Quantitative Evaluation of Overlaying Discrepancies in Mobile Augmented Reality Applications for AEC/FM

Valentin GOMEZ-JAUREGUI [1]*, Cristina MANCHADO [2], Jesús DEL CASTILLO [3], Cesar OTERO [4]

[1]* PhD Civil Engineering, Research Group EGICAD, School of Civil Engineering, Universidad de Cantabria. Avda. Los Castros, 39005, Santander, Spain. valen.gomez.jauregui@unican.es

[2] PhD Civil Engineering, Research Group EGICAD, School of Civil Engineering, Universidad de Cantabria. Avda. Los Castros, 39005, Santander, Spain. manchadoc@unican.es

[3] Auto Drive Solutions, C/ Chile 10, Local 248 - Edificio 92, 28290 Las Rozas (Madrid), Spain. jdelcastillo@seysmedioambiente.com

[4] PhD Civil Engineering, Research Group EGICAD, School of Civil Engineering, Universidad de Cantabria. Avda. Los Castros, 39005, Santander, Spain. oteroc@unican.es

Abstract

Augmented Reality (AR) is a trending technology that provides a live view of the real and physical environment augmented by virtual elements, enhancing the information of the scene with digital information (sound, video, graphics, text or geo-location). Its application to architecture, engineering and construction, and facility management (AEC/FM) is straightforward and can be very useful to improve the on-site work at different stages of the projects. However, one of the most important limitations of Mobile Augmented Reality (MAR) is the lack of accuracy when the screen overlays the virtual models on the real images captured by the camera. The main reasons are errors related to tracking (positioning and orientation of the mobile device) and image capture and processing (projection and distortion issues).

This paper shows a new methodology to mathematically perform a quantitative evaluation, in world coordinates, of those overlaying discrepancies on the screen, obtaining the real-scale distances from any real point to the sightlines of its virtual projections for any AR application. Additionally, a new utility for filtering built-in sensor signals in mobile devices is presented: the Drift-Vibration-Threshold function (DVT), a straightforward tool to filter the drift suffered by most sensor-based tracking systems.

Keywords

Augmented Reality; Mobile Augmented Reality; Error estimation; Tracking; CAD; BIM; Civil Engineering; AEC/FM; Geo-Location; Sensors.

1. Introduction

Representation applied to construction has been evolving continuously during thousands of years. For example, the way in which Egyptians represented the construction of the pyramids was improved by the Romans for designing and erecting the Aqueduct of Segovia, and subsequently by the architects of the Amiens gothic cathedral in France. This evolution did not stop, and representation techniques continued to progress during the last centuries until today. Probably, the most drastic evolution took place at the end of the 20th century, thanks to the Computer Aided Design (CAD). Later, information technology was integrated to the digital design, giving birth to the Building Information Modelling (BIM), which makes it possible to make decisions about physical and functional characteristics of a facility during all its life-cycle, from conception to dismantling.

However, the most sophisticated and up-to-date 3D techniques for designing, modelling and representing construction projects have not been able to substitute definitely paper layouts on site yet. Digital devices, such as tablets, smartphones or laptops, are often used on site for illustrating the traditional 2D blueprints that have traditionally been used in projects, usually by means of 2D on-screen pdf or CAD files. Even though mobile computing is a field in evolution, its possibilities are not widely spread in current practices. Human skills (e.g. spatial relations, spatial orientation, spatial visualization, etc.) are still required for processing the 2D documents and understanding their meaning in real world, i.e. interpreting classical 2D representations to recognize their 3D implications. These limitations can be overcome by means of adequate technologies, for example the Augmented Reality.

This paper presents an efficient solution for developing an outdoor application for portable devices, based on Mobile Augmented Reality (MAR), to represent the virtual model of a project in its construction site. As will be explained throughout this work, there are different factors that affect the accuracy and efficiency of this technology, i.e. the geo-location of the mobile device, its orientation, and the techniques for accurately overlaying virtual models (dealing with projection and distortion issues), which motivated the authors to obtain a manner for evaluating and measuring the imprecision due to those causes.

1.1. Aim and motivation

The authors have found a gap-in-knowledge in the underlying theories and current practices existing today to measure the alignment imprecision of virtual and real objects on the screen. For many applications, where accuracy of restitution is key, it is necessary to know the degree of misalignment of the objects represented on the screen by evaluating distances in 3D world coordinate system. Of course, there have been some authors dealing with the error estimation of the 2D deviations on the screen, measured in pixels, as will be exposed consequently.

However we have not found in the scientific literature any relevant study dealing with mathematical estimation of absolute measurements, in world coordinates, of those discrepancies.

Therefore, the main aim of this work is to present a replicable methodology to mathematically obtain a quantitative evaluation of those overlaying discrepancies, obtaining the real distances from any real point to the sightlines of its virtual projections, for any AR application. Following this evaluation process, it will be possible to have an error estimation of the distance between real and virtual points in world coordinates.

1.2. Augmented Reality (AR) and Mobile Augmented Reality (MAR)

Augmented Reality (AR) is a technology that permits the user to improve and enhance the subjective perception of reality. AR provides a live view of the environment in such a way that the components of the real and physical scene are augmented by virtual elements added to the scene and seem to co-exist with the real world (Azuma 1997). These virtual additions (sound, video, graphics, text or geo-location), interactive in real-time, are generated and inserted by means of specialized software and are visualized by means of different types of hardware, like computers, tablets, smartphones or wearables (e.g. head-mounted displays HMDs).

A number of authors and sources use different terms to name what is also called Mixed Reality (Milgram and Kishino 1994; Azuma 1997; Schnabel 2009). According to Schnabel (2009) there are several subdivisions of Mixed Reality; in a scale, from reality to virtuality, it is distinguished between amplified reality, augmented reality, mediated reality, diminished reality, augmented virtuality, virtualized reality and finally virtual reality. This work will be dealing with technologies framed between the augmented and the mediated reality.

There are nowadays many AR browsers and AR uses applied to many different fields of interest, like biomedicine, tourism, linguistics, education, sports, entertainment, gaming, etc. (van Krevelen and Poelman 2010). Architecture, Engineering and Construction, and Facility Management (AEC/FM) are some other fields of application for AR, which can give many advantages to improve and enhance representation techniques on site (Behzadan, Dong, and Kamat 2015; Meza, Turk, and Dolenc 2015; Webster et al. 1996; Zollmann et al. 2014).

A step further is the Mobile Augmented Reality (MAR), a subset of the AR technology. MAR allows the user to move freely in an open space and see virtual elements added to the user perspective. During the last two decades, many MAR technologies and applications have been developed, as well as some surveys and literature reviews to present the state of the art (Azuma 1997; Papagiannakis, Singh, and Magnenat-Thalmann 2008; van Krevelen and Poelman 2010; Billinghurst, Clark, and Lee 2015; Chatzopoulos et al. 2017; Li et al. 2018). Readers willing to have a deeper knowledge about this technology may refer to any of these sources.

1.3. Potential of MAR applied to AEC/FM

Some potentials and benefits of MAR applied to AEC/FM include, but are not limited to: contribution to the understanding of PID (~~p~~Project ~~i~~Information ~~d~~Documents) in the different stages of AEC/FM projects, especially in the visualization of 3D models on-site; identification and location of existing construction elements, components and materials; improvement of the communication between the experts (e.g. architects, engineers, etc.) and the investors (e.g. clients, customers, stakeholders, etc.); better analysis of the work on-site according with the expectations defined in the schedule; finally, possibility of free movement within the real space, seeing the virtual model in real time on site from different perspectives (Meza, Turk, and Dolenc 2015).

Abboud (2014) differentiates the opportunities of MAR in three different stages of the project: i) In the Design, as a virtual tour of the project (full scale design visualization in situ, component scaling & clash detection, augmenting physical presentation media, informing the design process and communicating architectural narrative as a new interface between the virtual and real scenes). ii) In the Construction (Geo-Locating BIM data on the construction site, task supporting for construction processes, way finding & site navigation, real-time field reporting and 4D phasing of construction work sites). iii) In the Post-Completion (training for maintenance and repair, facilities management, etc.)

1.4. Limitations of MAR in AEC/FM

Construction is a traditional and slightly conservative sector, usually quite cautious to include innovative technologies, such as AR in this case, being one of the most important shortcomings that currently prevent their adoption to AEC/FM. Another human factor that restricts its acceptance is that holding and interacting with portable devices like tablets or smartphones requires the use of both hands, which can be impractical in some cases when working on-site. In case of using wearable devices (e.g. AR glasses or HMDs), special care should be taken because it could be dangerous on-site, as they may limit or overlay the user's field of view. Some other sources of problems are related to the learning and adaptation to new technologies or devices, risk of providing too much information on screen that could overwhelm the final user or missing the depth perception when using single-screen devices (with no stereoscopic effect).

Related to technical issues, tracking and registration is still a challenge, even if there are many solutions and technologies. For example, the most direct and inexpensive method for positioning, geo-locating the device by means of GPS, cannot be easily applied to indoor applications. Additionally, most built-in sensors lack adequate accuracy, which derives to imprecise positioning and orientation. Another technical problem is related to the mismatch between the level of detail managed by BIM and MAR software; BIM models have a lot of information and detail, generating big model files that are very difficult to be managed by MAR applications (Wang et al. 2014).

Finally, as has already been mentioned before and will be exposed in detail in section 2.2, there are not many methods to quantitatively evaluate the overlaying deviations, in real-scale, between the real and the virtual images on the screen. The so-called registration error occurs when the virtual objects displayed in the AR device appear in the wrong position relative to the real environment. In these cases, it would be desirable to know the accuracy of a MAR application when trying to obtain the deviation of a virtual point in the world coordinate system.

2. Literature review

2.1. Existing precedents in AEC/FM

There have been several examples of research on outdoor MAR for AEC/FM applications. However, the literature review of this section will not be very exhaustive as there are many other works explaining in full detail its background on AEC (Abboud 2014; Rankohi and Waugh 2013; Li et al. 2018) and facility management (Palmarini et al. 2018). A more exhaustive review will be offered in the next section when dealing with prior scholarly works related to error estimations in MAR.

The early prototypes were conceived and designed in the 90s (Webster et al. 1996), proposing an AR system using see-through head-worn displays to overlay graphics and sounds on the user's vision and hearing. It was applied to inspection of concrete reinforcement and monitoring the assembly of space structures. One year later, Azuma published "A Survey of

Augmented Reality" (Azuma 1997), a key work for AR researchers. In the same decade, the first MAR system for exploring the urban environment was proposed (Feiner et al. 1997; Höllerer et al. 1999), developing indoor and outdoor user wearable interfaces by means of a real-time-kinematic (RTK) GPS system.

Since then, there have been many researches dealing with the junction of AR + AEC (Abboud 2014). Some examples are the use of AR as a communication tool for urban design processes (Broschart, Zeile, and Streich 2013), for displaying information and data about building technologies and management (Dong, Feng, and Kamat 2013), assisting in the assembly of complex mechanisms or installations (Hou, Wang, and Truijens 2015), performing maintenance and repair (Henderson and Feiner 2007), providing visualization of underground infrastructures (Schall, Zollmann, and Reitmayr 2013), improving safety in construction (Li et al. 2018), etc.

Some projects have developed new vision-based MAR technologies that allow users to query and access 3D cyber-information on-site by using photographs taken from standard mobile devices, which are used to create or match 3D point cloud models (Bae, Golparvar-Fard, and White 2013; Golparvar-Fard, Pena-Mora, and Savarese 2009).

There is also commercial software like Augment, designed to show 3D models or media (e.g. buildings, structures or facilities) with which the users can interact. Bimar is another AR tool for AEC projects, allowing to visualize and interact with customized BIM models. However, both of them lack the ability of geo-locating the users and the 3D models. Trimble's SiteVision is a new AR prototype that combines a software GNSS receiver and a Google Tango-enabled phone, therefore providing positioning tracking capabilities in order to accurately align the design models to the real world (Aviad 2017). Recently, two of the most important multinational technology companies have released new solutions for MAR: Apple presented

ARKit in June 2017 and Google did the same with ARCore in March 2018. They can provide very interesting performances, like stability by means of visual-based tracking and surfaces recognition. However, they are currently supported only by some models of high-end mobile devices.

2.2. *Existing precedents in registration errors*

A few studies have been carried out in the last decades for estimating error analysis in AR. Holloway (1997) characterized the nature and sensitivity of the errors that cause misregistration in AR displays (HMD in this case): system delay (latency), tracker error, calibration error, optical distortion, misalignment of the model, etc. However, his analysis does not provide a model for estimating the overall error of the AR system, neither on the screen nor in the world coordinate system.

MacIntyre et al. (2002) described a method for real-time estimation of dynamically changing registration errors, according to the noise and errors of the tracker measurements. However, this method is not valid for quantitatively evaluating, in world coordinates, the discrepancies detected on the screen, as to it does not measure the actual deviation between real and virtual points. It only takes into consideration the statistical properties of the registration errors of the hardware (mean and covariance of the registration errors, provided by the tracker devices and modified by the authors for a more conservative error bound). Therefore, it just provides a 2D region on the screen where the object could be found, but no information about how far is the real object from its virtual representation. Additionally, their error propagation algorithm is used to generate an error estimation, for each vertex, as a 2D ellipse on the screen (after projecting its vertices into 2D screen coordinates, and then taking the convex hull of the 2D points). This method can be useful when working with compact objects whose vertices are at a similar distance to the view point. However, if the object is very deep, with close and far vertices from the view point, the error estimate (2D ellipse) should not be the same size for all of them. For instance, an error on the location of the camera (e.g. GPS precision) induces a larger discrepancy on the screen to those points that are closer to the view point (this issue is explained with a real example in section 4, Discussion and synthesis).

Vigueras Gomez et al. (2005) focused only on the suitability of the theoretical pinhole model of the cameras to accurately represent the virtual objects on the screen. They evaluated the influence of the camera in the AR context measuring pixel errors on the screen, but without analyzing the discrepancies of their representation in the real scene in world coordinates (real-scale distances).

Up to the last decade, some tracking error estimation methods had been developed, but they could not be integrated because of computational speed and accuracy. Bian et al. (2008) created a real-time tracking error estimation (RTEE) algorithm, simulating the multiple causes that can produce them. Then, they compared these results with the errors measured on the screen, in order to warn the user about them and to implement their correction to the tracking method, improving accuracy. The discrepancy of the error on the screen was computed by means of a linecode marker-based tracking method, using longitudinal fiducial marks adhered to the pipes of the facilities. This methodology has several limitations, like the need of disposing markers along the site, affected by multiple factors like distance, size, spatial disposition, visibility occlusions, etc. Moreover, it is useful for estimating the pose of the user, but not for the position of the objects of the scene.

For the project Smart Vidente, Schall et al. (2013) used a visual procedure: For assessing the overall re-projection error, they set a bullseye as a reference grid (concentric circular rings plotted with an offset of 5 cm) over a highly accurate surveyed reference point (Fig. 1). The virtual flag of the reference point, in this figure a red cross with a vertical line, should be visualized in the real world over the exact center of the grid if the precision was perfect. Then, they took screenshots from several positions around the reference point, visually recording the apparent distance of the virtual flag from the center of the grid. This technique could work for achieving the aim of the present work, but it is a rough approximation and does not take into consideration that the virtual flag is not really placed on the plane of the bullseye, but at any point of the sightline crossing that plane. As a result, the virtual flag could represent the projection of any of the points of that sightline, not only the one intersecting the bullseye. Fig. 1 graphically explains the problem: the user would see the perspective shown at the top right corner, interpreting that the 2D screen representation of the virtual flag (Pv1=Pv2=Pv3) is on the bullseye plane and its distance to the real point is nearly 15 units (Pv1). However, the user would have seen exactly the same perspective if the virtual flag would have been located in the 3D scene at any of the two other locations shown in the main image (Pv2 and Pv3). In one case, the distance of the virtual flag (Pv2) to the center of the bullseye (Pr) would be nearly 15 units in horizontal and 10 in vertical over the bullseye plane; and in the other case, the distance of the virtual flag (Pv3) would be more than 20 units in horizontal and 26 in vertical under the bullseye plane.

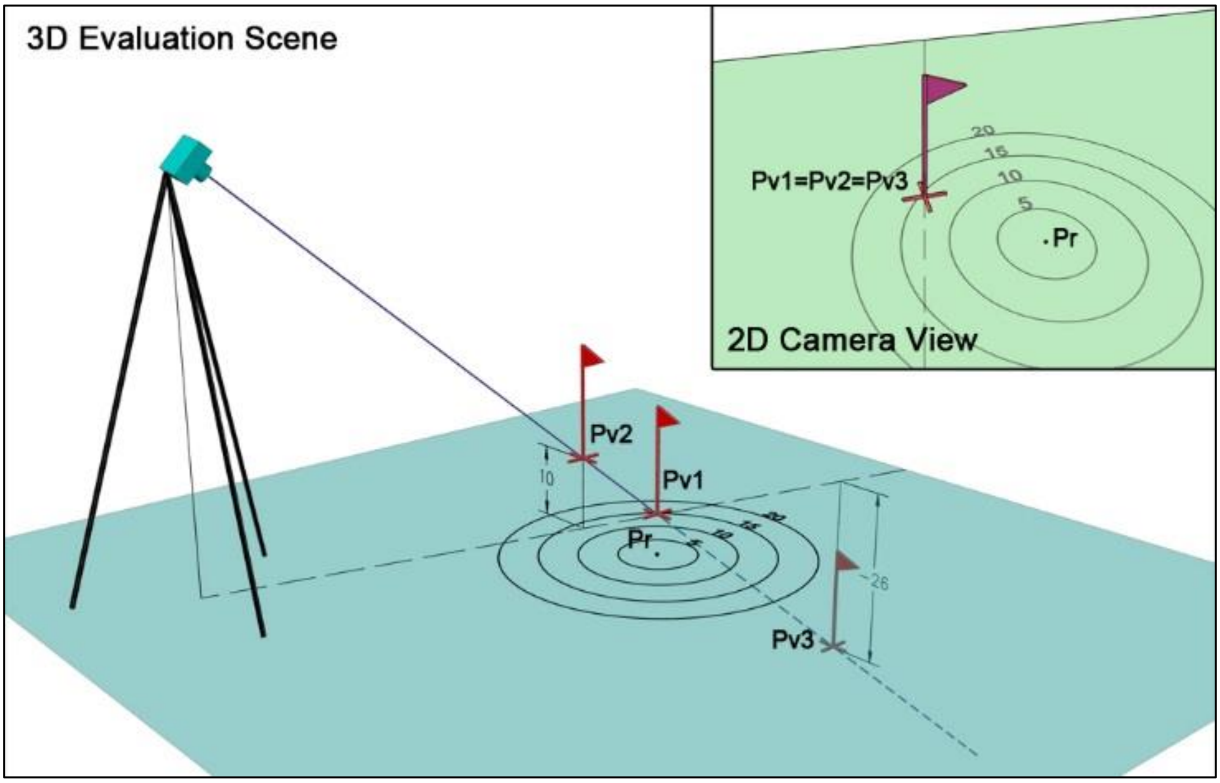


Fig. 1. *Depth problem with quantitative evaluation of AR discrepancies. Main image: possible locations of the virtual flags. Top right corner: perspective by the user, the same 2D screen representation of the virtual flag for three possible locations in the 3D scene.*

3. Methodology and results

3.1. CEsARe, the MAR application

This paper will show the benefits of its contributions applied to a new MAR application, CEsARe (Construction Engineering software for Augmented Reality). This is a software-tool specifically designed to represent in AR, by means of a portable electronic device, the 3D model of the project in the construction site or in any other environment. As a result, the virtual model (and its attached attributes) can be seen superposed to the real scenario of the construction site taken by the camera (Fig. 2). The application permits interaction with the virtual objects on the screen, representing existing elements of the environment, already built elements of the project or future elements still to be erected. Therefore, it is possible to obtain real-time information about all the elements represented by the digital device, such as spatial characteristics (position, geometry, interior not in-sight dispositions, etc.), physical properties (material, volume, weight, etc.), construction schedule, history, technical comments by the project team, etc. The amount of information retrieved on the screen is defined by the designer, because the project documentation can be updated continuously in a server and gathered by the application if it is connected directly to the internet.

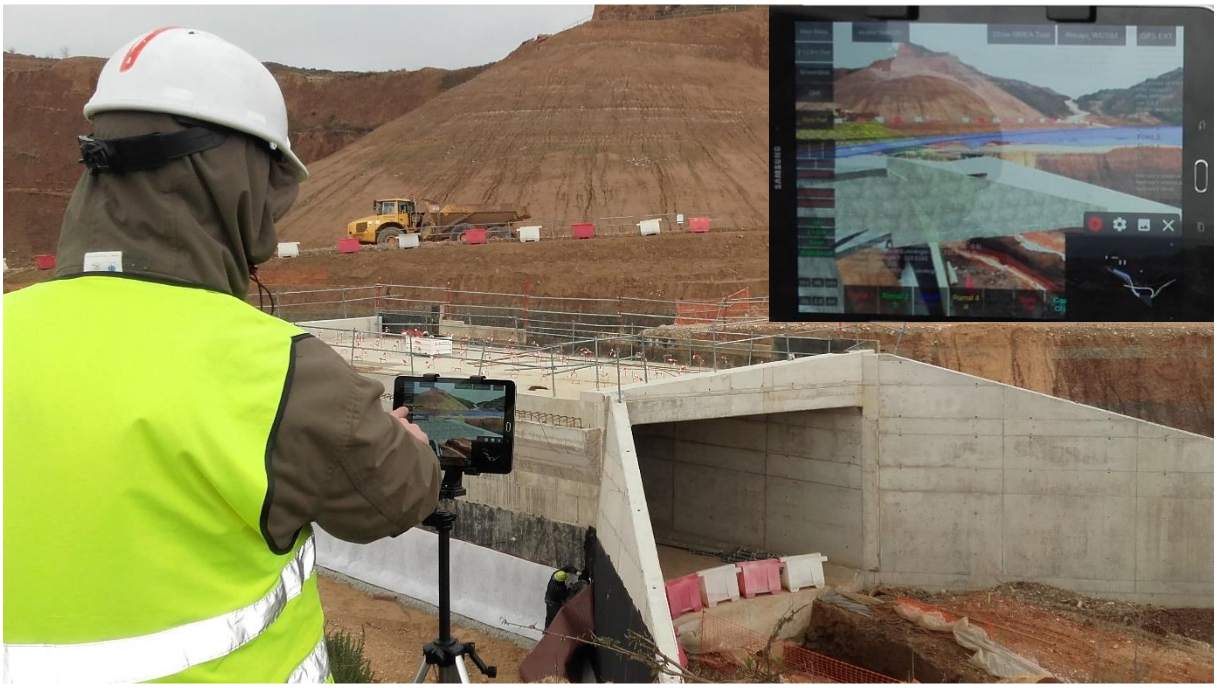


Fig. 2. *On-site verification of a concrete structure with CEsARe*

Overall, CEsARe is conceived essentially for outdoor applications and must respond to some technical and functional requirements that allow its use on-site in any construction project: i) accurate real-time geo-location, orientation, integration of real-time data and information streaming; ii) correct and stable virtual information overlaying real-time camera images; iii) complete real-time field reporting, giving the user updated and enhanced information of the elements shown on the screen; iv) multi-platform application, ready to run on several operating systems including Windows, Mac OS, Linux, IOS and Android.

3.2. Functional scheme of CEsARe, the MAR application

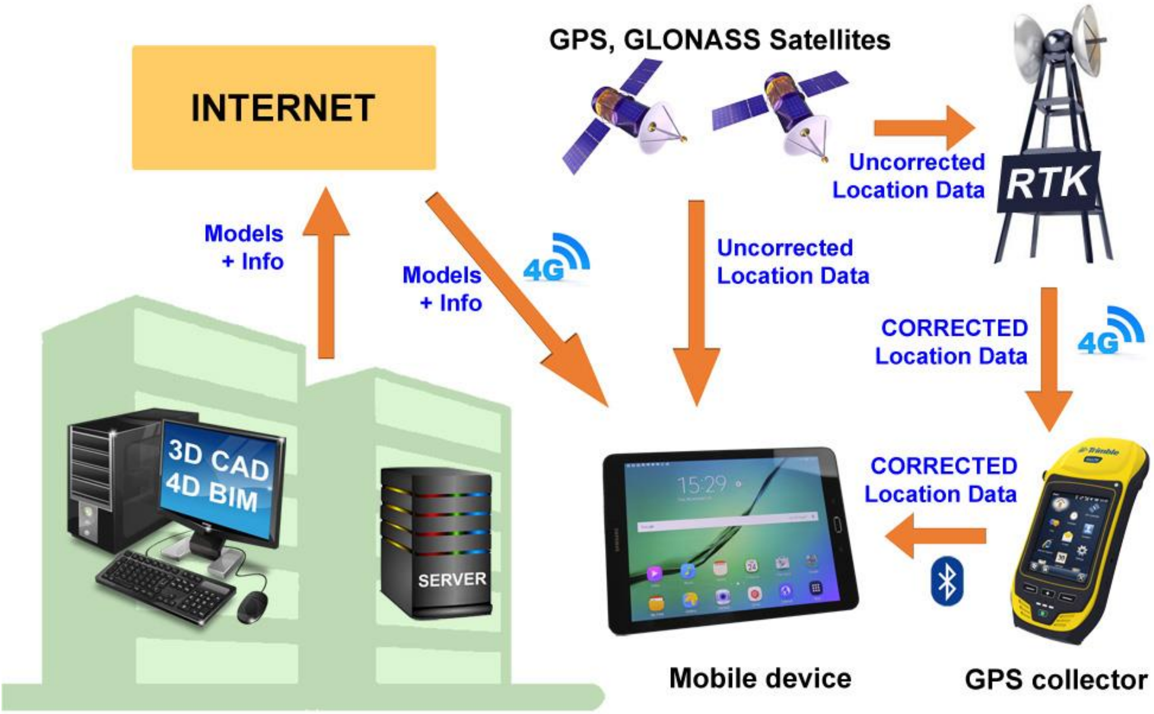


Fig. 3. *Scheme of the functioning of CEsARe, the MAR application*

Fig. 3 represents the functional scheme of CEsARe. From left to right, the first step is to create the model of the elements from 3D CAD or 4D BIM data, generating and geo-locating a virtual scene that has to be implemented with all the information available for the user. Then, the virtual models and the additional information (images, texts, web pages, documents, etc.) have to be stored in a web server, permitting access to the authorized users of the application. Information and virtual data can be downloaded in real-time from the server in such a way that it can be previously added to the repository by another designer at the studio and, from then on, can also be incorporated to the mobile device via 3G/4G or Wi-Fi. This quick-response function allows the user to ask for changes to the technical office that can be visualized in the application almost immediately.

The mobile device can receive continuous information about its position via GPS, either directly through the internal GPS receiver (uncorrected location data) or indirectly by Bluetooth from an external GPS collector, providing higher accuracy (corrected location data). This auxiliary GPS device requires data connection, which can be provided by the mobile device using tethering over Wi-Fi or directly by means of a 4G connection.

Therefore, for obtaining an accurate superposition of the virtual models over the reality captured by the mobile camera, four main challenges had to be fulfilled: i) generation of the virtual scene in an AR platform after modelling it by means of CAD or BIM, ii) exact geo-location of the device, iii) correct orientation of the scene and iv) precise overlaying or superposition of the virtual models over the real image through the camera lens.

3.3. Generation of the AR scene

The need of creating a multi-platform application led, among other factors, to choose Unity 3D (Unity Technologies 2015) as the AR engine for developing it. Unity 3D allows the deployment of the code in C# or JavaScript to the full range of mobile, VR, desktop, Web, Console and TV platforms. Nevertheless, all the different tests and trials for this work have been performed on Android operating system with a tablet Samsung Galaxy Tab S2 9.7”.

In order to produce the full virtual scene for the implementation of each project, it is necessary to generate and locate the 3D models previously, which can be imported to the scene in different formats. For this project, Autodesk Civil 3D was used to create the BIM models of the linear infrastructures. Then, after a post-processing phase, they have been segregated upon certain criteria, e.g. constructions phase, material, type of infrastructure, etc. Subsequently, these virtual objects have been converted to OBJ because this format permits importing them before compiling in the engine platform or after the compilation, in run-time on the actual MAR application.

3.4. Geo-location: accuracy test and assessment

The combination of position and orientation is referred to as the pose of an object or user. MAR applications make use of two methods of tracking and registration: sensor-based and vision-based tracking systems. The method using the combination of both of them is defined as hybrid tracking system (Chatzopoulos et al. 2017). Vision-based applications are difficult to be run on wearables due to their limited GPUs capacities, therefore CEsARe only uses sensor-based orientation. This section explains how to obtain the position or geo-location using electromagnetic methods (GPS), while the next section will deal with the orientation by means of inertial-based methods.

The accuracy of the position of the user is essential in MAR. Therefore, this goal has been achieved by means of Real Time Kinematic (RTK) satellite navigation, already used in other projects (Höllerer et al. 2001; Schall, Zollmann, and Reitmayr 2013; Dong and Kamat 2013). RTK is a technique used to improve the precision of position data derived from satellite-based positioning systems (Global Navigation Satellite Systems, GNSS) such as GPS, GLONASS and Galileo, thus providing submetric-level accuracy.

CEsARe offers three different ways to geo-locate the user: i) Static coordinates, either pre-established or set by the user on the way. ii) Internal GPS sensor of the mobile device. iii) External GPS collector. The first option allows the users to manually introduce the coordinates of their position, from which the scene must be observed. The second option lets the users to move around the scene, although the accuracy of this positioning is quite low, around 6 m in horizontal. For the last option, a Trimble Geo 7X has been used, an integrated, rugged, and high-accuracy GNSS handheld device that enables faster and productive geospatial data collection. It achieves high accuracy in real-time with the reliance of a traditional reference station-based infrastructure or VRS network, providing internet-delivered, centimeter to sub-meter GNSS positioning horizontal accuracy wherever cellular communications are available. In order to perform real-time GNSS corrections, the external GPS collector receives data streams from the supporting broadcaster of the area, via NTRIP (Networked Transport of RTCM via Internet Protocol). The program handles the HTTP communication and transfers received GNSS data to a RTK application. Once the location has been calculated and corrected, it is sent via Bluetooth in NMEA (National Marine Electronics Association) format. The GGA sentence sends, within a certain frequency (e.g. 1 second), the complete PVT (position, velocity, time) along with some other parameters. CEsARe is able to receive and process those NMEA sentences in real-time and, thus, locate the position of the user within a theoretical horizontal accuracy of approximately 2 cm + 1 ppm HRMS (Horizontal Root-Mean-Square 1-sigma).

3.4.1. Test No. 1: Geo-location precision with GPS and RTK

Fig. 4 and Fig. 5 show an experiment carried out in the test field of the School of Civil Engineering of Santander, where control points between P100 and P109 (Fig. 6) were horizontally located according their X and Y UTM coordinates (obtained within-centimeter precision by means of topographical tools) and compared with the horizontal measurements taken with the GPS handheld device (without additional external antenna nor survey rod). Fig. 4 shows that the average measurements are not always inside the limits of the precision contour in X and Y coordinates (3 cm, for a theoretical distance with the RTK base station of 10 km), reaching sometimes differences up to 5.5 cm with theoretical values (coordinate Y of P101). Vertical accuracy is always worse, with differences in elevations compared to theoretical values up to nearly 10 cm (coordinate Z of P109). However, the standard deviations are mostly under 3 cm in horizontal and under 5 cm in vertical, after taking 50 measurements at each point (Fig. 5).

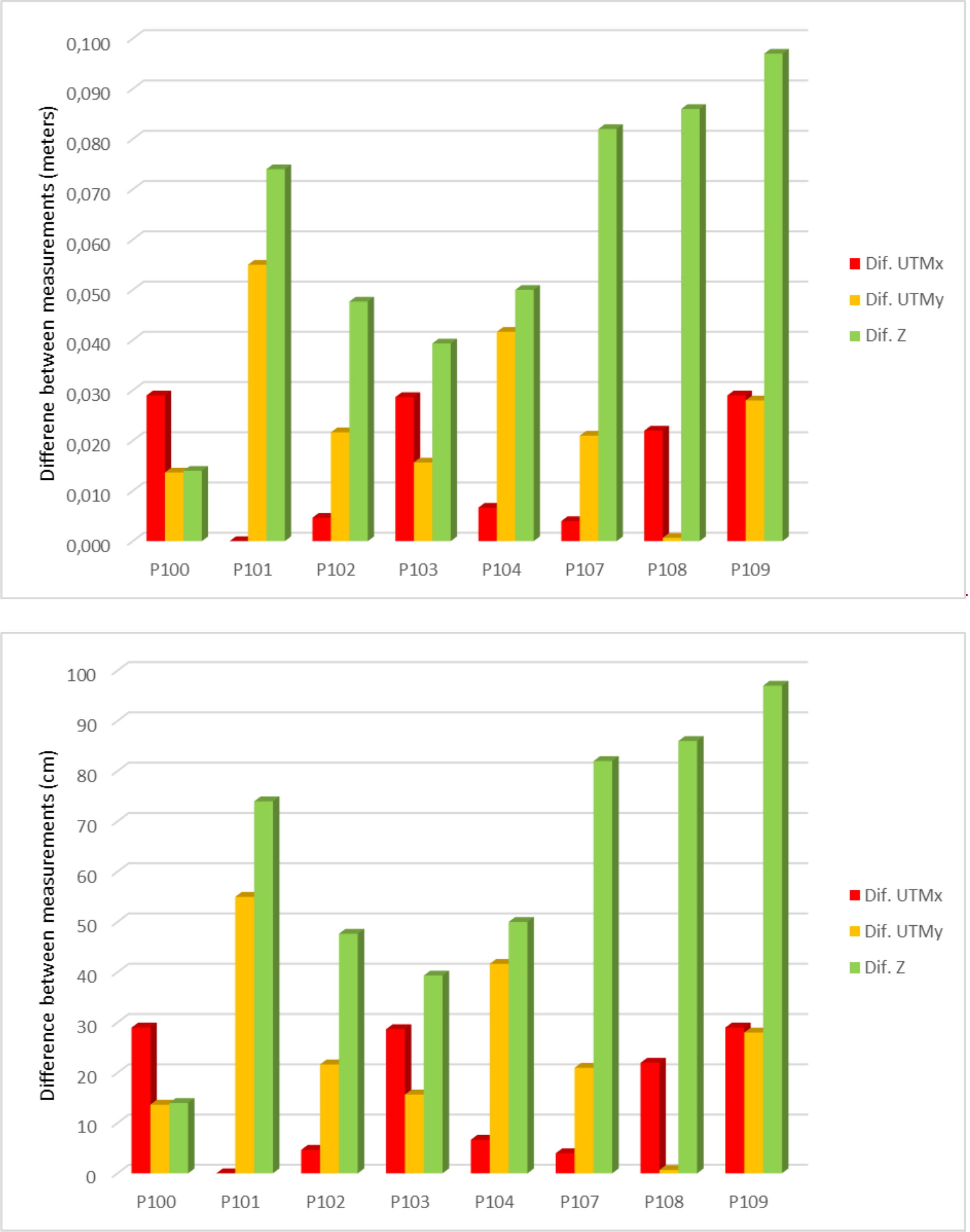


Fig. 4. *Precision test using a GPS handheld device with 2 cm + 1ppm HRMS.*

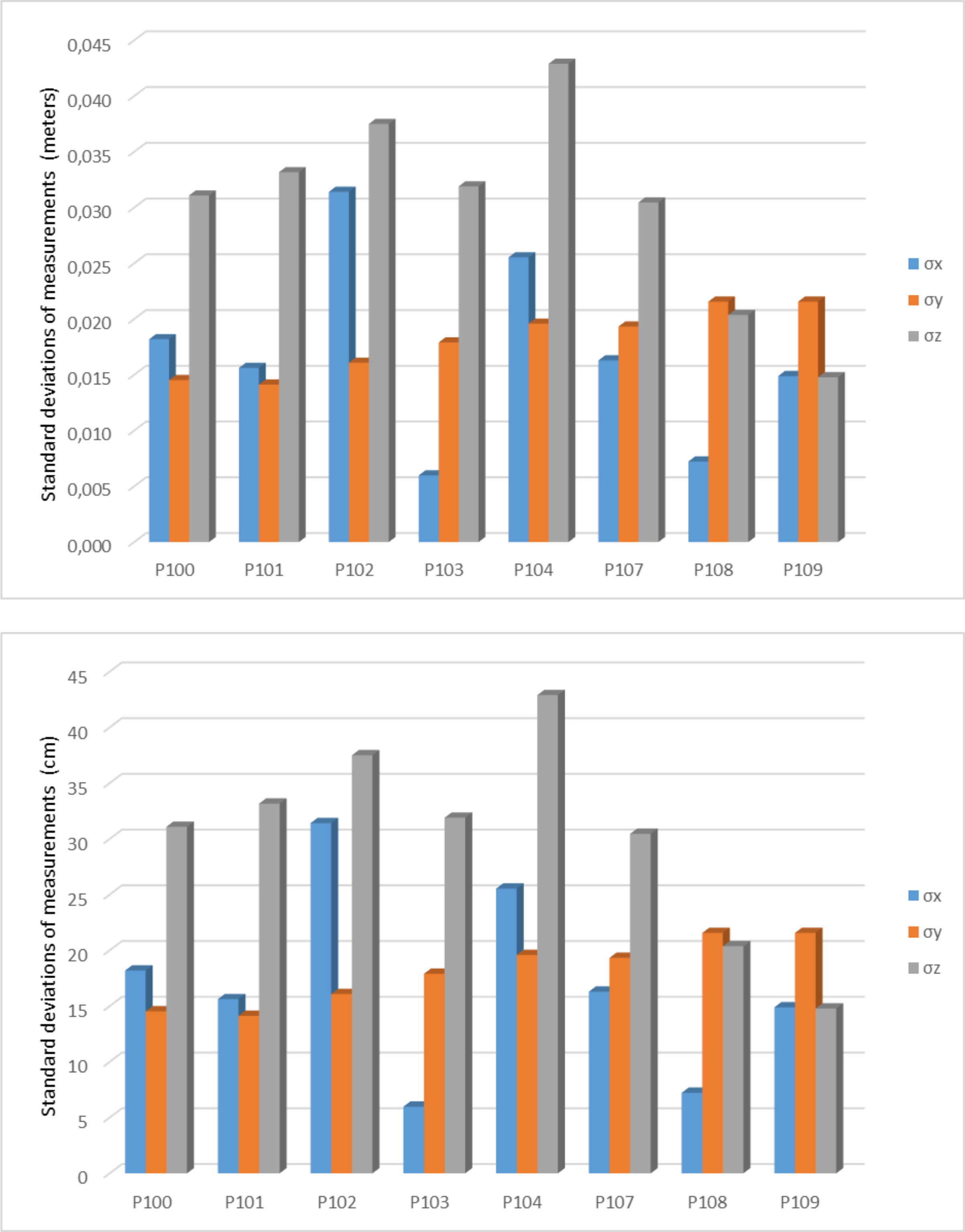


Fig. 5. *Standard deviations of measurements of coordinates X, Y and Z in the survey points.*

3.5. Orientation: evaluation of magnetometer and gyroscope

Once the mobile device is correctly geo-located in place, it is necessary to know where it is focusing at. Therefore, one of the main challenges of the MAR is the correct orientation of the mobile device in the real scene with regard to the six degrees of freedom, e.g. position X, Y, Z and rotations around these axis: pitch, yaw (or heading) and roll respectively (Fig. 6).

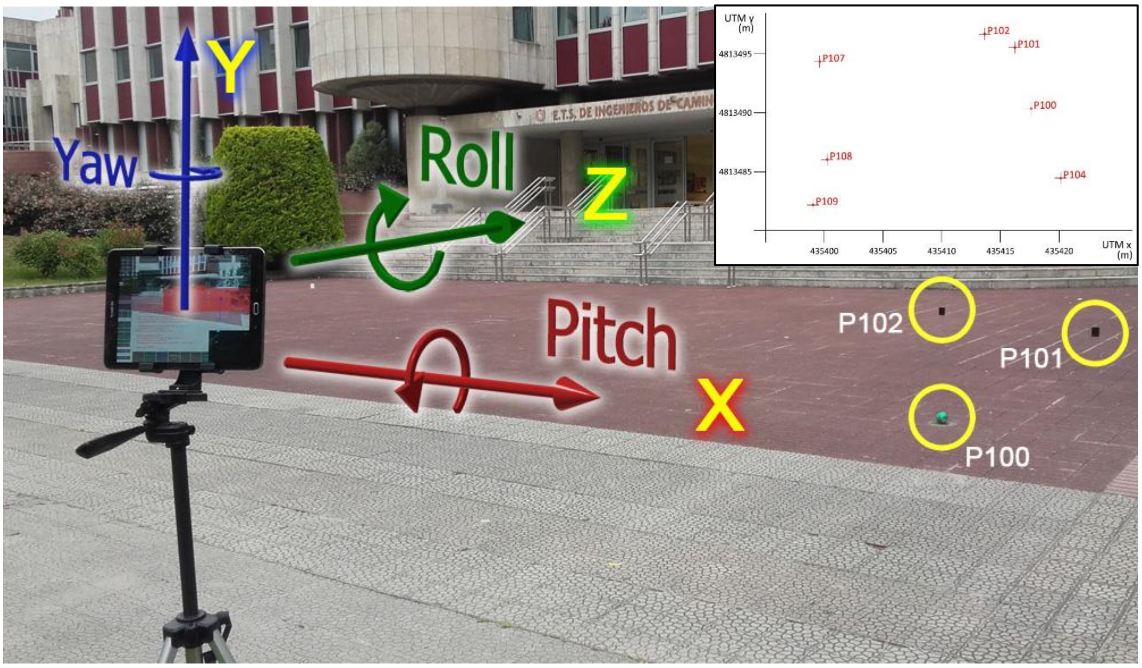


Fig. 6. *Principal axis in the mobile device (X, Y, Z) and its main rotations (Pitch, Yaw and Roll), positioned in the control point P104 of the test field, in front of control points P100 to P102.*

When using sensor-based methods, this performance is dependent on the quality and accuracy of the built-in MEMS sensors (Micro Electro Mechanical Systems) of the device (gyroscope, accelerometer and magnetometer). These sensors allow the application to know which vector represents the sightline from the user's position, and thus it would facilitate to overlay the virtual model over the real scene captured by the device camera. However, the Magnetic, Angular Rate, and Gravity (MARG) signals are affected by environmental electromagnetic influences and by the limited precision of the built-in sensors. As a result, there could arise two main kind of inaccuracies: i) orientation is not perfectly aligned with the magnetic or true north because magnetometers suffer from noise, jittering and temporal magnetic influences (Schall, Mulloni, and Reitmayr 2010) and ii) there could exist a drift of the 3D models related to the background camera image (Schall, Zollmann, and Reitmayr 2013). Related to the first issue, the magnetometer of a high-end mobile device may have a precision of not less than ±2 degrees, which could be insufficient accuracy for some measuring purposes.

3.5.1. Test No. 2: Magnetometer precision

Fig. 7 shows an orientation test carried out with a tablet Samsung Galaxy Tab S2 9.7”. The experiment consisted on measuring the values of the magnetic North during one minute (400 values in total) when the tablet was oriented to the geographical North. Those values were converted to true North by means of adding the magnetic declination (0° 58’ W in Santander in August 2017). The graph shows five sets of measurements, separated by pauses of 30”, without changing the mobile device position and orientation. The instability of its internal compass can be proved, with range deviation of 2.1, 3.3, 2.7, 2.6 and 2.8 degrees for sets 1 to 5 respectively and standard deviations of 0.33, 0.52, 0.45, 0.51 and 0.43 respectively. In this case, and for this mobile device, accuracy is not better than 9°.

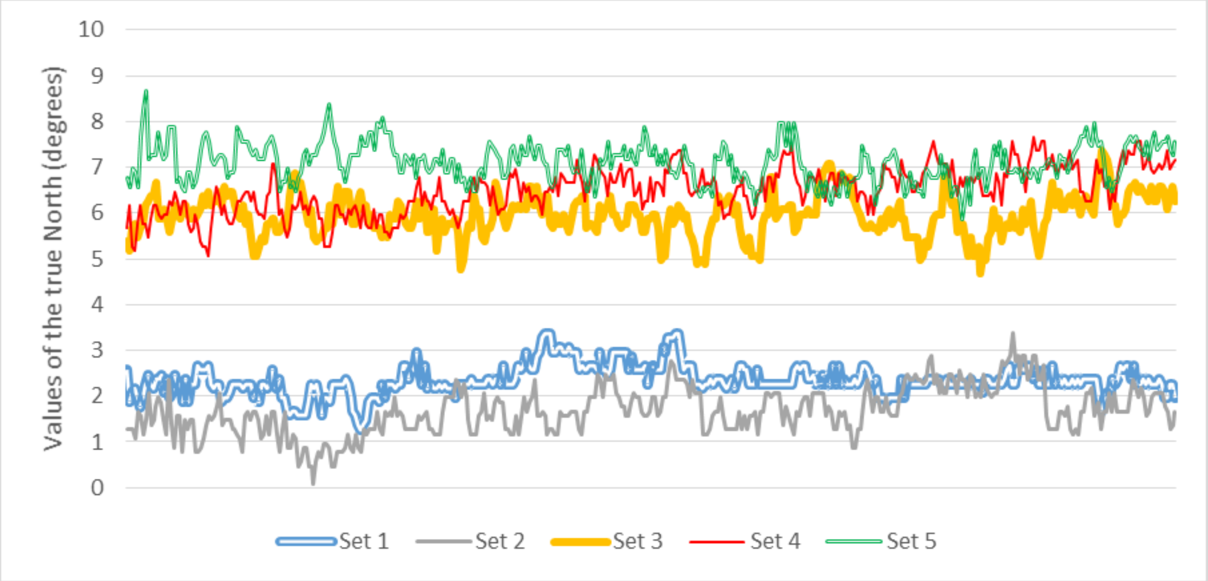


Fig. 7. *Variation of geographic North depending on the accuracy of the magnetometer.*

3.5.2. Test No. 3: Influence of Pitch in North signal

As can be seen in Fig. 8, pitching also influences the reading of the magnetic sensor (which only measures one axis), while the yaw or heading is constant. Therefore value of the North signal is not constant when tilting the mobile device; the graph shows the variation of the angle with the magnetic North starting at different values (heading N20, N125, N170, N190 and N320, where N20 means heading 20° North) when the mobile device is rotated along its X axis, varying the pitch from 0° (looking forward) to 90° (looking downward). This variation is not consistent, increasing in some cases (N190 and N320) and decreasing in others (starting N20, N125 and N170), changing the North signal in only 15° (N20) or up to 160° (N170). Finally, another limitation is that the signal from the magnetometer is not smooth and shows a lot of trepidation, which can be appreciated in Fig. 8, where some lines are broken and spasmodic (N190, N170).

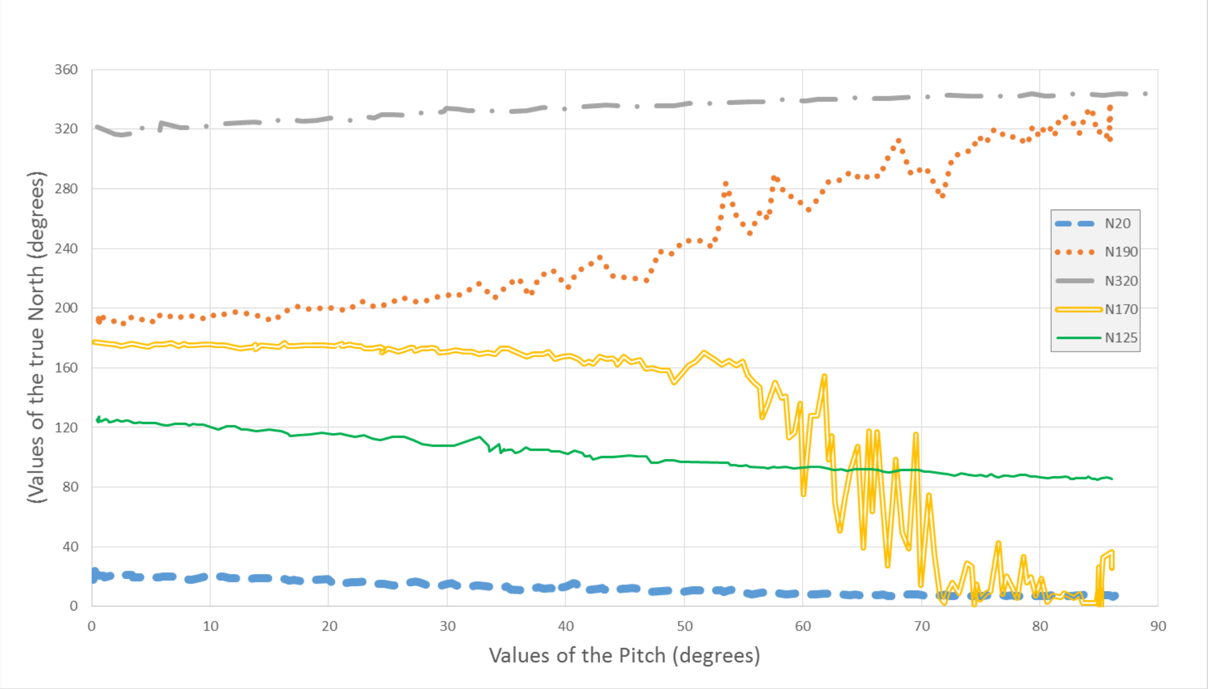


Fig. 8. *Influence of the Pitch signal over the North signal*

Related to the second issue, the angular rate, gyroscopes suffer the characteristic “drift” effect, a bias that appears after integration as an angular drift, increasing rise linearly over time. Several solutions have been applied over the years to solve this problem, based on algorithms combining the data provided by all the sensors, especially accelerometer and gyroscope, obtaining Inertial Measurement Units (IMU) that can be used to define the correct orientation of the device. Some of this procedures are based on the Kalman filter (in extended or discrete versions), which act as sensor fusion and data fusion algorithms (Schall, Mulloni, and Reitmayr 2010). Another common option is using the complementary filter, simpler than Kalman’s and involving less computation (Goslinski, Nowicki, and Skrzypczynski 2015; Higgins 1975), which uses the data from the gyroscope on the short term (high accuracy and independency of external forces) and the data from the accelerometer on the long term (it does not drift). In short, accelerometer and gyroscope can compensate for each other in the frequency domain (Wu et al. 2016). However, even using these fusion algorithms, error of the sensors can reach values close to 2 degrees (Schall, Zollmann, and Reitmayr 2013), which can have a strong influence on the accuracy of the screen registration.

The limitation of the previously mentioned filters is that the signals provided by the accelerometer are not suitable for calculating the yaw (or heading), because gravity conditions do not change when rotating the mobile device around Y axis. Moreover, its computational implementation, especially the Kalman filter, is not very simple and straightforward. Other methods used for compensating the drift are based on visual tracking (Chatzopoulos et al. 2017), giving place to the hybrid tracking methods, although they need more complex computational resources.

Therefore, in this work, a more direct and adequate approach has been taken: the Drift-Vibration-Threshold function (DVT). The main advantage of this method is that it is less computationally expensive than the others and it does not require a time-consuming effort for being implemented. The Kalman Filter and its derivations require to perform several matrix multiplications, additions, subtractions, transpositions and inversions, being the total time complexity of a single application O($n^{2.376}$) (Neto et al. 2009). Young (2009) simulated both the Kalman and the Complementary filters, and the latter performed up to nine times faster than the former. Our DVT algorithm deals with n-digit numbers rather than matrices, performing only two comparisons, three multiplications and one square-root for each step (Eqs. 1, 2). The total time complexity of these arithmetic functions (used by the DVT function) is, therefore, considerably lower than matrix algebra functions (Knuth 1993).

For the DVT function, two variables define the sensitivity of the IMU: the Drift Threshold (*dTh*) and the Vibration Threshold (*vTh*). The former defines the minimum value of the gyroscope angular rate that is not considered drift effect; the latter defines the minimum value of the accelerometer signal that is not considered a trepidation or involuntary trembling. It has been experimented that, in most cases, the drift affects only to the yaw (rotation along the Y axis). Therefore, both thresholds are used to define the yaw variation (*ΔYw*) of the camera according the following step function:

$$\Delta Yw(vTh, dTh) = \begin{cases} 0 & \rightarrow \quad \text{if } (\Delta accXZ \ \leq vTh) \text{ and } (\Delta Gy \leq dTh) \\ \\ \Delta Gy \cdot dt & \rightarrow \quad \text{if } (\Delta accXZ \ > vTh) \text{ or } (\Delta Gy > dTh) \end{cases} \qquad Eq.\ 1$$

$$accXZ = \sqrt{accX^2 + accZ^2} \qquad Eq.\ 2$$

being *accX* and *accZ* the values of the accelerometer in the X and Z axes respectively, *ΔaccXZ* the increment of the resultant of both accelerations*, ΔGy* the increment of the gyroscope signal in the Y axis, *dt* the increment of time at each step, *dTh* the drift threshold and *vTh* the vibration threshold.

3.5.3. Test No. 4: Panning test for the DVT function

Fig. 9 represents a panning test (swiveling the mobile device around Y axis, on a tripod, turning left – right – left) keeping it motionless at the beginning and between rotations. It is possible to observe the drift effect of the gyroscope, as the *Original GyroY* changes its value even when the device is stopped (this effect is more evident at the start). The *Corrected GiroY* shows the result of the signal after applying the DVT function, fixing the sensor bias when the variations of GyroY and accelerations are very low, but adjusting it in other case. It is also possible to observe the correspondence between the actual swiveling of the device and the value of the increment of AccXZ over the Vibration Threshold (*vTh*). The results show that the DVT function is able to eliminate the drift effect of the original gyroscope signal, which reaches average deviations up to -0.415°/sec (first plateau) and values of 0.072, -0.104 and 0.052 °/sec in the following segments of Fig. 9.

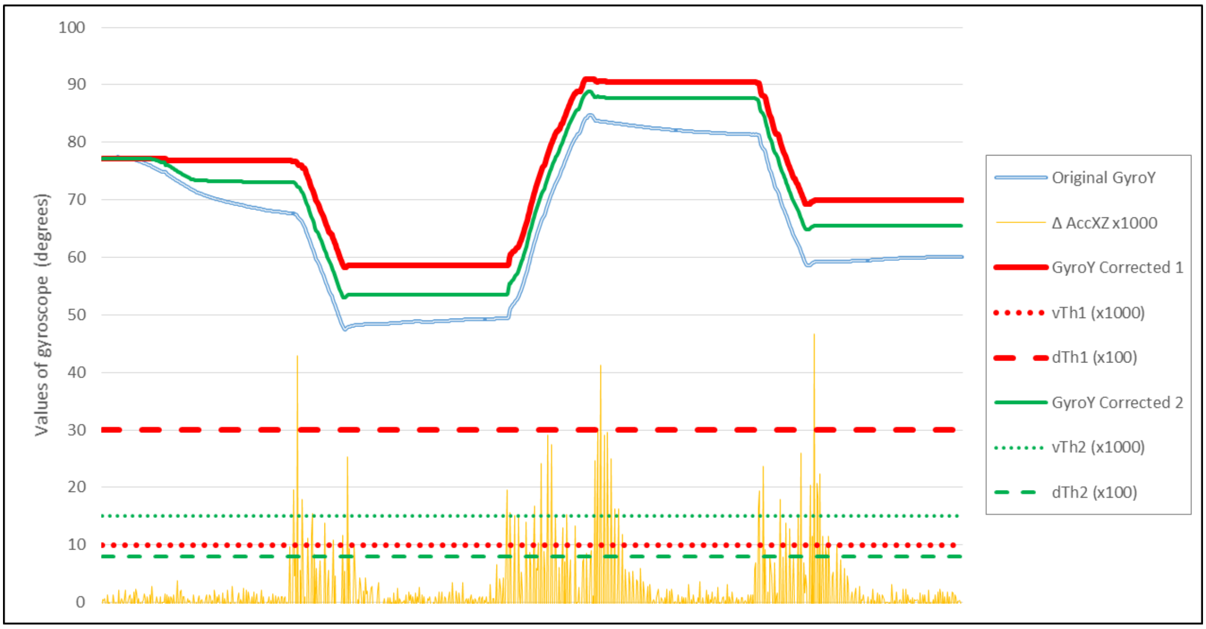


Fig. 9. *Comparison of the signal from the gyroscope in the Y axis before and after being processed with the DVT function, selecting two different sets of values for dTh and vTh (in red and green).*

3.6. Cameras: correspondence of real and virtual projections

Once the MAR device has been correctly geo-located and orientated, there may be some misalignments between the contours of the real objects and the virtual objects. In Fig. 10 it is possible to observe these small inconsistencies at the superimposition of the virtual elements, where the diagonal line of the fill is properly aligned at the left-hand side of the screenshot, while at the right-hand side the virtual water tower (in red) is slightly displaced compared to the location of the real water tower (in white). This, assuming that the virtual models are correctly generated and positioned in the scene, can be due to two sources of error: i) different projection parameters of real and virtual cameras and ii) distortion of the image due to the real camera lens.

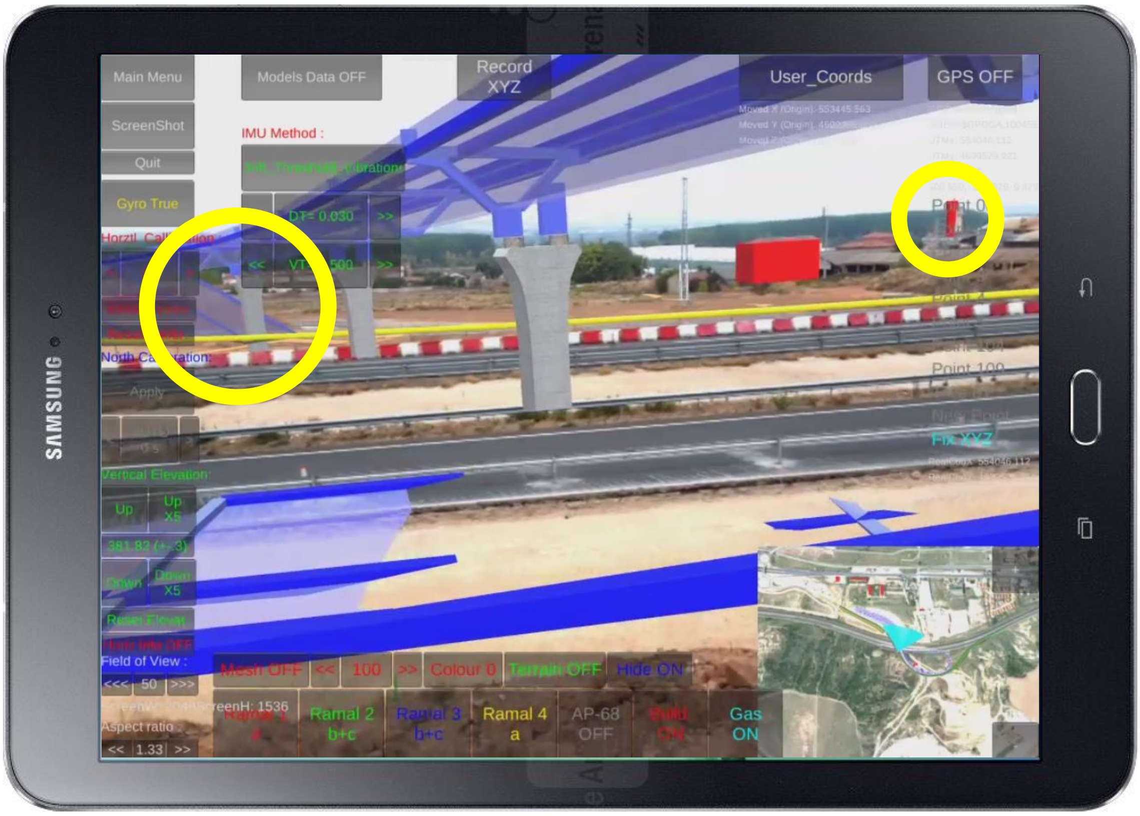


Fig. 10. *Misalignments in the scene between the virtual and real objects.*

Projection

Virtual projection of a 3D scene onto a 2D plane on the AR engine is achieved through a perspective projection camera (Unity Technologies 2015). Therefore, it was necessary to apply the same projection model of the real camera to the virtual camera configured in the AR engine.

The first concern affects specially to the angular field of view (AFoV). Even though some mobile devices identify the optical characteristics of their built-in cameras, sometimes the specifications are not reliable or unambiguous enough to be included as input data in the MAR. For instance, the AFoV can be different in horizontal and vertical axes, depending on the proportions of the screen or sensor. Therefore, CEsARe lets the user define both parameters: vertical AFoV and horizontal/vertical proportion of the virtual scene.

3.6.1. Test No. 5: Angular Field of View of the camera

These parameters were calculated on the device camera by means of an empirical test and then implemented on the virtual camera by editing the default projection matrix (Unity Technologies 2015) of the virtual camera. The experiment was very simple, based on capturing with the camera a tabulated grid from different distances and thus obtaining the angular size of the view cone. It was observed that AFoV changed depending on the distance to the panorama captured by the real camera, being slightly wider when the tabulated grid was further (Fig. 11). In fact, it could be observed that in all the cases the squares of the tabulated grid appeared more expanded at the edges of the picture than at the center. It was thus concluded that the most influent deviation had to be originated by the distortion produced by the lens, which will be analyzed in the following section.

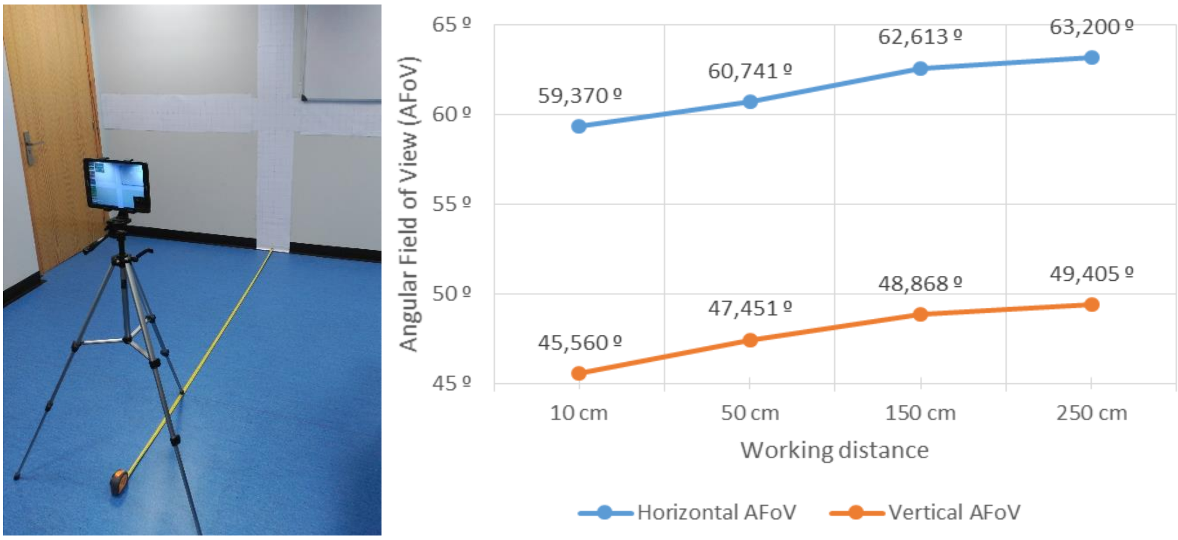


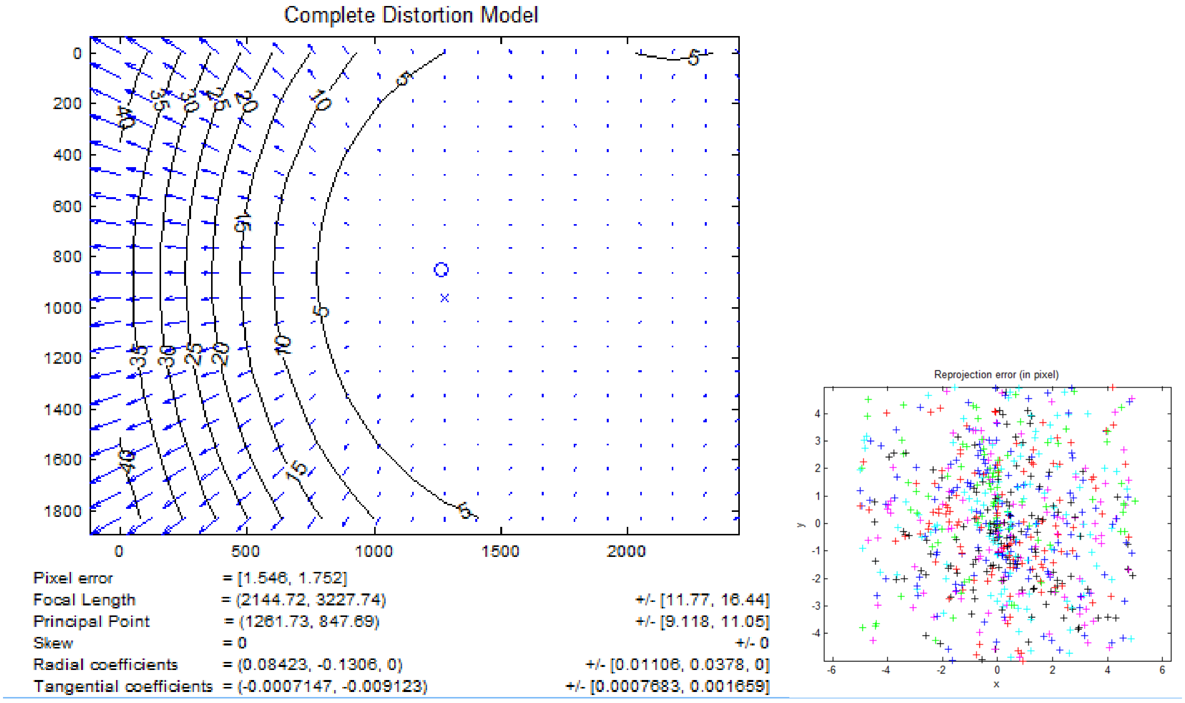
Fig. 11. *Variation of the angular fields of view of the device real camera depending on the working distance to the target.*

Distortion

It is well known that optical lens may produce deviation from rectilinear projection, arising to a deformation of the image captured by the device camera. The most commonly encountered distortions are radially symmetric, classified as either barrel, pincushion or moustache distortions, depending on the shape of the optical aberration. The deformation of the image, especially in its perimeter, modifies the theoretical AFoV and makes it impossible to measure angles and distances. Additionally, it creates some misalignments between the real and virtual objects of the scene, which is more relevant for this application.

3.6.2. Test No. 6: Distortion of the camera lens

Therefore, it was necessary to define the distortion of the device camera and apply it to the virtual camera. To do so, it was used the Brown-Conrady distortion model (Brown 1966), calculating the parameters that rule the angular and tangential distortions produced by the lens by means of a Matlab Toolbox (Bouguet 2015). Fig. 12 shows the complete distortion model of the camera of the tablet Samsung Galaxy Tab S2 9.7”, its calibration parameters (focal length, principal point and the skew, radial and tangential coefficients) and the reprojection error. In the figure of the left hand-side, each arrow represents the effective displacement of a pixel induced by the lens distortion, being as much as 45 pixels in the left-upper corner. This value represents, on a 2560x1920 px screen, a translation of the nearly 2.8 % of the distance to the center point (the cross indicates the center of the image, and the circle the location of the principal point). The distortion map is predominantly radial, although not symmetrical, proving that the tangential component could not be possibly neglected.

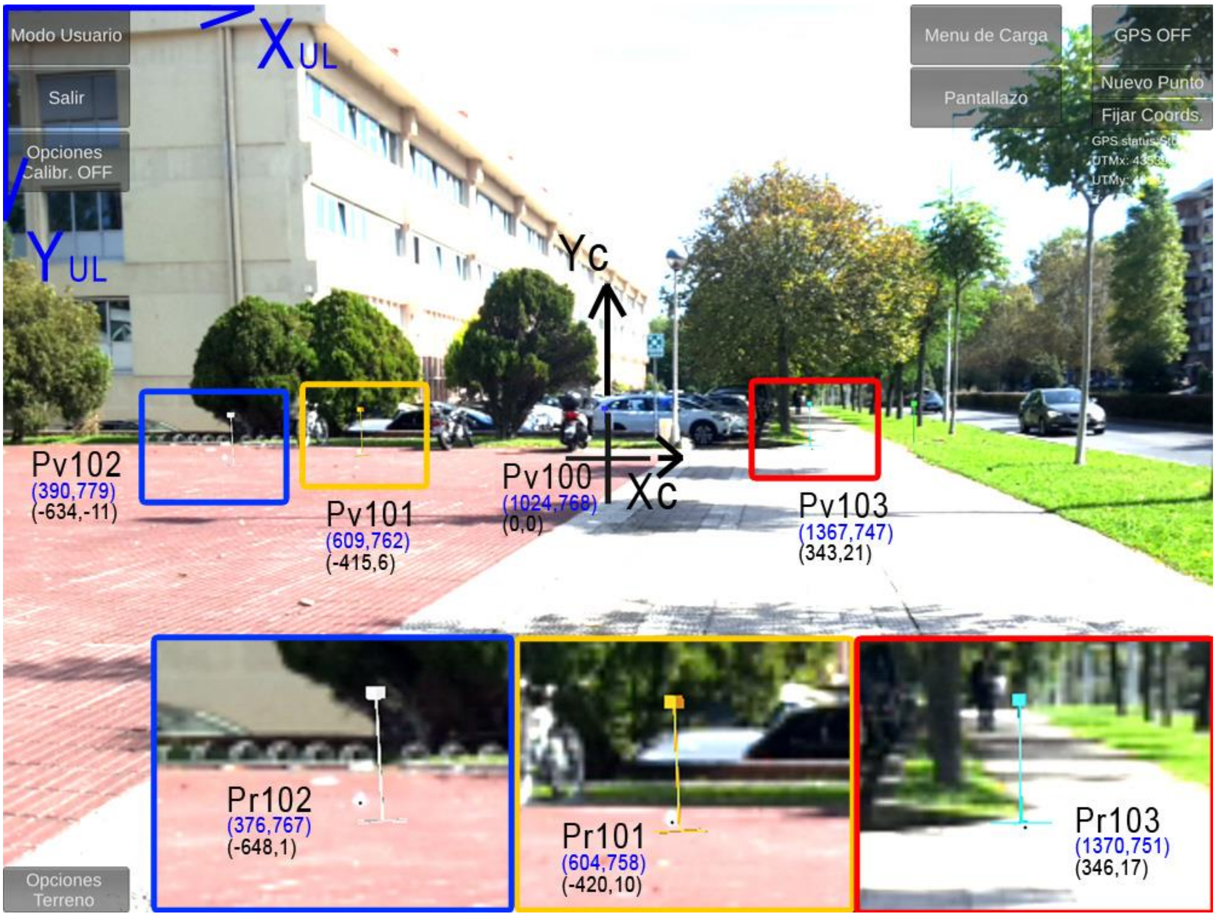


```
Pixel error            = [1.546, 1.752]
Focal Length           = (2144.72, 3227.74)          +/- [11.77, 16.44]
Principal Point        = (1261.73, 847.69)           +/- [9.118, 11.05]
Skew                   = 0                                      +/- 0
Radial coefficients    = (0.08423, -0.1306, 0)    +/- [0.01106, 0.0378, 0]
Tangential coefficients = (-0.0007147, -0.009123)  +/- [0.0007683, 0.001659]
```

Fig. 12. *a) Complete distortion model (tangential + radial) of the device camera; b) reprojection error of the calibration parameters; c) scenes for the experiment computed by the software.*

3.7. Quantitative evaluation of overlaying discrepancies

~~The last experiment was carried out in the same test field, where virtual and real points were strategically positioned to calculate the overall accuracy of the superposition.~~ Fig. 13 shows several scenes taken with CEsARe: above, there is a screenshot placing the mobile device at point P109 and targeting point P100; below, there is another view taken from point P104 and aiming P100. In both of them there are virtual flags overlaying the control points, remarked and amplyfied in the coloured rectangles. The images include information about the coordinates in pixels of both the real positions taken by the the camera (Pr) and the positions of the virtual flag bases (Pv). These coordinates were measured on the screeshots in a post-processesing step.



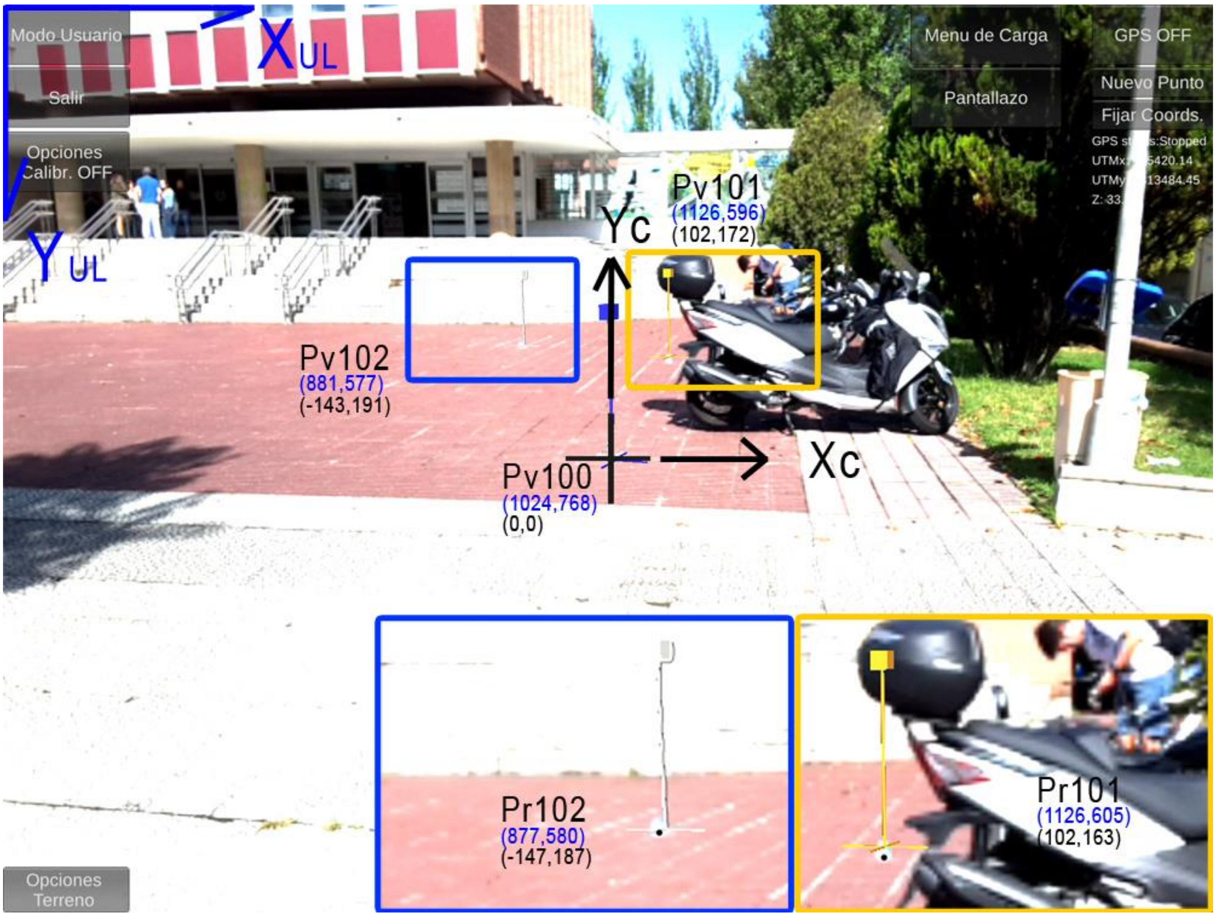


Fig. 13. *Verification of the overlaying of the application from control points P109 (above) and P104 (below), showing the coordinates (in pixels) of virtual flag bases (Pv) and real control points (Pr) with respect to upper-left screen axis (in blue) and central screen axis (in black).*

It should be stated that the deviation shown in pixels in a 2D ~~image~~ screenshot cannot be used to measure the real displacement of the virtual points with respect to the real points. The sightline between the observer and the virtual point holds infinite positions in the 3D space. It would be necessary to combine two or more different perspectives to calculate the actual position of the virtual point. However, in practice, it could not be possible because of two reasons: i) it is very unlikely that those sightlines intersect in a single point and ii) each one of those virtual points in the different pictures would be affected differently by the lens (they would be placed in different positions of the the distortion map).

Therefore, in order to assess the deviation in real scale (not in screen pixels) of each sightline of a virtual point (Pv) with respect to the real position of that point taken by the camera (Pr), it is necessary to reverse the projection process. Real cameras are ruled by the symmetrical perspective projection, which is schematically represented in Fig. 14. The image shows the representation of a certain point P (corresponding in this case with P100) and the different reference systems that are taken into consideraton: i) absolute axes (Xw, Yw, Zw), corresponding to the world-coordinate frame (UTM x,y and z over sea level), ii) local axes (Xv, Yv, Zv), corresponding to the viewing-coordinate system, referred to the center of projection (e.g. camera position) and iii) screen axes, corresponding to the coordinates in pixels, either referred to upper-left corner (Xul, Yul, Zul) or center of the screen (Xcs, Ycs, Zcs).

The methodology to obtain the transformation of world to screen coordinates (Hearn and Baker 2011) is illustrated in the real scene of Fig. 14: the camera of the mobile device is located at the projection reference point (Pprp), over the point P109; the camera is aiming the reference point (Pref), in this scene the point P102, therefore the middle axis of the frustum (view pyramide) is aligned with that point.

The transformation from world to viewing coordinates is achived by two steps: i) Translating the viewing-coordinate origin to the origin of the world coordinate system, by means of a Translation matrix **T**; and ii) Alingning the viewing axes (Xv, Yv, Zv) with the world axes (Xw, Yw, Zw), by means of a Rotation matrix **R**.

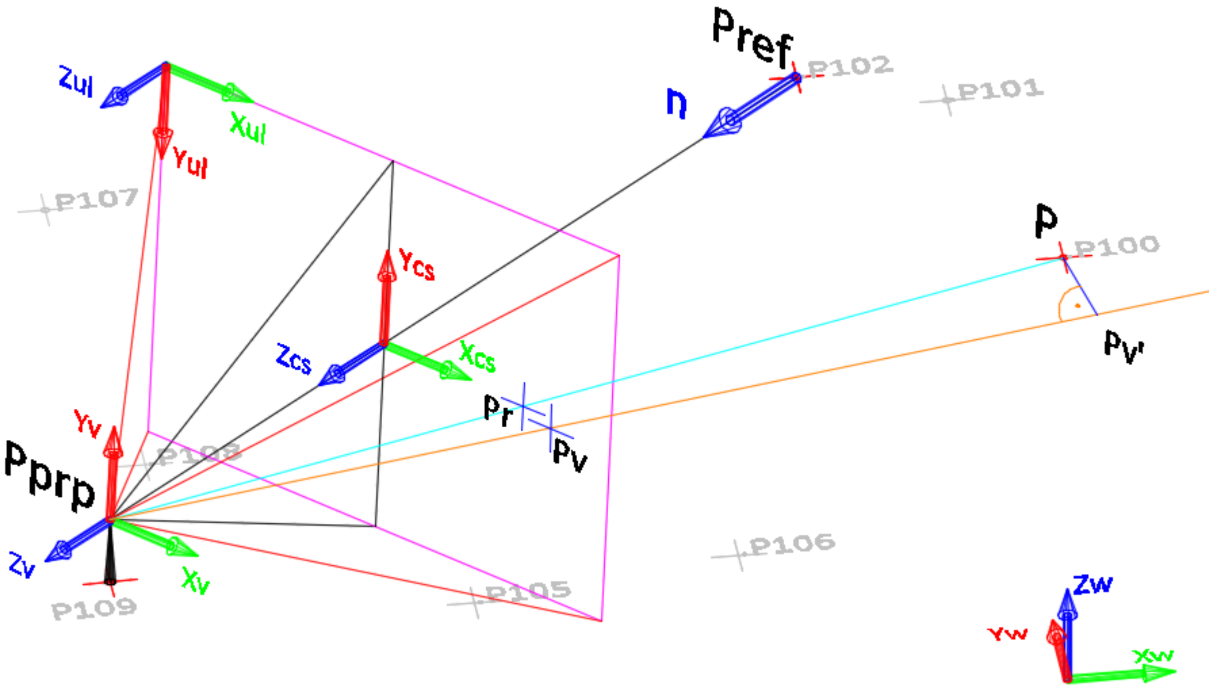


Fig. 14. *Symmetrical perspective projection of a real camera and reference systems for perspective transformations.*

The viewing-coordinate origin is at world position Pprp = (Xprp, Yprp, Zprp), thus the translation matrix **T** in homogeneous coordinates is defined as:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -Xprp \\ 0 & 1 & 0 & -Yprp \\ 0 & 0 & 1 & -Zprp \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \textit{Eq. 3}$$

Homogeneous coordinates are a system of coordinates used in projective geometry, where any point, including points at infinity, can be represented using finite coordinates. The rotation matrix **R** that superimposes the viewing axes onto the world frame is defined by the unit vectors **u**, **v**, and **n** as follows:

$$\mathbf{R} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \textit{Eq. 4}$$

The unit vector **n** comes from the vector **N** (Eq. 5)**,** being **N** = (Pprp - Pref) the direction for the Zv axis. **V** is the view-up vector, which in our case should be (0,0,1) if the camera is correctly balanced with null roll. Then, **u** is defined as a unit vector perpendicular to both **v** and **n** (Eq. 6). Finally, **v** is the cross product of **n** and **u** (Eq. 7).

$$\mathbf{n} = \frac{\mathbf{N}}{|\mathbf{N}|} = (n_x, n_y, n_z) \tag*{Eq. 5}$$

$$\mathbf{u} = \frac{\mathbf{V} \times \mathbf{N}}{|\mathbf{V} \times \mathbf{N}|} = (u_x, u_y, u_z) \tag*{Eq. 6}$$

$$\mathbf{v} = \mathbf{n} \times \mathbf{u} = (v_x, v_y, v_z) \tag*{Eq. 7}$$

The transformation from viewing to perspective-projection coordinates is defined by the following perspective matrix:

$$\mathbf{Mp} = \begin{bmatrix} \frac{cot\left(\frac{\theta}{2}\right)}{AR} & 0 & 0 & 0 \\ 0 & cot\left(\frac{\theta}{2}\right) & 0 & 0 \\ 0 & 0 & \frac{Znear + Zfar}{Znear - Zfar} & \frac{2 \cdot Znear \cdot Zfar}{Znear - Zfar} \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag*{Eq. 8}$$

Being θ the field-of-view angle of the cone of vision of the camera, and AR the Aspect Ratio (width / height) of the view plane. Znear and Zfar are the distances from the projection reference point (Pprp) to the near clipping plane and the far clipping plane of the frustum view volume.

The transformation from perspective-projection coordinates to screen pixels (referred to the center of the screen) is defined by the following matrix:

$$\mathbf{S_{CS}} = \begin{bmatrix} \frac{xVmax - xVmin}{2} & 0 & 0 & \frac{xVmax + xVmin}{2} \\ 0 & \frac{yVmax - yVmin}{2} & 0 & \frac{yVmax + yVmin}{2} \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag*{Eq. 9}$$

Being xVmax=w/2, xVmin=-w/2, yVmax=h/2 and yVmin=-h/2 the corner positions of the screen, defined by the resolution of the screen in pixels.

Finally, the last transformation changes coordinates from center-screen (Xcs, Ycs, Zcs) to upper-left-screen (Xul, Yul, Zul) referenced pixels, as measured in most image editor software. It is necessary to translate the origin of coordinates and to mirror the Y axis, so the transformation matrix is defined by:

$$\boldsymbol{S_{UL}} = \begin{bmatrix} 1 & 0 & 0 & -w/2 \\ 0 & -1 & 0 & h/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \textit{Eq. 10}$$

The complete transformation from world coordinates to upper-left screen coordinates is the composite matrix formed by concatening all the previous transformation matrices (Eq. 1, Eq. 2, Eq. 6, Eq. 7, Eq. 8):

$$\mathbf{M} = \mathrm{S_{UL}} \cdot \mathrm{S_{CS}} \cdot \mathbf{Mp} \cdot \mathbf{R} \cdot \mathbf{T} \qquad \textit{Eq. 11}$$

It is possible now to reverse this transformation and to obtain the world coordinates of the virtual flag base of any point (**Pv$_{\mathbf{WC}}$**), whose coordinates in upper-left-screen pixels (**Pv$_{\mathbf{UL}}$**) are known because they can be measured on the image:

$$\mathbf{Pv_{UL}} = \mathbf{M} \cdot \mathbf{Pv_{WC}} \qquad \rightarrow \qquad \mathbf{Pv_{WC}} = \mathbf{M}^{-1} \cdot \mathbf{Pv_{UL}} \qquad \textit{Eq. 12}$$

$$\mathrm{Pv}_z = -\mathrm{Zv} = -\frac{\mathrm{h}}{2} \cdot \cot(\theta/2) \qquad \textit{Eq. 13}$$

The screen coordinates of the images have only 2 dimensions, so for obtaining the conversion to the viewing coordinates it is needed to add a third one: the distance from the Pprp to the plane of view (Zv), directly calculated in Eq. 13 from the height of screen resolution *h* and the vertical field of view θ. When operating with matrices of dimension 4x4, points are expressed in homogeneous coordinates, being complemented so with the number one as the forth element, e.g. **Pv$_{\mathbf{UL}}$** = (Pv$_X$, Pv$_Y$, -Zv, 1).

The sightline between the view point and the representation of the virtual flag on the screen is now defined by the vector Pprp-Pv$_{\mathrm{WC}}$, which in the Fig. 14 is represented by the line connecting Pprp and Pv. It is possible, therefore, to measure the distance between the real point P and this sightline (Pprp-Pv), by calculating the shortest distance between point and line (segment P-Pv' in Fig. 14):

$$dist\,(\mathrm{P},\mathrm{Pv}') = min\,dist\,(P,\overline{Pprp\,Pv}) \;=\; \frac{|(\mathrm{Pprp}\;-\;\mathrm{Pv})\times(\mathrm{P}\;-\;\mathrm{Pv})|}{|(\mathrm{Pprp}\;-\;\mathrm{Pv})|} \qquad \textit{Eq. 14}$$

We consider this value as the deviation in world coordinates of the superposition between two points for a certain scene. However, due to the distortion is different depending on the position of the screen, the deviation of a point can be different depending on the scene. Therefore, several scenes are needed to better assess the deviation on the superposition, which imposes another problem because, in the general case, a set of sightlines will not intersect at a single point. Consequently, in the following, we propose the least-squares intersection of lines (Traa 2013) as the methodology to calculate the point that better fits the intersection of the sightlines (Pprp-Pv). A least-squares solution minimizes the sum of perpendicular distances from the unique solution point to all the sightlines.

Let's say that we have *k* different scenes where the point P is observable. For a certain scene *j*, there are the following elements: Pprp$_j$ are the homogeneous coordinates of the projection reference point of the scene (camera location), Pv$_j$ are the coordinates of the virtual flag base of point P, $\mathbf{H_j}$ is the vector between Pprp$_j$ and Pv$_j$ (sightline of point Pv$_j$), and $\mathbf{h_j}$ is its unit vector. According to Traa (2013), the point $\dot{\mathrm{P}}$ that minimizes the sum of perpendicular distances to the sightlines of the *k* scenes is the solution to the following linear system of equations:

$$\mathbf{R}\cdot\dot{\mathrm{P}} = \mathbf{q} \qquad \textit{Eq. 15}$$

$$\mathbf{R} = \sum_{j=1}^{k} (\boldsymbol{I} - \mathbf{h}_j \cdot \mathbf{h}_j^T) \ , \qquad \mathbf{q} = \sum_{j=1}^{k} (\mathbf{I} - \mathbf{h}_j \cdot \mathbf{h}_j^T) \cdot \mathbf{Pprp}_j \qquad \textit{Eq. 16}$$

Finally, we propose two values for estimate the superposition accuracy of the application:

- $D_{L\text{-}SQ}$: Distance between the optimum point achieved at the least-squares solution ($\dot{P}$) and the real position of the point (P).
- D_M: Maximum distance between the sightlines (Pprp-Pv) and the real position of the point (P), calculated in Eq. 14.

Summarizing, the quantitative evaluation, in world coordinates, of overlying discrepancies on the screen is based on the analysis of the scenes and the comparison between the real 3D position of certain elements and their virtual 2D projections on the screen. These are the steps to be followed ~~in order to perform~~ for estimating the evaluation on any AR application:

1) Identifying the intrinsic parameters of the mobile device:
    a) Resolution of the screen (in pixels), (e.g. height h, width w and aspect ratio AR=w/h) to be used in eq. 8, 9, 10 and 13.
    b) Angular field of view of the camera (θ), to be used in eq. 8 and 13.
    c) Distances to the near and far clipping plane (Znear and Zfar) of the frustum view volume, to be used in eq. 8.
2) Identifying the position, in 3D world coordinates, of the mobile device camera (**Pprp**), to be used in eq. 3.
3) Identifying the position, in 3D world coordinates, of the point aimed by the camera at the center of the screen (**Pref**), to be used in eq. 5 and 6.
4) Identifying the orientation of the mobile device, especially the view-up vector (**V**) obtained from the roll angle, to be used in eq. 6.
5) Obtaining the global transformation matrix (eq. 11) from world coordinates to coordinates in upper-left-screen pixels.
6) Identifying the 2D position on the screen (in pixels) of the real elements to be evaluated (Pr, the real camera representation) and their respective virtual flags (Pv, the virtual AR representation), to be used in eq. 12.
7) Identifying the position, in 3D world coordinates, of the real elements of the scene (**P**), to be used in eq. 14.

8) Calculating the shortest distance, in 3D world coordinates, between the sightline (Pprp-Pv) and the position of the real element (**P**) (eq. 14).

9) Applying steps 3 to 6 to several scenes, from different points of view, capturing one or several same points.

10) Calculating the least-squares intersection of the sightline (Pprp-Pv) of each scene to find the point that better fits the intersection of those sightlines of a same point from different points of view (one for each scene) (eq. 15 and 16).

~~Scene~~	~~**Pprp**~~	~~**Pref**~~	~~**Pv_{LL}**~~	~~**Pv_{CS}**~~	~~**V**~~	~~**Pv_{WC}**~~ ~~*(Eq. 12)*~~	~~*(Eq. 14)*~~ ~~dist (P,Pv')~~	~~*(Eq. 15-16)*~~ ~~**$\dot{P}$**~~	~~$D_{L\text{-}SQ}$~~ ~~dist (P, $\dot{P}$)~~
~~1~~	~~P109up (435399.046, 4813482.273, 35.498)~~	~~P100 (435417.629, 4813490.363, 33.887)~~	~~(609, 762, -1647)~~	~~(-415, 6, -1647)~~	~~(0, -0.018, 1)~~	~~(436739.731, 4814518.383, -82.171)~~	~~**0.085**~~	~~(435416.226, 4813495.603, 33.966)~~	~~**0.054**~~
~~2~~	~~P109up (435399.046, 4813482.273, 35.498)~~	~~P102 (435413.663, 4813496.695, 33.974~~	~~(1223, 762, -1647)~~	~~(199, 6, -1647)~~	~~(0, 0, 1)~~	~~(436705.196, 4814499.319, -72.170)~~	~~0.066~~		
~~3~~	~~P104up (435420.14, 4813484.445, 35.253)~~	~~P100 (435417.629, 4813490.363, 33.887)~~	~~(1126, 596, -1647)~~	~~(102, 172, -1647)~~	~~(0.02, 0, 1)~~	~~(434873.948, 4815041.121, -140.731)~~	~~0.054~~		

~~**Table 1.** *Analysis of results for P101 for three different scenes.*~~

3.7.1. Test No. 7: Quantitative assessment of absolute distances of overlying flags

The last experiment was carried out in the same test field of test No. 1, where virtual and real points were strategically positioned to calculate the overall accuracy of the superposition.

This quantitative evaluation can be illustrated in the following example, taking into consideration three scenes (the first and third scenes shown in Fig. 13), taken with the tablet Samsung Galaxy Tab S2 (screen width w=2048, height h=1536, vertical field-of-view θ=50º). The point P chosen for the estimation of discrepancies is P101 (435416.240, 4813495.555, 33.987), as this element is observable in the three screenshots. Table 1 exposes the initial parameters, conditions and final results after the calculations for every scene. It should be remarked that V is not always (0,0,1) exactly, as it depends on the levelling of the tripod. Attending to the outcomes, it can be concluded that the $D_{L\text{-}SQ}$, the distance between the optimum point achieved at the least-squares solution ($\dot{P}$) and the real position of the point (P), is 0.054 m (5.4 cm), while D_M, the maximum distance between the sightlines (Pprp-Pv) and the real position of the point (P), is 0.085 m (8.5 cm).

Scene	**Pprp**	**Pref**	**Pv_{UL}**	**Pv_{CS}**	**V**	**Pv_{WC}** *(Eq. 12)*	*(Eq. 14)* dist (P,Pv')	*(Eq. 15-16)* $\dot{P}$	D_{L-SQ} dist (P, $\dot{P}$)
1	P109up (435399.046, 4813482.273, 35.498)	P100 (435417.629, 4813490.363, 33.887)	(609, 762, -1647)	(-415, 6, -1647)	(0, -0.018, 1)	(436739.731, 4814518.383, -82.171)	**0.085**	(435416.226, 4813495.603, 33.966)	**0.054**
2	P109up (435399.046, 4813482.273, 35.498)	P102 (435413.663, 4813496.695, 33.974	(1223, 762, -1647)	(199, 6, -1647)	(0, 0, 1)	(436705.196, 4814499.319, -72.170)	0.066		
3	P104up (435420.14, 4813484.445, 35.253)	P100 (435417.629, 4813490.363, 33.887)	(1126, 596, -1647)	(102, 172, -1647)	(0.02, 0, 1)	(434873.948, 4815041.121, -140.731)	0.054		

Table 1. *Analysis of results for P101 for three different scenes.*

4. Discussion and synthesis

It has been stated that there are several sources of possible flaws that do not permit to obtain a perfect superposition of virtual models over their corresponding real entities. The synthesis of the results, including factors, methodology for contrast and evaluation, partial accuracy and remedial actions is presented in Table 2.

	AR Scene	**Geo-location**	**Orientation**	**Cameras alteration**
Factors	Coordinate reference system (ED50, ETRS89, WGS84, etc.), precision of modelled elements of the project	Precision of GPS receiver, RTK corrections, environment conditions, meteorology. Precision of NMEA transmission.	Precision and stability of magnetometer and gyroscope	Lens of real camera. Virtual camera parameters: field of view, proportion ratio.

Methodology for contrast and evaluation	Comparison of virtual and real coordinates of elements on site.	Survey and comparison with geo-located surveying points	Magnetometer: Comparison with real North. Gyroscope: drift analysis	Analysis of map of distortions and calculation of pixel deviation.
Partial Accuracy	0 cm	Under 6 cm in horizontal and under 10 cm in vertical.	Magnetometer: Up to 9° deviation with North. Gyroscope: drift up to 0.415°/sec	Pixels in most deformed corner: 45.
Remedial actions	Unnecessary.	Use of more accurate GPS devices and antennas. Data transfer without NMEA restrictions.	Gyroscope: DVT function, Kalman filter, complementary filter. Visual tracking. Magnetometer: manual calibration by visual superposition	Implementation of reverse distortion on the AR engine code.

Table 2. *Synthesis of factors in the AR application*

For the case of geo-location, it is possible to obtain accurate results in coordinates X and Y that do not affect the general precision of the system when the application is not used for very short distances. This was stated by moving the external GPS collector up to 5 cm and checking that the overlaying was exactly the same. However, a precision of 5 cm in horizontal and 10 cm in vertical could be not accurate enough for applying AR technologies in short distances or for identifying small elements on site. Moreover, it has been clearly proved that, in terms of geo-location, vertical accuracy is always the most disruptive input.

One of the main limitations of this study is the problem with the inaccuracy of the orientation, although it can be corrected under certain circumstances. The drift effect of the gyroscope can be completely eliminated by means of the DVT function when using a tripod in a static orientation and position. However, when holding the mobile device in the hands, it is not as efficient as the Kalman Filter or the Complementary Filter (because some users' shakings over the vibration threshold are filtered as movements rather than as jerking). The other limitation was due to the inaccuracy of the magnetometer, which does not let automatically orientate the scene in horizontal with enough precision. This issue is solved by using pre-existing real entities as guides that should be aligned with their corresponding virtual models. This operation is manual and delicate, and for that reason it should be essential to find another method to obtain an automatic and precise orientation of the scene (e.g. visual-based tracking methods).

Camera alterations are mainly due to lens distortion, which imposes another limitation to this study. According to the overlaying test of scene 1 shown in Fig. 13 (above), the distortion on the point P102 is 18 px. This deviation is very close to the distorsions discovered on the lens of the device camera at that position of the screen (Fig. 12). However, the translation of the virtual flag bases (Pv) with respect to the real control points (Pr) does not follow the map of distorsions reproduced in Fig. 12; for example, for that same point P102, distortions are not only horizontal but also vertical. The explanation could be, again, that the precision of the position of the control points in altimetrics is not good enough and the virtual flags are consequently not positioned correctly along the Z axis. This should be studied more deeply. After this analysis has been performed, it would be advisable to understand better and to correct those distortions automatically in real time. This could be done by warping the image with a reverse distortion by means of coding applied to the AR engine, which could be achieved by using certain methodologies (de Villiers, Leuschner, and Geldenhuys 2008). However, it could also be possible that computational correction of optical distortions could produce more delay-induced registration error than the distortion error it corrects (Holloway 1997).

CEsARe permits to correct some of these inaccuracies, either manual or automatically. For instance, the most disruptive data provided by the GPS, the elevation Z, can be corrected easily by the user by means of tactile controls on the screen. In terms of orientation, the North heading can also be adjusted by the user manually and the drift can be eliminated automatically by applying the DVT function when the mobile device reposes statically on a tripod.

5. Conclusions

In this paper, it has been shown that Mobile Augmented Reality (MAR) can be very useful to improve and accelerate specific tasks within Architecture, Engineering and Construction, and Facility Management (AEC/FM) projects. Some of its applications could give valuable input to on-site planning, interactive data identification, and on-site visualizations.

We have exposed several techniques and methodologies to respond to the main challenges proposed at the beginning of the project: i) obtaining an accurate real-time geo-location, ii) showing correct and stable virtual information overlaying real-time camera images, iii) providing interactive real-time field reporting and iv) delivering it as a multi-platform application for many operative systems and interfaces.

We further explained that one of the most important issues to resolve is the correct orientation of the mobile device related to the real scenario, because as has been widely proved, pure built-in sensor-based systems are not able to provide the required accuracy and performance without relying on a model of the environment. The focus of attention was also directed to the projection and distortion issues of the real and virtual cameras, which have to be addressed properly in order to achieve an accurate superimposition of the 3D models over the captured real scene.

Two main contributions have been proposed in this paper. The first one is a new methodology to perform a quantitative evaluation, in world coordinates, of the overlaying discrepancies on the screen by calculating mathematically two indicators: i) the distance from any real point to the sightlines from the observer to the virtual projections of that point and ii) the distance between the real position of any point and the optimum point achieved by the least-squares solution for all the sightlines of that point in different scenes.

The second original contribution is a new utility for filtering built-in sensor signals in mobile devices: the Drift-Vibration-Threshold function (DVT), a straightforward tool to filter the drift suffered by most sensor-based tracking systems. The DVT function corrects the sensor bias when the variations of the gyroscope and accelerator signals are under a certain threshold.

Opportunities for future research of the current application are constantly explored and developed in different real AEC/FM projects. Special efforts are being addressed in two main directions: i) to improve the automatic orientation, by calibration of mobile device sensors and / or by vision-based tracking and ii) to automatically correct those inaccuracies after being estimated in real time.

References

Abboud, Rana. 2014. “Architecture in an Age of Augmented Reality: Opportunities and Obstacles for Mobile AR in Design, Construction, and Post-Completion.” Australia: NAWIC National Association of Women in Construction. <https://www.inglobetechnologies.com/wp-content/uploads/2014/06/IWDS2013-AR-PAPER-R-ABBOUD-MARCH.pdf>. [accessed 21 January 2018].

Aviad, Almagor. 2017. “Trimble’s SiteVision Prototype - Highly Accurate Outdoor AR with Trimble Catalyst and Google Tango.” LinkedIn. May 19, 2017. <https://www.linkedin.com/pulse/trimbles-sitevision-prototype-highly-accurate-outdoor-almagor-aviad/>. [accessed 12 November 2017].

Azuma, R.T. 1997. “A Survey of Augmented Reality.” *Presence: Teleoperators and Virtual Environments* 4 (6): 355–85.

Bae, Hyojoon, Mani Golparvar-Fard, and Jules White. 2013. “High-Precision Vision-Based Mobile Augmented Reality System for Context-Aware Architectural, Engineering, Construction and Facility Management (AEC/FM) Applications.” *Visualization in Engineering* 1 (1): 3. https://doi.org/10.1186/2213-7459-1-3.

Behzadan, Amir H., Suyang Dong, and Vineet R. Kamat. 2015. “Augmented Reality Visualization: A Review of Civil Infrastructure System Applications.” *Advanced Engineering Informatics* 29 (2): 252–67. https://doi.org/10.1016/j.aei.2015.03.005.

Bian, Z., H. Ishii, H. Shimoda, and M. Izumi. 2008. “Real-Time Tracking Error Estimation for Augmented Reality for Registration with Linecode Markers.” *IEICE Transactions on Information and Systems* E91-D (7): 2041–50. https://doi.org/10.1093/ietisy/e91-d.7.2041.

Billinghurst, Mark, Adrian Clark, and Gun Lee. 2015. “A Survey of Augmented Reality.” *Foundations and Trends® in Human–Computer Interaction* 8 (2–3): 73–272. https://doi.org/10.1561/1100000049.

Bouguet, J-Y. 2015. “Camera Calibration Toolbox for Matlab.” Camera Calibration Toolbox for Matlab. 2015. <http://www.vision.caltech.edu/bouguetj/calib_doc/>. [accessed 23 March 2017].

Broschart, Daniel, Peter Zeile, and Bernd Streich. 2013. “Augmented Reality as a Communication Tool in Urban Design Processes.” In *Proceedings of the 18th International Conference on Urban Planning, Regional Development and Information Society, REAL CORP 2013*. Rome.

Brown, DC. 1966. “Decentering Distortion of Lenses.” *Photometric Engineering* 32 (3): 444–62.

Chatzopoulos, Dimitris, Carlos Bermejo, Zhanpeng Huang, and Pan Hui. 2017. “Mobile Augmented Reality Survey: From Where We Are to Where We Go.” *IEEE Access* 5: 6917–50. https://doi.org/10.1109/ACCESS.2017.2698164.

Dong, Suyang, Chen Feng, and Vineet R. Kamat. 2013. “Sensitivity Analysis of Augmented Reality-Assisted Building Damage Reconnaissance Using Virtual Prototyping.” *Automation in Construction* 33 (August): 24–36. https://doi.org/10.1016/j.autcon.2012.09.005.

Dong, Suyang, and Vineet R Kamat. 2013. “SMART: Scalable and Modular Augmented Reality Template for Rapid Development of Engineering Visualization Applications.” *Visualization in Engineering* 1 (1): 1. https://doi.org/10.1186/2213-7459-1-1.

Feiner, S., B. MacIntyre, T. Hollerer, and A. Webster. 1997. “A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment.” In *First International Symposium on Wearable Computers*, 74–81. Cambridge, MA, USA: IEEE Comput. Soc. ISBN: 978-0-8186-8192-9. https://doi.org/10.1109/ISWC.1997.629922.

Golparvar-Fard, M., F. Pena-Mora, and S. Savarese. 2009. “D4AR- A 4-Dimensional Augmented Reality Model for Automating Construction Progress Data Collection, Processing and Communication.” *Electronic Journal of Information Technology in Construction* 14: 129–53.

Goslinski, Jaroslaw, Michal Nowicki, and Piotr Skrzypczynski. 2015. “Performance Comparison of EKF-Based Algorithms for Orientation Estimation on Android Platform.” *IEEE Sensors Journal* 15 (7): 3781–92. https://doi.org/10.1109/JSEN.2015.2397397.

Hearn, Donald, and M. Pauline Baker. 2011. *Computer Graphics with OpenGL*. 4th ed. Boston: Addison Wesley. ISBN: 978-0-13-605358-3.

Henderson, Steven J., and Steven K. Feiner. 2007. “Augmented Reality for Maintenance and Repair (ARMAR).” AFRL-RH-WP-TR-2007-0112. Air Force Research Laboratory (USA). <http://www.dtic.mil/dtic/tr/fulltext/u2/a475548.pdf>. [accessed 21 January 2018].

Higgins, Walter. 1975. “A Comparison of Complementary and Kalman Filtering.” *IEEE Transactions on Aerospace and Electronic Systems* AES-11 (3): 321–25. https://doi.org/10.1109/TAES.1975.308081.

Höllerer, Tobias, Steven Feiner, Drexel Hallaway, Blaine Bell, Marco Lanzagorta, Dennis Brown, Simon Julier, Yohan Baillot, and Lawrence Rosenblum. 2001. “User Interface Management Techniques for Collaborative Mobile Augmented Reality.” *Computers & Graphics* 25 (5): 799–810. https://doi.org/10.1016/S0097-8493(01)00122-4.

Höllerer, Tobias, Steven Feiner, Tachio Terauchi, Gus Rashid, and Drexel Hallaway. 1999. “Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System.” *Computers and Graphics* 23: 779–785.

Holloway, Richard L. 1997. “Registration Error Analysis for Augmented Reality.” *Presence: Teleoper. Virtual Environ.* 6 (4): 413–432. https://doi.org/10.1162/pres.1997.6.4.413.

Hou, Lei, Xiangyu Wang, and Martijn Truijens. 2015. “Using Augmented Reality to Facilitate Piping Assembly: An Experiment-Based Evaluation.” *Journal of Computing in Civil Engineering* 29 (1): 05014007. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000344.

Knuth, Donald E. 1993. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Reading, Mass.: Addison-Wesley. ISBN: 0-201-89684-2.

Krevelen, D. W. F. (Rick) van, and Ronald Poelman. 2010. “A Survey of Augmented Reality Technologies, Applications and Limitations.” *International Journal of Virtual Reality* 9 (2): 1–20.

Li, Xiao, Wen Yi, Hung-Lin Chi, Xiangyu Wang, and Albert P.C. Chan. 2018. “A Critical Review of Virtual and Augmented Reality (VR/AR) Applications in Construction Safety.” *Automation in Construction* 86 (February): 150–62. https://doi.org/10.1016/j.autcon.2017.11.003.

MacIntyre, B., E.M. Coelho, and S.J. Julier. 2002. “Estimating and Adapting to Registration Errors in Augmented Reality Systems.” In *Virtual Reality, 2002. Proceedings. IEEE.*, 73–80. Orlando, FL, USA. ISBN: 978-0-7695-1492-5. https://doi.org/10.1109/VR.2002.996507.

Meza, Sebastjan, Ziga Turk, and Matevz Dolenc. 2015. “Measuring the Potential of Augmented Reality in Civil Engineering.” *Advances in Engineering Software* 90 (December): 1–10. https://doi.org/10.1016/j.advengsoft.2015.06.005.

Milgram, Paul, and Fumio Kishino. 1994. “A Taxonomy of Mixed Reality Visual Displays.” In *IEICE Transactions on Information Systems*, E77, No. 12:1321–29.

Neto, Armando Alves, Douglas Guimarães Macharet, Víctor Costa da Silva Campos, and Mario Fernando Montenegro Campos. 2009. “Adaptive Complementary Filtering Algorithm for Mobile Robot Localization.” *Journal of the Brazilian Computer Society* 15 (3): 19–31. https://doi.org/10.1007/BF03194503.

Palmarini, Riccardo, John Ahmet Erkoyuncu, Rajkumar Roy, and Hosein Torabmostaedi. 2018. “A Systematic Review of Augmented Reality Applications in Maintenance.” *Robotics and Computer-Integrated Manufacturing* 49 (February): 215–28. https://doi.org/10.1016/j.rcim.2017.06.002.

Papagiannakis, George, Gurminder Singh, and Nadia Magnenat-Thalmann. 2008. “A Survey of Mobile and Wireless Technologies for Augmented Reality Systems.” *Computer Animation and Virtual Worlds* 19 (1): 3–22. https://doi.org/10.1002/cav.221.

Rankohi, Sara, and Lloyd Waugh. 2013. “Review and Analysis of Augmented Reality Literature for Construction Industry.” *Visualization in Engineering* 1 (1): 9. https://doi.org/10.1186/2213-7459-1-9.

Schall, Gerhard, Alessandro Mulloni, and Gerhard Reitmayr. 2010. “North-Centred Orientation Tracking on Mobile Phones.” In *9th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2010*, 267–68. Seoul, South Korea: IEEE. ISBN: 978-1-4244-9343-2. https://doi.org/10.1109/ISMAR.2010.5643600.

Schall, Gerhard, Stefanie Zollmann, and Gerhard Reitmayr. 2013. “Smart Vidente: Advances in Mobile Augmented Reality for Interactive Visualization of Underground Infrastructure.” *Personal and Ubiquitous Computing* 17 (7): 1533–49. https://doi.org/10.1007/s00779-012-0599-x.

Schnabel, Marc Aurel. 2009. “Framing Mixed Realities.” In *Mixed Reality In Architecture, Design And Construction*, edited by Xiangyu Wang and Marc Aurel Schnabel, 3–11. Dordrecht: Springer Netherlands. ISBN: 978-1-4020-9087-5. https://doi.org/10.1007/978-1-4020-9088-2_1.

Traa, Johannes. 2013. “Least-Squares Line Intersection.” University of Illinois at Urbana-Champaign. <http://cal.cs.illinois.edu/~johannes/research/LS_line_intersect.pdf>. [accessed 1 September 2017].

Unity Technologies. 2015. “Unity 3D.” Unity 3D. 2015. <https://unity3d.com/>. [accessed 23 March 2016].

Vigueras Gomez, J.-F., G. Simon, and M.-O. Berger. 2005. “Calibration Errors in Augmented Reality: A Practical Study.” In *Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality, 2005. Proceedings.*, 154–63. IEEE. ISBN: 978-0-7695-2459-7. https://doi.org/10.1109/ISMAR.2005.23.

Villiers, Jason P. de, F. Wilhelm Leuschner, and Ronelle Geldenhuys. 2008. “Centi-Pixel Accurate Real-Time Inverse Distortion Correction.” In *Proceedings of the International Symposium on Optomechatronic Technologies 2008*, edited by Yukitoshi Otani, Yves Bellouard, John T. Wen, Dalibor Hodko, Yoshitada Katagiri, Samuel K. Kassegne, Jonathan Kofman, et al., 726611. https://doi.org/10.1117/12.804771.

Wang, Xiangyu, Martijn Truijens, Lei Hou, Ying Wang, and Ying Zhou. 2014. “Integrating Augmented Reality with Building Information Modeling: Onsite Construction Process Controlling for Liquefied Natural Gas Industry.” *Automation in Construction* 40 (April): 96–105. https://doi.org/10.1016/j.autcon.2013.12.003.

Webster, Anthony, Steven Feiner, Blair Macintyre, William Massie, and Theodore Krueger. 1996. “Augmented Reality in Architectural Construction, Inspection and Renovation.” In *Proc. ASCE Third Congress on Computing in Civil Engineering*, 913–919.

Wu, Jin, Zebo Zhou, Jingjun Chen, Hassen Fourati, and Rui Li. 2016. "Fast Complementary Filter for Attitude Estimation Using Low-Cost MARG Sensors." *IEEE Sensors Journal* 16 (18): 6997–7007. https://doi.org/10.1109/JSEN.2016.2589660.

Young, A.D. 2009. "Comparison of Orientation Filter Algorithms for Realtime Wireless Inertial Posture Tracking." In *2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*, 59–64. Berkeley, CA.: IEEE. ISBN: 978-0-7695-3644-6. https://doi.org/10.1109/BSN.2009.25.

Zollmann, Stefanie, Christof Hoppe, Stefan Kluckner, Christian Poglitsch, Horst Bischof, and Gerhard Reitmayr. 2014. "Augmented Reality for Construction Site Monitoring and Documentation." *Proceedings of the IEEE* 102 (2): 137–54. https://doi.org/10.1109/JPROC.2013.2294314.

List of figures

Fig. 14. Symmetrical perspective projection of a real camera and reference systems for perspective transformations.