Autonomous Vehicle Video Aided Navigation – Coupling INS and Video Approaches

Chris Baker[1], Chris Debrunner[1], Sean Gooding[2], William Hoff[2], William Severson[1]

[1]PercepTek, Inc. Littleton, CO
[2]Colorado School of Mines, Golden CO

Abstract. As autonomous vehicle systems become more prevalent, their navigation capabilities become increasingly critical. Currently most systems rely on a combined GPS/INS solution for vehicle pose computation, while some systems use a video-based approach. One problem with a GPS/INS approach is the possible loss of GPS data, especially in urban environments. Using only INS in this case causes significant drift in the computed pose. The video-based approach is not always reliable due to its heavy dependence on image texture. Our approach to autonomous vehicle navigation exploits the best of both of these by coupling an outlier-robust video-based solution with INS when GPS is unavailable. This allows accurate computation of the system's current pose in these situations. In this paper we describe our system design and provide an analysis of its performance, using simulated data with a range of different noise levels.

1 Introduction

Global Positioning System (GPS) outages are a serious problem to any autonomous vehicle system. Many autonomous navigation systems today rely on the consistent availability of GPS to compute an ongoing estimate of vehicle pose (position and orientation). GPS dropouts are not uncommon, typically occurring next to buildings or occluding land formations. GPS can also easily be jammed or spoofed. To accommodate GPS dropouts, a robust autonomous navigation system must additionally exploit an inertial navigation system (INS) approach using accelerometers and gyroscopes, and it must be tolerant of varying amounts of INS drift. Angular drift varies with INS quality and associated expense, and lower cost solutions typically rely on corrections from GPS to provide the best performance. Alternatively, video data can be used to compute vehicle pose in the absence of GPS. A video-based approach used alone is problematic, however, due its reliance on image texture. The presence and amount of image texture is context dependent and cannot be characterized in general.

The video-based pose approach that we use performs well on data containing many outliers due to our use of a statistical sampling analysis of the data. While this approach is common in video-based 2-frame pose estimation, it has not been used before as a method for outlier filtering for a Kalman based pose tracking system. Video-based solutions are not computed for every frame of data and hence our system handles the multi-rate approach providing an estimated pose at the highest sensor rate.

This problem is in general an example of an approach to the simultaneous localization and mapping (SLAM) problem, as it is sometimes referred to in the computer vision community [1-3] which simultaneously estimates both the camera motion as well as the 3D locations (structure) of the tracked features – typically using only imagery. The addition of inertial measurements for improved accuracy is not new in computer vision [4-18]. Most of these approaches merge inertial measurements in a Kalman-based filter. Many approaches use a minimization technique on the image plane error to estimate the pose of the camera and combine this with the inertial measurements while some incorporate the features directly in the Kalman. Our approach differs from these in how we identify features to insert into the Kalman filter. By first performing outlier-robust pose estimation, we guarantee that the features added in the Kalman are inliers and initialize the 3D location of the feature points within the Kalman.

The autonomous navigation system described in this paper has three general areas of consideration: 1) the use of video sensor data, 2) computation of video-based vehicle pose, and 3) integration of video-based and INS-based vehicle pose. The presentation of the system in this paper is organized in terms of these three areas. Section 2.1 presents the simulated data generation capability that is used to provide data as input to the system. Video-based pose computation is described in Section 2.2. Integration of video-based pose and INS-based pose using a Kalman filter is described in Section 2.3. Section 3 presents analysis and results comparing the integrated system with an INS-only and a video-only approach and demonstrates that the combined system performs better than either system independently.

2 System Description

We have developed a Video Aided Navigation (VAN) system that provides an ongoing estimate of vehicle pose. In the general system, GPS, INS, and video data sources are used. Dropouts of GPS data or periods of non-useful video data are expected and accommodated. In this paper, we focus specifically on the point where GPS data becomes unavailable or unreliable. We are currently focusing our attention on an airborne platform scenario, specifically an unmanned air vehicle (UAV).

The system diagram is shown in Fig. 1. An imaging sensor is mounted on a UAV platform directed toward the ground. Features are extracted from the initial image in the video stream (as in [19]) and tracked as the images are collected using a correlation-based feature tracker. These feature tracks are used to compute the vehicle pose using either an essential matrix or a homography matrix estimation. The set of feature tracks will tend to include outlier features arising for various reasons (see Section 2.1.1). The image-based pose estimation module removes these outliers by performing a RANSAC-based sampling analysis which uses a robust error measure for outlier rejection (see Section 2.2 as well as [3, 20]). The Kalman filter maintains an estimate of the vehicle pose by combining the vision-based measurements with the INS measurements from the platform after initializing with the GPS corrected INS system pose.

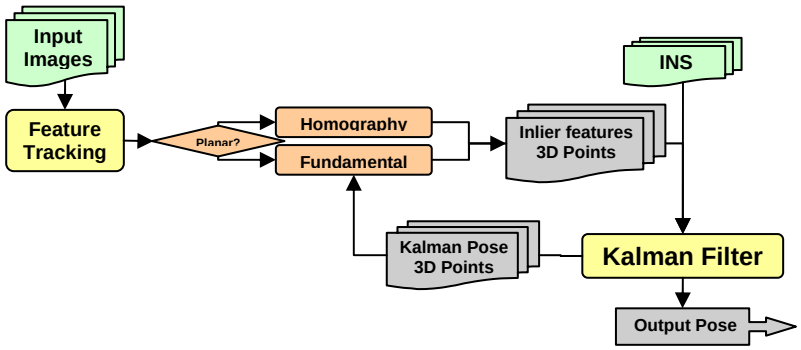

Fig. 1. Video Aided Navigation System Diagram

2.1 Full System Model

In order to systematically vary data inputs and noise levels, we have created a system model simulation package. The entire simulation consists of the following pieces.

- Ground plane of points
- Aerial vehicle (UAV)
- Camera system with fully defined intrinsic parameters
- Camera attached to the UAV platform
- One or more movable objects on the ground plane
- Ground truth transformations between all system objects
- INS system connected to the UAV
- GPS system connected to the UAV
- Complete noise models for each sensor in the simulation

The ground plane consists of a grid of 3D representing features extracted and tracked in a video stream. The terrain is modeled using sine wave functions. The 3D position of a given point on the ground plane dictates where it projects into an oriented camera of a given field-of-view. A trajectory specifies a sequence of platform poses, and each video frame consists of the projection of the collection of ground features that are visible. In order to accurately simulate what we would expect to see in real data, noise models have been developed for each simulated sensor.

2.1.1 Simulated Feature Tracks

The objective in feature extraction and tracking is to first find a large number of unique (trackable) features and track them over as many subsequent frames as possible. The vision community has spent a great deal of effort over its history developing robust feature extraction techniques. One of the more successful of these approaches makes use of the eigenvalues of the covariance of the local image gradient computed over a neighborhood at each image point [19, 21, 22]. Feature points are selected at locations where the smaller eigenvalue is greater than a threshold. This selects points that have significant intensity variation along both image dimensions.

Our data generation system creates track information that simulates a feature detector and tracker's output by adding noise to the true projected image positions of the 3D ground points. For a given simulation the ground features are partitioned into a set

of inliers and outliers. Inliers are feature points with small position errors that are tracked well and that correspond to a real fixed 3D feature. Outlier points have gross errors and model features not associated with a fixed 3D ground point (such as a moving object) or features that are not reliably tracked. Separate image plane noise models are applied to inliers and outliers. The following list outlines various causes of outliers typically seen in visual imagery, each of which we can simulate either by adding Gaussian image plane noise, or by adding moving objects to the simulation.

- Bouncing: Trackers can find high similarity in nearby locations and track points back and forth between these two locations.
- False Feature: Features drift due to an apparent feature at a depth discontinuity.
- Moving Objects: Arises from a moving object being in the field of view.
- Appearance Change: Drifting features as the appearance changes due to varying viewpoint.

2.1.2 Simulated INS

Along with the visual input, we rely on input from the INS system onboard the vehicle. We assume that the INS consists of a tri-axial gyroscopic sensor and a tri-axial accelerometer. The simulated sensor outputs are computed from first and second derivatives of the ground truth vehicle poses. Our noise model for a gyroscope output, consists of the true vehicle rotation rate, a constant offset, a moving or walking bias, and a wide band sensor noise such as that described in [23]. The wideband sensor noise is assumed to be zero mean Gaussian and is therefore specified by the sample covariance. The gyroscope's moving bias is modeled as a first order Markov process. The noise that drives the bias is zero mean Gaussian with sample covariance of one. In order for us to use this approach to model the gyroscope, an iterative solution to the process has to be used. We use the approach described by Brown in [24]. The resolution of the synthetic noise can be increased by either increasing the number of samples or by taking averages of completely separate random seeds.

2.2 Image-Based 2-Frame Pose Solution

There has been much previous work (summarized well in [25]) on computing camera pose from corresponding image points in two frames. When the internal calibration parameters of the camera are known, the motion can be captured in a matrix knows as the essential matrix. To solve for the essential matrix from points in two frames, we use the standard 7-point linear solution method described in [25] and the 4-point method for the estimation of planar homographies also given in [25]. Another notable approach is the 5-point method for the direct computation of the essential matrix [26]. The seven and four-point algorithms require a linear system solution to find the essential matrix or planar homography, which produces a solution for the given correspondences. Up to two motion and structure solutions can be extracted from the homography using the method described in Section 5.3.3 of [27]. One can also recover pose solutions for a rotation-only motion from a homography, which is not possible from the essential matrix. All of these algorithms can be applied to more than the minimum number of points, which can reduce the noise-sensitivity of the algorithm. If more than the minimum number of points is used, the solution spaces are just taken as the minimum norm subspaces of the desired dimension.

One of the difficult problems in this video-based pose estimation is that outliers (feature points with gross errors) can bias a motion solution sufficiently to make it useless, as is true with any least-squares-based solution. To avoid this deleterious effect of outliers, our approach uses the MSAC (M-estimator Sample Consensus [28] algorithm (a variant of RANSAC [20] which uses a robust error measure) to find a solution based on a reliable subset of the image feature points. Given a set of image point trajectories and a pair of image frames, MSAC randomly selects a minimal set of points and solves for the camera projection matrices and 3D structure describing the image feature motion over the two frames. We currently use either the 7-point fundamental matrix solution, which is converted into an essential matrix when the internal camera parameters are known, or the 4-point homography solution. These motion estimation algorithms are run on many (in our implementation 400-500) random samples of seven or four points (depending on the algorithm chosen) and the best solution is selected and returned. The choice of algorithm is based on the rank properties of the data matrix as described in [25] Section 10.9.2.

The criterion for the selection of the best solution for the two frame problem is measured in two stages. In the first stage the solution quality is characterized in terms of the symmetric epipolar distance [25], which is computed from the residual error between the image points and the epipolar lines in both images. The contribution to this error of a point is $\rho\left(e^2\right)$ where

$$\rho\left(e^2\right)=\begin{cases} e^2 & e^2 < T_h^{\,2} \\ T_h^{\,2} & otherwise \end{cases} \qquad \textbf{(1)}$$

and e is the distance of the point's image position to the epipolar line (averaged over both images), and T_h is a threshold based on the expected feature point error. The set of points with $e^2 < T_h^{\,2}$ is the set of inliers for a particular solution. The quality of a particular solution is then based on its first stage residual error, which is computed as the sum of the $\rho\left(e^2\right)$ over all points. The solutions are then further tested in order of increasing first stage residual error by re-computing the solution over all inliers and, in the case of the seven point algorithm, finding the essential matrix nearest to the computed fundamental matrix. Given the 3D locations of the points, we compute the second stage error as the geometric error [25], which is the deviation between the reprojections of the 3D points and the original image feature locations. If no solutions are found with more than the desired number of inliers, the algorithm returns with a failure. Otherwise, the first solution (lowest error in the first stage of screening) with the required number of inliers is returned.

Because of MSAC's iterative nature, the computation time becomes an issue. In order to maintain faster running time, we do not run the MSAC algorithm on every frame, but on a subset of the frames. Detailed descriptions of our method for selecting frames can be found in [3] which enforces the constraint that a minimum set of corresponding features are maintained. This 2-frame essential matrix estimation provides two functions. First, it provides a 3D structure and motion solution that is used to initialize the 3D points in the Kalman. Second, it provides a filtering mechanism for removing the outliers from the feature tracks before they are fed into the Kalman filter.

2.3 Sensor Integration using a Kalman Filter

Because the function relating a feature point position to an image observation is non-linear, we have evaluated an Extended Kalman Filter (EKF) to simultaneously estimate the motion of the camera and the locations of 3D points in the scene. As input, the Kalman filter uses feature point locations from the tracker, as well as INS data (i.e., raw gyroscope and accelerometer readings). The state vector x to be estimated consists of the following elements:

- Vehicle pose (3 translational and 3 rotational degrees of freedom)
- Translational and angular velocities (six additional degrees of freedom)
- Translational accelerations (3 translational degrees of freedom)
- The (x, y, z) locations of each of the N feature points being tracked

Therefore, the full state vector has a total of $15+3N$ elements. By combining these into a single state vector, we can represent cross-correlation between different axes of the motion, the cross-correlation of the uncertainty between different points, and the cross-correlation between the motion and the structure.

We have found that the EKF convergence is sensitive to the initial state. We have a good initial estimate of the pose of the camera at the beginning of the VAN process from the vehicle's GPS corrected INS pose estimation. However, we also need a good estimate of the 3D structure as feature points are added to the state. In addition, since the EKF assumes a Gaussian state distribution and uses a linearized model for the update step, outliers in measurement data can strongly distort the state estimate. This is why we use the 2-frame solution to provide initial estimates of the feature point locations in 3D and to eliminate outliers from the points provided to the Kalman filter.

We assume that the sensors are asynchronous and have independent noise, so each sensor can be incorporated into the state estimate using a separate measurement update equation. The filter will perform the time update projecting the state from the previous update time to the current update when gyroscope data, accelerometer data, or camera data (in the form of filtered image plane features) is available. The measurement update step will follow to update the filter's state based on the new measurement input. This is a recursive process and it will run whenever the measurement input data is available. The time and measurement update equations are given in [29].

Table 1. Gyroscope Specifications

Rate Gyro	Attribute	Units	Specs
Tactical (CASE I)	Random walk	°/sec/√Hz	0.0017
	Bias Time Constant	Sec	100
	Bias Variation	°/hr	0.35
Automotive (CASE II)	Random walk	°/sec/√Hz	0.05
	Bias Time Constant	sec	300
	Bias Variation	°/hr	180
Consumer (CASE III)	Random walk	°/sec/√Hz	0.05
	Bias Time Constant	sec	300
	Bias Variation	°/hr	360

New features are added to the state vector as they are observed, and old features that are no longer visible are removed. Thus, the state vector and its covariance matrix

can dynamically expand and contract depending on the number of trackable features. The 3D feature locations are initialized in the state vector based on the 2-frame solution. The covariance ellipsoid for the feature location is elongated along the projection line of the feature to capture the larger uncertainty of the feature depth.

3 Results and Analysis

Here we compare three different quality sensor systems – consumer, automotive, and tactical grad - as shown in Table 1 and summarized in [23]. The camera qualities corresponding to the different typical camera systems with resolutions of 2048x2048, 1280x1024, and 640x480 for the tactical, automotive, and consumer grade respectively. We assume in each case that the camera has a lens with a field of view held constant at 45°, and that our sub-pixel feature matching algorithm can match features to $^1/_{10}$ of a pixel. For each of the runs, the inlier ratio is 85%. Since outliers are derived from tracking problems (as described in 2.1.1), not camera quality, the outliers will exhibit different amounts of pixel error for the different camera systems. For these experiments, the outlier noise level has been set to 28.47 pixels, 22.78 pixels, and 11.12 pixels for the tactical, automotive, and consumer grade camera systems respectively. For each of these sensor systems we show the computed pose from the 2-frame video-only solution, the INS-only solution, and our combined VAN system. In this experiment we show that the combined VAN approach produces superior results than either of the two systems running independently.

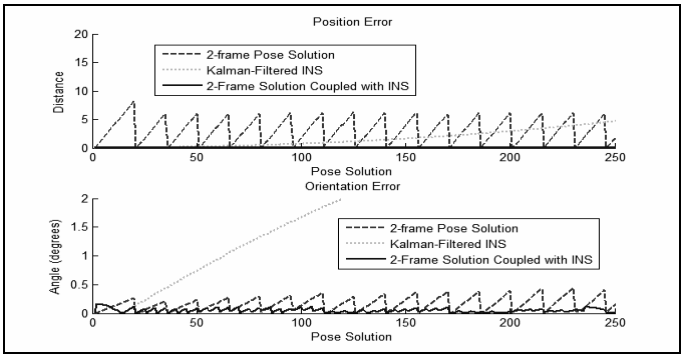

Fig. 2. Pose solutions for comparison of 2-frame video-only solution, INS-based solution, and our VAN system. This plot shows typical results for errors corresponding to a tactical grade (CASE I) INS and imaging system. The distances given are in meters.

3.1 CASE I: Low Noise – Tactical Grade Camera and INS Sensor System

Three pose estimation approaches are tested; the 2-frame video-only pose solution, the INS-based pose solution, and our VAN solution. For the low noise case, the INS data was set to the tactical grade sensor settings given in Table 1 and the camera is set

as described in the previous section. Notice in Fig. 2 the video-only solution presents a saw-tooth wave form. This is due to the fact that a solution is computed on only every 3rd frame, thus causing the intermittent frames to have increasing error as the true pose moves further from the last MSAC solution. Notice also the drift in the INS-only derived pose. This is typical of INS-only systems.

3.2 CASE II: Medium Noise – Automotive Grade INS

The medium noise case corresponds to the values in Table 1 for an automotive grade INS sensor and corresponding imaging system. Notice Fig. 3 shows similar results to the CASE I setup. The combined VAN algorithm performs much better than either of the individual systems. The INS-based system running alone is only slightly worse in this case when compared to the previous, likely due to the higher INS system noise.

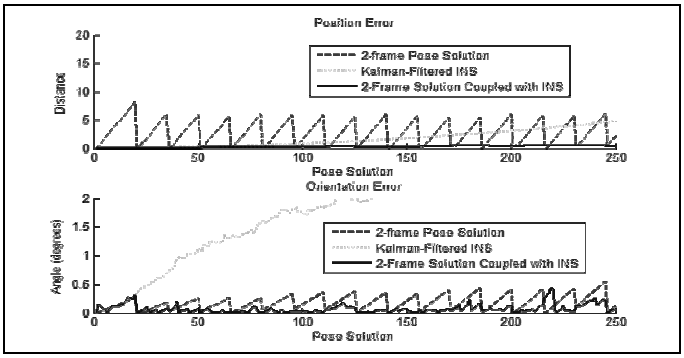

Fig. 3. Pose solutions for comparison of the 2-frame video-only pose solution, the INS-based pose solution, and our VAN system. This plot shows typical results for errors corresponding to an automotive grade (CASE II) INS and imaging system. The distances given are in meters

3.3 CASE III: High Noise – Consumer Grade INS

For the high noise case, the noise values used on the INS data correspond to the values in Table 1 for the consumer grade INS system and corresponding imaging system. Notice in Fig. 4 the trends are as expected with higher noise in the combined VAN approach, but still much better than either of the systems individually. Notice in this case, video-only occasionally failed to find a solution. This is likely due to the fact that at these higher noise levels, many of the inliers begin to look like outliers, and if there are not enough inliers, no solution is returned.

4 Discussion and Conclusions

In this work we show that autonomous navigation systems consisting of INS/GPS systems are not adequate due to INS drift when GPS signals are lost. Video-only solutions will not perform robustly when image texture is inadequate for feature tracking or flow based methods. We have proposed and demonstrated a combined VAN approach which merges both INS and vision systems to exploits the benefits of each using a new outlier filtering technique. Our system is still in early development and further testing on real data sequences will be necessary for full validation.

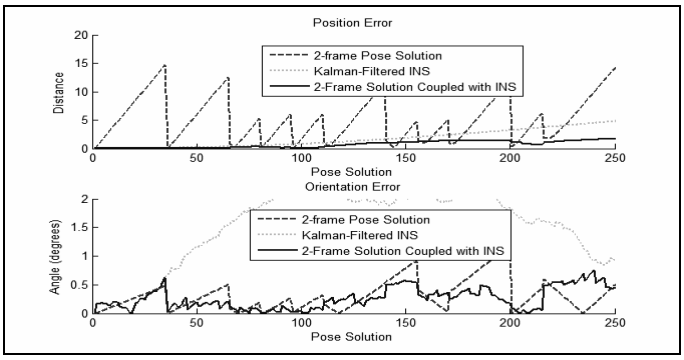

Fig. 4. Pose solutions for comparison of the 2-frame video-only pose solution, the INS-based pose solution and our VAN system. This plot show typical results for errors corresponding to a consumer (CASE III) grade INS and imaging system. The distances provided are in meters.

References

1. Debrunner, C.H. and N. Ahuja, Segmentation and factorization-based motion and structure estimation for long image sequences. IEEE Trans Pattern Anal Mach Intell, 1998. **20**(2): p. 206-211.
2. Davison, A.J. *Real-time simultaneous localisation and mapping with a single camera*. in *International Conference on Computer Vision*. 2003. Nice, France.
3. Baker, C., C. Debrunner, and M. Whitehorn. *3D model generation using unconstrained motion of a hand-held video camera*. in *The International Society for Optical Engineering*. 2006. San Jose, CA: Springer.
4. Alenya, G., E. Martnez, and C. Torras, *Fusing visual and inertial sensing to recover robot egomotion.* Journal of Robotic Systems, 2004. **21**: p. 23-32.
5. Diel, D.D., P. DeBitetto, and S. Teller. *Epipolar Constraints for Vision-Aided Inertial Navigation*. in *IEEE Workshop on Motion and Video Computing (WAVC/MOTION'05)*. 2005: IEEE Computer Society.
6. Deil, D.D., *Stochastic Constraints for Vision-Aided Inertial Navigation*, in *Mechanical Engineering*. 2005, Massachusetts Institute of Technology. p. 110.
7. Chai, L., *3-D Motion and Structure Estimation Using Inertial Sensors and Computer Vision for Augmented Reality*, in *Engineering Division*. 2000, Colorado School of Mines: Golden, CO. p. 110.

8. Domke, J. and Y. Aloimonos, *Integration of Visual and Inertial Information for Egomotion: a Stochastic Approach*. 2006, Dept. of Computer Science, University of Maryland: College Park, MD.
9. Huster, A. and S.M. Rock. *Relative Position Sensing by Fusing Monocular Vision and Inertial Rate Sensors*. in *International Conference on Advanced Robotics*. 2003. Coimbra, Portugal: Proceedings of ICAR 2003.
10. Makadia, A. and K. Daniilidis. *Correspondenceless Ego-Motion Estimation*. in *IEEE International Conference on Robotics and Automation*. 2005.
11. Nguyen, K., *Inertial Data Fusion using Kalman Filter Methods for Augmented Reality*, in *Engineering*. 1998, Colorado School of Mines: Golden.
12. Qian, G., R. Chellappa, and Q. Zheng, *Robust structure from motion estimation using inertial data.* J. Opt. Soc. Am., 2001. **18**(12): p. 2982-2997.
13. Rehbinder, H. and B.K. Ghosh. *Mulit-rate fusion of visual and inertial data*. in *International Conference on Multisensor Fusion and Integration for Intelligent Systems*. 2001.
14. Rihbinder, H. and B. Ghosh, *Pose Estimation Using Line-Based Dynamic Vision and Inertial Sensors.* IEEE Transactions on Automatic Control, 2003. **48**(2): p. 186 - 199.
15. Roumeliotis, S.I., A.E. Johnson, and J.F. Motgomery. *Augmenting inertial navigation with image-based motion estimation*. in *IEEE International Conference on Robitics and Automation*. 2002.
16. Strelow, D. and S. Singh. *Online Motion Estimation from Image and Inertial Measurements*. in *Workshop on Integration of Vision and Inertial Sensors (INERVIS)*. 2002.
17. You, S. and U. Neumann. *Integrated Inertial and Vision Tracking for Augmented Reality Registration*. in *Virtual Reality Annual International Symposium*. 1999. Houston**,**TX: IEEE.
18. You, S., U. Neumann, and R. Azuma. *Hybrid inertial and vision tracking for augmented reality registration*. in *IEEE Virtual Reality '99*. 1999: IEEE.
19. Harris, C. and M. Stephens, *A combined corner and edge detector.* Alvey Vision Conference, 1988.
20. Fischler, M.A. and R.C. Bolles, *Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography.* Communications of the Association of Computing Machinery, 1981. **24**: p. 381-395.
21. Lucas, B.D. and T. Kanade, *An iterative image registration technique with an application to stereo vision.* International Joint Conference on Artificial Intelligence, 1981.
22. Shi, J. and C. Tomasi. *Good Features to Track*. in *IEEE Conference on Computer Visioin and Pattern Recognition (CVPR94) Seattle*. 1994.
23. Flenniken, W., *Modeling Inertial Measurement Units and Analyzing the Effect of their Errors in Navigation Applications*. 2005, Auburn University: Auburn, Al.
24. Brown, R.G. and P.Y.C. Hwang, *Introduction to random signals and applied Kalman filtering*. 2nd ed. 1997, New York: J. Wiley. 502.
25. Hartley, R. and A. Zisserman, *Multiple View Geometry in Computer Vision*. 2001, Cambridge: Cambridge University Press.
26. Nister, D., *An efficient solution to the five-point relative pose problem.* Computer Vision and Pattern Recognition, 2003.
27. Ma, Y. and S. Soatto, *An invitation to 3-d vision: from images to geometric models*. 2004, New York: Springer.
28. Torr, P.H., *MLESAC: A new robust estimator with application to estimating image geometry.* Computer Vision and Image Understanding, 2000. **78**: p. 138-156.
29. Brown, R. and P. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*. 3rd ed. 1992, New York: Wiley.