Can Smart Devices Assist In Geometric Model Building?

Richard Milliken[1], Jim Cordwell[2], Stephen Anderson[2], Ralph R. Martin[1] and David Marshall[1]

Abstract—**The creation of precise three dimensional geometric models produced from visual information from low cost *smart* devices has many potential applications. There is a need to establish the capability of such devices, to determine if they are suitable. We explain how a typical smart device could be used in the creation of precise 3D geometric models from visual information, and establish a benchmark for evaluating such devices via a simulator which allows for parametric exploration of device capabilities.**

Various vision based algorithms (e.g. SLAM) could benefit from auxiliary sensor inputs, e.g. to solve the problem of determining the scale of the object being sensed. We discuss how the ancillary sensors in smart devices can assist in 3D mensuration to underpin 3D scene generation and 3D model building. We describe a number of tests performed on the inertial sensors integrated into consumer smart devices to ascertain their performance characteristics. These findings are then incorporated in our simulations. We also explore the extent to which future improvements to technologies are likely to be beneficial.

Our simulator allows modelling of a 3D scene, various sensors and the motion trajectory of the device whilst capturing data. We assess individual sensor outputs, and the quality of the fused sensor outputs in the context of 3D model building requirements.

Index Terms—**3D Model Building, Smart Devices, Kalman Filter, Structure from Motion, Inertial Sensor, Simultaneous Localisation And Mapping, Odometry**

I. Introduction

We would like to build three dimensional (3D) geometric models, potentially to milli-metric accuracy, using the low cost technologies that are found in today's *smart* devices. Portable consumer devices such as phones and tablets offer greater capability than ever before, particularly in terms of the hardware sensors now integrated, and used to advantage in a variety of applications. Alternatively bespoke portable devices may be created that take advantage of low-cost, readily available sensors. We have a particular interest in constructing CAD models—3D boundary representation (B-Rep) models of real world objects based on vertices, edges, and faces. Such models have uses in fields such as design, advertising, virtual reality, reverse engineering, and metrology.

Sensors commonly found on consumer-grade smart devices usually include a camera (albeit one having an optical system of limited quality), inertial sensors which can measure linear and rotational accelerations, as well as a GPS unit and magnetometer which can provide absolute location and orientation in the world. Furthermore, additional sensors are likely to soon become available: e.g. the Google "Tango" device has a 3D active sensor [1]. However, in this paper we restrict our investigation to currently available sensors.

In its simplest form, 3D model construction requires the capture of a collection of points at different depths relative to an arbitrary datum (e.g. the location of the observer). Stereo vision, using the principles of triangulation, is a well understood means of establishing the depth of a point from two different viewpoints. Triangulation requires that the baseline displacement between the two viewpoints is known; the error in estimated depth is directly proportional to the error in the measurement of the baseline displacement. A dedicated stereo sensor could in principle know this displacement to a high accuracy, whereas a single camera that is relocated arbitrarily would not: we need to measure the camera displacement between viewpoints. The question naturally arises as to whether the inertial sensors can help provide this information; we may also wonder whether the other auxiliary sensors can also help. As the accuracy of measuring the baseline displacement is a limiting factor on the precision of any constructed model, we wish to know how well that can be done with the inertial and auxiliary sensors in a smart device. The camera odometry—including the displacement of the camera—an be calculated by fusing the outputs from the sensor suite using an *extended Kalman Filter* (EKF) [2] (or some similar optimal estimator). The problem of extracting object structure from sequences of images is well known in the machine vision community, and is usually referred to as the *structure from motion* problem [3].

For the purposes of this research, two different commercial smart devices were selected and the basic performance characteristics of their in-built sensors were investigated. These characterisation experiments aimed to provide sufficient information to inform the simulation of typical operating characteristics. Once a basic characterisation of the sensors was established, simple sensor models were simulated, along with a dynamic model of the camera motion. This allowed an assessment of the overall sensitivity of the estimated camera odometry to changes in sensor performance. The simulation also provided simulated image based information fully synchronised with the camera motion, a capability not evident in the literature surveyed; this information can be used as a basis for investigating use of the sensors in structure from motion algorithms.

The rest of this paper contains sections discussing:

1) structure from motion, 2) sensor characterisation experiments, 3) the odometry extended Kalman Filter, 4) an assessment, and 5) conclusions.

The authors are with [1]the School of Computer Science & Informatics, Cardiff University, Queen's Buildings, 5 The Parade, Roath, Cardiff CF24 3AA, UK (MillikenR@cardiff.ac.uk, MarshallAD@cardiff.ac.uk, MartinRR@cardiff.ac.uk)

and [2]Renishaw PLC, New Mills, Wotton-under-Edge, Gloucester, GL12 8JR, UK (Jim.Cordwell@Renishaw.com, Stephen.Anderson@Renishaw.com)

II. Structure From Motion

Structure from motion involves determining 3D structures from 2D image sequences. Various categories of algorithm address this problem, one kind being *simultaneous localisation and mapping* (SLAM) maethods. They aim to concurrently estimate both the observed scene structure and the odometry of the observing sensor. We not describe these algorithms in any depth, but they are generally based upon the use of extended Kalman Filters [4], [5], [6]. Other formulations employ an unscented Kalman Filter [7] or a particle filter [8]. These algorithms have the characteristic that multiple sensors of differing characteristics may be used to different effect. In SLAM terminology, the 3D structure is often referred to as a *map*, populated by a set of *landmarks*.

The *MonoSLAM* [9], [10] formulation is both well documented and understood. Given a sequence of monocular images, it simultaneously reconstructs (within limitations) object structure along with the odometry of the camera. Its capabilities are compatible with our objective of constructing a 3D geometric CAD model. It also has the advantage that MATLAB code is freely available [10]. A limitation of *MonoSLAM* and similar algorithms is that they estimate depth and odometry up to a scale factor. The overall scaling has to be resolved by use of additional information, either determined from the observed scene or from an understanding of the camera motion as it captures the images. An *inertial measuring unit* (IMU) can in principle help to determine the camera rotation and displacement, by integrating the outputs from the gyroscopes and accelerometers respectively. Sensors such as GPS and magnetometers may also be used to provide supplementary information for fusion in the odometry EKF. The key requirement for the odometry is to measure relative distances; absolute position and orientation, as may be provided by GPS or a magnetometer, are not essential. The error in estimated point depths is proportional to the error in the camera displacement. Thus, changes over time in the camera displacement error will lead to temporal changes in point depths, potentially leading to distortion in the estimated object geometry. The use of an IMU in a SLAM formulation has been studied by a number of researchers [11], [12], but these studies did not examine the low-cost, consumer grade inertial sensors assessed in this study.

III. Sensor Characterisation Experiments

A. Introduction

The objective of sensor characterisation was to conduct some simple tests to establish the levels of random noise and bias present in the inertial sensors. These experiments were not intended as a detailed appraisal of the sensors, and so the tests have not followed the procedure outlined in [13]. The outputs of our experiments were intended to determine appropriate noise and bias statistics needed for the odometry EKF, for a particular sensor contained in a particular smart device. As we expect the device to be handheld in our application, and hence not on a rigid stable mount, some simple inertial stability tests were conducted to ascertain the likely consequences.

Device	***Accel.*** ($\mathrm{m \cdot s^{-2}}$)		***Gyro.*** ($^{\circ}\mathrm{s^{-1}}$)	
	Bias	***Noise***	***Bias***	***Noise***
Sony	0.103	0.164	0.642	0.290
Nexus	0.092	0.091	0.086	0.118

TABLE I: **Inertial Units Experimental Results**

The two readily available off the shelf devices selected for this work were

1) Google Nexus 7 (2013 model),
2) Sony Xperia S tablet (2012 model).

Manufacturers' data was readily available for the sensors in the Sony uni . However, those used in the Nexus could not be readily identified and thus the nominal design specifications of its sensors could not be established. Hence, the need for this test was particularly important to establish the basic performance characteristics of the Nexus sensors. The tests also provided confirmatory evidence for the Sony sensors.

B. Method

Static tests were conducted. The device was placed on a level surface and the accelerometer, gyroscope, GPS and magnetometer outputs were written to a file on the device. Under these conditions the only external input into the sensor is the acceleration due to gravity. Initially a custom Android application was used to collect the inertial data, but later tests used a freely available Android application. The tests were generally conducted for an hour, and repeated with the device in one of its three principal orthogonal orientations. Once the tests had been conducted, the saved files were copied to a PC for analysis. Given the general objective of this research, this was regarded as sufficient; for example, no attempt was made to assess the drift in sensor bias, or estimate any scale factor errors.

C. Observed Sensor Characteristics

We now describe the findings of our sensor assessment.

1) Inertial Units: We first outline the experimental findings concerning the inertial units on the two devices. In this simple analysis the bias was assumed to be the overall mean of the sensor outputs, and the noise was the standard deviation of those outputs.

Figures 1 and 2 are plots typical of the responses acquired from the Sony inertial sensors. The noise in both the accelerometer and gyroscope is clearly observable. There may be issues with the gyroscope—a number of transients were noted, and there was also an observable change in the level of noise. Table I provides a statistical analysis of the inertial data collected from the accelerometers and gyroscopes of the two devices. The results indicate that the Nexus IMU generally outperforms the Sony's; the latter under test performed better than indicated by the manufacturer's data sheet, particularly in terms of observed biases.

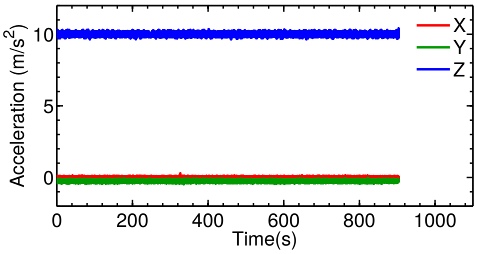


Fig. 1: **Accelerometer Output**

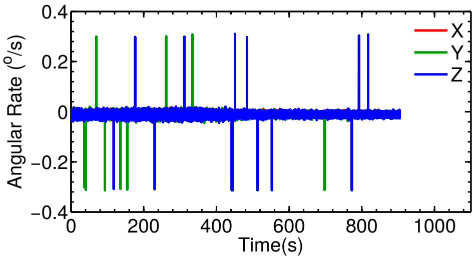


Fig. 2: **Gyroscope Output**

GPS Channel	*Device*	
	Nexus	*Sony*
Horizontal (m)	0.54	3.56
Altitude (m)	0.27	4.32
RSS (m)	0.61	5.60

TABLE II: **GPS Position Error Standard Deviations**

Magnetic Field (Unit Vector)			
Axis	*Mean*	*Std Dev'n*	*WMM 2010*
X (East)	−0.002	0.006	−0.0139
Y (North)	0.342	0.005	0.3993
Z (Vertical)	−0.940	0.002	0.9167

TABLE III: **Magnetic Sensor Field (Nexus)**

RMS Noise		
Sensor	*Static*	*Hand held*
Accel. ($\mathrm{m \cdot s^{-2}}$)	0.025	0.187
Gyro. ($^\circ\mathrm{s}^{-1}$)	0.021	0.855

TABLE IV: **Inertial Sensor Noise Test**

2) GPS and Magnetometer: In a similar manner, static tests were conducted on the GPS and magnetometer sensors on both devices. We define the GPS position error as the difference between the current measured position and the first position measurement; changes in position error are considered to be the GPS position noise. Table II indicates that the Nexus GPS unit provides a much more stable estimate of current position: for the device samples examined, the Nexus' position errors were nearly an order of magnitude better than those of the Sony device. Furthermore the stability of the Nexus unit was better than 0.2 metres for short periods of time (about 250 seconds), but steps of up to 0.5 metres in position were noted.

We also tested the magnetic sensor contained in the Nexus, recording the three orthogonal magnetic field components. The results, in µT, were then normalised to give a unit vector, as shown in Table III. This shows the unit vector field components alongside the corresponding components computed from the *World Magnetic Model, 2010* (WMM 2010) [14] for the location and time of the test. It is evident that there is approximate agreement between the measured field component values and those derived from the reference WMM 2010 model. However, for the purposes of subsequent filtering, it is the variation in field, and hence orientation, which is of importance, not the absolute estimate of orientation. An additional test was conducted where a metallic object was brought into the close vicinity of the magnetic sensor. This caused a substantial change in the magnetic unit vector, corresponding to a change in apparent orientation in excess of 100°. It is clear that the field was significantly distorted, thus limiting the use of such a sensor for this type of application, in for example an industrial environment.

D. Manual Stability Tests

As we expect the device to be handheld in our intended application, some basic tests were performed to try to establish the impact of such a mode of operation. A number of tests were performed where the operator attempted to hold the device steadily pointing in a fixed direction, and the responses from the accelerometers and gyroscopes were recorded. Table IV shows the impact of such a mode, where there is a substantial increase in the noise present in the sensor output compared to when the device is truly static.

E. Summary

1) The sensors in the Nexus clearly performed better than those in the Sony device. This may be expected as the Nexus is a newer device, and may well take advantage of advances in MEMS technology.
2) The GPS unit in the Nexus appeared to be substantially more stable than that noted for the Sony unit. However, the absolute positioning accuracy of these GPS units was not assessed.
3) There are some concerns over the stability of the temporal characteristics of the sensors contained in the Sony device.
4) As the sensor performance of the Sony device was generally inferior to that for the Nexus, further evaluation of the Sony device was not pursued.
5) The magnetic sensor is unlikely to be of practical use in this type of application.
6) Only one sample of each device was examined, and these samples may not be representative.

IV. Sensor Simulation

Our MATLAB simulation has three distinct compnents:

- A vehicle model which is intended to simulate the motion of the 'vehicle', i.e. tablet, that contains the sensors. The vehicle dynamic model is based upon the desired user determined rates of translational and rotational change, along with an Euler integrator to provide the updated true vehicle position (*S*) and attitude (Ω). Using dot notation to denote differentiation with respect to time,

$$S_{(t+\delta t)} = S_{(t)} + \dot{S_{(t)}}\delta t \tag{1}$$

$$\dot{S}_{(t+\delta t)} = \dot{S}_{(t)} + \ddot{S}_{(t)}\delta t \tag{2}$$

$$\Omega_{(t+\delta t)} = \Omega_{(t)} + \omega_{(t)}\delta t \tag{3}$$

where ω is the angular rate and δt is the time step.

- The sensor model calculates from the true vehicle motion the outputs of the sensors. At time *t*, the sensor output for the three principal axes, *Y(t)*, is given by

  $$Y_{sensor}(t) = Y_{true}(t) + Y_{rand}(t) + Y_{bias}(0) \tag{4}$$

  where

  $$Y_{sensor} \in \{\ddot{S}, \omega, S\} \tag{5}$$

  $$Y_{rand} = N(0, \sigma_{noise}) \text{ and } Y_{bias} = N(0, \sigma_{bias}) \tag{6}$$

  and $\ddot{S}, \omega, S$ represent the accelerometers, gyroscopes and GPS respectively, while $N(0, \sigma_{noise/bias})$ indicates zero mean Gaussian distributed noise.

- The simulation also has the capability to produce synthetic image features (points) that are representative of real world features and are suitable for processing in many vision applications, such as *MonoSLAM*. The motion of the synthetic features is fully synchronised with the vehicle model and how it relates to the model of the real world. Further details of this facility lie outside the scope of this paper.

V. Odometry: Kalman Filter

This filter is a classic 16 state EKF [2] whose function is to integrate the outputs from the accelerometers and gyroscopes to provide estimates of the current vehicle position (*S*) and orientation (Ω). The filter used a simple filter, and it is not really the purpose of this paper to dwell on any details of the mechanics of such a filter [2], but key aspects of the filter are highlighted. The filter states are defined as

$$X(t) = [Q, S, \dot{S}, \mathit{\Omega}_b, A_b] \tag{7}$$

$$\dot{X}(t) = [\dot{Q}, \dot{S}, \ddot{S}, \mathit{0}, \mathit{0}] \tag{8}$$

where

Q is the sensor attitude quaternion
$\dot{Q} = \frac{1}{2}\omega \cdot Q$
ω is the measured sensor angular rates
S is the vehicle position
$\dot{S}$ is the vehicle velocity
$\ddot{S}$ is the measured vehicle acceleration
$\mathit{\Omega}_b$ are the IMU gyroscope biases
A_b are the IMU accelerometer biases

Given the available sensors, the most useful additional data sources are the GPS and magnetometer units. In addition, a *zero velocity update* (ZUPT) was also incorporated, based upon the ideas presented in [15]. However, the *likelihood ratio tests* (LRT) described in [16] were not found to reliably determine zero velocity given the level of noise coupled with the relatively low rates inherent in practical handheld devices. An alternative scheme that relied upon the visual system being able to detect zero velocity was generally considered more suitable. This feature is easy to implement using the well understood techniques of optical flow [17]. Thus the augmented Kalman Filter measurement vector (Z) is

$$Z = \left[\, \mathit{\Lambda} \,\vdots\, x \,\vdots\, \dot{x} \,\right] \tag{9}$$

where

$\mathit{\Lambda}$ is the *North, East & Down* (NED) attitude unit vector, as given by the magnetometer.
x are the NED coordinates, as given by GPS.
$\dot{x}$ is the NED velocity ($\equiv$ 0 during a ZUPT cycle).

As previously noted the magnetometer was not considered reliable, so the orientation measurement vector (λ) can be removed from the measurement vector (Z). This means that there is now no longer any independent measurement of orientation, but the simplified measurement vector does mean that the Kalman Filter measurement matrix (*H*) is now linear.

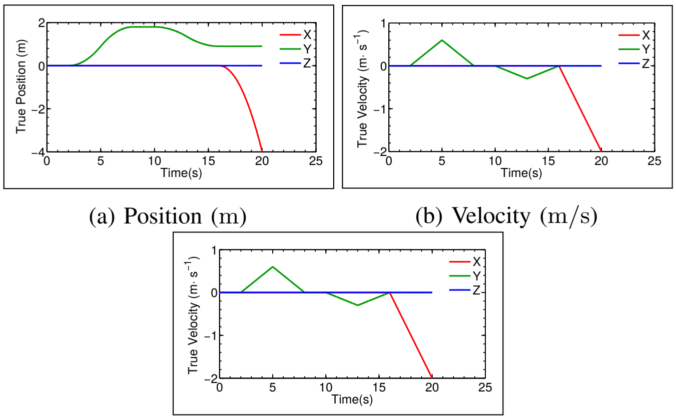
VI. Assessment

A. Experimental Method

Initially a number of experimental single shot runs were conducted to establish a working set of parameters for the Kalman Filter. The main assessment of the quality of the odometry produced by the Kalman Filter was via Monte-Carlo simulation, with sensor noise and biases randomized from run to run (usually, 1000 runs were performed), with sensor noise and bias statistics in accordance with the sensor being investigated. A pre-programmed trajectory was employed: a horizontal lateral motion along the x axis, firstly in one direction and then the other, was then followed by a longitudinal motion along the y axis. There was no vertical motion along the z axis. Details of the changes in position, along with the associated velocities and accelerations, are shown in Figure 3. The trajectory has three periods of zero velocity (one between 0 and 2 seconds, another from 8 to 10 seconds and an instantaneous one at 16 seconds). No rotational motion was present. Figure 4 shows a typical simulated sensor response for the Nexus device, where the noise and bias statistics are in accordance with those found during the sensor characterisation tests.

B. Results

1) Unfiltered Results: If the Kalman Filter does not go through an update cycle, then the filter acts as an inertial integrator. Tests indicate that if the unit is operated at the specification observed for the Nexus device (the baseline case), then after 20 seconds of operation, the results are very poor, with the error in position estimate being close to 20 metres. If the noise and biases of the Nexus inertial unit are reduced by two orders of magnitude, then unsurprisingly the position errors also reduce by about two orders of magnitude. An inertial unit of this quality is of a grade that might be found

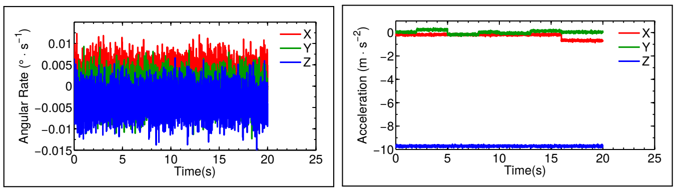


(a) Position (m)

(b) Velocity (m/s)

(c) Acceleration $(\mathrm{m/s^2})$

Fig. 3: **Simulated Trajectory**



(a) Angular Rate (°/s)

(b) Acceleration $(\mathrm{m/s^2})$

Fig. 4: **Simulated Sensor Outputs (Nexus)**

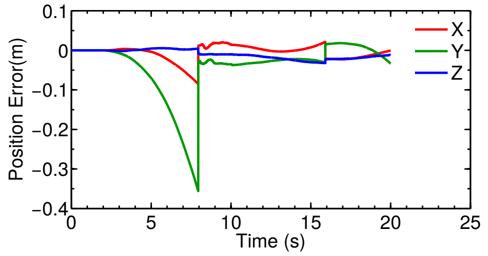


Fig. 5: **Position Estimate Error (ZUPT Filter)**

in aerospace applications, but is not currently achievable in present day high volume, low cost devices. The attitude errors were noted as being up to 1°at 2σ on the three axes after 20 seconds.

It is abundantly clear that the unaided inertial unit is not adequate, and, unsurprisingly, some form of filtering is essential.

2) Filtered Results: The previously indicated baseline case was run where the filter was updated by exclusively using the ZUPT update over the three separate intervals. Figure 5 shows the impact of the three ZUPT updates, where the overall positional error is now significantly improved. The Monte-Carlo results shown in Figure 6 confirm the significant improvement in performance offered by the ZUPT updates, but the position uncertainties change with time, which could be problematic. A case was run where the baseline sensor noise and biases was reduced by two orders of magnitude. In this case, the overall trend of the filtered position errors produced by ZUPT update remained as for the filtered baseline case, but the overall magnitude of errors reduced by approximately a factor of 10.

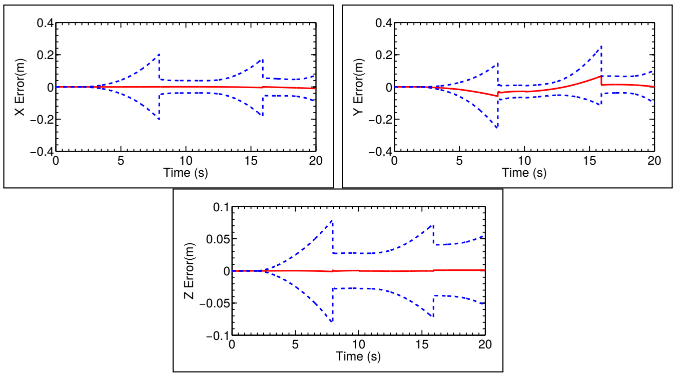


Fig. 6: **Ensemble Position Errors (ZUPT Filter)**

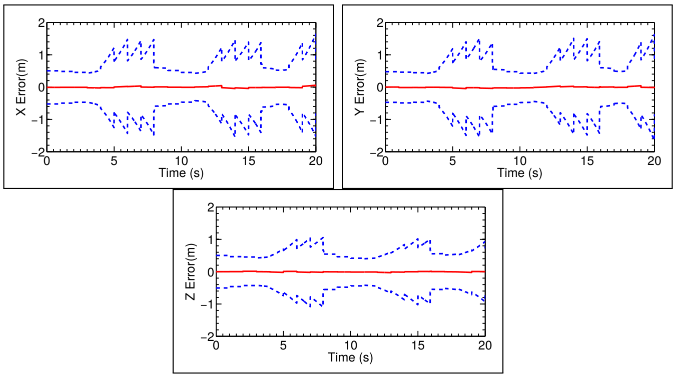


Fig. 7: **Ensemble Position Errors (ZUPT Filter + GPS)**

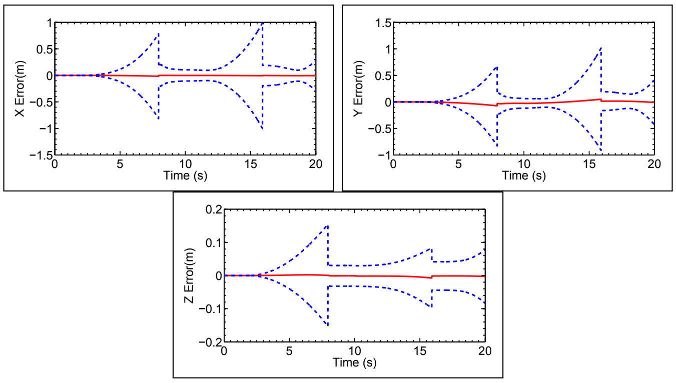


Fig. 8: **Ensemble Position Errors (ZUPT Filter + Shake)**

Many systems advocate GPS as source of a relatively high positional accuracy update [13]; Figure 7 shows the results of using GPS update alongside the ZUPT updates. The results indicate that the overall uncertainties in position are now influenced by the uncertainty in GPS position. The use of GPS did not improve any estimates of attitude.

In fact, a zero velocity condition would be quite difficult to achieve in practice with a handheld device. Figure 8 shows the case where during the period of zero velocity, the simulated accelerometer and gyroscope noise was increased to simulate hand-shake in the device during the period of ZUPT update. It is evident that the increase in noise has adversely affected the quality of the position estimates, which are now approximately five to six times worse than the baseline case in Figure 6. However, this result was sensitive to the initial conditions when the filter was initially estimating the sensor biases.

The results also indicate that the filtering generally did not have a uniform impact on the attitude estimation errors. The attitude errors were only reduced for the axes aroubd which vehicle moved (in this case x (roll) and y (pitch) axes), from about 1° (effectively the unfiltered attitude error) down to about 0.1° immediately after a ZUPT update. Like errors in displacement, an error in the estimated orientation has an impact on the estimation of the depth of a point when making 3D models of an observed scene.

VII. Conclusions

This study indicates that even low cost inertial sensors can be made to give what can be considered a good performance for applications such as a robot navigating around a car park. However, there are difficulties in directly using such sensors to provide scale for a structure-from-motion algorithm where much greater accuracy is required.

The use of the ZUPT update Kalman Filter was found to make a significant improvement in the quality of the odometry estimate, but requires the user of a hand held device to periodically remain still. However, the hand-shake likely to be present during a filter update would compromise the quality of the position estimates. The errors also increase during the period between filter updates which is likely to cause problems when trying to produce a geometric model. If the device was mounted on a rigid mechanism, such as a robot arm, then the zero velocity assumption could be better met, with a corresponding positive impact on position estimation performance.

The use of GPS did not significantly improve the quality of position or attitude estimates.

We may conclude that low-cost inertial sensors do not offer a simple means of addressing the SLAM scale factor problem to the requisite degree of accuracy needed to build a geometric model of an engineering object whose characteristic size is in the range 0.1–1 m. This work suggest that closer coupling with the visual system will be required. A vision system is able to measure well surveyed objects or other fiducial marks present in a scene. Such scene based knowledge will greatly assist in establishing the camera odometry to an accuracy that could potentially create geometric models to a high fidelity. The inertial sensors in such a system could be used in other ways than to provide absolute positions and orientations, and for example, could benefit a vision system in basic tasks such as guiding a visual object tracker.

Acknowledgements

The authors would like to thank the support of The Engineering and Physical Sciences Research Council (UK) and Renishaw PLC.

References

[1] "Google Tango," 2014. [Online]. Available: http://en.wikipedia.org/wiki/Project_Tango

[2] P. Zarchan and H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach*, 3rd ed. AIAA, 2009.

[3] R. Szeliski, *Computer Vision : Algorithms and Applications*. Springer, 2011. [Online]. Available: http://www.springer.com/computer/image+processing/book/978-1-84882-934-3

[4] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): Part I," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99 – 110, 2006.

[5] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 3, pp. 108–117, 2006. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1678144

[6] G. Dissanayake, S. Huang, Z. Wang, and R. Ranasinghe, "A review of recent developments in Simultaneous Localization and Mapping," *2011 6th International Conference on Industrial and Information Systems*, pp. 477–482, Aug. 2011. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6038117

[7] S. Julier and J. Uhlmann, "Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations," *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, vol. 2, 2002.

[8] M. Montemerlo, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Jul. 2003. [Online]. Available: http://www.ri.cmu.edu/pub_files/pub4/montemerlo_michael_2003_1/montemerlo_michael_2003_1.pdf

[9] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4160954

[10] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-Point{ RANSAC} for {E}xtended {K}alman {F}iltering: Application to real-time structure from motion and visual odometry," *Journal of Field Robotics*, vol. 27, no. 5, pp. 609–631, 2010. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/rob.20345/fullhttp://webdiis.unizar.es/~jcivera/code/1p-ransac-ekf-monoslam

[11] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004, pp. I—-652. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1315094

[12] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 519–535, 2007. [Online]. Available: http://ijr.sagepub.com/content/26/6/519.short

[13] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed. IET, 2005.

[14] "WMM2010," British Geological Survey, 2010. [Online]. Available: http://www.geomag.bgs.ac.uk/data_service/models_compass/wmm_calc.html

[15] E. Foxlin, "Pedestrian Tracking with Shoe-Mounted Inertial Sensors," *IEEE Computer Graphics and Applications*, vol. 25, no. 6, pp. 38–46, Nov. 2005. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1528431

[16] I. Skog, P. Händel, J.-O. Nilsson, and J. Rantakokko, "Zero-velocity detection — an algorithm evaluation." *IEEE transactions on bio-medical engineering*, vol. 57, no. 11, pp. 2657–2666, Nov. 2010. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5523938http://www.ncbi.nlm.nih.gov/pubmed/20667801

[17] B. K. P. Horn and B. G. Schunck, "Determining optical flow-a retrospective," *Artif. Intell. v59*, pp. 81–87, 1994. [Online]. Available: http://deepblue.lib.umich.edu/bitstream/handle/2027.42/30981/0000654.pdf?sequence=1