# An Efficient Reduction of IMU Drift for Registration Error Free Augmented Reality Maintenance Application

Lakshmiprabha N. S.[1,2] ns.lakshmiprabha@cern.ch
Alexander Santos[1] a.alvsantos@cern.ch
Olga Beltramello[1] olga.beltramello@cern.ch
[1]European Organization for Nuclear Research, CERN, Switzerland
[2]University of Rome Tor Vergata, Italy

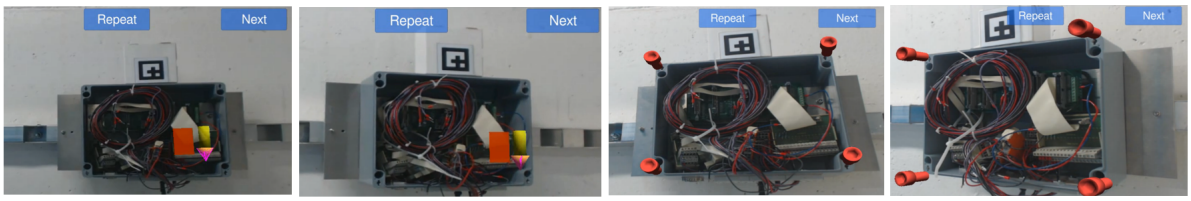



Figure 1: Augmented reality application showing user perceptive view of two different maintenance procedures.

## ABSTRACT

Augmented reality (AR) is a technology that overlays virtual 3D content in the real world to enhance a user's perception. This AR virtual content must be registered properly with less jitter, drift or lag to create a more immersive feeling for the user. The object pose can be determined using different pose estimation techniques using the data from sensors cameras and inertial measurement units (IMUs). Camera based vision algorithms detect the features in a given environment to calculate the relative pose of an object with respect to the camera. However, these algorithms often take a longer time to calculate the pose and can only operate at lower rates. On the other hand, an IMU can provide fast data rates from which an absolute pose can be determined with fewer calculations. This pose is usually subjected to drift which leads to registration errors. The IMU drift can be substantially reduced by fusing periodic pose updates from a vision algorithm. This work investigates various factors that affect the rendering registration error and to find the trade-off between the vision algorithm pose update rate and the IMU drift to efficiently reduce this registration error. The experimental evaluation details the impact of IMU drift with different vision algorithm pose update rates. The results show that the careful selection of vision algorithm pose updates not only reduces IMU drift but also reduces the registration error. Furthermore, this reduces the computation required for processing the vision algorithm.

## Keywords

Augmented reality; Pose estimation; Inertial measurement unit; Marker tracking; Sensor fusion; Registration error

## 1 INTRODUCTION

In past decades, virtual reality (VR) and augmented reality (AR) have seen a wide range of applications starting from entertainment (games) to medical fields. Many companies like Google (Google glass[1]), Microsoft (Hololens[2]), Epson (Moverio[3]) for example brought this AR technology from research experiment projects to daily use commercial products. The AR technology enables one to perceive reality in a more informative dimension. This supplementary information can be useful in many applications, one such is a maintenance application [Feiner11]. This paper concentrates on maintenance operation in ATLAS[4] particle physics detector (in Large Hadron Collider (LHC)) as a use case environment. The operators and technicians in this high energy physics radiation environment have to finish the maintenance job rapidly to reduce the exposure time. The AR technology can help in replacing paper manuals and actually display each maintenance procedure in the operator's field of view. Figure 1 shows the user perceptive view of

[1] https://support.google.com/glass/
[2] https://www.microsoft.com/microsoft-hololens/
[3] http://www.epson.com/cgi-bin/Store/jsp/Landing/moverio-bt-200-smart-glasses.do
[4] http://atlas.ch/

 ISBN 978-80-86943-65-7

two maintenance procedures for repairing a FPIAA (Finding people in ATLAS environment) system.

AR technology relays on the pose information provided by the sensors to overlay the virtual content. Amongst many available sensors, camera and Inertial measurement units (IMUs) are complete sensors, as they provide position and orientation in both indoor and outdoor applications. Also, these sensors can be compared closely with the humans vision (camera) and vestibular (IMU) system [Corke07]. The vision and vestibular systems provide key information about spatial orientation, body posture, equilibrium, reflex behaviours such as eye movement coordination, and navigation. This vestibular system that does inertial sensing is protected in the inner ear. The proper fusion of the camera and the IMU data can help us to mimic the human way of perceiving the environment. In this work, pose estimation is carried out by fusing camera and IMU data.

Camera based vision algorithms are very well suited for the AR application. However, based on the available processing power, the computation time for a vision based algorithm (other than fiducial marker tracking) can vary from several milliseconds to seconds. It is well known that the IMU provides faster pose updates than camera image based pose estimation. The next obvious measure to know about is the accuracy in terms of rendering an AR virtual content. The major disadvantage of IMU is that they typically suffer from drift. One well known method to overcome this drift is to use other low drift data (for example pose computed from the vision images) and then fuse the two data together. The fusion of vision relative pose and IMU absolute pose data will help to predict the object location assuming that the camera and IMU are very well calibrated. There have been different methods proposed in AR community for camera and IMU related sensor fusion [Azuma94, Chai99, Davison03, Hol06]. Nevertheless, one of the aspects that significantly affects the registration error for AR applications is the pose update rate from vision algorithm that will efficiently minimize the IMU drift. The focus of this paper is to define the different parameters that affect the rendering registration error and also to determine the pose update rate from the vision algorithm required to reduce the IMU drift and the registration error for an AR application. The experimental results show the impact of pose updates from the vision algorithm on IMU drift and registration error.

## 2 RELATED WORK

The usage of camera and IMU sensors is very well known in augmented reality applications for pose estimations. Starting from [Azuma94], there have been different work that discussed about the fusion of the pose from the camera and the IMU to reduce the registration error [Chai99, Davison03, Hol06]. IMU data has also been used in boosting vision feature matching and these features are used as an individual measurement as opposed to the more traditional approaches where camera pose estimates are first extracted by means of feature tracking and then used as measurement updates in a filter framework as in [Oskiper12]. Recently, Kriti et al [Kumar14] highlighted the usage of IMU data for the occlusion problem in vision algorithm by defining the fusion techniques. Another interesting application of augmented reality is binoculars [Oskiper13] using stereo camera, GPS and IMU for pose estimation. Their results show that the vision tracking algorithm computation takes 30 milliseconds which is more acceptable even without the need of an IMU. One different application of an IMU is used in sensing the user movement to correct the registration errors [Lo10].

In this work, the application of our interest is a maintenance operation in a complex and an extreme environment. A most recent work on a similar application is [Zhu14], which presented a whole system from tracking to rendering with the delay measurements at different modules. In this work IMU data was fused with vision algorithm pose using Extended Kalman Filter but there was limited highlight on the vision pose updates used for fusion and its effect on the registration error. The goal of our work is to find a trade-off between vision algorithm pose updates and the IMU drift to efficiently reduce the registration error. This will help to use only the required number of pose updates from the vision algorithm to have a good visualization of AR content at the correct location. Hence this optimizes the use of processing power that is required for vision algorithm computation.

In this paper, the different parameters affecting registration of AR visualization is discussed in section 3. In the subsequent sections, pose estimation using a camera and an IMU is elaborated. In section 6, the sensor fusion using extended kalman filter is explained. The experimental evaluation of determining the correct pose update rate from a vision algorithm required to reduce the drift and further the registration error is illustrated in section 7. The findings from this work and future direction is summarized in conclusion.

## 3 AR VIRTUAL CONTENT REGISTRATION

Augmented reality (AR) is user centric, where the user evaluates the system based on what he/she sees in the display device (either hand-held or head mounted display). Most common problem in rendering the AR virtual content is the registration between a real and a virtual object in the scene. The misalignment of the AR virtual content with their desired real world object is referred to as registration error. Figure 2 shows the registration error between real screws (dark gray) and the

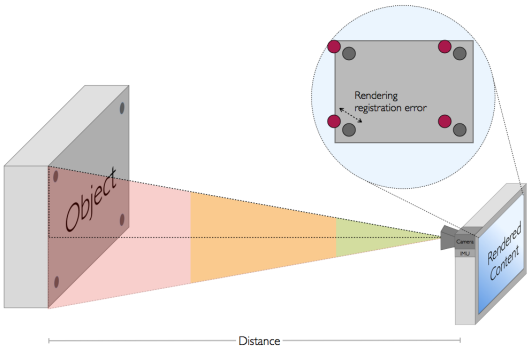



Figure 2: Augmented reality video display showing registration error in rendering the virtual object (red) on real object (dark gray) is shown.

virtual screws (red). This registration error must be kept quite minimal even during fast movements. Otherwise the system will not provide any help to the user. Assuming that the camera, IMU and the display device are calibrated to perfection, the registration error can be expressed as,

$$E = f(P_{cam}, D_{IMU}, D, S) \quad (1)$$

where, $P_{cam}$ is the accuracy of vision algorithm pose estimation, $D_{IMU}$ is the drift developed in the IMU over time, $D$ is the distance between the system (camera+IMU) and the target object (refer Figure 2 ) and $S$ is the size of the object of interest.

If the vision algorithm detects object A as object B (for example, blue cable detected as red in Figure 1) from the captured images, this will result in highlighting a different object. If the pose update from the vision algorithm is at every 1s, then between $t_0$ to $t_0 + 1s$ the drift accumulated in the IMU ($D_{IMU}$) can influence the registration error. If the distance ($D$) is increasing the error is also increasing, this can result in pointing A object (say red cable) as B (say blue cable) to the user. If the object of interest and the AR content is large (virtual box lid over the FPIAA box in Figure 1), a small registration error can be acceptable by the user. In other case, if the object of interest and virtual content is small, say highlighting a cable where there are identical cables next to each other, a small registration error can point a different cable to the user.

The main goal of this paper is to study the IMU drift related registration error developed over time and efficiently reduce that error using vision pose update. In order to have an accurate pose measurement from the vision algorithm, this work uses marker tracking instead of real object pose estimation. This will also help us to use marker pose as the ground truth and evaluate the error from other parameters (i.e. from the IMU drift).

As a foundation to the above discussed point, it is required to have pose estimation from a camera, IMU and sensor fusion to carry out the experiments. In the next sections, pose estimation using a camera and an IMU is discussed.

# 4 POSE ESTIMATION USING CAMERA IMAGES

In order to have accurate pose estimation from the vision algorithm, a prepared environment with fiducial markers are used. There are several libraries mainly used to resolve the marker tracking issues in the given scene. Most well known marker tracking libraries are ARToolKit [Kato99], osgART [Looser06], DWART [Bauer01], ARTag [Fiala04], Ubitrack [Ubitrack04] etc. In this work, marker tracking from Ubitrack[5] library is used. The precise marker tracking is possible when it is provided with the camera parameters (intrinsic and extrinsic) calculated using camera calibration[6] [Salvi02].

# 5 POSE ESTIMATION USING IMU

An inertial measurement unit has three accelerometers, three gyroscopes to measure acceleration and angular velocity along X, Y and Z axis. Some of the IMU also include three magnetometers to calibrate against earth's magnetic field. The cost and size of inertial sensors increase with the accuracy and reduced drift range. Following subsections detail the calculation of orientation and position from angular velocity and acceleration data from IMU.

## 5.1 Orientation calculation

The orientation in terms of quaternions was calculated using explicit complimentary filter (ECF) discussed by M. Euston et al in [Mahony08]. Please refer [Mahony08] for more details on orientation calculation. In the next section, position calculation from accelerometer data is explained.

## 5.2 Position calculation

The process of integrating acceleration data twice to calculate position is not so direct, since the accelerometer data contains the body acceleration and gravity component. As a first step, the gravity component was removed and the resulting linear acceleration was used for further processing. The absolute position is then calculated by double integrating this linear acceleration. The relative 6Dof pose from camera and absolute 6Dof from IMU is fused using the Extended Kalman Filter detailed in sensor fusion section.

[5] http://campar.in.tum.de/UbiTrack/WebHome
[6] http://www.vision.caltech.edu/bouguetj/calib_doc/

# 6 SENSOR FUSION

The fusion of pose data is performed with an Extended Kalman filter (EKF) [Bishop01] approach, a technique widely used in state estimation problems such as pose estimation in robotics, aviation and augmented reality [Azuma94, Chai99, Davison03, Hol06]. The kalman filter can be either used at the output of a pre-built sensor [Azuma94, Chai99] or as an integral part of the vision algorithm for pose estimation [Davison03]. The difference is based on the orientation representation, as in [Azuma94, Davison03] it is quaternion and [Chai99] uses euler representation. Most importantly the filter should cope up with unsynchronized pose data coming from the different sources particularly IMU and camera in our work. In this paper, marker tracking is used as vision algorithm since it provides accurate pose estimation which can serve as ground truth to measure the registration error developed by an IMU drift. Further for other applications this marker tracking can be replaced by any real object pose estimation algorithms within the same sensor fusion frame work detailed below.

## 6.1 Extended Kalman Filter (EKF)

A nonlinear version of the Kalman filter that linearizes an estimate of the current mean and covariance is referred as an Extended Kalman filter (EKF). This filtering technique allows us to estimate the parameters from multiple measurements without completely discarding information from previous sensor readings.

Let us assume that the process has a state vector $x \in R^n$, and is governed by the non-linear stochastic difference equation

$$x_k = f(x_{k-1}, u_k, w_{k-1}) \tag{2}$$

with a measurement $z \in R^m$ that is

$$z_k = h(x_k, v_k) \tag{3}$$

where the random variables $w_k$ and $v_k$ represent the process and measurement noise. In this case the non-linear function $f$ in the difference equation 2 relates the state at the previous time step $k-1$ to the state at the current time step $k$. It includes driving function $u_k$ and the zero-mean process noise $w_k$ as parameters. The non-linear function $h$ in the measurement equation 3 relates the state to the measurement $z_k$. In this case, the state vector consists of pose from camera and pose from IMU transformed to the camera coordinate frame using the camera-IMU calibration.

To estimate a process with non-linear difference and measurement relationships, we begin by writing new governing equations that linearise an estimate about equation 2 and equation 3

$$x_k \approx \tilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + W w_{k-1} \tag{4}$$

$$z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_k) + V v_k \tag{5}$$

where $\hat{x}_k$ is a posteriori estimate of the state at step $k$. $A$ is the Jacobian matrix of partial derivatives of $f$ with respect to x, that is

$$A_{[i,j]} = \frac{df_{[i]}}{dx_{[j]}}(\hat{x}_{k-1}, u_k, 0) \tag{6}$$

$W$ is the Jacobian matrix of partial derivatives of $f$ with respect to $w$,

$$W_{[i,j]} = \frac{df_{[i]}}{dw_{[j]}}(\hat{x}_{k-1}, u_k, 0) \tag{7}$$

$H$ is the Jacobian matrix of partial derivatives of $h$ with respect to $x$,

$$H_{[i,j]} = \frac{dh_{[i]}}{dx_{[j]}}(\tilde{x}_k, 0) \tag{8}$$

$V$ is the Jacobian matrix of partial derivatives of $h$ with respect to $v$,

$$V_{[i,j]} = \frac{dh_{[i]}}{dv_{[j]}}(\tilde{x}_k, 0) \tag{9}$$

The complete set of EKF equations is given in equations below,

**EKF Time update**

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k, 0) \tag{10}$$

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \tag{11}$$

the time update equations project the state and covariance estimates from the previous time step $k-1$ to the current time step $k$. $A_k$ and $W_k$ are the process Jacobians at step $k$ (see equation 6 and 7) and $Q_k$ is the process noise covariance equation at step $k$. Careful selection of process noise is a must to have good performance. The high value causes old measurements to decay quickly and new measurements are given a higher weighting.

**EKF measurement update**

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \tag{12}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0)) \tag{13}$$

$$P_k = (1 - K_k H_k) P_k^- \tag{14}$$

The measurement update equations (12 to 14) corrects the state and covariance estimates with the measurement $z_k$. $H_k$ and $V_k$ are the measurement Jacobians at step $k$ (see equation 8 and 9). $R_k$ is the measurement noise covariance at step $k$.

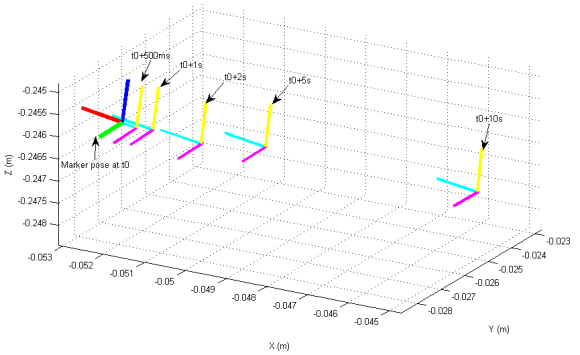


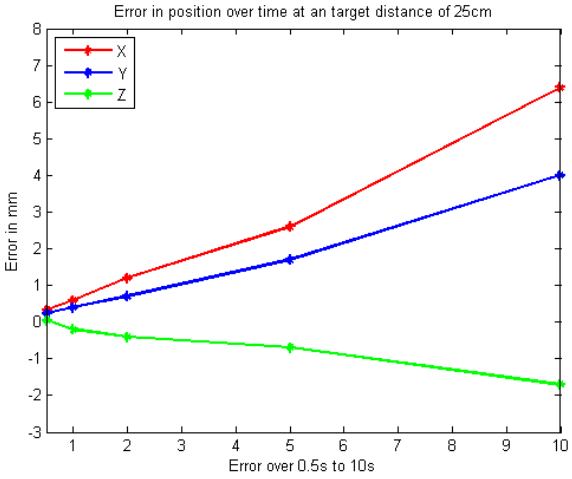

Figure 3: Error calculation over time at a target distance of 25cm. RGB is the marker pose (ground truth) and CMY is the sensor fusion pose after 500ms, 1s, 2s, 5s and 10s.

## 7 EXPERIMENTAL EVALUATION

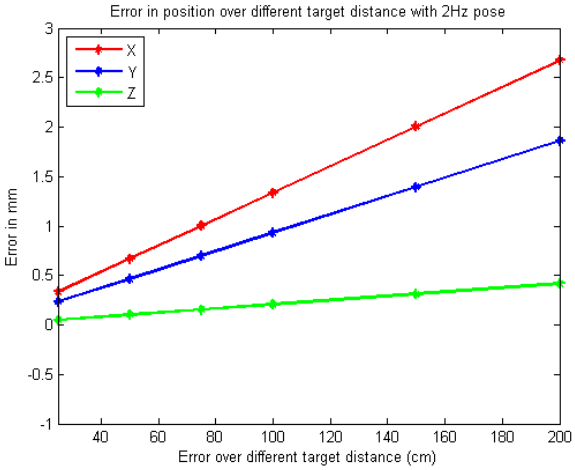
The framework is implemented in C and C++ and executed in Advantech MIO-5271 with Intel corei5-4300U CPU and 8GB of RAM. The sensors consist of a Logitech C920 camera and an Xsens MTi-100 IMU. Camera images are captured at 640x480 pixels resolution at 30fps and the IMU data at 100Hz frequency. The AR virtual content rendering is implemented using Unity3D[7] game engine.

This work concentrates on AR for a maintenance application as shown in Figure 1, the operating distance between system and target object can vary from 0.25 meter to 1.5 meter. The size of the object over which the AR content is rendered vary from 2cm to 8cm for the considered maintenance use case. The IMU drift related registration error was studied by varying one parameter and keeping all other constant. For example, the pose update rate from the vision algorithm to sensor fusion was varied to see the registration error produced due to the IMU drift under static conditions and at a fixed distance. The pose obtained from the marker remains as the ground truth pose that can be used to measure the drift from the IMU. The marker pose from camera images is taken at a fixed rate for fusing with the IMU data. The IMU pose through EKF is used to know the object location between the intermediate marker pose updates. The IMU drift developed in this duration is corrected using the upcoming marker pose from the camera. This drift over time was measured by comparing the marker pose with the sensor fusion pose. Since the camera and IMU are in a static condition (at 25cm away from the marker), the deviation between the marker pose and the sensor fusion pose is caused by the IMU drift. In the experiments, the marker pose is taken at 0.5s, 1s, 2s, 5s and 10s update rates. The drift between these updates are shown in the Figure 3 where the

[7] http://unity3d.com/




(a) Error in position over time at a target distance of 25cm.




(b) Error in position at different target distance with the 2Hz marker pose update.

Figure 4: Position error over time and at different target location.

marker pose is shown in RGB and sensor fusion pose is shown in CMY. The drift increases with the decrease in marker pose update rate, i.e. the drift is higher for every 5s update than the 2s marker pose update. As it can be seen in Figure 3, the deviation in orientation between marker pose (RGB) and the sensor fusion pose (CMY) is considerably lower than the position (XYZ) values. The closer look for the deviation in position (XYZ) values for different marker pose update rate is shown in figure 4 (a). This deviation in position is measured at a target marker placed 25cm away from the camera and the IMU. The registration error in terms of percentage along the three axis is given in Table 1.

 ISBN 978-80-86943-65-7

As mentioned in equation 1, the registration error is also related with the distance between the system and the target object. The influence of the different target distance and the related registration error is shown in Figure 4 (b). As the distance increases the registration error along all the three axis also increases steadily. These results are measured with the vision algorithm pose update of 0.5s to the sensor fusion since 2Hz pose update is possible in most real object pose estimation algorithms. Thus the careful selection of a vision algorithm is required to efficiently reduce the IMU drift and this in turn reduces the AR content registration errors. Further, this reduces the computation required for vision based algorithms to process several images per second to the number that is sufficiently enough to correct this registration error.

| Marker pose update (s) | X(%) | Y(%) | Z(%) |
|---|---|---|---|
| 0.5 | 0.64 | 0.82 | 0.02 |
| 1 | 1.23 | 1.54 | 0.067 |
| 2 | 2.4 | 2.69 | 0.147 |
| 5 | 5.9 | 6.99 | 0.29 |
| 10 | 12.2 | 14.33 | 0.686 |

Table 1: Registration error in percentage along X, Y, Z axis for different pose updates from marker.

The drift cannot be the same for different IMU. To test this statement, the above mentioned experiment was repeated again with another Xsens IMU and the measured registration error over time is shown in Figure 5. The result shows that the registration error developed over time due to IMU drift is purely random. The direction along which the error was happening is opposite in both the cases. The registration error was happening more in north east direction using the first IMU and it was moving in south west for the second case. Having said this, a method that reduces the registration error in the first case may not really serve as a solution for the other case. Thus, it is wiser to use other sources of information (vision algorithm in our work) to eliminate this error.

The maintenance application of our interest has a working distance of 0.5m to 1.5m. For this working distance, the IMU drift is 0.3mm to 2mm along the X axis with 0.5s marker pose updates and slightly less in other two axis. These measurements are taken under static conditions, but in real situation the system will be moved dynamically by the user. The size of the object on which the virtual rendering has to be shown is 2cm (screw in Figure 1). Considering these facts, we select the marker pose updates of 4Hz to be fused with the IMU data. The marker tracking pose at 4Hz frequency along with the sensor fusion pose is shown in Figure 6 . There is a slow and smooth transition of the sensor fusion pose between the two marker poses as shown in Figure 7 . The application of this pose data for a maintenance operation in ATLAS environment is shown in Figure 1. First two images show a cable operation and in next two images the box closure is highlighted by pointing the screw locations.

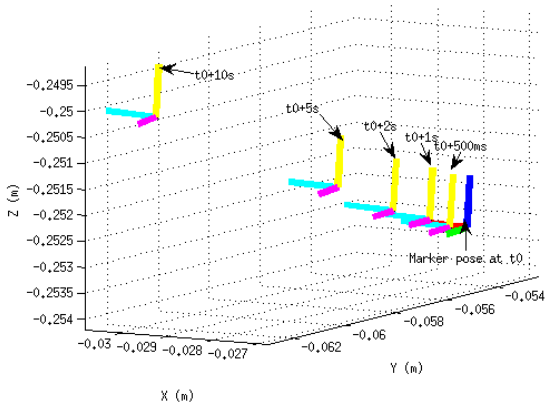
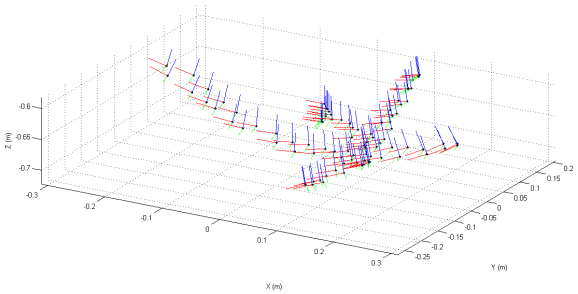



Figure 5: Error calculation over time at a target distance of 25cm. RGP is the marker pose (ground truth) and CMY is the sensor fusion pose after 500ms, 1s, 2s, 5s and 10s. Experiment performed with different IMU from Figure 3.

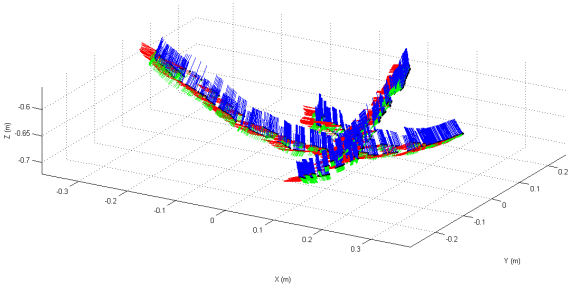
The camera and IMU together with the processor board mentioned above and a handheld video display showing the AR content was demonstrated to the users. They were asked to try with two similar setups, first one with only the marker tracking pose on all images (30Hz) and the second one with the marker pose at 4Hz and the IMU fused pose. The feedback was equal for both the setups with an advantage observed in fused pose setup during fast movements. This is because the motion blur in camera image due to the fast movements affected marker detection and the AR content was lagging behind or swimming around the corresponding real objects. Thus the efficient selection of marker pose updates rate for the fusion with the IMU data is performing better even in fast movements.

## 8 CONCLUSION

In this paper, the factors affecting the registration error in an augmented reality (AR) application was discussed. The real object pose estimation using camera and IMU was detailed with the formulation of sensor fusion using Extended Kalman filter (EKF). The advantage of having IMU data was to have fast pose updates. This data was rather affected by drift and needs to be corrected by the pose from a vision algorithm. Between this periodic correction of poses from a vision algorithm, the virtual object rendering was based on IMU data and the drift causes the registration error. This error in rendering can be defined as a function of pose accuracy from the vision algorithm, drift from IMU over



(c) Marker tracking pose at 4Hz.



(d) Sensor fusion with 4Hz marker pose and IMU data.

Figure 6: Marker tracking and sensor fusion results.

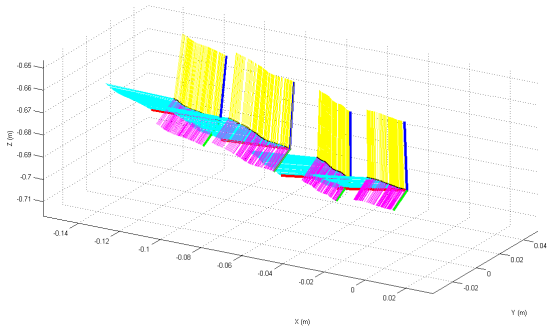


Figure 7: Close comparison of marker pose 4Hz (RGB) pose updates with the Sensor fusion pose (CMY).

time, size of the real object on which virtual content is overlaid and the distance between the target object and the system. Experimental evaluation detailed the effect of IMU drift on registration error over different pose update rates from the vision algorithm. The results also showed the effect of registration error over the distance and size factors. Based on the studies, a vision algorithm pose update rate that efficiently reduces the IMU drift was selected for the AR maintenance application. As a result, the registration error was minimized and secondarily that also optimized the processing power required for the vision algorithm computation.

The AR maintenance application was demonstrated to the end-user for their feedback with two setups, one with only marker tracking and the other using IMU fusion with fixed pose updates from the marker. Both the setup's performance was satisfactory during slow movements with the IMU fused marker pose setup showing a clear advantage during dynamic movements. In future work, the IMU drift during dynamic movements and its effect in registration error is in the pipeline to be studied by performing different experiments.

## ACKNOWLEDGEMENTS

The authors wish to thank all other members of the EDUSAFE consortium. This research has been supported by a Marie Curie Initial Training Network Fellowship of the European Commission FP7 Programme under contract number PITN-GA-2012-316919-EDUSAFE.

## 9 REFERENCES

[Oskiper12] T. Oskiper, S. Samarasekera and R. Kumar, Multi-Sensor Navigation Algorithm Using Monocular Camera, IMU and GPS for Large Scale Augmented Reality, IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2012.

[Kumar14] K. Kumar et al, An Improved Tracking using IMU and vision fusion for mobile augmented reality application, The International Journal of Multimedia & Its Applications (IJMA) Vol.6, No.5, October 2014.

[Lo10] C.-C. A. Lo, T.-C. Lin, Y.-C. Wang, Y.-C. Tseng, L.-C. Ko, and L.-C. Kuo, Using intelligent mobile devices for indoor wireless location tracking, navigation, and mobile augmented reality, in IEEE VTS Asia Pacific Wireless Commun. Symposium (APWCS), 2010.

[Oskiper13] Oskiper, M. Sizintsev, V. Branzoi, S. Samarasekera, and R. Kumar, Augmented reality binoculars, IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2013.

[Zhu14] Z. Zhu et al, AR-Mentor: Augmented Reality Based Mentoring System, IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2014.

[Feiner11] S. Henderson and S. Feiner. Exploring the benefits of augmented reality documentation for maintenance and repair, IEEE Transactions on Visualization and Computer Graphics, 2011.

[Ubitrack04] Newman, Joseph and Wagner, Martin and et al. Ubiquitous tracking for augmented reality, IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp.192-201, 2004.

[Mahony08] Euston, Mark and Coote, Paul and et al. A complementary filter for attitude estimation of a fixed-wing UAV, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008.

[Hol06] Hol, J.D. and Schon, T.B. and et al. Sensor Fusion for Augmented Reality, 9th International Conference on Information Fusion, pp.1-6, 2006.

[Bishop01] Bishop, Gary and Welch, Greg. An introduction to the kalman filter, Proceeding of SIGGRAPH, Course 8, 2001.

[Azuma94] R. Azuma and G. Bishop, Improving Static and Dynamic Registration in an Optical See-through HMD, in Proceeding of Siggraph, Orlando, July 1994, pp. 194-204.

[Chai99] L. Chai, K. N Guyen, B. Hoff, and T. Vincent, An Adaptive Estimator for Registration in Augmented Reality, International Workshop on Augmented Reality (IWAR), October 1999.

[Davison03] A. Davison, W. Mayol, and D. Murray, Real-Time Localisation and Mapping with Wearable Active Vision, IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2003.

[Kato99] H. Kato and M. Billinghurst, Marker tracking and hmd calibration for a video-based augmented reality conferencing system, International Workshop on Augmented Reality (IWAR 99), pages 85-94, 1999.

[Looser06] J. Looser, R. Grasset, H. Seichter, and M. Billinghurst, OSGART - A pragmatic approach to MR, IEEE International Symposium of Mixed and Augmented Reality (ISMAR), 2006.

[Bauer01] M. Bauer, B. Bruegge, G. Klinker, A. MacWilliams, T. Reicher, S. Riss, C. Sandor, and M. Wagner, Design of a component-based augmented reality framework, IEEE International Symposium on Augmented Reality (ISAR), 2001.

[Fiala04] Fiala, M, ARTag Revision 1, A fiducial marker system using digital Techniques, NRC Technical Report (NRC 47419), National Research Council of Canada, 2004.

[Salvi02] J. Salvi and X. ArmanguÃ© and J. Batlle, A comparative review of camera calibrating methods with accuracy evaluation, Pattern Recognition, 35(7), pp. 1617-1635, 2002.

[Corke07] P. Corke and J. Lobo and J. Dias, "An Introduction to inertial and visual sensing", The International Journal of Robotics, pp. 519-535, 2007.