A proposal of a multi-view environment for markerless augmented reality

Caio Sacramento de Britto Almeida, Antônio Lopes Apolinário Júnior
Computer Science Department
Federal University of Bahia
Salvador - Bahia - Brazil
{caiosba, apolinario}@dcc.ufba.br

***Abstract*—Augmented reality is a technology which allows 2D and 3D computer graphics to be aligned or registered with scenes of the real-world in real-time. This projection of virtual images requires a reference in the captured real image, which is often achieved by using one or more markers. But, there are situations where using markers can be unsuitable. This work aims to present a multi-view augmented reality environment, composed of augmented reality glasses and two Kinect devices, one for capturing the observer and the other to capture the observed model. The references for virtual images projection are obtained from the information gathered by the Kinects. In this context, calibration and devices positioning in a common coordinates system, and the resolution of possible interferences, are important concerns for the viability of this proposal.**

***Keywords*-augmented reality; kinect; markers; calibration; multi-view**

I. Introduction

Augmented reality has benefited from the progress of multimedia and virtual reality, making feasible new ways of interaction between humans and machines. Differently from virtual reality, which takes the user to the virtual environment, augmented reality keeps the user at his physic environment and takes the virtual environment to the user's space, allowing interaction with the virtual world, in a more natural way, without need of training or adaptation [2]. This interaction often means merging virtual images with images captured from a real environment.

One of the biggest challenges on the augmented reality field is determining, in real time, which virtual image to be displayed, in which position, and how it should be represented. In order to get the illusion of integration between real objects and virtual objects, the generated object must stay aligned with the three-dimensional position and orientation of the real objects [3]. So, camera pose must be estimated.

In order to determine this estimative, in many situations fiducial markers[1] are used (mainly because augmented reality applications usually require real time performance) [4], which are designed in a way that they are easily recognized. Those markers must be positioned on the scene to be captured and achieve good results using few computer resources. However, besides requiring human interference, there are situations where using such markers wouldn't be possible, feasible or comfortable for the observed model. This is the case, for example, of medical applications, on which the model is a patient. Other limitations of using fiducial markers can be listed, for example: occlusion (a virtual image could not be rendered if the marker was not completely visible) and illumination (the intensity of light reflected by the marker could make it difficult to be identified). Less common, there are approaches that don't use fiducial markers [6], [7] and are based, for example, on GPS, gyroscopes, accelerometers, cameras, among others [4], [8]. Such approaches have as advantage not requiring human interference on the scene.

This work proposes an augmented reality multi-view system, of direct view, composed by two Kinects [5] and augmented reality glasses. This system should allow an observer to see, in real time, virtual images merged with real images of the model being observed. No fiducial marker will be used in this implementation. Instead of such markers, a geometric approach will be used based on data captured by each Kinect. This proposed system could be applied, for example, in medical field (real situation, training and education) or in any other situation where a markerless multi-view augmented reality environment could fit. The implementation is ongoing and the preliminary results shows that the proposal is feasible and promising.

The rest of this paper is organized the following way. Section II will present some related works, for better understanding of the state-of-art. The proposal will be discussed on section III, followed by its implementation on section IV. Lastly, on section V, conclusions will be discussed and future works will be indicated.

II. Related works

In order to identify the position to place a virtual image, there are two approaches that can be used by augmented reality applications: the ones that use fiducial markers and the ones that don't use them.

Many of the augmented reality applications use fiducial markers to calculate the camera real position in relation to the marker real position. This is done, for example, on ARBioMed system, where a virtual heart is represented over a marker positioned on an individual's chest and has its pulsation

[1] A fiducial marker is an object positioned on the field of view of an imaging system, for use as a positioning point or reference point.

simulated accordingly to a signal received from a computer [9].

Early vision-based tracking used fiducial markers in prepared environments, but currently, vision-based tracking research is based on markerless approach [20]. The work shown on [10] proposes a tracker algorithm based on 3D models to calculate the distance between camera and objects. Based on this calculation, objects can be positioned on the scene. Although this method is robust in terms of occlusion and luminosity changes, which are weaknesses of fiducial markers, it still has limitations. This is an example of models-based application, which uses 3D geometric data to identify where to render the virtual image on the real scene.

Disadvantages of fiducial markers are listed on [11], for example, the fact that they are invasive, have limited interactivity and need to be printed before using and stored for future use. On the other hand, advantages of markerless augmented reality are presented, for example, parts of the real environment can be used as targets and even informations can be extracted from this environment and used by the augmented reality system.

The system developed by [26] allows any designated object from the environment to be used as a marker. Besides that, this system was designed for low-contrast surfaces (like marks on the user's hands). In order to place a 3D object, the system uses salient points from the environment merged with a local texture, which gives more stability on the detection.

An approach that uses the user's hand as a marker is presented on [12]. On the calibration step, an algorithm detects the edges of the fingers and uses them as a reference pattern, providing a six-degrees of freedom camera positioned on the user's palm, on which virtual objects will be projected. From this point on, the user can move his hand randomly, and the virtual object will move accordingly. This is an example of augmented reality application that uses image processing to identify where to project a virtual object. Another system that uses the user's hand as reference is the one presented on [13], which allows a user to interact with menus and 3D objects in a markerless way. This system is activated from a sequence of movements done by the user's hand, which are captured by a Kinect. Interaction is obtained by selecting one option from a set of available commands, which allow selecting and controlling virtual objects. There are works on which more than one Kinect is used, so new challenges appear, for example, the low quality of data on the overlapping region between two or more Kinects due the interference between them [14]. On the other hand, using multiple Kinects can result on better data quality when each device is responsible for capturing a specific part of an object, or a specific object from the scene. The system proposed by [15] implements an improved version of KinectFusion [24] algorithm to achieve better 3D reconstructions done by many Kinects. Issues like calibration and interference reduction are discussed on this work as well. Other system for 3D reconstruction is also proposed by [14], which uses three Kinects to reconstruct the human body.

A few studies about multiple Kinects calibration were found. The most notable way to calibrate multiple RGB cameras in relation to a common reference point is by processing captured images of a chessboard. The approach on [21] became popular by providing a friendly interface to detect the corners of the chessboard. When multiple Kinects are positioned on a multi-view configuration, the challenge is to develop methods that simultaneously calibrate both RGB sensor and depth sensor using an appropriated calibration pattern. On [19] the effects of using multiple Kinects for motion capture is evaluated. Two approaches are used, one based on checkerboard calibration and one based on time-varying point correspondences.

As noticed, the works that propose alternatives to fiducial markers use algorithms that have limitations of environment, while the works that deal with multi-view environments for 3D reconstruction don't use this information to track an object from the scene. The approach proposed by this work aims to use a multi-view environment for augmented reality based on a 3D reconstructed model which will be compared to a captured model. This comparison will determine where and how to render a virtual image with the real image.

III. Environment

The environment proposed by this work aims to contribute to augmented reality applications on which virtual images should be merged with real images in real time, without using fiducial markers, and considering the angle of view of the observer and the position of the object of interest.

Based on the study of the related works, summarized on section II, the following scope of the initial version of this markerless augmented reality environment was defined, graphically represented on Figure 1.

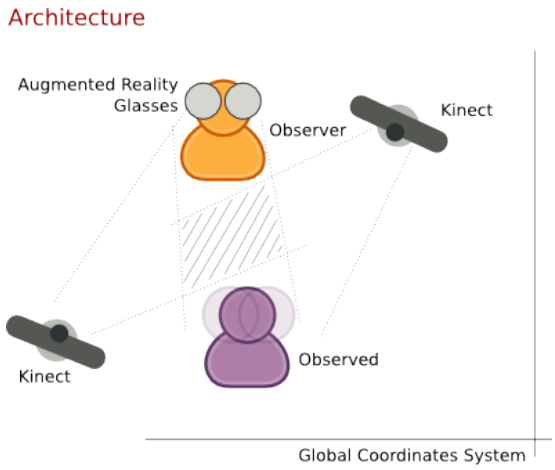

Fig. 1. Global view of the environment

The observer is a user of the system which wear augmented reality glasses and is positioned in front of the observed object (a person, for example). On an application of this environment for medical purposes, the observer would be the specialized doctor, responsible for observing the patient and for analyzing

the combination of the real image (part of the patient body) with the virtual image (from magnetic resonance imaging, for example). The observer can move his head. He wears augmented reality glasses, which have two cameras, whose images captured from the observed object will be merged with the virtual image and displayed on their lens (also two). The movement of the observer is determined by the sensors of the glasses. Based on sensor data, it's possible to determine the variation on glasses orientation and so the movement of the observer's head. This calculation returns values that define movements along longitudinal, transversal and vertical axises. The virtual image must be re-rendered in real time according to the movements.

The observed object (a person, for example) is positioned in front of the observer and doesn't use any fiducial marker. The objective is that the virtual image should be rendered over the observed object. In order to calculate this position, it's necessary to identify the pose of the real object. This is done based on two sensors present on the environment. One captures the observer and the other captures the observed object. They are positioned on the environment and capture data of the observer and the observed object (a sensor for each of them). The sensor that gets information from the observed object contains its model, which will be used to render the virtual image.

Each device present on this multi-view environment (glasses and sensors) has its own coordinates system, but these informations should be converted to the global coordinates system. The initial proposal is that the origin of this global coordinates system should be the observer. Intrinsic and extrinsic parameters of the camera can be used to determine the relationship between the different coordinates systems. The process of determining these parameters is known as the camera calibration problem [16]. Calibration strategies, configuration and implementation of this environment will be described with details on the following section IV.

IV. Implementation

The environment described in section III is being implemented in a GNU/Linux environment, using C++ as language and open libraries and technologies such as OpenCV [23], OpenNI [25] and KinectFusion [24]. The glasses are Vuzix Wrap 920AR [17], which has movement tracker. The sensors are Kinect [5] devices.

The glasses have an API which works only under Microsoft Windows or Macintosh [22]. So, in order to run it under Linux, a driver was developed. The driver is written in C++ and identifies the glasses when attached on the USB port and maps it to an entry under /dev. Data from the glasses are parsed and converted into values of yaw, pitch and roll, which define three degrees of freedom. Raw data from the glasses are formatted as 42 bytes blocks, related to x, y and z coordinates of accelerometer, magnetometer and high and low density gyros (a tuple for each of the four sensors). Raw returned values are in range from -32768 to 32768 while the ones processed by the driver are in range from -180°to 180°(yaw and roll) or in range from -90°to 90°(pitch). The tracker of the glasses is calibrated by moving it in all possible directions. Minimum and maximum values for each of the four sensors are considered at this step. Data from the tracker is used to determine the movement of the observer's head. Other positions are determined by Kinect devices.

Two Kinect devices are used to avoid the usage of fiducial markers. Each Kinect provides data streams of the sensors, from which is possible to obtain the RGB map and the depth map. For this reason, Kinect is known as an RGB-D camera. At the first step, the depth map is not ready for immediate usage, because there is no correspondence between each pixel of both maps. In order to process the depth map, the OpenNI library was chosen. OpenNI provides a calibration feature between RGB map and depth map automatically, by using configuration data stored in Kinect's firmware. This calibration (called *depth registration*) is enough for most of the applications, but better precision can be achieved by using other calibration methods. Initially, this calibration is used on this work. For the next step, the OpenCV library was used to determine the position of the Kinect in relation to a common reference point. The first step was to calibrate the RGB image using a chessboard, in order to get the distortion parameters of the camera. Intrinsic parameters are obtained for each Kinect. This step needs to be done only once for each Kinect. The Kinects are not calibrated simultaneously, this way, interference can be avoided at this point.

The translation t (3x1) and rotation R (3x3) of an object relative to the camera, equals to the transformation of the object to the camera space, given by:

$$v' = R \cdot v + t \tag{1}$$

The reversal of the rotation matrix is simply its transpose, so it's possible to get the transformation of the camera on the space of the chessboard:

$$R^{-1} = R^T \therefore v = R^T \cdot v' - R^T \cdot t \tag{2}$$

Finally, it's possible to get the homogeneous transformation matrix (4x4):

$$M = \begin{bmatrix} R^T & -R^T \cdot t \\ 0 & 1 \end{bmatrix} \tag{3}$$

Since the reference point is the same for both, the position of a Kinect in relation to another can be estimated. Figure 2 shows a chessboard being captured by two Kinects.

The next step was to integrate the glasses with the model captured by the Kinect. As a multi-view environment, there are multiple cameras and they need to communicate with each other. There is a main program which receives data from each device, combines them, and generates the output, which will be sent to the glasses' lens. Each device is controlled by a separated program, written in C++, which gets its data and releases it in a broadcast interface, using UDP sockets. This way, each program can be run on its own machine (virtual or

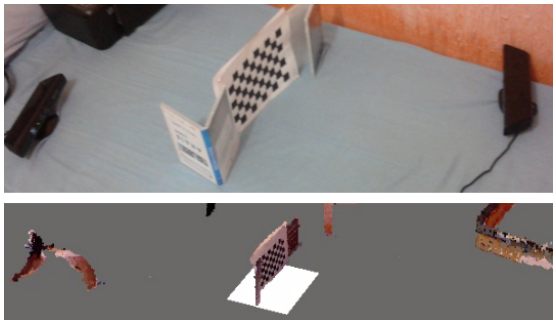

Fig. 2. Two Kinects in action: above, the real configuration (two Kinects capturing the same object) and below, the virtual image combined

physic) on the same network, or even on the same machine. Experiments using those different scenarios are planned but were not done yet.

The current status of implementation is the following. A Kinect captures a model of the observed object and sends its point cloud to a UDP socket. The glasses tracks the movements of the observer and sends the values of yaw, pitch and roll to a UDP socket. The main program reads data from those sockets and then the 3D model captured by the Kinect is controlled by the movement of the observer's head. The next step is to render a virtual image over the observed model based on tracking data from the glasses and the point cloud captured by the Kinects. The alignment of the virtual image with the real one will be model-based, as implemented by [27].

V. Conclusion

This work proposed a multi-view environment for markerless augmented reality to merge virtual images with real images captured from an observed object. The state-of-art study showed that most of the applications still use fiducial markers, while the ones that don't use them implement algorithms not suitable for the presented scenario. So, it was identified a possibility to develop a multi-view environment where the cameras would be used as mechanisms to avoid fiducial markers. The implementation of this environment was initiated, and current results show that it's possible to use more than one Kinect for pose estimation and it's possible to use them in conjunction with augmented reality glasses. After the implementation is done, it will be possible to have results and to evaluate the performance of this environment in comparison to other multi-view environments for markerless augmented reality.

VI. Acknowledgements

This work has been financially supported by FAPESB and CAPES.

References

[1] A. Liu; F. Tendick; K. Cleary and C. Kaufmann, *A Survey of Surgical Simulation: Applications, Technology, and Education*, Presence, vol. 12(6) pp. 599-614. Cambridge: MIT Press, 2003.

[2] R. Tori, C. Kirner and R. Siscouto, *Fundamentos e Tecnologia de Realidade Virtual e Aumentada*, vol. 1.369, 1. ed. Porto Alegre: Sociedade Brasileira de Computação - SBC, 2006.

[3] A. Placitelli and L. Gallo, *Low-Cost Augmented Reality Systems via 3D Point Cloud Sensors, Signal-Image Technology and Internet-Based Systems (SITIS)*, Seventh International Conference on Signal Image Technology & Internet-Based Systems, pp. 188-192, 2011.

[4] R. Azuma, *A survey of augmented reality, Presence: Teleoperations and Virtual Environments*, 6(4), pp. 355-385, 1997.

[5] Kinect for Windows. Available at http://www.microsoft.com/en-us/kinectforwindows. Accessed on April 04th, 2013.

[6] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani and M. Ivkovic, *Augmented reality technologies, systems and applications*, Multimedia Tools and Applications, vol. 51 no. 1 pp. 341–377, 2011.

[7] L. Gallo and M. Ciampi, *Wii Remote-enhanced hand-computer interaction for 3D medical image analysis*, in *Proceedings of International conference on the Current Trends in Information Technology*, ser. CTIT '09, pp. 85-90. Los Alamitos, CA, USA: IEEE Computer Society, 2009.

[8] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier and B. MacIntyre, *Recent advances in augmented reality*, IEEE Computer Graphics and Application, vol. 21(6) pp. 34–47, 2001.

[9] A. Bucioli; E. Lamounier Jr. and A. Cardoso, *A Utilização da Realidade Aumentada no Tratamento e Simulação de Sinais Cardiológicos com BioFeedBack em Tempo Real*, 5º Workshop de Realidade Virtual e Aumentada, 2008.

[10] A. Comport, E. Marchand and F. Chaumette, *A real-time tracker for markerless augmented reality*, The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 36-45, 2003.

[11] J. Dolz, *Markerless Augmented Reality*. Available at http://www.arlab.com/blog/markerless-augmented-reality. Accessed on April 02nd, 2013.

[12] T. Lee and T. Hollerer, *Handy AR: Markerless Inspection of Augmented Reality Objects Using Fingertip Tracking*, 11th IEEE International Symposium on Wearable Computers, pp. 83-90, 2007.

[13] E. Santos, A. Lamounier and A. Cardoso, *Interaction in Augmented Reality Environments Using Kinect*, XIII Symposium on Virtual Reality (SVR), pp. 112-121, 2011.

[14] J. Tong, J. Zhou, L. Liu, Z. Pan and H. Yan, *Scanning 3D Full Human Bodies Using Kinects*, IEEE Transactions on Visualization and Computer Graphics, vol. 18, no. 4, pp. 643-650, 2012 .

[15] K. Bernhard, S. Hauswiesner, G. Reitmayr, M. Steinberger, R. Grasset, L. Gruber, E. Veas, D. Kalkofen, H. Seichter and D. Schmalstieg, *OmniKinect: real-time dense volumetric data acquisition and applications*, in Proceedings of the 18th ACM symposium on Virtual reality software and technology, ser. VRST '12. New York, NY, USA: ACM, pp. 25-32, 2012.

[16] E. Trucco and A. Verri, *Introductory Techniques for 3D Computer Vision*. Prentice-Hall, 1998.

[17] Vuzix 920AR. Available at http://tinyurl.com/wrap920ar. Accessed on March 04th, 2013.

[18] OpenKinect Project. Available at http://openkinect.org/. Accessed on March 04th, 2013.

[19] K. Berger, K. Ruhl, C. Brümmer, Y. Schröder, A. Scholz and M. Magnor, *Markerless motion capture using multiple color-depth sensors*, in Proc. Vision, Modeling and Visualization (VMV), pp. 317-324, 2011.

[20] I. Rabbi and S. Ullah, *A Survey on Augmented Reality Challenges and Tracking*, in Acta Graphica 24, pp. 29-46, 2013.

[21] J. Bouguet, *Camera calibration toolbox*. Available at http://www.vision.caltech.edu/bouguetj/calib_doc. Accessed on March 30th, 2013.

[22] V. Corporation, *Vuzix Eyewear Software Development Kit Version 3.1*. Vuzix Corporation, 2011.

[23] Open Source Computer Vision. Available at http://opencv.org. Accessed on March 30th, 2013.

[24] KinectFusion Project Page. Available at http://research.microsoft.com/en-us/projects/surfacerecon. Accessed on April 02nd, 2013.

[25] OpenNI. Available at http://www.openni.org. Accessed on April 02nd, 2013.

[26] Q. Zhang and M. Lew, *The leiden augmented reality system (LARS)*, in Proceedings of the 12th international conference on Computer Vision, pp. 639-642, 2012.

[27] M. Macedo, A. Apolinário Jr. and A. Souza, *A Robust Real-Time Face Tracking using Head Pose Estimation for a Markerless AR System*, XV Symposium on Virtual and Augmented Reality (SVR), 2013.