Journal of Global Positioning Systems (2011)
Vol.10, No.1 :3-10
DOI: 10.5081/jgps.10.1.3

Journal of Global Positioning Systems
2011
© CPGPS

Low Cost Vision-Aided IMU for Pedestrian Navigation

Chris Hide[1], Terry Moore[1], Tom Botterill[2]
[1] *IESSG, University of Nottingham, Nottingham, UK*
[2] *Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand*

Abstract

Modern smartphones contain a number of sensors that can be used for navigation when GPS signals are unavailable. Low cost MEMS gyros and accelerometers are increasingly becoming available in modern devices, however when used for positioning, they typically result in large errors after very short periods of time. This paper investigates using measurements from a computer vision algorithm that uses successive frames from a camera approximately looking at the ground to compute the translation between frames. The measurements can be used to control the drift of inertial sensor measurements when measurements from GPS are not available. The concept is convenient since it uses sensors already available on smartphones and pedestrians will naturally hold the smartphone in the required position when using it for navigation. This paper demonstrates that computer vision measurements can significantly reduce the drift of inertial-only positioning for pedestrian navigation in areas where GPS is unavailable. Issues such as computational requirements and operation in low light areas are also discussed.

Keywords: inertial, computer vision, integration, GPS, indoor, navigation

1. Introduction

Indoor pedestrian navigation is a difficult problem due the unavailability of accurate Global Navigation Satellite System (GNSS) signals. Furthermore, it is desirable an indoor positioning system should require no additional infrastructure due to the cost and expense of installation and maintenance. Other issues for indoor navigation include the strong requirement for using low cost sensors, therefore it is desirable that we can make use of the sensors that are already available on modern smartphones. Such sensors include GPS, Wi-Fi, Bluetooth, mobile phone network, microphones, gyros, accelerometers, magnetometers, light sensors and cameras, all of which make measurements that may be useful for navigation.

Low cost MEMS gyros and accelerometers are frequently considered as a potential solution for indoor navigation. However, the reality is that such sensors are only sufficient to provide positioning for very short periods of time (typically a few seconds) because of sensor errors such as biases and scale factor errors that are not necessarily constant over time and difficult to separate from the true navigation signal (Hide, 2003). However, if external measurements are available, these can be used to restrict the drift. Low cost MEMS sensors have recently been demonstrated to provide useful levels of performance through innovative ideas such as mounting an Inertial Measurement Unit (IMU) on a user's foot and using zero velocity updates every time a user takes a step such as in Foxlin (2005). This frequent application of accurate velocity measurements has demonstrated that even low cost sensors can provide potentially useful position. Using this algorithm, the main issue that influences position accuracy is yaw accuracy, which is based on the quality of gyro used and accuracy of any aiding sensors used such as magnetometers (Abdulrahim, 2010).

One source of external measurements to aid a low cost IMU comes from the computer vision community. Cameras can be used to provide measurements such as translation and rotation between frames by tracking features in successive images. This paper looks at the use of aiding measurements from a camera attached to an IMU where the user is walking with the device held out in front of them with the camera pointing approximately towards the ground. This is a typical use of smartphones where the user is reading navigation information from the display. The camera therefore has a view of the ground beneath, and immediately in front of the user. Sequential images can then be used to compute the 3-dimensional body frame translation direction of the camera as well as 3-d rotation. The images can be captured at a relatively low rate (a few per second) provided sufficient common features exist between successive frames. From a single camera alone, the body-frame translation can only be computed up-to an unknown scale factor, however if the height of the camera above the ground is known, the absolute velocity of the camera can be computed. This

measurement is used to correct the drift of the IMU-derived position.

This paper investigates the use of computer vision derived velocity measurements to frequently correct the drift of a low cost IMU. The computer vision algorithm provides a 3-d camera frame translation which, when scaled by the height of the camera above the ground, provides a 3-d camera frame velocity measurement which is very closely related to the IMU body frame. The measurements are combined using a Kalman filter that models the errors of the inertial sensor including position, velocity, attitude and sensor biases. An initial guess for the camera height can be taken from the average height that a user holds the camera, and the estimate is refined by adding an additional state to the Kalman filter which can be estimated when measurements from sensors such as GPS are available. The paper extends the work previously presented in Hide (2010), where the algorithm was previously demonstrated with a tactical grade IMU.

In this paper, the algorithm is tested using real-world measurements from a Microstrain 3DM-GX3 IMU attached to a commercial off-the-shelf camera. It is demonstrated that during a 6 minute outage, the maximum position error experienced is 2.9m for the integrated IMU and computer vision system, whereas position errors are greater than 200m after only 1 minute using IMU-only. These results are derived from a trial walking around the perimeter of a tennis court which highlights the characteristics of the errors involved which are shown to be yaw drift and scale errors. The positioning performance from such a system would be suitable for integration with other systems such as GPS when available or Wi-Fi position estimates such as from fingerprinting algorithms (see, for example, Mok and Retscher). Such a combination of sensors, particularly when combined with map matching, could provide high accuracy indoor navigation using sensors that are available on smartphones. Issues such as using the camera in low light conditions and the computational requirements of the computer vision algorithm are also discussed.

2. Inertial Navigation

Inertial Navigation provides the foundation of the proposed algorithm. An IMU is used that consists of three gyros and accelerometers that are used to compute the position and orientation of the mobile device. The process of integrating the gyro measurements to generate attitude, and combining the attitude with double integrated accelerometer measurements is known as the INS mechanisation, and is described in, for example, Hide (2003), Titterton and Weston (2004) and Farrell and Barth (1999).

In order to start the INS mechanisation, it is necessary to know the initial position, velocity and attitude of the IMU. Obtaining the initial position and velocity is trivial if GPS measurements are available. However this is not always the situation if operating indoors and this becomes a significant limitation of the technology. The initial attitude is also not trivial to compute. The roll and pitch of the IMU can be computed from the accelerometers by comparing the measurements to the gravity vector assuming that gravity is the only force being measured. Obtaining the heading of the IMU is a more difficult task since the gyros are not sensitive enough to measure the rotation of the Earth which is commonly used for initialising higher quality devices (although such techniques are still impractical for pedestrian navigation considering the time required for alignment). Instead, a 3-axis magnetometer can be used to initialise heading, although this can be very inaccurate when operating in areas with large magnetic disturbances.

A Kalman filter is used to estimate the navigation and IMU errors. The state vector is defined as:

$$x = \begin{pmatrix} \delta p & \delta v^n & \delta \phi & \delta g^b & \delta a^b \end{pmatrix}^T \quad (1)$$

where δp is vector of latitude, longitude and height errors; δv^n is the vector of navigation frame velocity errors; $\delta \phi$ is the navigation frame rotational misalignment; δg^b is the vector of gyro bias errors; and δa^b is the vector of accelerometer bias errors. The Kalman filter is used to estimate the errors using a linearised inertial navigation model such as that described in Titterton and Weston (2004). The model describes the interaction between different error states and can be used to estimate the full state vector using position or velocity measurements and sufficient dynamics. Dynamics are required in order to separate some of the error states; for example, heading error can only be estimated if there is sufficient horizontal acceleration when using position and velocity measurements from GPS (Hide, 2003).

The filter is used in feedback form so that when a measurement is available from a sensor, the error is computed using the Kalman filter which is then used to correct the inertial sensor measurements and navigation parameters. This is to ensure the navigation errors remain small and hence approximately linear. More information on Kalman filters and Kalman filters for inertial navigation can be found in Hide (2003), Titterton and Weston (2004) and Farrell and Barth (1999).

3. Computer Vision

This section describes how computer vision is used to compute the motion of the camera which will be used to aid the IMU. The camera is pointed roughly towards the ground, and a sequence of images is captured. The ground is assumed to be approximately planar. For each image, the relative position and orientation of the camera is estimated relative to its position when it captured the previous image. The images contain features from the ground plane, but also features from the pedestrian's moving legs, feet and shadow. The camera's relative position and orientation are computed by matching features lying on the ground plane with the same feature from previous frames. The algorithm also has to automatically detect and avoid incorrectly matched features, and matches between moving features and features not lying on the ground plane..

When an image is captured, the first stage is to detect point features in the image. The FAST corner detector (Rosten et al, 2010) is used to detect approximately 300 points in each image which are likely to be detected in other images showing the same scene. The area around each FAST corner is described using a small patch of the image: a patch sized 27x27 pixels centered on each corner is scaled down to 9x9 pixels. The similarity of two of these 9x9 patches is measured by computing the sum-of-squared differences (SSD) between corresponding pixel values.

Each patch feature from an image is matched to the most similar patch feature in the previous image. These feature matches ('correspondences') are found by computing the SSD between all pairs of patch features, and choosing the closest match to each (although a more efficient procedure could be used, such as a kd-tree as described in Beis et al; 1999). When a patch feature appears similar to several patch features in the other image, all possible matches between pairs of patch features are used as correspondences.

Many of these correspondences will give the location in each image of some feature visible in both images. Each detected feature location is transformed using the camera's calibration matrix, and shifted to correct radial lens distortion. When these correspondences also lie on the ground plane, they are related by a perspective homography, H, which is a 3x3 matrix mapping homogeneous point locations in one image, $(x,y,1)$, to homogeneous point locations in the other image, $(x',y',1)$ (following normalization so that the third component is one). H can be computed from four or more correspondences using the Discrete Linear Transform, or DLT (Hartley and Zisserman, 2003), a least-squares approach.

Some correspondences are not on the ground plane however, and many others will be incorrect matches caused by similar-looking features, and matches between moving features. These outlier correspondences must be removed before a least-squares approach can be used. To remove outliers while simultaneously fitting a homography to inliers, the BaySAC framework is used (Botterill, 2009). BaySAC is based on the RANSAC framework (Fischler and Bolles, 1981), but enables matches between multiple similar-looking points to be used efficiently. To compute H using RANSAC, many random subsets ('hypothesis sets') of four points are selected. Each hypothesis set is used to generate a homography. The number of correspondences compatible with each homography is counted. When a homography compatible with many correspondences is found, this model is usually correct, and the correspondences found are inliers.

In BaySAC, hypothesis sets are selected based on the prior inlier probabilities of each match (estimated from the number of potential match candidates) and the history of hypothesis sets which have been tried. This reduces the number of iterations needed to find a good hypothesis set, and can efficiently remove low-quality matches between multiple similar features. Once a homography and inlier set are found, the inlier set is refined by using the DLT to fit a new homography to all of the inliers found, then re-computing which points are compatible with the new model. The DLT is then used again to fit a homography to all of these points.

The homography, H, has the property that

$$H = R + d^{-1}\boldsymbol{t}\boldsymbol{n}^T \tag{2}$$

where R is the rotation from the previous camera location, t is the camera motion vector, n is a unit vector normal to the ground plane, and, and d is the distance between the camera and the ground. H is decomposed to find R, n and t/d, using Levenberg-Marquardt's algorithm (Hartley and Zisserman, 2003), t is calculated from the estimated height of the camera above the ground.

Occasionally, when few matches between frames are correct (for example because there are no distinctive features on the ground, or motion is too fast and either frames contain motion blur, or consecutive frames do not overlap), BaySAC will fail to compute a homography compatible with many correspondences. In this case, no update will be made and we will rely on the INS-only solution. By only accepting estimates compatible with many correspondences, incorrect measurements are avoided.

4. Integration

In order to use the computer vision measurements to correct the drift of the IMU, it is necessary to develop observation equations that relate the computer vision measurements to the INS error states modeled in the Kalman filter. For this work, we do not consider the rotational information from the computer vision algorithm since the measurements from the MEMS gyros used are considerably more accurate. Instead, we focus on using the translation information to restrict the drift of the IMU.

For the camera measurements, we use the following error model:

$$\tilde{v}^b = (1+\delta s) C_c^b v^c + e_v \tag{3}$$

where $\tilde{v}^b$ is the estimate of the IMU body frame velocity from the camera; v^c is the true velocity in the camera frame; δs is the scale factor error for the camera height; e_v is the measurement noise; and C_c^b is the rotation matrix from the camera frame to the body frame. Since, for this work, we assume that the axes of the IMU and the camera are perfectly aligned, we set C_c^b equal to the 3 by 3 identity matrix. Again for simplicity and considering the accuracy of the sensors used, we do not consider the offset of the camera axes to the IMU axes. Both of these parameters can potentially be calibrated using algorithms such as those used for boresight calibration in aerial photogrammetry such as in Tao (2007).

Following the derivation in Shin (2005) to use vehicle frame measurements, and assuming no axes offset, we have the following INS error equation:

$$\hat{v}^b = \hat{C}_n^b \hat{v}^n \tag{4}$$

$$\approx C_n^b \left[I + (\delta\phi\times) \right] (v^n + \delta v^n) \tag{5}$$

$$\approx v^b + C_n^b \delta v^n - C_n^b (v^n \times) \delta\phi \tag{6}$$

where $\hat{}$ indicates a predicted value from the IMU, v is the velocity with the superscripts b and n denoting the body and navigation frame respectively; and C_n^b is the direction cosine matrix from the navigation frame to the body frame. Therefore the observation equation can be formed as the difference of the IMU and camera velocities:

$$\delta z = \hat{v}^b - \tilde{v}^b \tag{7}$$

$$= C_n^b \delta v^n - C_n^b (v^n \times) \delta\phi - v^b \delta s - e_v \tag{8}$$

This observation equation is used to relate the body frame measurements from the IMU and camera to the states that are being estimated in Equation 1; therefore Equation 1 is extended to include the scale factor error term.

5. Experiment

An experiment was conducted at the University of Nottingham UK in January 2011 to test the developed algorithm. The equipment used is shown in Figure 1 which consists of a back pack containing a GPS receiver, data logger, power supply and a handheld IMU attached to a camera. The data logger used was a Precise Time Data Logger (PTDL) developed by the Geospatial Research Centre in New Zealand. The PTDL is able to record measurements from different sensors and log the data to SD card with an accurate GPS time stamp using an integrated u-blox ANTARIS 4 high sensitivity GPS receiver. The u-blox receiver recorded single frequency pseudorange, Doppler and carrier phase measurements which were used to generate a differential solution using data from the National GPS Network Ordnance Survey network at Keyworth forming a baseline of approximately 10km.

Figure 1: Experimental equipment

A Microstrain 3DM-GX3-25 MEMS IMU was used with the particular model having a rotation range of ±1200deg/s and an accelerometer range of ±18g. The IMU is typical of the current range of calibrated low cost MEMS sensors with gyro bias stability quoted as ±0.2deg/s for the equivalent 300 deg/s model and accelerometer biases of 0.01g. The IMU was fixed to a Canon IXUS 750 digital camera as shown in Figure 1 using the edge of the camera display for alignment. The small misalignment between sensor axes of the camera and IMU was not calibrated due to the accuracy of the sensors used. The camera was used to record 640x480 pixel images at 15 frames per second and the IMU sample rate was approximately 100Hz.

While the GPS and IMU measurements were synchronized using the PTDL, no timing information was available for synchronising the images from the camera.

Instead, software was developed in MATLAB to perform a cross correlation of the z-axis gyro measurement with the z-axis camera rotation estimate from the computer vision algorithm. The maximum cross correlation coefficient for different time offsets was calculated using 20 second sections of data at regular points throughout the dataset. A linear model was fitted to this data which was used to align the computer vision and IMU datasets. While this approach is not suitable for real-time application, hardware synchronisation of IMU, GPS and camera data is possible and could be developed for future research.

A dataset was collected at a tennis court at the University of Nottingham. The camera was initially still for 10 seconds, after which a user walked five times around the outside of a single tennis court. The data was collected outside so that GPS measurements were available throughout the test to assess the performance of the integrated solution. Furthermore, a rectangular trajectory was selected to minimise the effects of heading drift on the solution, this is discussed further in the following sections.

The datasets were processed using software developed at University of Canterbury and University of Nottingham. Computer vision software was developed to compute the translation and rotation of successive frames using the algorithm described in this paper. The translation information was time stamped using cross correlation and scaled by an approximate initial camera height with the results written to a log file. Every 2nd image from the 15 frame per second video were used to reduce the processor requirements since each image contains a significant amount of overlap. The IESSG's POINT GNSS and INS integration software was modified to include the error model described in this paper. Files containing the GPS position and velocity and the computer vision log file were used in the software to correct the drift of the IMU. The results are presented in the following sections.

5.1 Results

Figure 2 shows an example of the computer vision algorithm operating as the user walks around the tennis court. On the image, the detected features are shown as red dots. The narrow red lines drawn on the figure show the incorrect feature correspondences between the current image and features detected on the subsequent image. These correspondences have been rejected by BaySAC. The dark blue lines show the correspondences identified as inliers by the BaySAC algorithm. These inlier correspondences are used to estimate the homography between the images, and hence the translation and rotation of the camera between images. The algorithm is shown to work robustly even on images that contain many features which are not on the ground plane, as well as on textures that contain many self-similar features. The average processing time on a 3GHz desktop PC using a single processor core was 120ms per pair of frames, which means that the algorithm is able to run slightly faster than real-time using a video frame rate of 7.5 frames per second. The optimisations which would be needed to attain real-time performance on the less powerful processors typically found in smartphones are considered in Section 3.2.

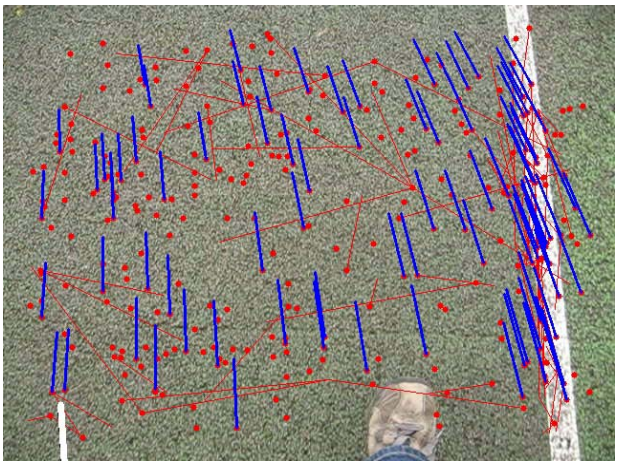

Figure 2 Computer vision algorithm (red dots are detected features, blue lines are inlier correspondences between frames, red lines are outliers).

Figure 3 shows the trajectory of the experiment. The reference solution is generated using the differential GPS position solution from the u-blox ANTARIS receiver. The computed standard deviation from the GPS processing software indicates the position accuracy to be between 0.1 and 0.5m. The GPS antenna and IMU are not collocated, therefore the position difference is not an exact estimate of the position error, however, given the accuracy of the sensors used, it is sufficient to provide a reference trajectory for this experiment.

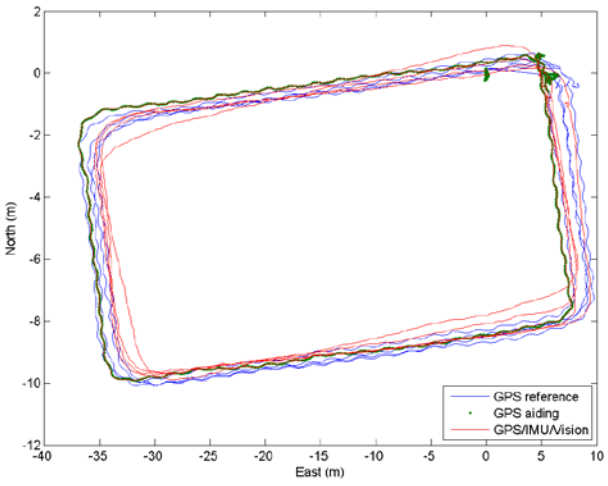

Figure 3 Trajectory around tennis court

The POINT integration software generates an INS solution that is corrected by measurements from the GPS and computer vision sensors. The camera measurements are used constantly in the filter whereas the GPS measurements are used for the first 97 seconds which corresponds to a short stationary period followed by one single circuit of the tennis court. Therefore, the GPS measurements are available to initialise the INS and help estimate the initial gyro and accelerometer biases, initial heading error and camera height. The INS heading was initialised using a coarse estimate from the magnetometer. Figure 3 shows the GPS measurements that are used in the integration filter from the first circuit of the tennis court as green dots.

Figure 3 also shows the integrated position solution in red. For the first circuit, the position estimate is close to the GPS position solution as the GPS measurements are being used in the filter. Once GPS measurements are removed, the INS and computer vision measurements are able to maintain the trajectory for the following 6 minutes until the end of the dataset. The position error for the experiment is shown in Figure 4 and Table 1. Figure 4 shows that the horizontal position error oscillates according to the position the user is walking round the circuit, but in general, the position error slowly grows with time. The maximum horizontal position error after 6 minutes without GPS corrections is 2.9m which is a significant result since this position accuracy would enable a range of applications where GPS signals are unavailable.

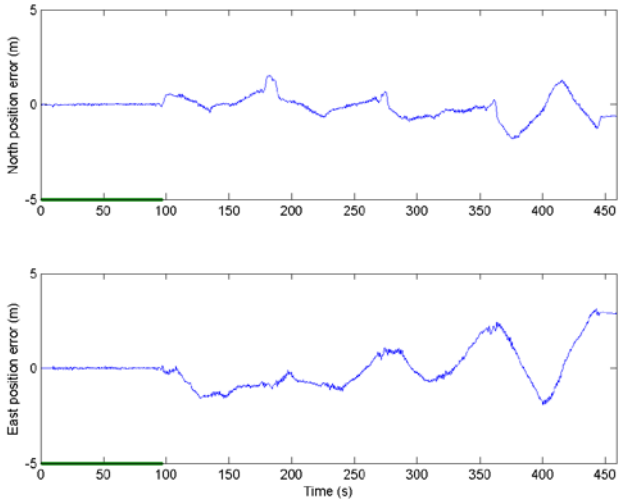

Figure 4 Position error for tennis court experiment

The experiment was carried out by walking around a tennis court which was designed specifically to explore two significant characteristics of the position error. Firstly, Figure 3 shows that the integrated solution appears to be slightly shorter than the reference trajectory. This is due to scale error where the exact height of the camera above the ground is not known. Secondly, the trajectory shows that the error of the integrated position solution increases as a result of yaw error since the track of the trajectory is not following the same direction in the latter circuits. This is because the yaw error of the INS is not observable through using the translation measurements from the camera. These two error characteristics are exposed to be the main issues for the proposed method of sensor integration.

Table 1 . Comparison of Horizontal Position Errors

	Outage length (s)	North error (m)	East error (m)	Horizontal error (m)
INS only	60	212	93	231
	120	595	663	891
	180	1474	2359	2782
	240	2723	5678	6297
	300	4349	10990	11819
	360	6551	18140	19287
Computer vision aided-INS	60	0.1	1.0	1.0
	120	-0.4	0.8	0.9
	180	0.0	0.8	0.8
	240	-0.3	0.5	0.6
	300	-0.6	1.4	1.6
	360	-0.6	2.9	2.9

Figure 5 shows the estimated scale factor for the integrated solution. The scale factor represents the percentage error in the height of the camera which is approximately 1.4m above the ground. After the user moves, at approximately 9 seconds, the scale factor is able to be estimated using the GPS measurements in the Kalman filter. However, when the GPS measurements are unavailable, after 97 seconds, the scale factor estimate slowly decreases with time compared to the scale factor estimate generated by integrating with GPS throughout the dataset. This scale factor issue can also be seen in Figure 3 where the scale of the trajectory is slightly shorter than the GPS solution. Further analysis is required to understand this change in scale factor when GPS measurements are unavailable.

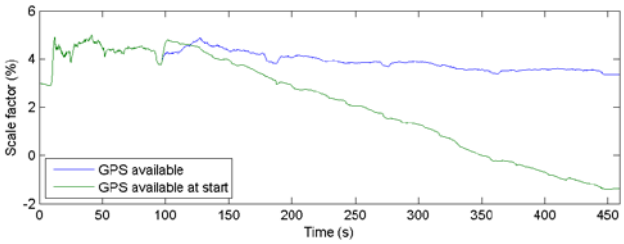

Figure 5 Estimated scale factor for tennis court experiment

6. Discussion

So far we have demonstrated that the computer vision measurements are able to significantly reduce the drift of

the low cost IMU when GPS measurements are unavailable. The aim of this work is to provide a pedestrian positioning system that can work inside or close to buildings where GPS is unavailable or significantly degraded. However, two issues remain before this work can be used for this application.

Firstly, as we have already described, the computer vision algorithm is relatively computationally expensive. Some simple modifications such as reducing the frame rate are already used in this paper, however further reduction in frame rate is not possible since the resultant images may not have sufficient overlap. The most computationally expensive part of the algorithm is finding candidate correspondences by matching features between frames. One possible optimisation would be to more closely integrate the INS and computer vision algorithms so that the predicted translation and rotation from the IMU could be used to predict where matching features will be found, hence limiting the search area. Such closer integration between the IMU and computer vision algorithm will be the focus of future studies.

A second significant issue is the problem of using the camera in low light conditions such as those that can occur indoors. In this situation, insufficient features are matched between frames, BaySAC fails to compute a relative position and orientation, and no position update will be available. The reason too few features are matched is that the camera used for this work has no control over the exposure time or sensitivity used, and the long exposure used when light levels are low leads to significant motion blur. Future research will investigate the selection of image features which can be localized and matched correctly in the presence of motion blur or image noise.

7. Conclusions

This paper has demonstrated the integration of GPS and computer vision measurements with measurements from a low cost inertial sensor. When GPS measurements are unavailable (typically due to obstructions such as when walking through buildings), low cost IMUs do not provide sufficient accuracy to navigate for more than a few seconds. This paper has introduced a computer vision algorithm that makes use of images collected from a camera looking approximately towards the ground, which is where a camera is typically positioned when navigating using a smartphone. During a GPS outage, measurements from the IMU and camera alone were sufficient to maintain a position accuracy of 2.9m after 6 minutes, when walking a circuit around a tennis court. This is compared to a position accuracy of 231m after 60 seconds, degrading to over 19km after 6 minutes using only the IMU for positioning.

The results in this paper demonstrate that a pedestrian can be positioned accurately for several minutes using only the measurements from a low-cost IMU integrated with relative position estimates from computer vision. This system can potentially be used to position a pedestrian indoors using only the camera and IMU available on a smartphone, however errors in scale, and errors in yaw still accumulate over distance. Future work will investigate the reduction of these errors through algorithms such as using IMU measurements to aid feature matching in the computer vision algorithm, and aiding the IMU with yaw corrections from map information such as that described in Abdulrahim et al (2010).

The successful integration of GPS, IMU and computer vision is significant since these sensors are already found in modern smartphones. Further work will look towards addressing issues such as reducing computational requirements of the algorithm through, for example, using the IMU measurements for outlier detection in the computer vision algorithm which is currently the most computationally intensive step. Additional work can further investigate the integration of this technology with other sensors and algorithms. One particularly promising area is through the integration with indoor maps which can significantly constrain the trajectory of the user as they walk along corridors and through doorways. In summary, this paper describes and demonstrates the feasibility of a promising technology for indoor navigation.

References

Abdulrahim, K., Hide, C., Moore, T., Hill, C., 2010, **Aiding MEMS IMU with Building Heading for Indoor Pedestrian Navigation**, Proceedings of UPINLBS 2010, Kirkkonummi. Finland.

Beis, J. S., Lowe, D. G., 1999, **Indexing without invariants in 3D object recognition**, IEEE Transactions on Pattern Analysis and Machine Intelligence 21-10, p1000-1015.

Botterill, T., Mills, S., Green, R., 2009, **New Conditional Sampling Strategies for Speeded-Up RANSAC**, In Proceedings of the British Machine Vision Conference.

Farrell, J. A., Barth, M., 1999**, The Global Positioning System and Inertial Navigation**, McGraw-Hill.

Fischler, M., Bolles, R., 1981, **Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography**, Communications of the ACM 24-6, p381-395.

Foxlin, E., November 2005, **Pedestrian tracking with shoemounted inertial sensors**, IEEE Computer Graphics and Applications 25, 38–46.

Hartley, R., Zisserman, A., 2003, **Multiple View Geometry in Computer Vision,** 2nd Edition. Cambridge University Press.

Hide, C., September 2003, **Integration of GPS and low cost INS measurements**, Ph.D. thesis, University of Nottingham.

Hide, C., Botterill, T., Andreotti, M., 2010, **Vision-aided IMU for handheld pedestrian navigation**, Proceedings of the Institute of Navigation GNSS 2010 Conference. Portland, Oregon.

Mok, E., and Retscher, G., 2007. **Location Determination using WiFi finterprinting versus WiFi trilateration. Journal of Location Based Services**, Vol. 1, No. 2, pp 145-159.

Rosten, E., Porter, R., Drummond, T., 2010, **FASTER and better: A machine learning approach to corner detection**, IEEE Trans. Pattern Analysis and Machine Intelligence. 32 pp. 105-119.

Shin, E. H., 2005, **Estimation Techniques for Low-Cost Inertial Navigation**, PhD Thesis, MMSS Research Group, Department of Geomatics Engineering, University of Calgary, Calgary, AB, Canada, UCGE Report No. 20219.

Titterton, D. H., Weston, J. L., 2004, **Strapdown Inertial Navigation Technology,** 2nd Edition. Institution of Electrical Engineers.

Tao, C. V., Li, J., Ed., 2007, **Advances in Mobile Mapping Technology**, ISPRS Series, Vol 4, London: Taylor and Francis.

Biography

Dr Chris Hide is a Senior Research Fellow at the IESSG, University of Nottingham. He has a degree in Mathematics and Topographic Science from the University of Wales, Swansea and a PhD in Engineering Surveying from the IESSG, University of Nottingham. He previously worked on secondment from University of Nottingham as a Senior Research Scientist at the Geospatial Research Centre in New Zealand before returning to IESSG in 2010. He has been working with GNSS, indoor positioning technologies and integrated systems for over 10 years.