmented Reality and Virtual Reality Head-Mounted Displays, the eye center location information plays a significant role in gaze tracking and point-of-regard[2] detection systems [3], [15]. These two systems, along with other systems such as Inertial Measurement Units (IMUs), are responsible for improving the virtual image overlay into the real environment, providing the correct registration of the virtual objects according to the user's point-of-view. These systems can also be used to minimize the vergence-accommodation conflict [3], which occurs in most AR and VR devices with stereoscopic displays currently existing on the market.

The use of pattern recognition and computer vision techniques in applications, which involve embedded systems and demand real-time processing using low computational resources, like AR HMD's applications, is a complex task. Therefore, the use of low-resolution images as the input to the computer vision and pattern recognition algorithms is a feasible way to enable real-time processing without increasing the computational resources needs and the hardware costs.

B. Classification of Eye Center Location Methods

Several methods for eye center detection using low-resolution images have been proposed in the literature and they can be grouped into basically three categories: model-based methods, feature-based methods and hybrid methods [4], [15], [16].

[2]Three-dimensional coordinate systems that represent the point in the space in which the user is focused.

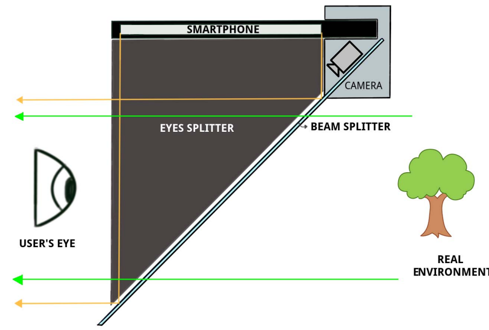

Fig. 1. Design of the AR Display Module

1) Model-based Methods: These methods use global information related to the eyes and face appearance. These approaches often use the classification of a set of attributes or the learning of a given model to estimate the eye location. By using the global appearance, the model-based methods present advantages such as accurate and robust detection of eye position. However, as the success of these methods requires the correct detection of different features or the convergence of a complete model, the importance of eye center location in the global information is often reduced, due to its variability. Thus, the eye center location is usually interpreted just as the central point of the eye model or as the midpoint between the two corners of the eye. Therefore, these methods are not very accurate in situations of sudden motion of the eye center [15].

2) Feature-based Methods: These methods use known characteristics for eyes to detect possible eye center locations through simple local features in the image (like the eye corner, borders and image gradient). Such methods do not require models usage nor any other way of learning. This way, noises or other features overlap do not disturb feature-based methods. Besides, they can achieve very precise results when locating the eye center. However, a lot of times the detected features may be wrong, so feature-based methods are less stable than model-based methods [4], [15].

As a way of combining advantages from both previous methods, researchers have developed some methods named hybrid methods.

C. Hybrid Methods

In these methods, a classifier (trained using a given eye modeling) receives multiple candidates to locate the eye center (these candidates are obtained from the feature-based method). The classifier is responsible for determining which is the correct eye center. This way, hybrid methods can achieve a better precision and greater robustness than previous methods [4], [15].

IV. Low-Cost AR HMD Prototype with Eye Center Location Capability

This section presents the assembled AR HMD prototype and its features. The system is composed of an AR display module working as user interface and an Intelligent Sensor

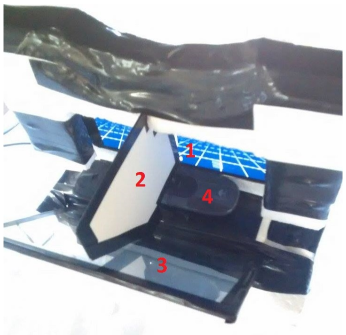

Fig. 2. Components of the AR Display module prototype

Board (ISB) [17] module running a digital image processing algorithm for eye center location. The AR display was assembled with an Android smartphone running a simple application with stereoscopic images, a beam splitter, an eye splitter, and a camera pointed to the user's eye. Figure 1 shows the design of the AR display module. The ISB is composed of a processing unit, which is implanted to the internal structure of a safety helmet. This processing unit is responsible for estimating the user's eye center location through the images captured by the camera in AR display module.

The eye center location information will be used to correct the right and left images position of the virtual object in the stereoscopic view generated by the smartphone application. After obtaining this eye center information, it is possible to infer the users gaze direction for each image of the stereoscopic view, and then correct the binocular disparity between these images. The correction of the binocular disparity using this approach guarantees the successful 3-D visualization of the virtual object independently of the users gaze. Therefore, we can minimize some known problems of AR and VR HMDs, like the vergence-accommodation conflict [3], using a feasible and low-cost solution.

The AR HMD prototype developed in this work can be classified as a binocular optical see-through HMD without pupil formation according to the HMD classification presented in Section II. This architecture is better than the video see-through approaches used in [17] and [18], because it enables the user to see virtual and real objects at the same time without blocking the user's peripheral vision or reducing the user's mobility.

The current version was assembled using pieces of Styrofoam material mounted on top of a safety helmet. As future work, we plan to design and print a more compact and lightweight version of this prototype using a 3-D printer.

We have implemented a state-of-the-art algorithm available in the literature, which was firstly proposed in [16] and later improved in [4], to perform the eye center location task. The eye center location algorithm was implemented in Python using the OPENCV Library. A sequential version and a parallel version of the algorithm were developed.

The prototype was designed using a modular approach.

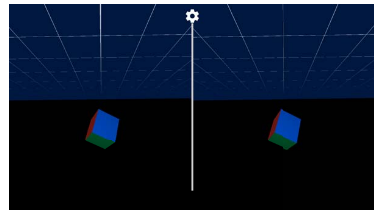

Fig. 3. Stereoscopic view of a virtual object (cube) provided by Google Cardboard demo application

First, we created a display module capable of providing augmented reality information. Then, we designed the ISB module, capable of running digital image processing algorithms.

Next, each module of the prototype is described.

A. AR Display Module

The AR display module is composed of four principal components. Figure 2 shows all these components where each component is identified using the enumeration described in the following.

1) Android smartphone: It is responsible for providing the virtual images of the augmented reality system through an application showing stereoscopic images. We use the demo application of Google Cardboard API [3] to provide this stereoscopic view. Figure 3 shows the application screen, which is vertically divided into two regions. The right region must show the virtual object image from the point of view of the user's right eye, whereas the left region from the point of view of the user's left eye. In this way the user can view a 3-D virtual object overlaid on the real world. In this prototype, we use the Samsung Galaxy Note 3 smartphone.

2) Eyes splitter: It is responsible for ensuring that the right eye sees only the virtual image of the right region and the left eye sees only the virtual image of the left area. Thus, it is responsible for ensuring the correct visualization of stereoscopic images. The eyes splitter is placed between the user's eyes and is aligned with the application screen division.

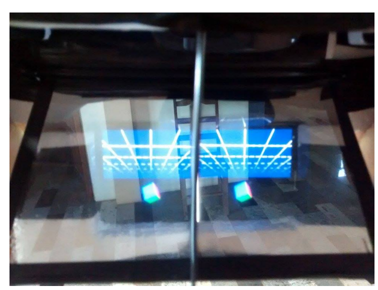

Fig. 4. Augmented reality view using the beam splitter

3) Beam splitter: It is responsible for combining the virtual images provided by the smartphone with the real-world view.

[3]Google Cardboard originally supports the building of 3D VR applications for Android Smartphones. More information at https://developers.google.com/cardboard/

Fig. 5. The ISB module processing unit sewed into the structure of a safety helmet

Figure 1 also shows the beam splitter functioning. The yellow arrows represent the beam of light emitted by the smartphone screen, i.e., the beam of light which forms the virtual image. When this light beam reaches the beam splitter surface, part of it is reflected directly into the user's eyes and the another part pass through the beam splitter. The green arrows represent the beam of light emitted by the real-world environment, i.e., the beam of light which forms the real-world view. When this light beam reaches the beam splitter surface, part of it is reflected directly into the user's eyes and the another part pass through the beam splitter. Therefore, by using the beam splitter, it is possible to overlay the real-world scene with virtual images and objects, enabling augmented reality features in this device. Figure 4 shows the user's point-of-view when wearing the prototype. In Figure 4, it is possible to see at the same time a real-world object (chair) and a virtual object (cube). Moreover, it is worth noting that the user's peripheral vision is not blocked by the prototype structure.

4) Camera: It is responsible for capturing the frontal image of the user's eye and sending these images to the processing unit located inside the safety helmet. We used Logitech C270 webcam in the prototype.

B. ISB Module

The AR HMD prototype uses a camera pointed to users eyes in order to estimate each eye center location using a digital image processing algorithm. This algorithm runs on the processing unit of the ISB module. Each eye center position calculated by the algorithm can be used to estimate the users gaze direction. This information can be used as a feedback data to the smartphone application. We use Bluetooth communication to send the eye center data from the ISB module to the smartphone. Then, the smartphone application will provide the correct stereoscopic view according to users gaze, which will guarantee the proper 3-D visualization.

As shown in Section V-A, different development boards were tested as the processing unit of the ISB module. Figure 5 shows the processing unit attached to the structure of a safety helmet. Together, the processing unit and the safety helmet compose the ISB module. We use a safety helmet to build this prototype because general AR HMDs are used in several industrial applications and this safety helmet is accordance with the industrial environment.

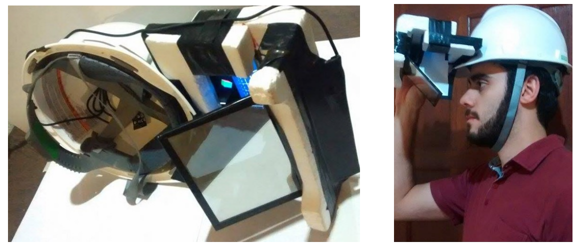

Fig. 6. Complete AR HMD Prototype (left) and an user wearing it (right)

Figure 6 shows the complete AR HMD prototype and a user wearing the prototype. The total prototype cost is less than US$100 (Smartphone cost is not considered), and the greater part of this value is related to the processing unit of the ISB module, which costs approximately half of the total value.

C. Eye Center Location Algorithm

In order to provide the eye center location information, we have implemented the eye center location algorithm proposed in [4], [16]. This algorithm provided high accuracy results and was developed using digital image processing and pattern recognition techniques. Our implementation of the algorithm does not achieve a high accuracy as the original does, but it does produce a satisfactory result.

The eye center location algorithm proposed in [4], [16] is based on invariant isocentric patterns formed by isophotes[4], which can be obtained from the gradient of an eye image in grayscale, and has three variants. The three variants of the algorithm consists of: (i) a basic feature-based method in which the eye center is estimated as the isophotes center; (ii) an intermediate feature-based method that uses the basic method along with Mean Shift (MS) algorithm [19] to improve the method's stability and precision; and (iii) an enhanced hybrid method that uses the basic method with a k-Nearest Neighbor (k-NN) classifier and with the Scale Invariant Feature Transform (SIFT) algorithm [20] to make the system's accuracy and robustness even better.

The results from experimentation and analysis of the method proposed in [4], [16] indicate that the most basic variants (in other words, the basic method and the basic method with Mean Shift algorithm) can achieve good results while keeping a performance that obeys real-time requirements. Meanwhile, the enhanced version of the method, in spite of achieving even better results, follows neither real-time requirements nor low computational cost, which are both demanded by the present work. Therefore, our implementation consists of the intermediate method variant, which uses the basic method along with Mean Shift (MS) algorithm to achieve satisfactory results for the eye center location problem even with low computational cost and real-time restrictions.

[4]Isophotes are curves connecting points of equal intensity in a grayscale image

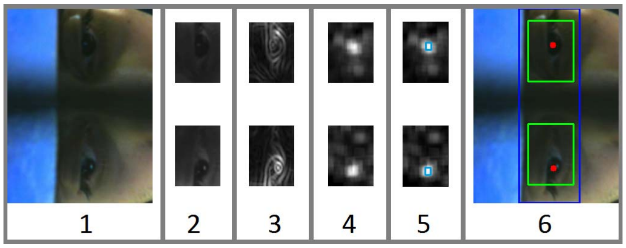

Fig. 7. The algorithm stages

Figure 7 depicts our implementation of the algorithm process. The difference between the sequential and the parallel implementations of the algorithm is related to the processing of each eye in the image. In the sequential version, each eye is processed per time. In the parallel version, both eyes are processed at the same time each one, in a processor core. Next, the six stages of the algorithm are explained.

1) Frame Capture and Grayscale Conversion: In this stage, the image frame of the camera is captured and converted from a three channel RGB image to a single channel grayscale image.

2) Crop Image: In this stage, the grayscale image is cropped into two regions. As the camera and eyes have a fixed position in this head-mounted device, each eye will always appear in the same area of the image. Then, we have two regions of interest in each frame, one for each eye. We chose this strategy to reduce the computational cost of the algorithm, by limiting the eye center search area.

3) Isophote Calculation: In this stage, the algorithm performs the calculation of the isophote curvatures, for each eye, using the method proposed in [16].

4) Center Voting and Centermap Calculation: In this stage, the algorithm performs the center voting mechanism, proposed in [16], to calculate the centermap[5]. The centermap acts like 2-dimensional probability distribution function, where the most voted coordinates have a greater chance of being the real eye center location coordinate. This stage is also performed for each eye. The fourth stage of Figure 7 shows the centermap image. The centermap's brightest regions have a greater probability of being the real eye center location.

5) Mean-Shift Algorithm: In this stage, the MS algorithm is applied over the centermap image of each eye. The MSs sliding window iterates over the centermap image looking for the area with high density of votes. The MSs sliding window is represented by a blue rectangle in the fifth stage of the algorithm in Figure 7.

6) Eye Center Location: In this stage, each eye center location is calculated as being the most voted coordinate, in the centermap, close to the center of MSs sliding window. The sixth stage of Figure 7 shows the estimated eye center location for each eye.

[5]Votes accumulator of the Center Voting mechanism

V. Experiments

This section presents the experiments we conducted to evaluate our implementation of the eye center location algorithm in different development boards. We evaluate the algorithm performance in different development boards. The goal of the experiments is to define the best development board for this AR HMD with eye center location capability. Also, it is essential the matching of the hardware with some of the wearable and the real-time requirements. For instance, the user interface accessibility, lightweight and compact size are some requirements needed for the wearable AR HMD. Meanwhile, to identify the user eye center in real time, the algorithm must provide the center eye information at least twice per second, in case of low eye motion speed, or at least four times per second, in case of high eye motion speed [21]. Therefore, the hardware and software need to have a minimum frame rate between 2 and 4 FPS to provide this requirement. All these requirements were considered in the selection of the hardware. Next, we present the experiments scenarios and results.

A. Development Boards Performance

We chose four embedded platforms to assess the hardware and software performance. Each one is in accordance with the wearable/HMD requirements and can work with a battery as power supply [17]. The following embedded platforms were selected: **Intel Edison** (Dual-threaded Intel Atom CPU at 500 MHz, a 32-bit Intel Quark microcontroller at 100 MHz and 1 GB LPDDR3 RAM, 1 MB cache L1), **Wandboard Quad** (Freescale i.MX6 Quad core processor at 1 GHz and 2GB DDR3 RAM), **Raspberry Pi 3 Model B** (Broadcom Quad-Core BCM2837 64 bits, 1.2 GHz and 1 GB SDRAM), **Cubie board** (ARM Cortex A8 at 1 GHz, 1 GB DDR3).

For each development board, we used the Linux as the operating system. For Intel Edison, we used the Yocto framework to build the operating systems. Also, each one uses the minimal core building, since our intention is to optimize the hardware performance. After the operating systems installation, we build the OpenCV with Python dependencies to execute the algorithm.

The frame per second is the metric defined to evaluate this system. We use a video stream to evaluate the hardware performance. Although the system uses a camera as sensor, we use this approach in the first evaluation to avoid camera interference on the algorithm performance. The equipment used executes at maximum 30 FPS.

Figure 8 shows the results obtained after the evaluation with 99% of confidence. We execute the sequential and parallel versions of the algorithm. The parallel one divides the image into two parts (one region of interest for each eye) and each one is sent to a core of the board. All development boards had a satisfactory frame rate for this application.

All development boards increased the frame rate when compared to the single-core and multi-core versions, except Cubieboard that is a single core device. This is an expected result due to the number of context switch occurences while the algorithm is in execution.

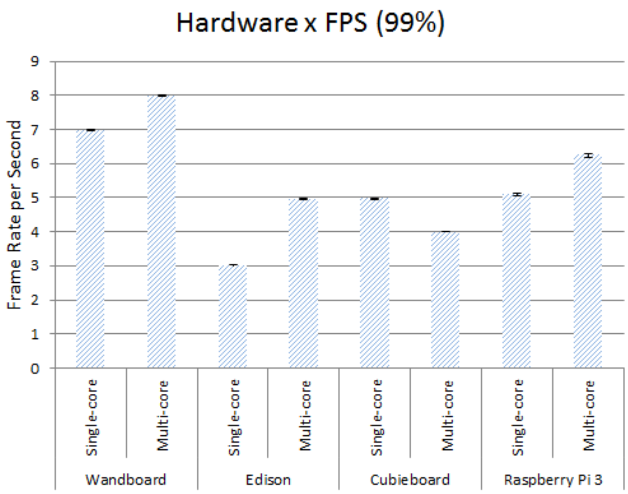

Fig. 8. Hardware Performance

The frame rate difference between single-core and multi-core versions is not significant. Intel Edison has the biggest difference with two frame rate more in the multi-core version than single-core version. The performance evaluation shows that all of the hardware, except Cubieboard, has computing power enough to receive new applications.

VI. Conclusions and Future Work

This paper presented an overview about AR HMD's and eye center location methods. Also, an evaluation of an AR HMD prototype with eye center location capability is presented. The prototype was built with low-cost components, and its construction cost is less than US$100. An augmented reality display was developed using a beam splitter and an Android smartphone with Google Card Board API. A camera pointed to the users eyes is used to estimate each eye center location through a digital image processing algorithm. The algorithm implemented for this function was evaluated in four different development boards. The Intel Edison development board had the best performance, considering both real-time and wearable requirements, while the Cubieboard presented the worst result, considering the same requirements. Therefore, the best development board for this project is the Intel Edison. As future work, we plan to implement the algorithm using other programming languages, such as C or C++, and add new functionalities, which will be evaluated in the development boards. Our intention is to implement a gaze tracking algorithm to detect the users point-of-regard and possible points of interest in the environment. We also want to develop our own augmented reality application, independent of the Google Cardboard API, so that we can do a full integration of all prototype modules.

Acknowledgment

The authors would like to thank UFOP and iMobilis Lab for supporting this project.

References

[1] "Key Challenges to Affordable See Through Wearable Displays: The Missing Link for Mobile AR Mass Deployment," http://optinvent.com/HUD-HMD-benchmark.

[2] "Microsoft Hololens," https://www.microsoft.com/microsoft-hololens/en-us.

[3] G. Kramida, "Resolving the vergence-accommodation conflict in head-mounted displays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 7, pp. 1912–1931, July 2016.

[4] R. Valenti and T. Gevers, "Accurate eye center location through invariant isocentric patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1785–1798, Sept 2012.

[5] K. Kiyokawa, "Head-mounted display technologies for augmented reality," in *Fundamentals of Wearable Computing and Augmented Reality*. CRC Press, 2015, pp. 59–84.

[6] B. Kress, "Optics for smart glasses, smart eyewear, augmented reality, and virtual reality headsets," in *Fundamentals of Wearable Computing and Augmented Reality*. CRC Press, 2015, pp. 85–124.

[7] Y. S. A. Takagi, S. Yamazaki and N. Taniguchi, "Development of a stereo video see-through HMD for AR systems," pp. 68–77, 2000.

[8] K. P. K. A. State and H. Fuchs, "Simulation-based design and rapid prototyping of a parallax-free, orthoscopic video see-through head-mounted display," pp. 28–31, 2005.

[9] S. Mann, "The eyetap principle: Effectively locating the camera inside the eye as an alternative to wearable camera systems," in *Intelligent Image Processing*. John Wiley & Sons, Inc., 2002, pp. 64–102.

[10] ——, "Vision 2.0," *IEEE Spectrum*, vol. 50, no. 3, pp. 42–47, 2013.

[11] E. Ackerman, "4-D light field displays are exactly what virtual reality needs," IEEE Spectrum, Aug 2015. [Online]. Available: http://spectrum.ieee.org/tech-talk/consumer-electronics/gaming/4d-light-field-displays-are-exactly-what-virtual-reality-needs

[12] F.-C. Huang, D. Luebke, and G. Wetzstein, "The light field stereoscope," in *ACM SIGGRAPH 2015 Emerging Technologies*, ser. SIGGRAPH '15. New York, NY, USA: ACM, 2015, pp. 24:1–24:1. [Online]. Available: http://doi.acm.org/10.1145/2782782.2792493

[13] J. P. Rolland and H. Fuchs, "Optical Versus Video See-Through Head Mounted Displays in Medical Visuailization," *Presence*, vol. 9, no. 3, pp. 287–309, 2000.

[14] A. C. M. Billinghurst and G. Lee, "A Survey of Augmented Reality," *Foundations and Trends in Human-Computer Interaction*, vol. 8, no. 2, pp. 73–272, 2014.

[15] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 478–500, March 2010.

[16] R. Valenti and T. Gevers, "Accurate eye center location and tracking using isophote curvature," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.

[17] S. Delabrida, T. D'Angelo, R. A. Oliveira, and A. A. Loureiro, "Building wearables for geology: An operating system approach," *SIGOPS Oper. Syst. Rev.*, vol. 50, no. 1, pp. 31–45, Mar. 2016. [Online]. Available: http://doi.acm.org/10.1145/2903267.2903275

[18] S. E. Delabrida, T. D'Angelo, R. A. R. Oliveira, and A. A. F. Loureiro, "Towards a wearable device for monitoring ecological environments," in *2015 Brazilian Symposium on Computing Systems Engineering (SBESC)*, Nov 2015, pp. 148–153.

[19] G. R. Bradski, "Real time face and object tracking as a component of a perceptual user interface," in *Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on*, Oct 1998, pp. 214–219.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[21] A. Al-Rahayfeh and M. Faezipour, "Enhanced frame rate for real-time eye tracking using circular hough transform," in *Systems, Applications and Technology Conference (LISAT), 2013 IEEE Long Island*, May 2013, pp. 1–6.