

Low cost vision-aided IMU for pedestrian navigation

Chris Hide

IESSG

University of Nottingham

Nottingham, UK

chris.hide@nottingham.ac.uk

Tom Botterill, Marcus Andretti

Geospatial Research Centre

University of Canterbury

Christchurch, New Zealand

tom@hilandtom.com, marcus.andretti@canterbury.ac.nz

Abstract—Low cost MEMS sensors typically result in large position errors after very short periods of time unless they are frequently corrected by measurements from other systems. One form of measurements comes from the computer vision community where successive frames from a camera approximately looking at the ground can be used to compute the translation between frames. These measurements can be used to control the drift of an Inertial Measurement Unit (IMU) when measurements from other systems such as GPS are not available. This configuration of sensors is preferable since they are already available on some smartphones. This paper demonstrates that computer vision measurements can significantly reduce the drift of IMU-only positioning with a view for pedestrian navigation indoors. Issues such as computational requirements and operation in low light areas are also discussed.

Keywords - *inertial, computer vision, integration, GPS, indoor, navigation*

I. INTRODUCTION

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) unless some form of external measurements are available to restrict the drift. This is due to sensor errors such as biases and scale factor errors that are not necessarily constant over time. Low cost MEMS sensors have recently been demonstrated to provide useful levels of performance through innovative ideas such as mounting an IMU on a user's foot and using zero velocity updates every time a user takes a step (see, for example [1],[2]). This frequent application of accurate measurements has demonstrated that even low cost sensors can provide potentially useful position accuracy provided that constant reliable measurements are available.

One form of frequent 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 that are contained in the image. 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 towards the ground. The idea being that these sensors are already available on some smartphones and a user would need to hold the camera in front of them to see the display. The camera therefore has a view of the ground beneath, and immediately in front of the user, and sequential

images are used to compute the 3-dimensional body frame translation 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. The 3-d body frame translation can be converted to velocity if the height of the camera above the ground is known.

A robust estimator is used to find corresponding features between subsequent images which is based on the assumption that the camera views two scenes that form a plane (i.e. the ground the user is walking over is approximately flat) and that the features in both images lie on the plane. This provides a strong model which can be used to remove features that do not lie on the plane, such as the user's feet and legs as they are taking a step. The robust estimator also deals with issues such as incorrect feature correspondences by removing measurements that do not conform to the model.

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 can be refined by adding an additional state to the Kalman filter. The paper extends the work previously presented in [3] by applying the algorithm to a low cost IMU.

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 3 minute outage, total horizontal positioning errors are reduced from 3740 metres for IMU-only positioning, to only 13 metres when computer vision measurements are used. Such positioning performance would be suitable for integration with other systems such as GPS when available or Wi-Fi position estimates such as from fingerprinting algorithms. Such a combination of sensors, particularly when combined with map matching, could provide high accuracy indoor navigation using sensors that are likely to be 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.

II. 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 [4],[5],[6].

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. 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. These are some of the reasons that make inertial navigation sensors difficult to use.

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

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

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 axis misalignment; δg^b is the vector of gyro bias errors; and $\delta \alpha^b$ 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 [4],[5],[6]. 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 [5].

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 keep the linearised model valid. More information on Kalman filters and Kalman filters for inertial navigation can be found in [4],[5],[6].

III. COMPUTER VISION ALGORITHM

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 captures a sequence of images which are approximately looking at the ground and therefore show a ground plane. 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 which we do not want to use in the estimation.

When an image is captured, the first stage is to detect point features in the image. The FAST corner detector [7] is used to detect approximately 300 points in each image which are likely to be detected in other images showing the same scene. The image 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 detected feature location is transformed using the camera's calibration matrix, and shifted to correct radial lens distortion.

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 (a more efficient procedure could be used, such as a kd-tree; [8]). 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. 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 [9], 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 [10]. BaySAC is based on the RANSAC framework [11], 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 enables many low-quality matches between multiple similar features to be used efficiently, without increasing the computational cost of RANSAC.

Once a homography and inlier set are found, the set is refined by using the DLT to fit a new homography to all of the inliers found and 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

$$H = R + d^{-1} \mathbf{t} \mathbf{m}^T \quad (2)$$

where R is the rotation from the previous camera location, \mathbf{t} is the camera motion vector, \mathbf{n} is a unit vector normal to the ground plane, and, d is the distance between the camera and the ground. H is decomposed to find R , \mathbf{n} and \mathbf{t}/d , using Levenberg-Marquardt's algorithm [9], \mathbf{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.

IV. 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 accuracy is unlikely to be comparable to the measurements from the MEMS gyros used. 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:

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

where $\tilde{\mathbf{v}}^b$ is the estimate of the IMU body frame velocity from the camera; \mathbf{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 (see for example [12]), and this will be the focus of future work.

Following the derivation in [13] to use vehicle frame measurements, and assuming no axes offset, we have the following INS error equation:

$$\hat{\mathbf{v}}^b = \hat{C}_n^b \hat{\mathbf{v}}^n \quad (4)$$

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

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

where $\hat{\cdot}$ indicates a predicted value from the IMU, \mathbf{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:

$$\delta \mathbf{z} = \hat{\mathbf{v}}^b - \tilde{\mathbf{v}}^b \quad (7)$$

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

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.

V. TRIAL

A trial was conducted at the University of Nottingham, UK in September 2010. A Microstrain 3DM-GX3 IMU was used, with the particular model having a rotation range of $\pm 1200 \text{ deg/s}$ and an accelerometer range of $\pm 18 \text{ g}$. The gyro bias stability is quoted at $\pm 0.2 \text{ deg/s}$ for the 300 deg/s model and may be larger for the unit used [14]. Accelerometer biases are quoted as 0.01 g [14]. The IMU data rate was 100 Hz . The IMU was fixed to a Canon IXUS 750 digital camera as shown in Figure 1. The IMU was fixed to the edge of the camera display which provided a flat edge for reference, however it is not possible to quantify the accuracy of the alignment between the camera and IMU axes without undertaking a calibration procedure. No calibration was undertaken therefore the results obtained must be considered with this in view. Video mode was used to record images at 30 frames per second with a resolution of 640×480 pixels.



Figure 1. Microstrain 3DM-GX3 IMU attached to Canon IXUS 750 camera

The full equipment set up is shown in Figure 2. . The IMU and camera were held in the user's hand. A backpack was carried containing a power supply and a Precise Time-stamping Data Logger (PTDL) from the Geospatial Research Centre, New Zealand. The Data Logger time stamps the IMU measurements with a GPS timestamp to sub-millisecond accuracy. The PTDL contains a u-blox ANTARIS 4 GPS receiver, and the u-blox binary format RXM-RAW messages were logged which contain pseudorange, Doppler and carrier phase measurements. A low cost patch antenna was used attached to the end of a survey pole at the top of the backpack. Time stamping of the images from the camera was achieved in post processing by cross correlating the z-axis gyro measurement from the IMU with the z-axis camera rotation measurement from the computer vision algorithm. The cross correlation was performed with 30 measurements at a time over many points throughout the dataset. A linear model was then used to generate a timestamp for the camera measurements.

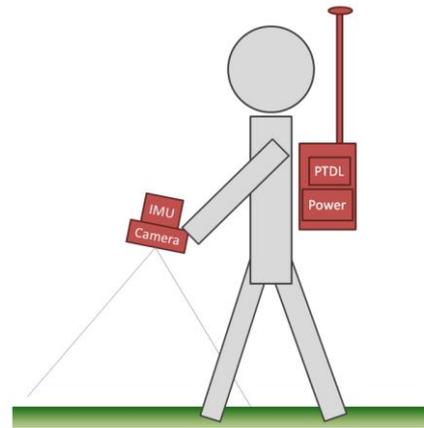


Figure 2. Field trial configuration

A trajectory was walked around a field with a clear view of the sky on the University of Nottingham Jubilee campus. Although the focus of this work is to generate a system suitable for indoor navigation, the outside area was used so that GPS measurements could be used to generate a reference trajectory in order to assess the performance of the algorithm. The initial roll and pitch for the IMU was computed using 1 second of static accelerometer data. An initial coarse estimate of the heading was obtained from Google Earth from the direction the user walked. The trajectory starts at position (0,0) with the user first walking in approximately a westerly direction. The total trajectory lasts approximately 9 minutes (540 seconds). After walking one circuit around the field (taking 3 minutes 20 seconds), a 3 minute GPS outage was simulated (shown in green) so that the combined IMU and computer vision performance could be evaluated. A second short outage occurs in the latter part of the dataset due to poor quality GPS.

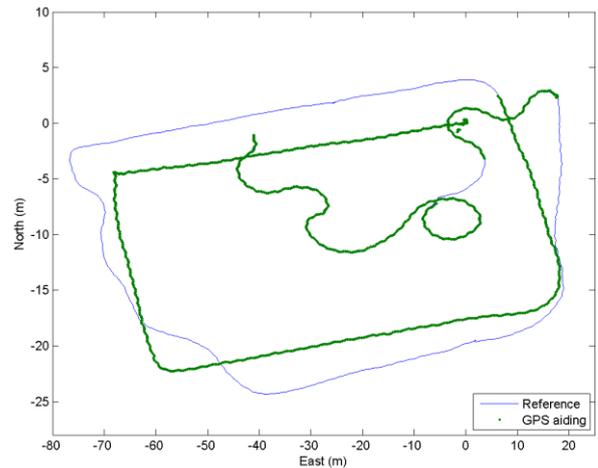


Figure 3. Trajectory of field trial

The u-blox GPS data was differentially processed using Waypoint GrafNav 8.10 software. Reference station data from the National GPS Network Ordnance Survey network at Keyworth was used forming a baseline of approximately 10km. A float solution was generated with an estimated accuracy of better than 0.4m. The remaining data was processed using software developed at GRC, University of Canterbury and IESSG, University of Nottingham. Firstly, computer vision software was developed to compute the translation and rotation of successive frames, with the estimates being recorded to a log file. Every 4th image from the 30 frame per second video were used to reduce the processor requirements since each image contains a significant amount of overlap. The IESSG's POINT software was modified to include the error model described in this paper. Files containing the GPS position and velocity from GrafNav, 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.

VI. RESULTS

Figure 4. shows an example of the computer vision algorithm in operation. As previously mentioned the 30 frames per second video were decimated to use every 4th image resulting in measurements at 7.5 frames per second. The figure shows an example image where the features are marked with red dots. The correct feature correspondences are marked with dark blue lines, and the outliers are marked with red lines. It is shown in the image that there are a significant amount of outliers identified by the BaySAC algorithm which are correctly removed. Throughout the testing, the algorithm was robust by, for example, excluding features on the user's foot and leg when they were moving. The average processing time on a 3GHz desktop PC was 125ms per pair of frames which means that the algorithm is able to run in real-time, although it would not be suitable to run on current mobile devices. Optimisation of the algorithm is required which could involve, for example, reducing the number of features detected or more tightly integrating the computer vision and INS integration code so that the INS can be used for outlier removal.



Figure 4. Features (red dots), inlier correspondences (blue lines) and outliers (red lines) marked on example frame

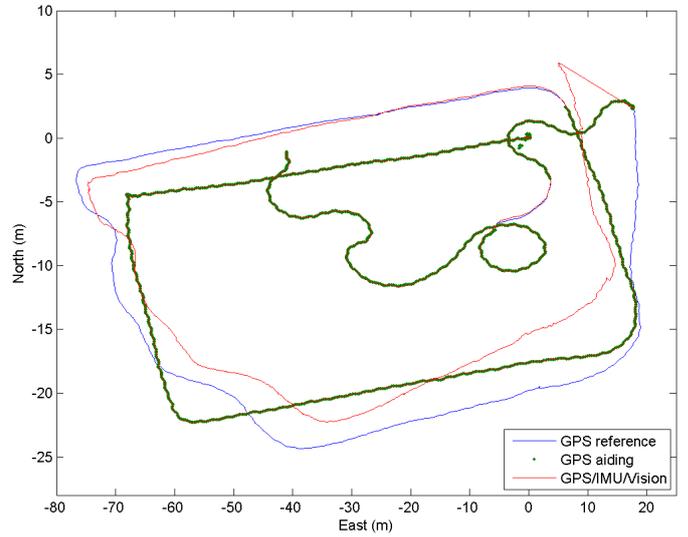


Figure 5. Comparison of integrated GPS/IMU/Vision solution to reference trajectory

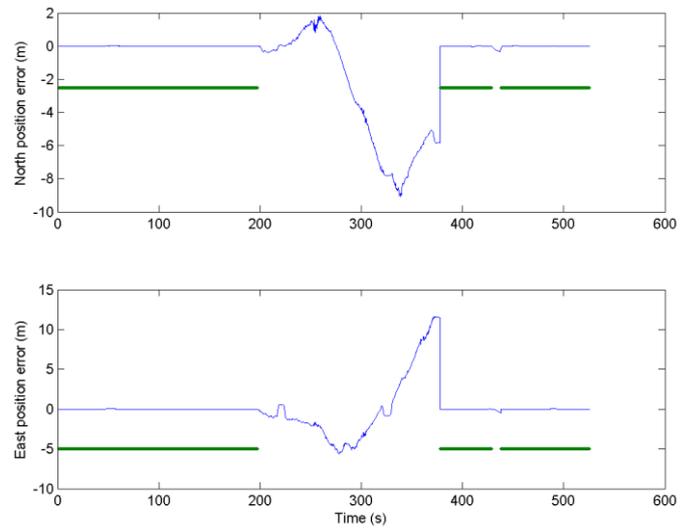


Figure 6. Position error during trial. Times where GPS position and velocity updates are used in the filter are marked in green.

Figure 5. shows the trajectory from the integrated computer vision/IMU/GPS solution. The figure shows that the computer vision aided low cost IMU is able to bridge the 3 minute gap relatively well. Figure 6. and TABLE I. quantify the obtained performance during the GPS outage. Furthermore, TABLE I. shows the INS only solution (without computer vision aiding) for the same outage. We can see in the table that after 30 seconds without velocity aiding the total horizontal position error reaches 35m. After 60 seconds, the error is greater than 200m which is too large to be useful for indoor positioning as other technologies are likely to be better suited such as Wi-Fi positioning. After 180 seconds, the total error is nearly 4km. These results appear very poor but are indicative of the performance obtained from lower cost sensors because the error growth is not linear. While IMU-only positioning may be possible for a few seconds, periods of 60 seconds or more cannot be reliably bridged using low cost IMUs on their own.

The table also shows the results obtained with the combined computer vision and IMU integration. Here we see that the largest total horizontal position error after 180 seconds is just under 13m, which is smaller than the INS-only drift after 30 seconds. The drift of the IMU is significantly reduced which would be ideal for applications such as indoor positioning. It is interesting to observe the characteristic of the position error in Figure 5. Here we see that the position drifts to the left of the true trajectory as the user walks from east to west from the start of the outage. Assuming that the dominant error that remains is due to heading drift, we would expect that the heading less than the true value. Figure 7. shows the comparison of the heading solution between the reference trajectory to the integrated GPS/IMU/Vision solution during the GPS outage. Here we see that the heading drifts by over 10 degrees from the reference heading towards the end of the outage which appears to explain much of the position error. If the position error that remains is mainly due to heading drift, then we return to a situation much like foot mounted IMU positioning where the challenge is to reduce heading errors. Novel algorithms such as that presented in [16] could therefore be applied to improve positioning, particularly indoors.

TABLE I. COMPARISON OF HORIZONTAL POSITION ERRORS WITH AND WITHOUT COMPUTER VISION AIDING DURING GPS OUTAGE

Outage length (s)	INS-only			Computer vision aiding		
	North error (m)	East error (m)	Total error (m)	North error (m)	East error (m)	Total error (m)
30	31.8	14.9	35.1	0.1	-1.3	1.3
60	157.5	149.2	216.9	1.7	-2.1	2.7
90	347.1	552.8	652.7	-2.3	-4.5	5.1
120	460.0	1354	1430	-7.1	-0.4	7.1
150	311.1	2506	2526	-7.4	5.0	8.9
180	281.6	3729	3740	-5.8	11.5	12.9

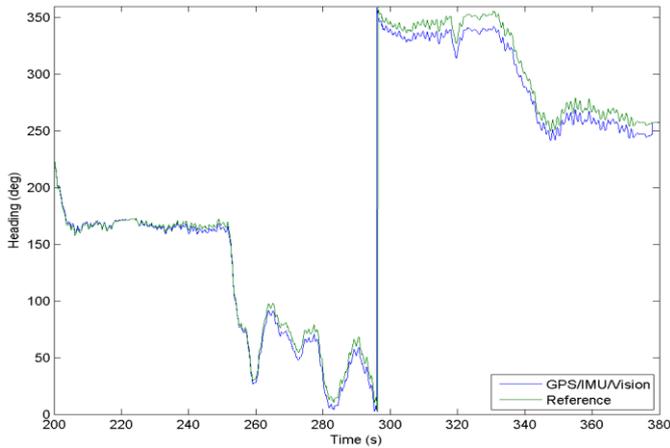


Figure 7. Comparison of heading estimate of reference solution to integrated GPS/IMU/Vision solution

Figure 8. shows the estimated scale factor error for the integrated solution. The scale factor error is only estimated when the camera is moving which accounts for the non-changing values at the start of the dataset, and also at 400s. When the scale factor error is first estimated, the error is estimated to be greater than 0.2 (20%) which is much larger than expected and does not correspond to a significant change in camera height. Over time the scale factor error converges to around 0. When the GPS outage occurs 200 seconds into the dataset, the scale factor error continues to be estimated. Careful analysis of the scale factor error is required in the future to assess the observability of this state.

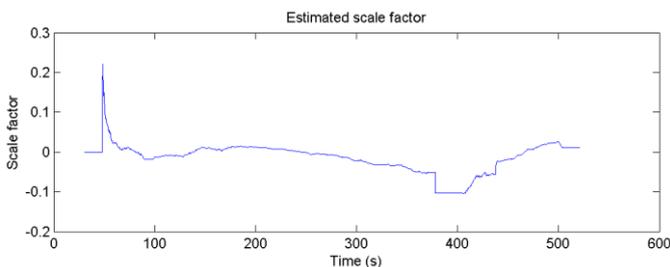


Figure 8. Estimated scale factor error

VII. 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 directly applied to this application.

Firstly, as we have already described, the computer vision algorithm is relatively computer intensive. Some simple modifications can be applied such as reducing the frame rate (as already applied in this paper, although the frame rate could be reduced further still), or reducing the number of features extracted (which may have an adverse effect if insufficient

correspondences can be identified). The main computational load occurs during the BaySAC framework required to remove outliers. 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 identify outliers. Investigation into some of these issues will form the focus of future work.

A second significant issue is the problem of using the camera in low light conditions such as those that can occur indoors. Since the relative motion of the camera with respect to the ground is relatively fast, images obtained indoors tend to have significant motion blur. The camera used for this work has no control over the exposure time or sensitivity used, therefore future work will need to explore this with other cameras. Furthermore, smartphone based camera sensors are typically low quality so, again, further work is required in this area.

VIII. CONCLUSIONS

This paper has demonstrated the integration of GPS and computer vision measurements with a low cost inertial sensor. We have been able to demonstrate that computer vision measurements are able to significantly reduce the INS position drift when GPS measurements are unavailable. The algorithm has been tested using real-world data using a Microstrain 3DM-GX3 IMU. During a simulated GPS outage, it has been shown that horizontal position errors have been reduced from 3740 meters for IMU only positioning, to only 13 meters when the computer vision measurements are also used. We have demonstrated that the work described in [3] can be applied to a low cost IMU. The measurements from this combination of sensors could form a valuable addition to indoor navigation, particularly if we extend this work to using other sensors such as Wi-Fi or map information when navigating indoors.

Future work will look towards issues such as more tightly integrating the computer vision and IMU processing so that the IMU measurements can be used, for example, to identify outliers in the sets of feature correspondences since the IMU provides predicted values of the translation and rotation of the camera. Other work could investigate further sensor integration using, for example, RFID, Wi-Fi or map matching algorithms.

REFERENCES

- [1] Foxlin, E., November 2005. Pedestrian tracking with shoemounted inertial sensors. *IEEE Computer Graphics and Applications* 25, 38–46.
- [2] Hide, C., Botterill, T., Andreotti, M., 2009. An integrated IMU, GNSS and image recognition sensor for pedestrian navigation. In *Proceedings of the Institute of Navigation GNSS 2009 Conference*. Fort Worth, Texas.
- [3] Hide, C., Botterill, T., Andreotti, M., 2010. Vision-aided IMU for handheld pedestrian navigation. In *Proceedings of the Institute of Navigation GNSS 2010 Conference*. Portland, Oregon.
- [4] Farrell, J. A., Barth, M., 1999. *The Global Positioning System and Inertial Navigation*. McGraw-Hill.
- [5] Hide, C., September 2003. *Integration of GPS and low cost INS measurements*. Ph.D. thesis, University of Nottingham.
- [6] Titterton, D. H., Weston, J. L., 1997. *Strapdown Inertial Navigation Technology*. Institution of Electrical Engineers.

- [7] 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.
- [8] 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.
- [9] Hartley, R., Zisserman, A., 2003. *Multiple View Geometry in Computer Vision 2nd Edition*. Cambridge University Press.
- [10] Botterill, T., Mills, S., Green, R., 2009. New Conditional Sampling Strategies for Speeded-Up RANSAC. In *Proceedings of the British Machine Vision Conference*.
- [11] 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.
- [12] Tao, C. V., Li, J., Ed., 2007. *Advances in Mobile Mapping Technology*. ISPRS Series, Vol 4, London: Taylor and Francis.
- [13] 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.
- [14] Microstrain Inc., 2010. 3DM-GX3-25 Miniature Attitude Heading Reference System datasheet. Retrieved from http://www.microstrain.com/product_datasheets/3DM-GX3-25_datasheet_version_1.06.pdf
- [15] Aiding MEMS IMU with Building Heading for Indoor Pedestrian Navigation", Khairi
- [16] Abdulrahim, K., Hide, C., Moore, T., Hill, C., 2010. Aiding MEMS IMU with Building Heading for Indoor Pedestrian Navigation, To be published in the *Proceedings of UPINLBS, Kirkkonummi, Finland*.