Augmented reality system for inventorying

João Real
School of Technology and Management
Polytechnic Institute of Leiria
Leiria, Portugal
jreal@redoute.pt

Luis Marcelino
Research Center for Informatics and Communications
Polytechnic Institute of Leiria
Leiria, Portugal
luis.marcelino@ipleiria.pt

Abstract—**This article presents and describes the implementation of an augmented reality system to assist in the task of warehouse inventory. The proposed system integrates an augmented reality engine (NyARToolkitCS) with a QR Code library. This integration enables users to capture product codes add the product's information to the view of the product itself. This article also presents the challenges for developing this system, the adopted architecture and the tested threshold algorithms. These algorithms were evaluated under real environment conditions.. (Abstract)**

Keywords-component; formatting; style; styling; insert (key words)

I. Introduction

Augmented reality fades the real and virtual worlds boundary, connecting them in such a way that "the manipulation of physical objects has consequences in the virtual world and, similarly, the in-world events have an impact or effect on the physical world" [1]. When we talk about augmented reality, and according to Azuma [2], we refer to a system that integrates virtual elements with real-world elements, creating a mixed environment in real time.

In any warehouse there is a need to control stock. The task of counting the items in stock is a long and arduous because it requires counting all items stored one by one. In an ideal vision of an assisting Augmented Reality system, the user points at a shelf for the system to acquire the image, and the system would verify all the articles and count them. Such a system would make an extremely time consuming and difficult task into something fast and straightforward.

The present study aims to assess the suitability of this technology to facilitate the counting of inventory in a warehouse in a simple and intuitive, so that the system operator can visually get a sense of real time state of the article count from merging with the virtual world. In order to achieve this objective, was carried out fieldwork that culminated in the development of a prototype to provide users a real application where they could experiment with this technology.

This article begins by making a brief background theory in the context of augmented reality, presenting briefly some work done in the area and problems / technical difficulties that arise in using this technology. In a second phase, describes the process of implementing a system of augmented reality for inventorying, showing the steps carried out, starting with the justification for the selection of hardware and software, and presentation of the system architecture, then moving the reasoning algorithms used, ending with the tests performed in a production environment and analyzing the results.

II. Barriers to the adoption of Augmented Reality

Working with a system that provides real-time contextual information such as augmented reality systems can have a significant impact on user performance. This concept of augmented reality was first applied in a business context, by Boeing in 1990 [8]. The project consisted in building a tool to assist in the mechanical connection of electrical conductors that form part of the aircraft electrical system, saving space and reducing costs. Another example of application of this technology in a job application is the application of maintenance and repair techniques being developed by BMW. In this case, augmented reality is used as a means to help employees of BMW Service in its work of high technical requirement. Several examples such as these may be found in the literature.

Despite the undoubted potential use of augmented reality in many professional areas, this application has not had much commercial success, which is mainly attributed to the fact that the technology still has some obstacles to both hardware and software.

A. *Hardware challenges*

Some of the difficulties in the use of augmented reality are related to the Hardware, especially with your fitness and weight equipment, which can affect the mobility of the user [5]. Fortunately, according to Moore's Law of the processors have been getting smaller and smaller and powerful and it lets you interact with a series of sensors. Currently, these systems are much more compact and lighter, but this reduction in size has led to other problems such as the interaction with the user, since, for example, the fingers may be too large for a device such as a small keyboard [3]. One of the solutions presented in this article is to use alternative means such as sensors, and create intelligent algorithms that minimize the number of necessary interactions with the user.

With regard to virtual reality goggles, Knight and Baber [3] reported that the weight of some eye muscle requires some effort, especially in front of the head. The human body can sustain higher loads where the weight is balanced. Thus, the challenge is to get lightweight glasses or have their weight evenly distributed so that they are comfortable for the user. The average field of view of a pair of virtual reality goggles trade is between 16º and 60º, forming a smaller field than that of normal vision is approximately 170º. This decrease in the field of vision may, in some cases, reduce performance in tasks such as object manipulation and navigation. The motor level, requires the user to move over the head to scan the environment. These studies show that the choice of hardware to be used is crucial to optimize the performance of the user, and the ideal would be to minimize the use of input.

B. *Software challenges*

The relative commercial success of AR systems is also related to the software. Until the development of ARToolkit [7] and making it available to the general public, this technology was difficult to use and was not easily accessible to all.

ARToolKit, originally developed by Dr. Hirokazu Kato, is a software library for building augmented reality applications. This library has been used as the basis for most of the libraries of augmented reality. This success is due in part to the large amount of documentation available and also available for test applications.

A major difficulty in developing augmented reality applications is the problem of calculating the actual position of the camera and orientation relative to the marker in real time (viewpoint). For the application can make this calculation it is necessary to know where the user is looking at the real world. The main functionality of ARToolKit is to make this calculation. Other features of this library include: detection of labels formed by simple black squares; ability to use any pattern within the simple black square, and methods to calibrate the camera simple.

Despite the features offered by this library it still has some limitations especially in terms of distance decoding, the great level of sensitivity to different lighting conditions and the difficulties of decoding when the marker partially erupts. However, the ARToolkit's success, gave a new impetus to the development of augmented reality applications, democratizing it and making it accessible to anyone with a learning curve fairly quickly.

III. IMPLEMENTATION OF AN AR INVENTORY SYSTEM

A. *System architecture*

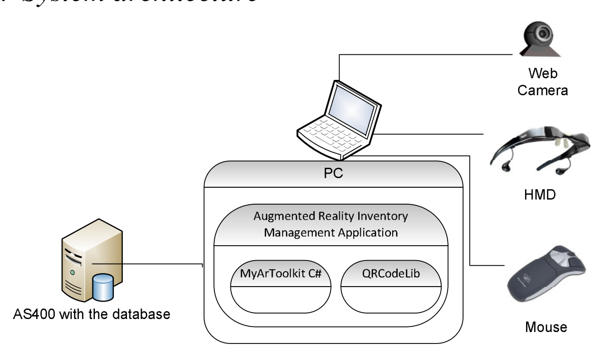

Figure 1 System architecture

The proposed system architecture tries to minimize the user input by integrating the information flow involved in inventory operations. The system interacts with the warehouse backoffice transparently, attempting to make its usability as simple as possible without compromising its performance. Figure 1 presents the system architecture of the developed system.

The system has a typical architecture of an augmented reality system and consists of a camera with a resolution of 640 by 480 (Logitech QuickCam Pro 9000) that allows to acquire images that will be processed, a pair of glasses with a HMD resolution of 640 by 480 (701 Myvu Crystal - Standard Edition - BLACK) that allows the user to display the information returned by the system, a laptop that has a higher computational power than a PDA and can be almost as light and compact as this; an optical mouse that lets you interact with the system, and a DB2 database hosted on a AS400 server which holds information of items to inventory.

The developed software includes two libraries, the NyARToolkitCS-2.4.0 that adds to the system the ability to have augmented reality and the library to decode QRCode QRCodeLib adding to the system the ability to decode QRCode.

Since NyARToolkitCS-2.4.0 is only capable of decoding a limited number of images that have to be preloaded into the system, it became obvious that it would be impractical to create an image for each article and its subsequent inclusion in the system like that would happen if they used the tools provided by this library. It is for this reason that the QRCodeLib was integrated into the system giving you the ability to decode QRCode, which allows the system to provide background information with minimal interaction by the user. I could have used any type of visual code reading faster. The choice of QRCode and not other types of recognition code faster, deals exclusively with libraries decoder open source available and increased amount of documentation available. The diagram of Figure 2 summarizes the operating mode of the system.

As can be seen in the diagram shown in Figure 2, after the user points the camcorder to the marker, the system acquires the image of the real world which is then transformed into an image in black and white. This operation is performed through a thresholding algorithm to be described later. The next step is to find the coordinates of the marker that surrounds the QRCode. This step is accomplished with the aid of augmented reality library NyARToolkitCS-2.4.0. Ideally do not use a marker around the QRCode. The implementation of this operation was initiated, but the algorithm is not fully operational. For this reason we decided to keep the marker around the QRCode.

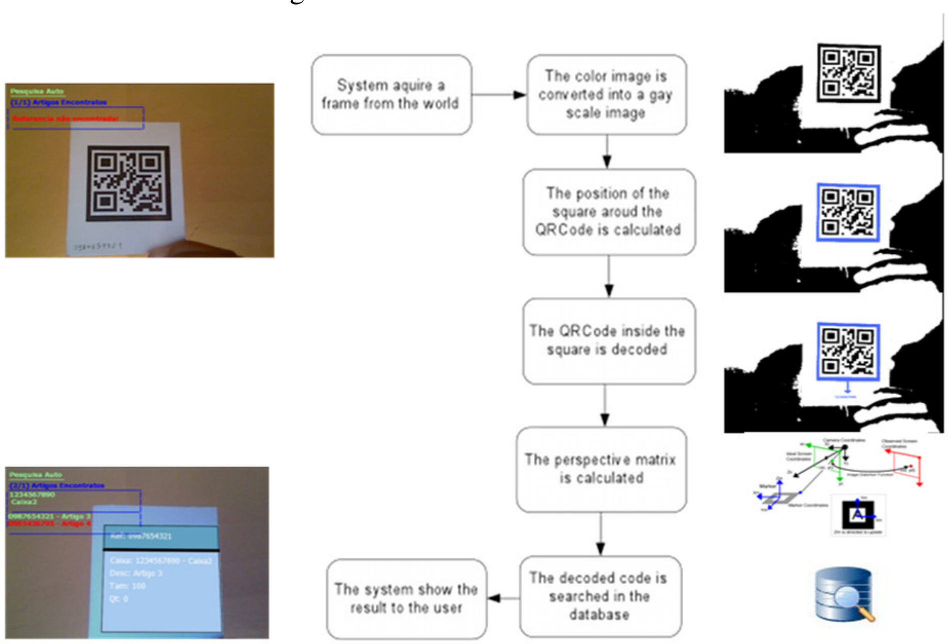

Figure 2 Fig. 1 System operation

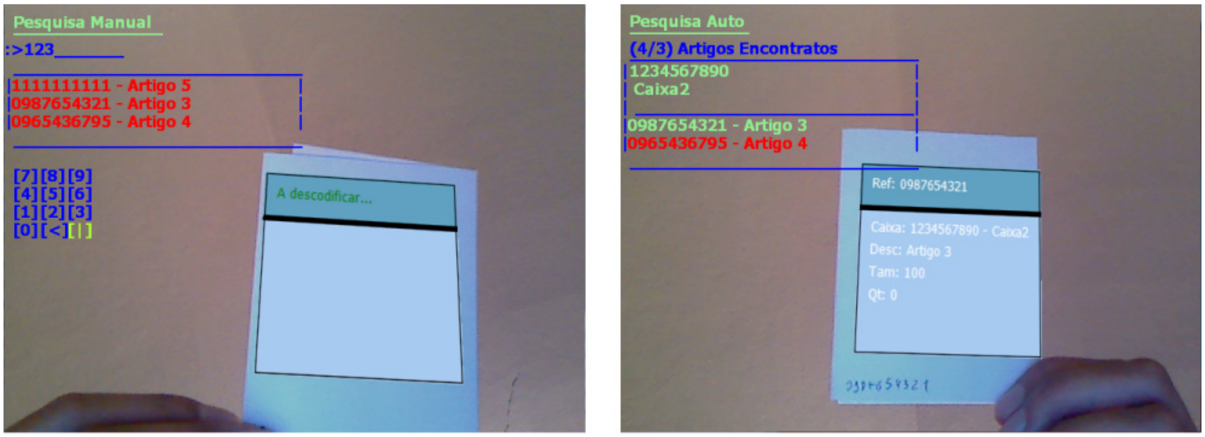

Figure 3 Left the manual mode, automatic mode on the right.

After processing the information of the code uses the library NyARToolkitCS-2.4.0 to calculate the matrix of perspective that will allow drawing the virtual object in the coordinates and the correct angle on the screen. The next step is to find the code obtained in the database to obtain more details about it and record your score.

Finally, the system displays the processed information to the user. This whole process has to be fast enough so that the image does not stand still. To be able to achieve this result they have created multiple threads. The main threads only deals with the presentation of the image to the user without interruption and for each tag that tries to be decoded is created a new thread. The main thread will wait until the threads that decode the codes have finished their work and this happens when the image is processed with the code information decoded.

To navigate through the boxes as counted by the time the user can use the mouse buttons. When the system is unable to decode the codes, can be made a manual search in the database as shown in the Figure 3.

The user interacts with the system using only the mouse buttons and scroll. Only with these buttons you can interact with the system that has a text mode GUI. The graphical user interface is text-mode for this type of interface is considerably lighter the computational level and users are more accustomed to working with this type of presentation of information. The left button lets you choose the option on the left and the right option on the right. The movement of the scroll lets you choose the options above and click the scroll down and lets you choose the option. As the items are numeric identifiers is needed only one keypad to search for articles. This keyboard is drawn on the screen and the characters are selected with the help of mouse.

IV. Thresholding Algorithms

Since the development libraries chosen to work with black and white images and the camera captures images in color, it was necessary to convert these images into binary images in black and white. To achieve this it was necessary to use an algorithm Thresholding, whereby each pixel color is converted to a zero or a one, black or white. This is done due to the operation of a computer that processes information faster that way. This point of the algorithm can be considered the "bottle neck" of the system, because the good performance of it depends directly on the quality of the image provided to the module QRCode decoding. It is here that the old problem of making something with an analog color image into something digital. As can be seen in Figure 4 the existence of a shadow or a light stronger than the ambient light makes it impossible to decode the code.

To determine the optimized threshold value to be used under warehouse lighting conditions we tried to recognize the codes with all possible threshold values (from 0 to 255). This experiment showed that threshold values between 100 and 120 produced better recognition rates. In these environments it was found that the best thresholding value was 110.

However, it was found that when the same image includes two codes or when there were disturbances in the lighting, application had some difficulties in decoding of the codes. The solution was to implement an adaptive algorithm with Thresholding, i.e. for each area of the screen thresholding is applied on the middle of this area the screen.

Figure 4 Problem with Thresholding algorithm

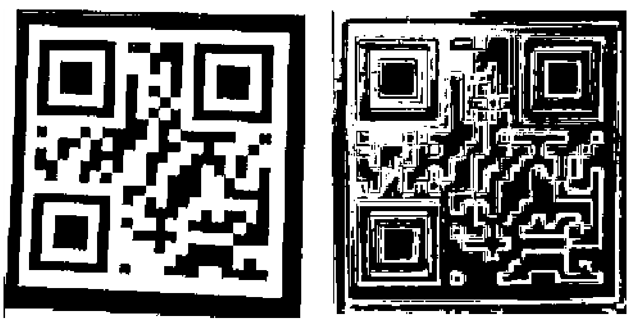

Figure 5 Adaptive Threshold: left Bradley and Roth's algorithms, right Zatepyakin's algorithm

We tested the Integral Image Adaptive Algorithm proposed by Derek Bradley and Gerhard Roth [4] algorithm and its modification from Eugene Zatepyaki [6]. Derek Bradley's algorithm is based on the algorithm used in the library and ARTAG and according to the author is more robust to illumination changes than the original. Eugene Zatepyakin algorithm is an improved version of the thresholding algorithm used in FLARToolkit. Figure 5 shows a comparison of results obtained using the different algorithms described.

The results obtained with 110 Thresholding are not much different from the results of Bradley an Roth's algorithm. Eugene Zatepyakin's algorithm does not preserve white and black pixels, which is essential to decode de QR Code.

V. TESTS ON PRODUCTION ENVIRONMENT

This section presents the results of performance tests for the implemented system. These tests were conducted at the site where the application will be used. The purpose of the tests was to collect data to measure the system performance under real circunstances.

The tasks consisted of pointing the camera at one or more markers and record the average time to decode (TTD) according to the distance from the camera to the code. The considered result was the average time obtained from 100 decoding trials.

In order to measure more effectively the system components that could be optimized, we decided to measure both the time consumed by the QRCode decoding module and the time consumed by both the QRCode decoding module and the AR module. We also measured the time to decode one or more tags simultaneously.

Average time to decode (TTD) in milliseconds after the aquisition of a quality image

Table I lists the average times of the system after decoding to obtain an image with sufficient quality, so it is possible to decode the code on the first try. To obtain these data was necessary to implement a counter in the code that discards all the images that were not successfully decoded on the first try. Note that it was not possible to decode 3 and 4 codes at 50 cm because not all images would fit in the camera's field of view.

The observed decoding times for the QRCode module are only a few milliseconds which is sufficient to maintain a fluid video stream.

Table II shows the number of attempts versus the number of correct decoding images and Table III shows the decoding time, since the system identifies a code until you can decode it. From these results it is possible to conclude that the bottleneck of the system is not the QRCode decoding module itself but is caused by the quality of images provided by capture and thresholding algorithm, since most of the images provided do not allow the decoding of your code.

TABLE I. AVERAGE TIME TO DECODE (TTD) IN MILLISECONDS AFTER THE AQUISITION OF A QUALITY IMAGE

Distance (cm)	TTD 1 code (ms)	TTD 2 Codes (ms)	TTD 3 Codes (ms)	TTD 4 codes (ms)
50	24.98	37.51	--	--
35	17.57	16.33	310.55	734.97
20	23.51	19.55	235.27	941.55

TABLE II. NUMBER OF ATTEMPTS / NUMBER OF CORRECT ANSWERS

Distance (cm)	1 Code	2 Codes	3 Codes	4 Codes
50	562/101	788/100	--	--
35	198/100	437/100	1339/100	2882/38
20	198/101	401/100	1349/100	2674/100

TABLE III. AVERAGE TIME TO DECODE (TTD) IN SECONDS

Maximum Distance	TTD 1 Code (Seconds)	TTD 2 Codes (Seconds)	TTD 3 Codes (Seconds)	TTD 4 Codes (Seconds)
50	2.5	3.3	--	--
35	2.06	2.19	3.29	17.76
20	2.08	2.10	2.83	9.88

The completion of these tests allowed to optimize the algorithm for management of threads so that the system was able to decode more than one label simultaneously, without degrading system performance exponentially. Before this system was constantly optimized to provide images to QRCode decoding algorithm without the prior processing was finished. To resolve this problem, we added a counter that only allows you to create a new thread to decode a new code, if the number of worker threads to decode, is less than the number of codes to decode the image.

VI. TESTS WITH END USERS

For a more complete analysis, it was considered pertinent to conduct a survey to assess the viability of the system in production environment

The sample universe consists of two colleagues from the IT department whose function is related to stock management and four colleagues who are potential users of the application, working in the warehouse. This population represents all users directly connected with the tasks of counting inventory.

All the users responded that they already knew what is the augmented reality and believe that this is a more valid for the completion of inventory adjustment. These results can be explained by the fact that before initiating this study has been made a short presentation on the objectives and the technologies used in order to gain their support in achieving it.

The next section of the survey contains a small booklet, which aims to guide the user to scroll through a corridor and do the inventory adjustment for some of the articles. Then it should answer a number of issues.

After analyzing the results it is concluded that four of the users assigned the highest level and two at low level, the question that asks whether the task was completed successfully. These results prove that the system works and successfully responds to key requirements originally proposed.

When users were asked whether the inventory is faster using the application of augmented reality, than using traditional methods, the responses were divided and one-half of the universe that works in the warehouse and the other half disagreed that the department works Computer agreed. This is explained by the significant difference of experience that both departments have in carrying out inventory. Probably with enough practice warehouse employees would be able to perform the inventory count with the application of augmented reality more quickly than with traditional. Despite the store's employees are quicker to realize the accuracy of inventory with traditional methods than with the application of augmented reality they recognize that this technology can be a useful tool to check inventory.

Regarding the information provided on the screen is its suitable to perform the proposed task and will ease the navigation system using only the mouse buttons, users responded mostly with the highest degree of harmony. This indicates that designates the user interface is adequate and the mouse buttons are sufficient to accomplish this task. In designating the graphical interface and the provision of information, we tried to make as similar as possible with the currently used application, so this is familiar to users.

Users considered that the distance to decode QRCodes must be increased. This is clearly one of the aspects to improve. The ideal distance for decoding labels with 5.5 cm, is 35 to 40 cm, but the distance between camera and the label is about 60 to 70 cm, which requires users to align the label of the camera.

One of the users also identified the system delay as a usability issue. However, the remaining users did not corroborate this opinion.

To evaluate the suitability of the used hardware, users answered five questions:

1) The ergonomics of the hardware (glasses, camera and laptop) is convenient for the specified task;
2) The weight of the glasses is acceptable to perform this task;
3) The weight of the camera is acceptable to perform this task;
4) The weight of the laptop is acceptable to perform this task;
5) The field of vision glasses are sufficient for this task.

The answers for these questions are presented in Table IV.

For the first hardware-related issue users responded mostly agree with the option. The explanation for this result lies mainly in the possibility of using this system you can have your hands free to be able to handle the items. This appealed greatly to the users. Perhaps for this reason did not consider relevant the fact that they carry more hardware such as glasses camera and a laptop on her back. Another fact, which may have been relevant to this outcome, was the fact that we have informed users that a future production system, the hardware would be much lighter and more compact.

All users felt that the weight of the glasses, camera and laptop computer is acceptable.

As for the field of vision glasses most of the users considered the acceptable field of view to the completion of inventory adjustment. This can be explained by the following observations made by one user: "After some time get used to the reduced field of vision glasses.".

In the last question of the survey sought to ascertain whether the overall system performance is adequate for this task. Only one of the users responded affirmatively. This can be explained with some of the comments made by some respondents:

- "Can be used in smaller equipment (eg PDAs or Netbooks)";
- "Reducing the required hardware, or minimize their size.";
- "Improving the viewing angles of the camera and adjust the reading distance of the recognition code.";
- "Should you use a camera with more range, the camera needs to be a small height in relation to the code. There are some problems focusing at the camera. ";
- "While there are such problems, both in scope and focus of the camera, the system can not compete with traditional methods."

With this last question and based on feedback from some of the respondents also could conclude that users were surprised by the positive and enjoyed the experience provided by this order of accuracy of inventory, but for the application to become truly productive and usable production environment would have to be improved especially at the hardware level and distance decoding.

VII. Conclusions and Future Work

This study demonstrated that it is possible to develop an inventory management system based on augmented reality, thus complied with the objectives as originally proposed. As proof of this fact are the tests performed in a production environment and the responses given by potential future users of this system made in the questionnaire.

TABLE IV. USER OPINION ABOUT USED HARDWARE

Question	Doesn't know	Totally disagree	Disagree	Agree	Totally agree
Q1: convenient hardware	0	0	1	4	1
Q2: glasses weight	0	0	1	2	3
Q3: camera weight	0	0	2	1	3
Q4: laptop weight	0	0	2	3	1
Q5: field of view	0	0	2	3	1

Despite a finding that the proposed objectives were achieved, the system still has some things to improve until you can use it in production environment in terms of both hardware and software.

Although the primary focus of this research to be related to the software, the hardware has a very significant influence on the successful implementation of the system. The problems associated with the hardware is easily solved with more funding, because there are several solutions on the market Lightweight compact and ergonomic, which can significantly improve accuracy of inventory experience for the user.

Regarding the software, as already mentioned, the "bottle neck" of the system is in the thresholding algorithm. In the future, we intend to explore all possibilities in converting color images into black and white images. There is still much work to develop this area, and can significantly improve this algorithm. With this improvement would be achieved by increasing the distance required to decode a tag, the speed of decoding and unwillingness to different lighting conditions.

Another aspect to improve the level of the software module is the identification of markers, since in reality a QRCode has a black square around it, as is used in the implemented system. This black square is used for augmented reality library to identify the area that contains the code. We tried to develop an algorithm to detect the area of the screen where the code, but despite the considerable time spent on its development, it was not finished in time to be used in this study.

Although the system is already capable of decoding multiple codes in real time without the image stops, as the number of codes that are decoded simultaneously increases the system undergoes a non-linear degradation in performance. Thus, it is also possible to improve the decoding speed with a thorough review of all code and the optimization of certain operations.

Launching a chronological look over the applications of augmented reality, there are two key moments when such applications have been studied. The first, in the nineties, when the PC and virtual reality goggles emerged and explored the most of your ability, but due to limitations in terms of hardware and software applications developed were not very successful. The second point is this new wave that we are now living and which is due in large part to MyARtoolkit which made it possible for anyone to develop applications of augmented reality and also because development of the hardware that is increasingly powerful, compact and low expensive.

One of the contributions that this study intends to leave this new wave is an augmented reality system capable of use as labels QRCode, opening a huge potential in terms of the areas that can be applied. This system also aims to make its contribution to the level of user interaction, since it was designed only to provide background information that the user needs at any given time, which allows the user to not spend time with little information relevant to the task to be performed.

Another of the contributions that this application was left to show is that actors involved in augmented reality, leading them to realize their potential, leaving in any one of them the will to use it and want to further explore this area. The reception to this technology was such that it was possible to obtain funding for the next two years to develop more applications in this area. At this point is already developing a new project called "Virtual Dressing Room," where the user can virtually try on.

As a final balance of the system developed, we believe that it is not yet ready to be used as a production business application. This is due to the limitations of hardware and software mentioned above. An enterprise application is not extremely fast unlikely to succeed. Probably would have developed an application of entertainment software performance would be more than acceptable.

REFERENCES

[1] Sérgio Oliveira; “Adopting Visual Codes as an Interaction Model”, Adopção de códigos visuais como modelo de interacção, Master’s thesis, Universidade do Minho, 2007.

[2] Ronald Azuma, “A survey of augmented reality”, In Presence: Teleoperators and Virtual Environments, Massachusetts Institute of Technology Press, 6(4):355–385, August 1997.

[3] Chris Baber and James Knight. Human Factors Problems of Wearable Computers, pages 1134 – 1153. Ashgate, England, 2005.

[4] Derek Bradley, Gerhard Roth, “Adaptive thresholding using the integral image”, Journal of graphics, gpu, and game tools, 12(2): pp. 13–21, 2007.

[5] Bonnie Kudrick, Carolina Zingale, Vicki Ahlstrom, “Human factors guidance for the use of handheld, portable, and wearable computing devices”, Technical Report DOT/FAA/CT-05/15, Atlantic City International Airport, NJ: Federal Aviation Administration William J. Hughes Technical Center, 2005.

[6] Eugene. Adaptive thresholding using integral image, August 2009

[7] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In 2nd International Workshop on Augmented Reality (IWAR 99), San Francisco, USA, 1999.

[8] David W. Mizell Adam L. Janin and Thomas P. Caudell. Calibration of head-mounted displays for augmented reality applications. In Proceedings of IEEE VRAIS ’93, pages 246–255, 1993.