J. Vis. Commun. Image R. 25 (2014) 227–237

Contents lists available at SciVerse ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Calibrated depth and color cameras for accurate 3D interaction in a stereoscopic augmented reality environment

Andrea Canessa, Manuela Chessa, Agostino Gibaldi, Silvio P. Sabatini, Fabio Solari *

Department of Informatics, Bioengineering, Robotics and System Engineering-DIBRIS, University of Genoa, Via all'Opera Pia 13, 16145 Genova, Italy

ARTICLE INFO

Article history:
Received 29 July 2012
Available online 5 March 2013

Keywords:
RGB-D cameras
Human–computer interactions
Calibration and data pre-processing
Kinect device
Virtual reality
Spatially variant depth correction
Eyes' tracking
Mixed reality

ABSTRACT

A Human–machine interaction system requires precise information about the user's body position, in order to allow a natural 3D interaction in stereoscopic augmented reality environments, where real and virtual objects should coherently coexist. The diffusion of RGB-D sensors seems to provide an effective solution to such a problem. Nevertheless, the interaction with stereoscopic 3D environments, in particular in peripersonal space, requires a higher degree of precision. To this end, a reliable calibration of such sensors and an accurate estimation of the relative pose of different RGB-D and visualization devices are crucial. Here, robust and straightforward procedures to calibrate a RGB-D camera, to improve the accuracy of its 3D measurements, and to co-register different calibrated devices are proposed. Quantitative measures validate the proposed approach. Moreover, calibrated devices have been used in an augmented reality system, based on a dynamic stereoscopic rendering technique that needs accurate information about the observer's eyes position.

1. Introduction

The recent diffusion of affordable depth and color (RGB-D) cameras (e.g. the Microsoft Kinect) has paved the way for the development of new Computer Vision algorithms and Human–machine interaction systems, which can take advantage from the robust depth estimates coming from such sensors.

Recently, many authors demonstrate the effectiveness of the use of RGB-D sensors in several fields of application, from real-time robotics control applications [1,2] to gesture recognition applications [3]. Specifically, hand and fingers tracking systems are presented in [4,5], thought in these papers a quantitative analysis on the finger position detection is missing. In [6,7], the authors analyze the use of Kinect for postural analysis and for recording human movements to assess musculoskeletal injuries, and they claim the validity of such a device for possible uses in clinical settings. Moreover, an assistive at-home device for passive fall risk assessment, which exploits RGB-D camera devices, is described in [8].

In the field of augmented reality (AR) or mixed reality, several authors present systems capable to accurately fuse real and virtual objects, e.g. [9,10], or to allow a user to act in a telepresence system [11], where a calibrated multi-Kinect setup is used.

In order to achieve an accurate 3D interaction in AR systems it is necessary to obtain the 3D coordinates of the real world with high accuracy. Indeed, the correct perception in an AR scenario is a challenging and still open problem [12,13]. This issue is particularly relevant in the peripersonal space where the distances among the observer and the virtual and real objects are small (below 1.5 meters), thus the stereoscopic cues play an important role [14]. Consequently, it is necessary to accurately render the virtual environment to obtain a coherent perception of both virtual and real objects, and at the same time to obtain precise measures of the positions of the user in the environment. To obtain such precise measures, the RGB-D sensor must be properly calibrated [9,15–17].

In a Human–machine interaction system, as the one proposed in this paper, we need to know the position of specific parts of the user's body, like hands or head, in order to achieve an interaction both among real and virtual objects, and between the user and the virtual environment. Such a 3D interaction can be addressed under different aspects described as follows.

– The detection and the tracking of the *user* acting in the virtual environment, e.g. of his/her *hands* for reaching and grasping tasks. This is a classical problem in virtual reality Human–machine interfaces that has been addressed by several authors. The different solutions are based on dedicated devices [18], or on the tracking of specific markers (e.g. gloves) [19]. Markerless hands tracking is still an open issue, due to the slowness of the solutions and to the difficulty in obtaining precise results, but see [20,21].

* Corresponding author. Fax: +39 010 353 2289.
E-mail address: fabio.solari@unige.it (F. Solari).
URL: http://www.pspc.unige.it (F. Solari).

1047-3203/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2013.02.011

- The detection of the different position of the *user's head*, in order to take into account the observation of the virtual scene from different points of view, i.e. the motion parallax. In [22] the authors present a visualization technique that copes with the motion parallax issue, but the described system is not based on stereoscopic visualization. Head tracking for a stereoscopic display is considered in [18]. Nevertheless, for stereoscopic visualization the tracking of the user's head position (motion parallax) is not sufficient to cope with the misperception issues related to the movements of the observer [23]. It is worth noting that several authors detect and track the position of the observer's head and eyes in order to obtain stereo views to be rendered for parallax barriers autostereoscopic displays [11], without changing the stereo projection technique that is implemented in the virtual reality engine and that does not depend on the specific visualization device.
- The change in position of the *user's eyes* has been addressed in [24,25], where we have presented a novel visualization technique (TD3D) to render stereoscopic 3D virtual stimuli to an observer moving in front of a display, in order to induce to him/her a veridical and natural 3D perception. In these previous works we have taken into account and solved both motion parallax and 3D misperception due to the different position of the observer's eyes.

In this paper, we develop an AR system based on such TD3D visualization technique that requires a high degree of precision, since virtual and real elements interact with each other. Thus, the use of multiple calibrated RGB-D cameras is relevant to cope with the requirements of such a system.

The main contributions of this paper are: (i) to design a calibration procedure that allows us to obtain reliable data from the RGB-D devices, and to use multiple devices in a coherent reference frame; (ii) to develop a robust stereoscopic AR system that implements the previously developed TD3D technique, and where an user experiences a natural and accurate interaction with the virtual objects.

The rest of the paper is organized as follows: in Section 2 an overview of the developed AR system is given, and in Section 3 the TD3D rendering technique that takes into account the eyes' position of the observer is briefly described. The technique adopted to detect the eyes and finger positions of the observer is presented in Section 4. The procedure that we have developed to calibrate the RGB-D cameras and to compute the relative poses among the components of the system (i.e. RGB-D devices and stereoscopic monitor) is detailed in Section 5. Finally, we present a quantitative validation of the calibration in Section 6.1, an application of the developed calibrated AR system in Section 6.2, and the conclusion of our work in Section 7.

2. The augmented reality system

In this section, we describe the system we have developed to implement the rendering technique briefly explained in Section 3. Fig. 1 shows the setup scheme of our system and a snapshot taken during its use. The RGB-D sensor used to develop the proposed AR system is the Microsoft Xbox Kinect, based on an RGB camera and on a depth sensor consisting of an infrared (IR) projector combined with a monochrome camera. Two RGB-D devices are used: one is located on the top of the monitor (Upper Kinect, $\mathbf{K}u$), and it acquires the position of the observer's eyes; the other one (Lateral Kinect, $\mathbf{K}l$) observes the scene from a lateral view, and it is used to measure the position of the hands. Two RGB-D sensors are necessary in order to overcome the limit on the minimum distance measurable by a Kinect. Indeed, as it can be seen in Fig. 1, the user acts in a portion of space near the monitor, at his/her reaching distance. The virtual stimuli are presented between the display and the observer him/herself, thus allowing an interaction with the augmented reality environment in the peripersonal space. Although it is well known that the use of multiple Kinect devices produces interference [26], our configuration minimizes the area where the structured light patterns overlap and thus where the depth cannot be computed. The upper Kinect captures the user's face (see Fig. 4(a)), whereas the lateral Kinect acquires the user's hand and the user's head is out of its field of view (see Fig. 4(b)). For this reason we decided not to implement any technique to reduce interferences.

Since a precise estimate of the eyes' position is necessary to accurately render the stereoscopic views, and the finger position has to be measured with high accuracy to allow a fine interaction with the 3D virtual environment, the RGB and D cameras' reference frames of the two Kinect devices and of the monitor must be registered (see Section 5 for details).

All the software modules have been developed in C++, using Microsoft Visual Studio 10, and the different modules are multithreaded. To render the stereoscopic virtual scene in quad buffer mode we use the Coin3D graphic toolkit,[1] a high level 3D graphic toolkit for developing cross-platform real time 3D visualization. To access the data provided by Microsoft XBox Kinect, we use the Kinect for Windows software development kit (SDK) and the related drivers provided by Microsoft.[2] The processing of the images acquired by Kinect RGB camera is performed through the OpenCV 2.4[3] library.

Both the development and the testing phases have been conducted on a PC equipped with an Intel Core i7 processor, 8 GB of RAM, a Nvidia FX 580 video card with 0.5 GB of RAM, and a LG passive stereo 3D TV 42-inch.

3. A novel rendering technique for dynamic stereoscopic visualization

In previous works [24,25], we have developed a novel rendering technique (TD3D) that allows an observer, which freely moves in front a display, to correctly perceive the position and the shape of virtual objects in a stereoscopic augmented reality environment. In the following, we briefly describe the salient features of such a technique with respect to the AR system proposed in this paper.

In the conventional systems used for presenting stereoscopic 3D stimuli, an observer looking at a stereoscopic screen misperceives the depth, the shape, and the layout of the virtual scene, when his/her eyes are in a different position with respect to the position of the virtual stereo cameras that are used to generate the left and right image pairs projected on the screen [27,23]. If the eyes are in the correct position, then the retinal images, originated by viewing the 3D stereo display and the ones originated by looking at the real scene, are identical. If this constraint is not satisfied, a misperception of the 3D positions of the scene points occurs, see Fig. 2a. The virtual stereo cameras positioned in C_0 (for the sake of simplicity, we omit the superscript, i.e. C_0^L and C_0^R denote the positions of the left and right cameras, respectively) determine the left and right projections t^L and t^R of the target T on the projection plane. The straight lines that link the projections and the observer's eyes are the visual rays. An observer located in the same position of the virtual camera ($O_0 = C_0$) will perceive the target in a position $\widehat{T}_0$ coincident with the true position. Otherwise, an observer located in a different position ($O_i \neq C_0$) will experience a misperception of the location of the target ($\widehat{T}_i \neq T$). This problem is always

[1] www.coin3D.org.
[2] www.microsoft.com/en-us/kinectforwindows/develop/developer-downloads.aspx.
[3] opencv.willowgarage.com.