ORIGINAL PAPER

Design and implementation of eye tracking application for generation of augmented reality content on spatial see through display of remote and virtual control tower (RVT)

Martino C. Moruzzi[1] · Sandhya Santhosh[1] · Marzia Corsi[1] · Sara Bagassi[1] · Francesca De Crescenzio[1]

Abstract

Airports are a fundamental node in the aviation system. The growth of this sector undergoes the evolution of the airports procedures and infrastructures. Air traffic control is considered to be one of the most important activity performed in an airport. This often involves numerous personnel, construction and maintenance costs etc. With the ongoing digitalisation process in various fields, the concept of remote and virtual control towers (RVT) has emerged owing to the innovation in this field. Technologies such as augmented reality (AR) have successfully paved their smooth way to bring in improvements to an RVT. The current work aims to enhance and improve the controller work in an RVT exploiting the adaption of virtual reality and AR systems. In particular, this concept is under investigation within the EU funded SESAR project: RETINA (Resilient Synthetic Vision for Advanced Control Tower Air Navigation Service Provision) and DTT (Digital Technologies for Tower). AR content generation by virtue of various display technologies are studied and, Optical Spatial See-through displays (O SSTDs) have been considered for its own advantages in developing this application. One of the necessary requirements of AR in this configuration is identified as the tracking of the operator's point of view (eye-tracking), to provide virtual content consistent with its real position. Thus, this paper elaborates the design of specific eye-tracking system using Microsoft Kinect V2 for the virtual control tower application. The need to have binocular vision to use AR content is assessed and the interface has been tested with few subjects to evaluate the precision of the measurements detected through the proposed solution.

Keywords Augmented reality · Eye-tracking · Air traffic control · Remote control tower · Microsoft kinect · Spatial see-through displays

1 Introduction

With the advent growth in air traffic, air transportation has spread to small towns and remote locations providing traffic link to major airports. This certainly increased the pressure on air navigation service providers in optimizing the activity of Air traffic management. A completely equipped and operational tower at a small airport servicing only few takeoffs and landings per day is an economic load. Thus, Remote Virtual Tower (RVT) serves as a solution to improve profitability and flexibility [1]. RVT is capable of collecting real time information and conditions of the controlled airport scene, weather conditions and traffic which are presented on a remote tower

scene [2, 3]. The RVT offers digital working ambience as the view of the runway is broadcast remotely via cameras which are located at the physical airport [4]. The possibility of using a remote virtual tower develops a path of innovation in the field.

The classic environment of a control tower is a 360° windowed room, which allows to have a complete view of the airport area and several people working from the room. Innovative technologies such as Augmented Reality (AR) and Virtual Reality (VR) can further reinforce and strengthen these instruments for air traffic control in the virtual control tower. In brief, Virtual Reality immerses a uses in a completely digital world where as Augmented Reality overlays digital information on to the real world [5]. In particular, these technologies can mitigate the visibility problem connected to the weather conditions, the distance and the presence of obstacles. Moreover, they can insert additional information about the aircraft or the weather on the display. Thus, AR and

---

✉ Martino C. Moruzzi
    martinocarlo.moruzz2@unibo.it

[1] Department of Industrial Engineering, Università di Bologna, Via Fontanelle, 40, 47121 Forlì, Italy

Springer

VR systems enhance the quantity and the quality of the information provided to the air traffic control operator (ATCO) in a direct and simple way for the user itself [6] and in a safer and more efficient way for his/her work performances. On one hand, integration of an AR system to improve the visual reproduced information of the tower view has been studied and proved beneficial through various works [7–9]. The general idea to enhance the physical view thanks to image sensors, GPS and radars [10, 11]. On the other hand, VR systems in air traffic control have also been exploited by a number of researchers in Remote and Virtual Tower (RVT).[1] A synthetic enhancement of the virtual view of the airport leads to similar improvements in the performances and in the operations safety to those of the AR on site control towers.

This work focuses on AR systems and is studied as a part of SESAR Joint Undertaking Project RETINA (Resilient Synthetic Vision for Advanced Control Tower Air Navigation Service Provision), which investigates the potential and methods of applying VR/AR technologies to air traffic control.[2] RETINA approach is based on superimposition of synthetic overlays on the real vision, which reports additional information for Air Traffic Controllers. This can be done through the use of AR display modes. There are a number of AR displays, with different maturity levels, extending the ability to see the information on the remote tower depending on their position between the observer and the object [12]: retinal display, head-mounted display, hand-held display and spatial optical see-though display. This study focuses on the last technology: the spatial see-through display (SSTD), in order to best reproduce the control tower windows. In particular, the Optical SSTD (O SSTD) exploits semi-transparent windows, with integrated mirrors, projectors and LCD or LED technologies. AR systems have to deal with different operations and activities of the controller and the transition (in terms of changes of the desired view) from an activity to an other [13]. One of the necessary requirements for AR in this configuration is the presence of a head- or eye-tracking system to calculate the operator's point of view in order to provide virtual content that is coherent with its real position and to allow to specify which part of image the user is looking at. Head tracking is a computer vision-based interface enabling to identify and monitor the movement of the user's head usually with basic camera or face-tracking software and is coupled with eye-tracking systems to enhance human-computer interaction.

Eye detection and tracking is extremely important when dealing with the development of human-computer interaction along with face detection and attentive user interface which remain as a challenging task. Light conditions, individuality of eyes, head pose, degree of eye openness are some of the factors that can affect the robustness of eye tracking systems [14]. The active research of the last decades, exploited different approaches on eye detection and tracking to implement robust and accurate eye tracking systems and many tracking methods are present in literature [15]. This paper demonstrates the development of a specific eye-tracking system for the control tower application, and collaterally, it sets some general reference points for the use of AR in this environment.

In Sect. 2, the classification of reference display technologies is elaborated highlighting the advantages of Optical Spatial See-through displays followed by the virtual content generation techniques. Section 3 elaborates the tools and methods of the concept along with the algorithm development in Sect. 4. Lastly, the tests performed and results obtained are exhibited in Sect. 5.

2 Classification of AR display technologies

The three basic techniques of displaying visuals in AR depend on the way the images are generated and displayed. More precisely, the classification is established on the position in which the AR contents are positioned along the optical path between the observed object and observer's eyes. As specified in the introduction, these are categorised to: Head Mounted Displays (HMD), Hand-Held Displays and Spatial Displays. Hand-held AR is the most common method which uses smartphones and tablets to show AR content [16]. HMDs are devices with one or two small displays in front of the eyes embedded in either glasses or helmet and therefore require the user to wear them. In contrast to the other two, spatial displays are fixed semi-transparent screens installed in the work area rather than being worn. Different approaches like Video See-through, Optical See-through or direct augmentation exist to augment this environment. However, in order for spatial displays to function properly, it is necessary to maintain a visual perspective aligned with the head or, rather, with the user's eyes. Therefore, at any time, the application that produces the overlap of AR content must know where the user is and where it is staring at. In fact, the content displayed on the screen is determined from the point of view of the observer, which is usually calculated by monitoring its position and orientation of the head or directly of the eyes [17]. The present study chose the Spatial display systems for its advantages as mentioned in the following sub-section.

2.1 Spatial see-through displays: advantages and content generation

The principal advantage of SSTD tools is the low bulk of the system itself for the users and absence of limits caused due

---

to the battery charge times as compared to HMDs. Moreover, it could be used by more people at same time [18, 19]. An important issue and challenge of this system is the mix between the real and the virtual image as the user/observer has the freedom of movement with respect to the device, which is not the case with the HMD systems. In particular, this issue could create a missed collimation between real and virtual images affecting visual performance or creating parallax between two images affecting position and as focus distance. Thus, the superposition at the same depth between the real image and the virtual one is a challenge of this system. Depending on the target quality (and on the application) of the system there are two different ways to generate virtual images:

- biocular disparity (same image for both the eyes, as in a traditional screen);
- binocular disparity (two different images for the two eyes, so a stereoscopic view).

The human perception quality depends on the type of disparity, but not only as described by Nagata [20]. There are various parameters that influence the perception quality such as: sources of information, occlusion, interposition, relative size and density. The reliability of these cues depend on the distance of the observer. The sensibility of the binocular disparity has an important value up to 20 m, although researchers are debating on its importance in the so called vista space (over 30 m) [21, 22]. Moreover, in an AR environment, conflicting sources of information could lead to uncomfortable situations. Therefore, for the purpose of our application i.e, to have AR content displayed on a SSTD, a binocular disparity rather than a biocular is preferred. In order to provide the user with the precision of a binocular disparity, the SSTD system has to track the head and eyes of the user in order to use them as virtual machine points for generation of synthetic contents. Head-tracking system generally calculate the area between the eyebrows, called glabella and returns only one point as a result. However, in certain situations the availability of the stereoscopic cue of visual perception may become important and therefore an eye-tracking technique allows to generate more refined AR contents with better system performance. Any eye-tracking technology uses a software with an implemented algorithm and sensors. Generally, the actual technologies track the eyes position from the coordinates of nose. In this study, an algorithm for eye-tracking with Microsoft Kinect ® device has been developed, which tracks the user eyes position directly allowing to provide more accurate data to the user.

3 Tools and methods

3.1 Microsoft kinect V2

In order to obtain the real time measurement of the position of an operator eyes, we used the Microsoft Kinect V2® (or Microsoft Kinect v2®[3]), Fig. 1, a motion sensing input device by Microsoft. This sensor hardware allows with a dedicated software to obtain and monitor the spatial coordinate of the various points of the user face. This information can be sent to a communication interface and then can be elaborated in order to generate AR contents. The software, named *KET - Kinect HD Eye Tracker*, is written through the Microsoft Visual Studio Integrated Development Environment (IDE), which supports multi-language programming (C#, Visual Basic, C++) and multi-platform hardware (PC, mobile, console). It is based on the.NET framework and can be integrated by numerous ad-hoc extensions for specific tasks. The programming language to develop this software is C# and markup XAML in order to produce a Window Presentation Foundation (WPF) application. The sensor calculates the distance between the camera and the object through a time of flight camera (TOF-camera) estimating the time taken by a light pulse to travel through the camera-object-camera path. Kinect® uses an infrared camera with a resolution equal to 512×424 pixels next to an infrared projector, which generates a reference pattern for the camera. Moreover, in order to acquire video stream the device has a RGB standard video-camera with resolution of 1080p at 30 fps joined by four microphones able to capture the position of sound sources and to delete echos or noise. Finally, thanks to the dedicated API, Kinect® is able to simultaneously distinguish up to six users, their body parts and gestures.

3.2 Method

As previously mentioned, the Kinect ® device is able to distinguish different body parts of the user and in particular up to 1347 reference points on user face in a 3D space. Between these reference points there are those of the eyes contour. The points are enumerated and viewable with the possibility of tracing their spatial coordinates. Among these references, there are also various points of the eye contour, but not the eye center itself. In order to obtain the center of any object, the midpoint between two opposite edges can be found, likewise for the observer eyes, average distance between the two opposite edges has been considered as the eye centre. The next step was to identify these edges for which, two possible sets of points were realised:

- the outer and inner corners of the eye (red in Fig. 2);

---

[3] simply called Kinect® in the article.

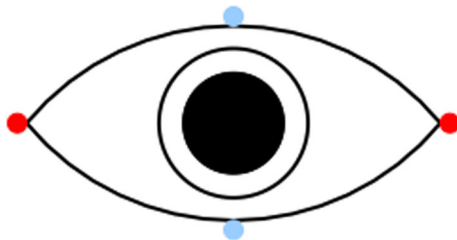**Fig. 1** The Microsoft Kinect® installed at V-Lab Unibo

Fig. 2 The two points set options: red the outer and inner corners, blue the bottom and top borders

- the top and bottom borders of the eye (blue in Fig. 2).

The outer and inner corners are discarded as it was noticed that these points are often shadowed by some face orientation. The bottom and top borders are more visible even with significant rotation of the head, as represented in Fig. 3.

In general, the human eye is not perfectly symmetric and so the "centre" can be estimated using the ocular angles of the real bottom and top borders as shown in Fig. 4. Even if there is a difference of few millimetres between the border and the angular method, this distance can be considered as negligible when compared with the operative one (order of meters) of the Kinect®. When creating AR content this discrepancy can be taken in to account to refine generated virtual images. In this study, the algorithm is developed and tested for both methods.

The reference points are now two: the centre's of each eye and in order to create a 3D vector, three points are required. This vector represents the direction of user gaze, in particular for the upward gaze and so to provide the user with coherent AR contents with the gaze direction (both for the position of the contents and for the type of the information). Moreover, the knowledge of the gaze direction is important to understand which areas of the screen could be left empty in order to reduce the computational cost or to leave these areas free for other users. Theoretically, the third point could be arbitrarily chosen. Although, for this study, a point is chosen that can be easily identified and has a clear exposition with respect to to the sensor. The best choice is the tip of the nose, that is already centred between the eyes.

From a practical point of view the following work is divided in three parts:

- the C# algorithm for the Kinect® interaction, the reading of data streams and their elaboration;
- the XAML interface in order to show the data;
- the Socket in order to communicate the data to the AR system.

4 The algorithm development

The source code is developed in C# and open source.[4] The code block diagram and the logic path is shown in Fig. 5. The following section will briefly describe these blocks.

From Kinect® the software acquires the face image and the reference points on the user face. Then the algorithm extracts the coordinates of the bottom and top borders points and of the nose tip point. The centre of the eye is calculated minimizing the number of operations for the three coordinates (x, y and z). The importance of the reduction of the operations number is due to the high refresh frequency of the image. Moreover, the software calculates the distance between the eyes (the ocular distance), which will be useful on one hand in the generation of stereoscopic contents, and on the other hand for the validation of the algorithm as presented in Sect. 5. On the tip nose coordinates no elaboration is required. Once the coordinates of the three points are obtained and so the components of the vector, the real time information is shown on the interface, called Kinect HD Eye Tracker (KET) in Fig. 6, and sent through the Socket to the AR device.

In order to validate the software results we performed several experimental tests, described in Sect. 5.

5 Tests and results

5.1 Experimental setup

In order to avoid any interference caused by different reference systems between the Kinect® system and our, it was preferred to compare an absolute distance, as the ocular one previously calculated.

---

[4] https://amslaurea.unibo.it/19049/.

Fig. 3 Different rotations and inclinations angles of the face can hide the reference borders and corners. The former are more visible than the latter (Image extracted from visagetechnologies.com and modified by the authors)
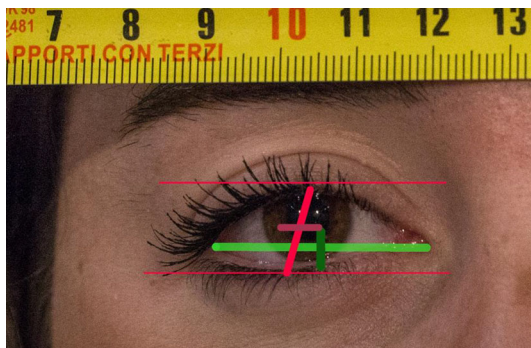

Fig. 4 Difference between the eye centre calculated with the ocular angles (green) and the bottom and top borders (red)

The distance between the top border of the eyes is measured, Fig. 7, in order to obtain the ocular distance. The two different measurements (the ocular distance from the software and from the experimental measurements) are then compared and the difference between the two distances is an absolute value. Greater distance of the user from the Kinect® admits greater differences between the measurements. Thus, relative errors are calculated, normalizing the difference with the distance between the nose tip (Nose Tip Norm NTN) of the user and the device. This distance is the module of the nose tip vector composed by the three spatial coordinates of the nose tip reference point previously extracted by the algorithm. The results are reported in terms

of absolute and relative errors. The absolute error E_{abs} is the difference between the measured distance and the distance calculated by the algorithm. The relative error is calculated in two different ways in order to take into account the different factors that may influence it:

- in relation to the NTN

$$E_{NTN} = \frac{E_{abs}}{NTN} \times 100 \tag{1}$$

- in relation to the measured ocular distance measured by the ruler $D_{measured}$

$$E_{ruler} = \frac{E_{abs}}{D_{measured}} \times 100 \tag{2}$$

The tests are preformed on four volunteers and six different positions, Fig. 8, and two sets of measurements are reported:

- in the first set the ocular distance is calculated through the border method;
- in the second set the distance is calculated through the angular method.

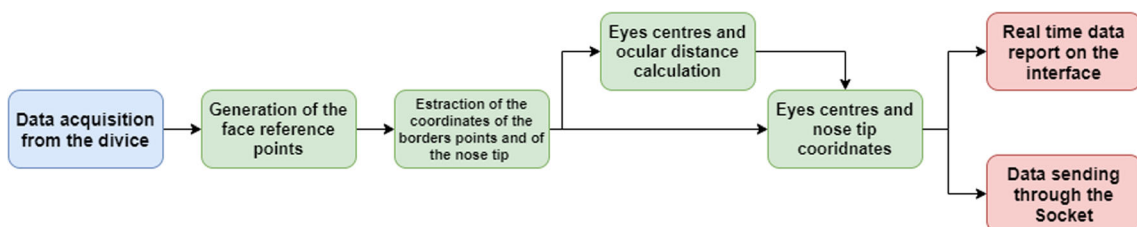Hence, the validation includes a total of 48 measures.

Fig. 5 The block diagram of the developed algorithm. Inputs are reported in blue and outputs in red
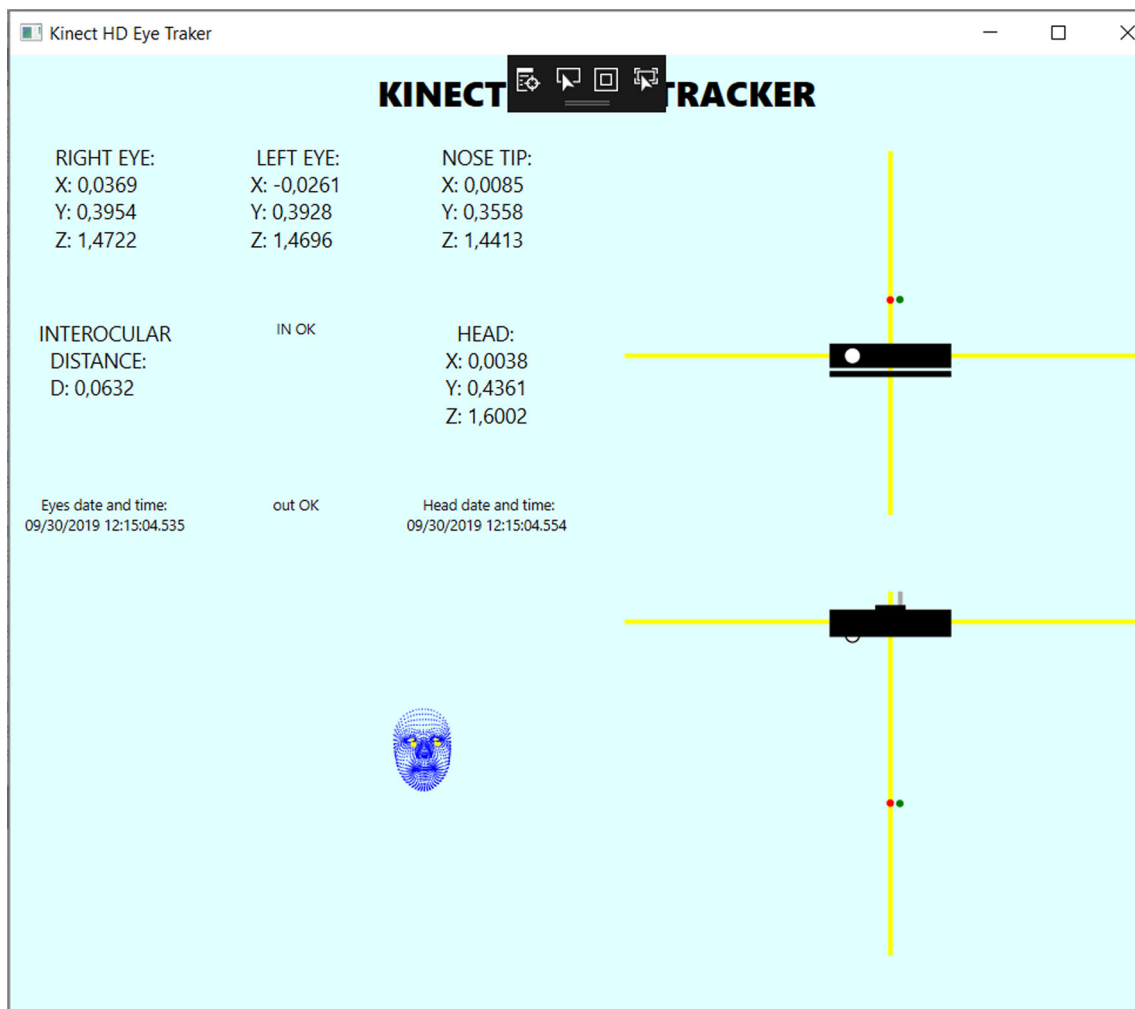
Fig. 6 The Kinect HD Eye Tracker (KET) interface. The centre eyes and nose tip coordinates are reported, as well as the ocular distance, the eyes orthogonal representation respect to the Kinect®, the 3D reconstruction of the user's face and the head position. All the distances are in metres
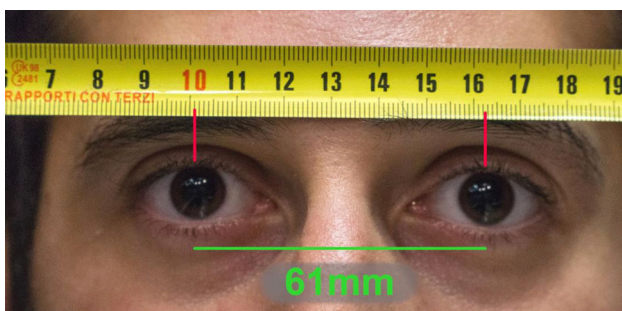

Fig. 7 Measurement of the eyes distance from the top borders

5.2 Results

According to the results, the algorithm shows a better precision with an estimation based on the eye border rather than on the ocular angle, as reported in Fig. 9. In fact, the errors are below the 6% for the former method and below the 11% for the latter respectively. Medium results for each volunteer are reported in Tab. 1. Furthermore, when comparing the error to the user distance, the relative error is below the 0.2% for both methods.

From Fig. 10, we can apparently conclude that there is an increase in the precision for higher NTN between the user and the Kinect®. This datum is caused by the operation of the TOF-camera of the Kinect®. At lower NTN the IR beam can not be opened wide enough to differentiate the reference points on the user's face, while for a higher distance it does not have any superposition problem. As reported in Table 2 for the six test positions in Fig. 8, the average error with respect to the user's distance, for the border method increases closer to the Kinect®, but is more user-influenced than distance-influenced, still maintaining low relative errors. Furthermore, the difference between the distance measured and calculated by the algorithm do not seem to be influenced by a lateral position. For the angular method, the errors are

Fig. 8 Test area with six different positions for the measurements
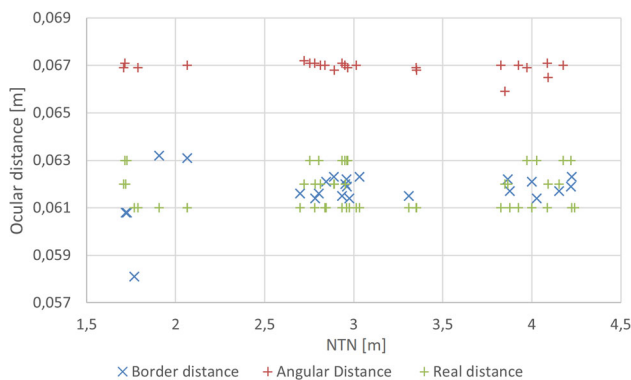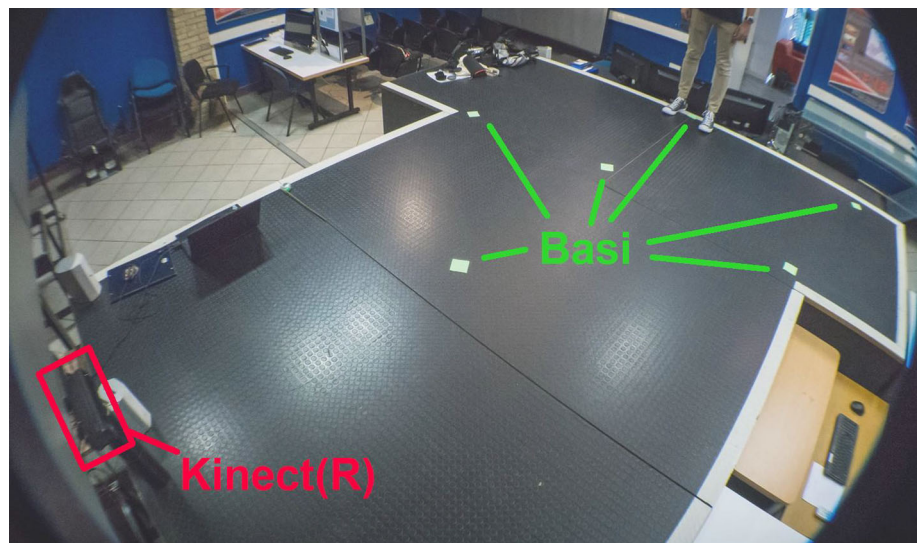

Fig. 9 The ocular distance obtained by the software (blue from the eye border and red from the eye angles) and the measured distance (green) as a function of the NTN from the Kinect®
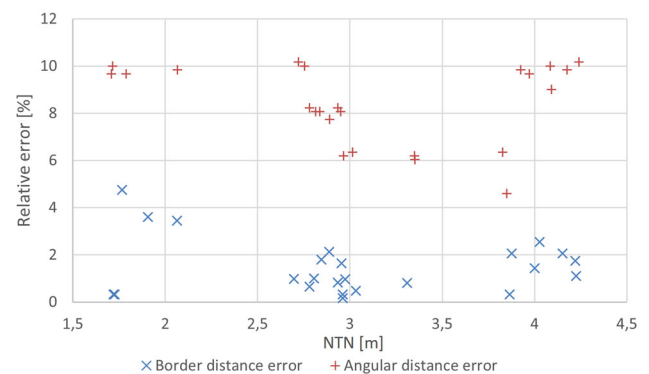

Fig. 10 The software errors for the ocular distance (blue for that obtained from the eye border and red from the eye angles) as a function of the NTN from the Kinect®

higher, therefore the influence of the user's distance is not appreciable.

6 Conclusions

This paper presents the work on the design of a eye-tracking system for overlaying Augmented Reality content on Remote and Virtual Control Tower (RVT). Primarily, the study focuses on head-tracking and eye-tracking technologies in superimposing such content on a Optical Spatial See-through Display (O SSTD) and comprehends the advantages of considering eye-tracking using Microsoft Kinect. For which, a significant algorithm is developed and tested. In this process, the need to have a binocular vision to provide AR contents inside the air traffic control towers was

Table 1 The average results for each volunteer in terms of absolute error E_{abs} in (m) and relative error (%) on the user distance E_{NTN} and on the ocular distance itself E_{ruler}

Volunteer	Borders Error			Angular Error		
	E_{abs}	E_{NTN}	E_{ruler}	E_{abs}	E_{NTN}	E_{ruler}
1	0.0011	0.0060	1.7760	9.8907	0.0477	0.1954
2	0.0004	0.0050	0.6989	8.0645	0.0190	0.1790
3	0.0015	0.0041	2.3519	5.9524	0.0552	0.1334
4	0.0038	0.0038	2.4317	9.7541	0.0574	0.2036
Average	0.0017	0.0047	1.8146	8.41543	0.0448	0.1778

In the last row the final average errors are reported

Table 2 The relative average errors (%) for the four volunteers on the six positions

Base	Position coordinates	Absolute distance	Border E_{ruler}	Angular E_{ruler}
1	(−1.2, 1.4)	1.844	1.4829	7.6643
2	(1.2, 1.4)	1.844	1.1742	7.7034
3	(−0.08, 2.48)	2.481	1.3719	7.6649
4	(−0.1, 1.4)	1.404	3.4471	7.3409
5	(0.0, 3.9)	3.900	1.4533	7.7872
6	(0.7, 3.85)	3.913	1.9509	6.5377

The coordinates and distance in [m] are referred to Fig 8 (the Kinect®is placed at the centre of the reference system)

assessed. It emerged by previous studies, that the visual sensibility given by the binocular disparity is also important at distances greater than 30 ms, especially in those situations where other information is lacking.

The software and the algorithm were therefore developed in C# language and XAML markup, capable of interfacing with the Kinect®device to measure the position of the user's eyes and nose tip in a precise and non-invasive way, obtaining their coordinates with respect to the sensor and adapting them in order to be sent via Sockets to the AR content generation system.

Experimental tests were performed in order to evaluate the goodness of the measurements detected, by measuring the ocular distance of four volunteers, both through the developed software and with a physical meter. The values obtained were highly compatible, with a relative measurement error always less than 6%, and, if the measurement distance is also taken into account even less than 0.2%.

The tests also showed that the measurements are less precise being in the vicinity of the device (Niose Tip Norm less than 2.5 ms), while the accuracy is stable over 2.5 ms. It is supposed that this is due to the way the Kinect®TOF-camera works, which uses a projected IR beam to generate light references for the IR camera. Therefore, apart from in the vicinity of the Kinect®, there is no decrease or increase in error related to the user's position.

For future software development, it would be desirable to make it more resistant to the presence of multiple people in the Kinect®field of view, possibly making the software capable of measuring multiple users at the same time.

References

1. Fürstenau, N.: Virtual and remote control tower: research, design, development, validation, and implementation (2022)
2. Pan, W., Luo, Y., Han, S., Zhong, L.: Summary of reliability analysis methods of remote tower system. IOP Conference Series: Earth and Environmental Science **693**(1), 012062 (2021)
3. Reynal, M., Imbert, J.-P., Aricò, P., Toupillier, J., Borghini, G., Hurter, C.: Audio focus: interactive spatial sound coupled with haptics to improve sound source location in poor visibility. Int. J. Hum. Comput Stud. **129**, 116–128 (2019)
4. Kaidi, Z.: Design and implementation of a centralized monitoring system for a remote control tower system. In: 2019 IEEE 1st International Conference on Civil Aviation Safety and Information Technology (ICCASIT), pp. 498–503 (2019)
5. Milgram, P., Takemura, H., Utsumi, A., Kishino, F.: Augmented reality: a class of displays on the reality-virtuality continuum. In: Das, H. (ed.) Telemanipulator and Telepresence Technologies. volume 2351, pp. 282–292. International Society for Optics and Photonics, SPIE (1995)
6. Reisman, R., Brown, D.: Design of augmented reality tools for air traffic control towers. In: 6th AIAA Aviation Technology, Integration and Operations Conference (ATIO) (2006)
7. Bagassi, S., De Crescenzio, F., Lucchi, F., Masotti, N.: Augmented and virtual reality in the airport control tower. In: 30th International council of aeronautical sciences, 09 (2016)
8. Bagassi, S., De Crescenzio, F., Piastra, S., Persiani, C.A., Ellejmi, M., Groskreutz, A.R., Higuera, J.: Human-in-the-loop evaluation of an augmented reality based interface for the airport control tower. Comput. Ind. **123**, 103291 (2020)
9. Teutsch, J., Bos, T., van Apeldoorn, M., Camara, L.: Attention guidance for tower atc using augmented reality devices. In: 2022 Integrated Communication, Navigation and Surveillance Conference (ICNS), pp. 1–12 (2022)
10. Rediess, H.: An augmented reality pilot display for airport operations under low and zero visibility conditions. In: Guidance, Navigation, and Control Conference (1997)
11. Ellis, S.R., Adelstein, B.D., Reisman, R., Schmidt-Ott, J.R., Gips, J., Krozel, J., Cohen, M.M.: Augmented reality in a simulated tower environment: effect of field of view on aircraft detection. In: NTRS - NASA Technical Reports Serve (2002)

12. Bimber, O., Raskar, R.: Chapter 3.1. 3. Head Mounted Projectors. Spatial Augmented Reality. AK Peters, pp. 76–79 (2005)

13. Pinska, E.: An investigation of the head-up time at tower and ground control positions. In: 5th Eurocontrol Innovative Research Workshop, pp. 81–86 (2006)

14. Hansen, D.W., Ji, Q.: In the eye of the beholder: a survey of models for eyes and gaze. IEEE Trans. Pattern Anal. Mach. Intell. **32**(3), 478–500 (2010)

15. Al-Rahayfeh, A., Faezipour, M.: Eye tracking and head movement detection: a state-of-art survey. IEEE J. Transl. Eng. Health Med. **1**, 2100212 (2013)

16. Heinrich, F., Schwenderling, L., Joeres, F., Lawonn, K., Hansen, C.: Comparison of augmented reality display techniques to support medical needle insertion. IEEE Trans. Visual Comput. Graph. **26**(12), 3568–3575 (2020)

17. Masotti, N., De Crescenzio, F., Bagassi, S.: Augmented reality in the control tower: A rendering pipeline for multiple head-tracked head-up displays. In Lucio Tommaso De Paolis and Antonio Mongelli, (ed), Augmented Reality, Virtual Reality, and Computer Graphics, pp. 321–338 (2016). Springer, Cham

18. Hopf, K., Neumann, F., Przewozny, D.: Multi-user eye tracking suitable for 3d display applications. In: 2011 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), pp. 1–4 (2011)

19. Kang, Suk-Ju: Multi-user identification-based eye-tracking algorithm using position estimation. Sensors, **17**(1) (2017)

20. Nagata, S.: Visual sensitivities to cues for depth perception. J. Inst. Telev Engineers Jpn. **31**, 649–655 (1977)

21. Allison, R.S., Gillam, B.J., Vecellio, E.: Binocular depth discrimination and estimation beyond interaction space. J. Vis. **9**(1), 10 (2009)

22. Palmisano, S., Gillam, B., Govan, D.G., Allison, R.S., Harris, J.M.: Stereoscopic perception of real depths at large distances. J. Vis. **10**(6), 19 (2010)