

Error Analysis of Algorithms for Camera Rotation Calculation in GPS/IMU/Camera Fusion for UAV Sense and Avoid Systems

Tamas Zsedrovits^{†*}, Peter Bauer^{*}, Akos Zarandy^{†*}, Balint Vanek^{*}, Jozsef Bokor^{*}, Tamas Roska^{†*}

[†]Pázmány Péter Catholic University, Faculty of Information Technology and Bionics, Budapest, Hungary,

^{*}Institute for Computer Science and Control of the Hungarian Academy of Sciences (MTA SZTAKI) Budapest, Hungary

Abstract— In this paper four camera pose estimation algorithms are investigated in simulations. The aim of the investigation is to show the strengths and weaknesses of these algorithms in the aircraft attitude estimation task. The work is part of a research project where a low cost UAV is developed which can be integrated into the national airspace. Two main issues are addressed with these measurements, one is the sense-and-avoid capability of the aircraft and the other is sensor redundancy. Both parts can benefit from a good attitude estimate. Thus, it is important to use the appropriate algorithm for the camera rotation estimation. Results show that many times even the simplest algorithm can perform at an acceptable level of precision for the sensor fusion.

Index Terms — UAV, sensor fusion, IMU, GPS, Camera, FPGA

I. INTRODUCTION

The UAV technology is close to offer great opportunity to run various commercial and public services like meteorological measurements, surveillance tasks, or agricultural services. Many of the possible applications can be run with low cost, small size UAVs and in good weather conditions. However, these applications are very limited today because of limited access to national airspace (NAS).

In order to integrate UAVs into NAS, two of the most important features are the sense-and-avoid (SAA) capability and hardware redundancy [1]. Provided that the size and the energy consumption of the UAV is limited, a camera based SAA system would provide cost and weight advantages against radar based solutions [2], [3]. The main drawbacks of camera based systems are the high computational need because of the complex algorithms that need to be run and that there is no direct range information. Lighting conditions provide another limitation which is not considered

in this article but should be part of a future work.

Fortunately, complex algorithms in real time with low power consumption can be implemented on today's kilo-processor chips and the range information can be estimated using Kalman filter based approaches.

As an example, in our previous work a camera-based autonomous on-board collision avoidance system and its implementation aspects on kilo-processor architectures were introduced. In [4] the feasibility study and the algorithms of electro-optical based collision avoidance are presented. In [5], [6] and [7] the performance of a solely camera based collision avoidance algorithm as well as its implementation on a specialised FPGA architecture are introduced. In [8] the developed camera-processor system is introduced.

The sense-and-avoid task has to be run in critical situations as well, for example when one or more sensor fails. One solution is redundancy in the sense of the number of similar sensor modules or in different sensor modalities. In this case the use of our camera can be broadened to localisation task besides its main function in collision avoidance.

On the other hand with an IMU/Camera fusion better accuracy can be achieved in the ego motion as shown in [9]. With these more accurate results our SAA algorithm can be speed-up which provides even higher separation distance or the avoidance of aircrafts with higher speed.

In [10] performance comparison of tight and loose (Kalman filter based), INS-Camera integration is studied by Chu et al. through simulations. The paper shows that tight coupling can provide higher accuracy but it is less stable due to the linearization methods of the filters. Thus loose integration is favourable in low cost systems.

In [11] a monocular camera, INS and GNSS integration is presented for ground vehicles by Chu et al. This system is validated through a real drive test and results show that the system based on camera-INS fusion outperforms the conventional INS-GNSS systems. However the GNSS measurements are not included in the camera-INS system. As stated in the paper this step can further improve the performance of the system. Furthermore, the real-time functionality is a challenging task because of the image processing algorithms involved.

For aircraft attitude estimation many different image processing algorithms can be used from a simple homography based calculation to the more complicated five point algorithm. The question is how these algorithms can be ranked based on their performance and computational complexity in realistic simulations.

The inventors of these algorithms provide information about their accuracy [12] [13], and there are other papers which assemble and compare different algorithms from some perspective [14]. To the best of our knowledge there is no analysis for these algorithms for GPS/IMU/Camera fusion which can easily show the strength and weaknesses of a specific algorithm in this scenario.

In this paper the error analysis of four algorithms is introduced. The analysis is done with realistic flight paths generated by the HIL simulator [15]. The camera model is based on the calibration of the camera used on board of our test aircraft. These results can give a general idea that in which situation which algorithm can be used effectively. As an application example simulation and measurement results from our camera-IMU (including GPS) sensor integration are shown.

The remainder of the paper is as follows: in the next chapter the coordinate frames, transformations and the algorithmic basics are introduced. In section III. the simulations and analysis results are shown regarding the image processing algorithms. In section IV. simulation examples are presented applying a GPS/IMU/Camera fusion algorithm, and finally the conclusions are drawn in section V.

II. ALGORITHMIC BACKGROUND

In this section the coordinate frames and specifically the transformation from world (earth)

coordinate system (coord. sys.) to the image plane are defined and the basics of used camera pose calculation algorithms are introduced. For the measurements four feature point based relative pose estimation algorithm are chosen. A homography based solution as a basic algorithm with small computational need but with less accuracy. The eight point algorithm, as standard algorithm in epipolar geometry. The five point algorithm, as one of the state of the art algorithms with higher computational need, but with promising stability over the various scenes. Finally, MLESAC, as an iterative, stochastic solution. Other algorithms can be tested in the future with the same framework.

A. Coordinate frames and transformations

In the application example a small UAV is considered, which flies only short distances (about 1km range). This allows considering the North-East-Down (NED) frame as an inertial (non-moving, non-rotating) frame (earth frame), which is defined in the convention used in [16].

The other two applied coordinate systems are the body and camera frames. The axes of the camera system are in general nonparallel with the axes of the body system but in the considered system set up for sake of simplicity they are parallel but the camera system is rotated (Figure 1.).

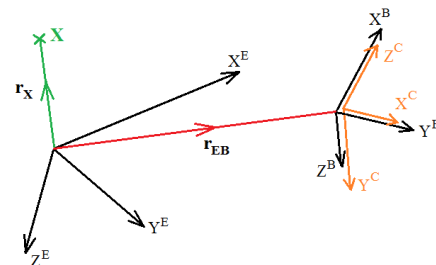


Figure 1. The earth, the body and the camera coordinate systems in this specific scenario where the origins of body and camera systems coincide

In Figure 1. X is a feature point in the earth coord. sys. characterized by vector r_X^E ($(\)^E$ means a vector with coordinates in earth coord. sys.). r_{EB}^E gives the position of the body frame relative to earth. The coordinates of point X in the camera frame can be calculated as follows:

$$X_{cam} = T_{CB}T_{BE}(r_X^E - r_{EB}^E)$$

Here, $T_{f_2f_1}$ defines a transformation matrix from frame f_1 to f_2 . In our special case the origins

of the body and camera system are assumed to coincide (see Figure 1.).

B. IMU models

Our IMU consists of sensors which are required for outdoor waypoint navigation. In our system the conventional accelerometer, rate gyro, differential and absolute pressure sensor and magnetometer are completed with a GPS unit [17].

C. Camera measurements

The electro optical sensor is modelled as a projective camera [18]. The camera matrix P consists of the internal and external parameters of the camera and can be decomposed as follows:

$$P = K[R|t]$$

where R and t are the rotation and translation of the camera, which are the extrinsic parameters. K contains the intrinsic parameters: the focal length f in pixels (it can be different in the x and y directions) and the position of camera principal point p in the image plane as follows:

$$K = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

Here the resolution of the camera is interesting as well, because the effect of pixelization and spatial resolution is studied. As results show a projective camera can be characterized by the angular resolution of the central pixel (or CPAR), which is defined as follows:

$$CPAR = \tan^{-1} \frac{1}{f}$$

where f is the focal length of the camera. With this measure cameras with different resolution and field of view can be compared.

D. Feature extraction and matching

On the consecutive frames a modified Harris corner feature extraction is used [19]. Corner features are extracted but two constraints are used: 1) the feature points should be farther to each other in the image than a given threshold and 2) feature points should be in the ground region, below the horizon. The latter constraint can be satisfied by an adaptive threshold, which is applied before the corner detection. With these two constraints the number of the feature points is limited. The first constraint can assure in most cases that degenerate

feature point combinations are avoided.

Our UAV will be used mainly in countryside, where there are only a few tall buildings (if any). It means that static features according to the NED frame are located on the ground. That is why feature points are searched for on the ground. This is viable, because except the take-off and a few manoeuvres, the ground can be seen by the camera.

E. Homography

As a basic solution for the problem of camera pose estimation a scene homography based algorithm is tested. In this case the assumption is made that the movement of the camera is so small that the effect of the movement can be neglected thus only the camera rotation is calculated.

The basic equations of the calculation are used for planar panoramic mosaicking as well and also known as inhomogeneous DLT. The equations are as follows:

$$A = \begin{bmatrix} 0 & 0 & 0 & -x_i w'_i & -y_i w'_i & \dots \\ x_i w'_i & y_i w'_i & w_i w'_i & 0 & 0 & \\ & -w_i w'_i & x_i y'_i & y_i y'_i & & \\ & 0 & -x_i x'_i & -y_i x'_i & & \end{bmatrix}$$

$$A * h = \begin{pmatrix} -w_i y'_i \\ w_i x'_i \end{pmatrix}$$

where $x_i \leftrightarrow x'_i$ and $y_i \leftrightarrow y'_i$ are the coordinates of the corresponding feature points on the consecutive frames, and the element of h vectors are the elements of the homography matrix up to an unknown scale. This scale is given by w_i and w'_i for each frame and each feature point.

An optimal solution for the homography can be yielded with the SVD of the A matrix. And again the optimal rotation can be calculated from the SVD of the resulting homography matrix. More details about the calculation can be found in [18].

F. Eight point algorithm

As a more promising variant the normalised eight point algorithm is tested [18]. From feature point pairs the fundamental matrix F can be calculated. F is defined by the epipolar constraint as follows:

$$x'^T F x = 0$$

If one has a calibrated camera the essential matrix E can be obtained from F by multiplying

with the camera matrix K such as:

$$E = K'^T FK$$

Here we have only one camera, so $K' = K$.

G. Five point algorithm

In the case of calibrated cameras the E matrix can be computed directly from five point correspondences because it has only five degrees of freedom. In [20] and [12] an efficient algorithm is presented, which is numerically more stable than other methods. Furthermore, the five point algorithm should be accurate in the case of pure rotational or pure translational movement as well.

H. MLESAC

As the member of the RANSAC family, the MLESAC algorithm is tested [13]. This is a more advanced RANSAC variant where the fundamental matrix is robustly calculated based on probability features.

This algorithm is not the best with respect to accuracy as stated in [14] but the computational complexity of the algorithm is reasonable and the implementation is available online.

I. Camera rotation and translation from epipolar matrices

With the eight point algorithm, the MLESAC and the five point algorithm the E matrix can be calculated from point correspondences. From E the two camera matrices can be calculated in canonical form (that is the first camera matrix is $P = [I|0]$ and the second is $P' = [R|t]$), because $E = [t]_{\times} R$, where $[t]_{\times}$ is a skew symmetric form of translation vector t representing vector cross product.

For the calculation, E has to be decomposed with SVD and four solutions result. Only one of them satisfy the chirality constraint [21] that is in only one arrangement are the reprojected feature points in front of both cameras [18].

J. Reconstruction of aircraft attitude change from camera rotation matrix

From the matched feature points in two consecutive camera frames the camera rotation matrix R and translation vector t (with scale ambiguity) can be reconstructed assuming canonical cameras. Here, normalised coordinates and calibrated cameras are considered as stated before.

This way the X_{cam} (not normalized) vector can be transformed into the first frame as (using homogenous coordinates):

$$x = P \begin{bmatrix} X_{cam} \\ 1 \end{bmatrix} = [I \quad 0] \begin{bmatrix} X_{cam} \\ 1 \end{bmatrix} = X_{cam}$$

The same X_{cam} vector can be transformed into the second frame considering the transformation between the two frames which is the P' camera matrix:

$$x' = P' \begin{bmatrix} X_{cam} \\ 1 \end{bmatrix} = [R \quad t] \begin{bmatrix} X_{cam} \\ 1 \end{bmatrix} = RX_{cam} + t$$

x' is the image of point X in the second (rotated and translated) camera frame which means the rotation and translation of the aircraft body frame. This way x' can be also constructed by considering the changed T_{BE}' matrix and $(r_{EB}^E)'$ vector:

$$x' = X_{cam}' = T_{CB} T_{BE}' (r_X^E - (r_{EB}^E)')$$

From the two representations of x' and the original expression for X_{cam} by considering $T_{BE}' = T_{\Delta} T_{BE}$ and $(r_{EB}^E) = r_{EB}^E + \Delta r_{EB}^E$ one gets the aircraft attitude change T_{Δ} as follows:

$$T_{\Delta} = T_{CB}^T \cdot R \cdot T_{CB}$$

The detailed calculation can be found in [9].

III. SIMULATION RESULTS FOR THE ERROR ANALYSIS OF IMAGE PROCESSING

In this section the methodology and results of the error analysis of image processing are introduced. The pose estimation algorithms introduced in the previous section are analysed in a realistic simulation environment. The algorithms are tested with different image resolutions and sampling time. This way the tendencies can be pointed out for each algorithm as well as the performance of these algorithms can be compared.

A. Simulation environment

The simulation environment is based on the MATLAB EGT toolbox [22]. For the tests realistic flight paths are used, which are generated by a HIL simulator [15]. In this paper the results of two simulated flight tests are introduced: 1) a sinusoidal path with almost constant altitude and 2) a zigzag path with also nearly constant altitude. The resulting error figures show similar phenomena, that is why only one of them is shown as an example.

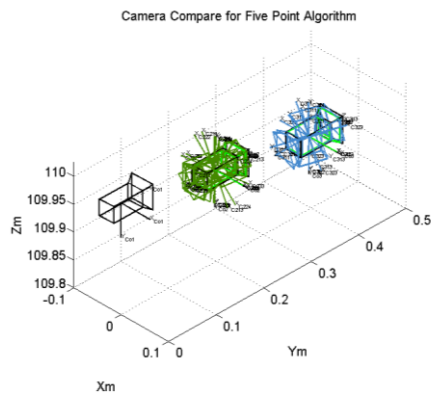


Figure 2. Cameras in the EGT frame

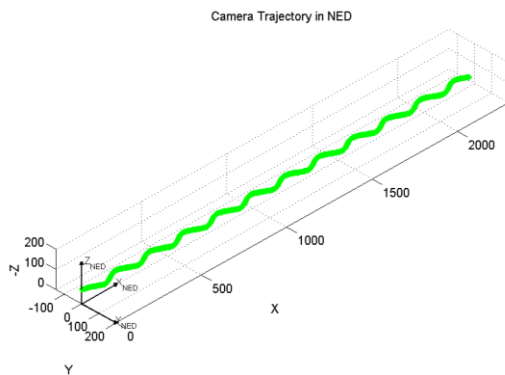


Figure 3. Sinusoidal path in the NED frame

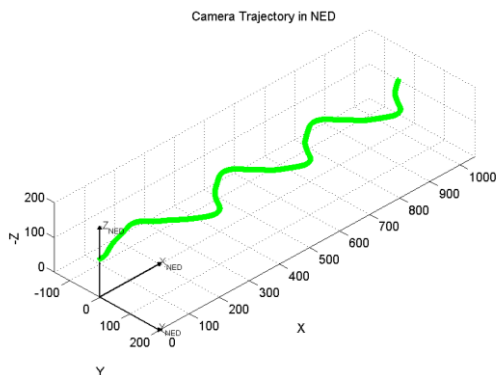


Figure 4. Zigzag path in the NED frame

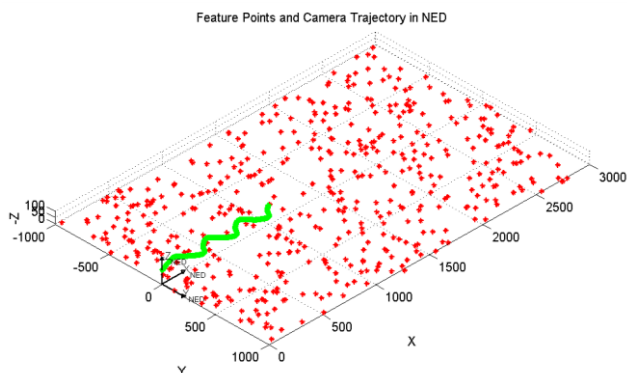


Figure 5. Camera trajectory and feature points in NED frame

For the tests 350 feature points are placed randomly with uniform distribution in a right prism which is 2000m wide, 3000m long and 30m tall.

The point coordinates are between -1000 and 1000 in the Y direction and from 0 to 3000 in the X direction. The maximum altitude of the points is 23 m and the Z coordinate starts from 3 m beyond the ground level to simulate small holes.

The camera can see only feature points which are closer than 800m. This way the dense feature point cloud can be avoided on the images near the horizon level. This is important, because in the real images feature points near the horizon cannot be extracted because the blurring effect of the distant objects.

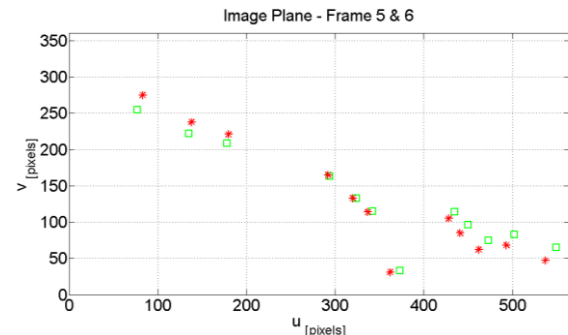


Figure 6. Feature points of two consecutive frames on the image plane; with green squares feature points of frame 5 and with red stars the feature points for frame 6; the camera resolution is 752×480

For the camera projection the calibration matrix of one of our miniature camera is used. The calibration was obtained using the Camera Calibration Toolbox in MATLAB [23]. The resolution is 752×480 pixel and the Field of View (FOV) is $\sim 63^\circ \times \sim 43^\circ$. Based on this calibration matrix 5 virtual cameras are generated with the same FOV and different resolution, that is with different CPAR as shown in Table 1.

Resolution [px]	564	752	1017	1280	1540	1692
	×	×	×	×	×	×
	360	480	649	817	960	1080
CPAR [°/px]	0.12	0.093	0.068	0.055	0.046	0.041

Table 1. Resolution and CPAR of cameras

For these cameras test cases are run with absolute feature point precision, which is after the projection, there is no pixelization. Thus, the coordinates are not rounded to integer pixel values. Because many feature point extraction algorithm support subpixel resolution, this case is simulated as well. To the exact feature point coordinates a noise with normal distribution, 0 mean and 0.5 pixel standard deviation is added later. This way the

luminance fluctuation and the nature of the point spread function in the real case is mimicked. Finally, the feature point coordinates are rounded to the nearest integer coordinate values to examine the effects of pixelization.

The simulations are run with different sampling frequencies. As in our test bed, the camera is running at its maximum with 56Hz. In the simulation this is approximated with 50Hz base sampling frequency that is with 20ms sampling time. Due to the processing steps or if we change the camera for another with bigger resolution, the frame rate can be dropped. The effect of the sampling frequency that is the effect of the translation on the different algorithms, is investigated in ten steps from 20 ms sampling time (50Hz) to 200 ms (5Hz).

Standard implementations of the aforementioned algorithms are used. The eight point algorithm and the MLESAC is implemented in the EGT toolbox [22] and the implementation of the five point algorithm is from its authors' website [24]. The homography algorithm was implemented in house according to [18].

B. Error measures

In each and every step the direction cosine matrix (DCM) between the two frames is extracted which describes the rotation from one camera orientation to another. Based on this DCM the Euler angles are calculated and these are compared to the ground truth. To characterize the performance of each algorithm the absolute error of the three Euler angles are used.

$$e_i = \sqrt{(\alpha_i - \alpha_i^m)^2}$$

where α_i is the ground truth angle for the i^{th} frame (roll, pitch or yaw) and α_i^m is the calculated angle. Additionally, for each run also the mean, the median and the corrected standard deviation of the absolute error are calculated.

C. Homography algorithm correction

To handle that the homography neglects the translation a simple correction algorithm is introduced based on the sampling time, the measured velocity and the altitude. Most of the time the error introduced by the translation has a bigger effect on the pitch and it has a smaller effect on the yaw angle, but the error is distributed

proportionally to the roll angle. Thus the correction term is as follows:

$$\begin{aligned} pitch_{correction} &= \frac{\cos(roll) + \sin(roll)}{\cos(roll)} \cdot f(\tau, alt, v) \\ yaw_{correction} &= \frac{\cos(roll) - \sin(roll)}{\cos(roll)} \cdot f(\tau, alt, v) \end{aligned}$$

where the correction terms are added to the calculated angle values and $f(\tau, alt, v)$ is an empirical function based on the linear interpolation of the measured error term for different τ (sample time), altitude and velocity values.

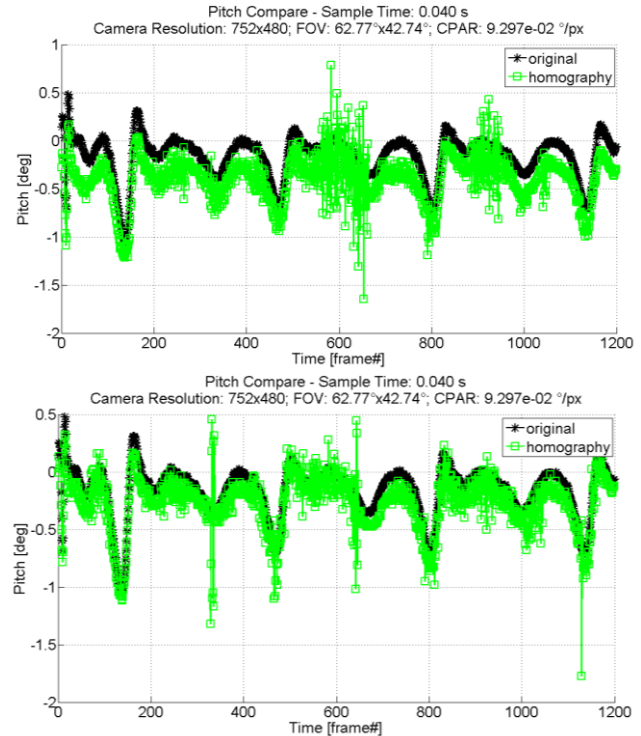


Figure 7. Pitch compare for homography on sinusoidal path; with black stars the ground truth, with green squares the homography results; top without correction, bottom with correction

As an example, in Figure 7. the correction of the pitch angle is shown. On the upper part, the pitch values are compared to the original values without correction and on the lower part with correction. As it can be seen in Figure 8. the error is almost twice without the correction. In this case the original camera matrix is used and the sample time is 40 ms.

D. Results with absolute feature point precision

First, tests with absolute feature point precision are run. In this case the best achievable results are obtained because there is practically no spatial discretization, the effect of the temporal resolution change can be investigated independently.

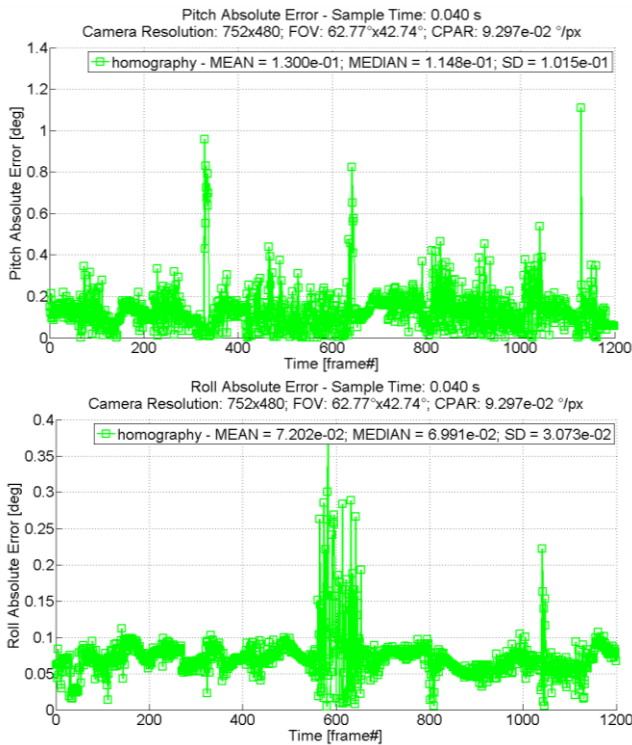


Figure 8. Pitch absolute error for homography on sinusoidal path; top without correction, bottom with correction

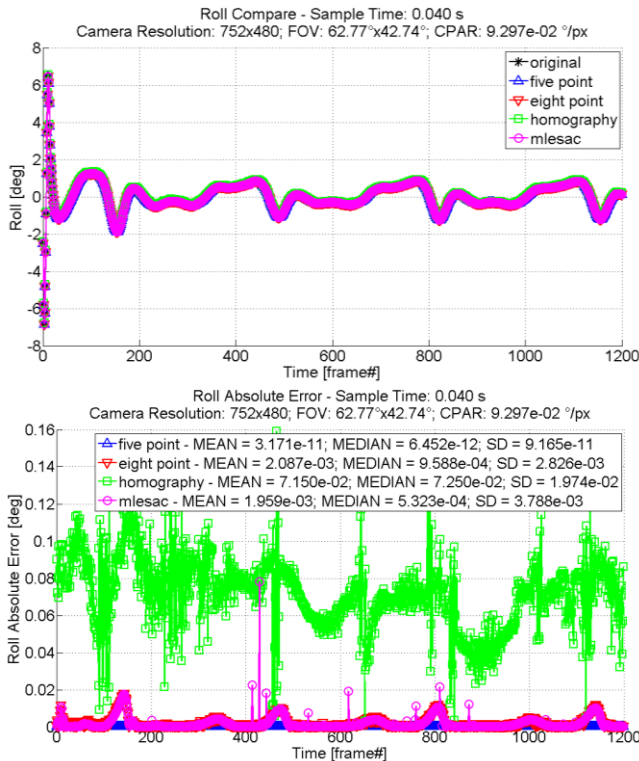


Figure 9. Compare of the four different algorithm with absolute feature point precision on sinusoidal; top the roll angle, bottom the error of the roll angle; with black star the original, with blue triangle the five point, with red triangle the eight point, with green square the homography and with magenta circle the MLESAC results

As shown in Figure 9., without any feature

point coordinate error the five point algorithm is the best. The error of the five point algorithm is close to the numerical precision of the calculations. The errors of other two epipolar geometry based solutions are also at least one order of magnitude smaller than the 1 pixel angular resolution. And the homography has got an error that remains below 1 pixel.

The effect of the translation is shown in the next figure with the pitch angle, which is most affected. Theoretically due to the bigger baseline separation bigger translation between the two frames could be advantageous for the three algorithms which are based on the epipolar constraint (5 point, 8 point and MLESAC). It can be seen in the figure practically this is not true, the error is bigger as the step is bigger in between the frames except for the five point algorithm in some situations. One possible explanation is that the number of the feature points which can be seen in both frames is reduced and the feature points are more drifted to the side of the image.

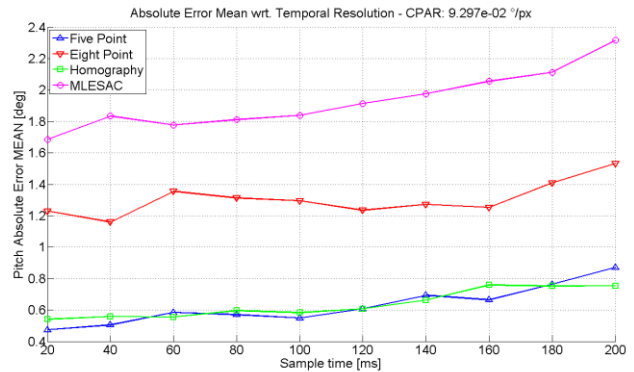


Figure 10. Effect of the translation through the sample time change on the pitch angle error; on sinusoidal; the pitch angle is most affected by the translation effect

E. Results with subpixel precision

As mentioned before, the subpixel feature point extraction is simulated by random, normal distribution noise on absolute precise feature point coordinates.

Surprisingly, the five point algorithm performs badly in this situation compared to the others (Figure 11.). This could be because this algorithm is less robust than the others. Usually high performance algorithms have lower robustness.

On the other hand if the spatial resolution is less (CPAR=0.093), again the mean error of the five point algorithm is smaller than of the eight

point and MLESAC with the noisy pixel data. (Figure 12.) The effect of the temporal resolution change is similar to the previous case and the standard deviation shows similar features.

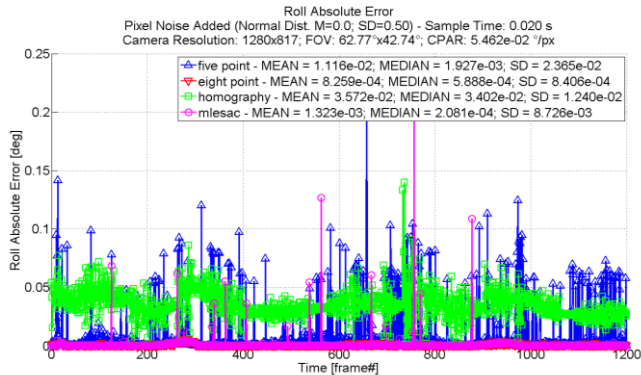


Figure 11. Roll error with subpixel resolution on sinusoidal; the five point algorithm performance is worse than expected

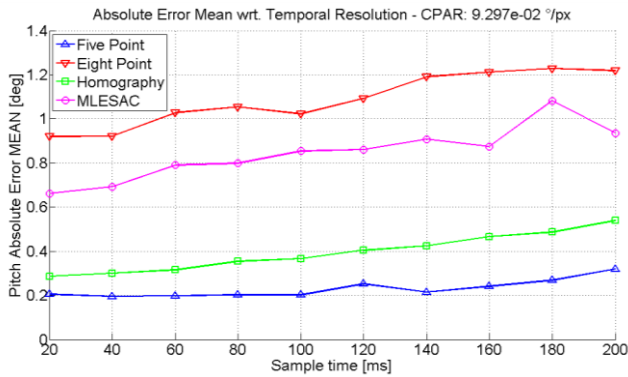


Figure 12. Effect of the translation through the sample time change on the pitch angle; on sinusoidal

F. Results with pixelized coordinates

In this case the performance of the algorithms changed again. The best performing algorithm is the five point, but most of the time the homography can keep up with its performance (see Figure 13.).

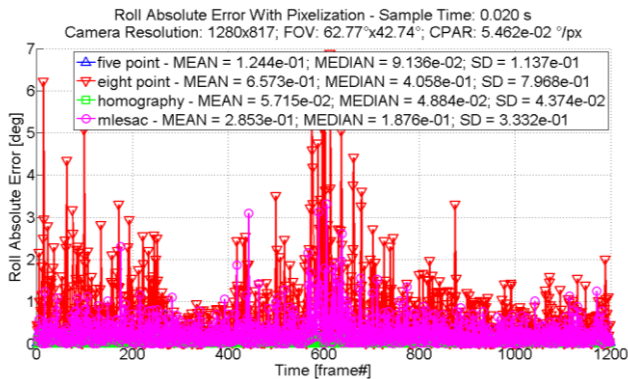


Figure 13. Roll error with pixelization on sinusoidal path; the homography is almost as good as the five point algorithm

This is important because the computational need of the homography algorithm is much less

than the others. That is the pixelization has got a smaller effect on the homography algorithm. An extreme example is the roll error of homography which is almost independent to the CPAR (see Figure 14.).

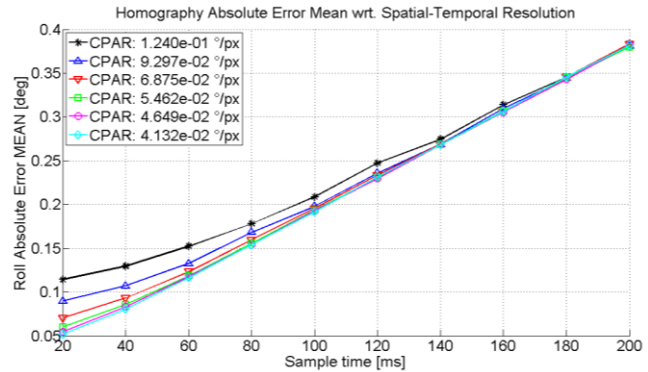


Figure 14. Roll error mean with pixelization on sinusoidal; the roll error mean of the homography is almost independent on the resolution

G. Results with pixelized coordinates and noise

The noise added to the pixelized coordinates causes only a slightly higher error level compared to the pixelization. The results here are very similar to the results in the previous section. For example the yaw error change of the homography can be seen in Table 2.

	mean	median	sd
Absolute precision	$4.422 \cdot 10^{-2}$	$3.119 \cdot 10^{-2}$	$4.670 \cdot 10^{-2}$
With noise	$4.609 \cdot 10^{-2}$	$3.278 \cdot 10^{-2}$	$5.065 \cdot 10^{-2}$
Pixelized	$6.036 \cdot 10^{-2}$	$3.845 \cdot 10^{-2}$	$6.924 \cdot 10^{-2}$
Pixelized & noise	$7.002 \cdot 10^{-2}$	$4.051 \cdot 10^{-2}$	$9.379 \cdot 10^{-2}$

Table 2. Yaw error of homography changing with different feature point precision for the CPAR=0.055°/px camera

IV. HIL SIMULATION AND MEASUREMENT RESULTS

In this section the coupled GPS/IMU/Camera attitude estimator system is introduced. As a measurement example some of the datasets from the HIL simulation tests are run in this system.

The HIL simulation includes the aircraft dynamical model in MATLAB Simulink completed with the RC transmitter, and on-board microcontroller. The control inputs from the transmitter and microcontroller are sent into the PC through an RS-232 or CAN interface. The sensory system of the aircraft is emulated in Simulink, the sensor data is sent to the microcontroller again through an RS-232 or CAN interface. This way the

real electronics controls the aircraft simulation (for details about a HIL configuration see [15]).

A. Coupled GPS/IMU/Camera attitude estimator implementation

In this section the coupling of a GPS/IMU-based aircraft attitude estimation algorithm (from [25]) with the camera-based rotation matrix increment estimate (T_{Δ}) is introduced.

The original estimator is an extended kalman filter (EKF) which uses the angular rate and acceleration measurements to propagate the attitude, velocity and (latitude, longitude, altitude = LLA) position dynamics of the aircraft. The Euler angles, earth relative velocity and position are predicted using system dynamic equations.

In the correction step of the EKF GPS position and velocity measurements are used to calculate the prediction error and update the attitude, velocity and position accordingly. The rate gyro and accelerometer biases are also estimated.

The camera based rotation increment can be included into the measurement step as an information about the change of the direction cosine matrix (DCM). This is explained in the forthcoming part.

The algorithm was implemented in MATLAB, and tested on the same data used in the previous section.

This data was generated in HIL excluding sensor noise and wind disturbance. The goal is to test the sensor fusion on exact data and so compare the performance of the different image processing algorithms in an ideal situation. From HIL, the real Euler angles are known. The attitude considers the error in the DCM (here T_{BE}) instead of the error of Euler angles. The aircraft orientation in the second camera frame can now be represented in two different ways:

$$T_{BE}' = T_{\Delta} T_{BE}(cam) \text{ from the camera}$$

$$T_{BE}' = (I + [\delta E^{\wedge}]) T_{BE}(-) \text{ from the GPS/IMU.}$$

Here $T_{BE}(cam)$ is the rotation matrix related to the first camera frame. $T_{BE}(-)$ is the rotation matrix predicted from actual IMU data. δE is the vector representing rotation errors and $[\delta E^{\wedge}]$ is the skew-symmetric matrix created from it. Comparing the two equations $[\delta E^{\wedge}]$ can be expressed:

$$T_{\Delta} T_{BE}(cam) (T_{BE}(-))^T - I = [\delta E^{\wedge}]'$$

Of course, because of measurement and numerical errors $[\delta E^{\wedge}]'$ will not be skew-symmetric (this is denoted by $()'$). But it can be made skew-symmetric with the following transformation:

$$[\delta E^{\wedge}] = \frac{[\delta E^{\wedge}]' - ([\delta E^{\wedge}]')^T}{2}$$

From $[\delta E^{\wedge}]$ the rotation error terms can be directly incorporated into the attitude estimator algorithm as measurements.

With the inclusion of camera data three working modes should be defined in the attitude estimator considering 5Hz GPS and 50Hz camera data:

1. Only GPS data, correction with GPS measurement
2. Only camera data, correction with camera measurement
3. Both GPS and camera data, correction with both of them

This means that the measurement equations of the attitude estimator Extended Kalman Filter (EKF) are changing according to the available data. In this application only the first and third modes are used, because the second mode needs some reformulation or tuning according to the simulation results.

B. Coupled GPS/IMU/Camera attitude estimator Results

Two examples are shown here. First the GPS/IMU solution and the error against the ground truth is plotted (Figure 15. and Figure 16.), and then the results of the homography and five point algorithm run with the random noise case are shown (Figure 17. and Figure 18.). In both the homography and the five point cases the sample time is minimum, that is 20ms, the CPAR is 0.093, and the sinusoidal path is used. For the five point algorithm only the errors are plotted (Figure 18.), because the angle comparison is very similar to the homography.

The comparison of the GPS/IMU results with the GPS/IMU/Camera solution shows that the latter has a better precision as with the inclusion of the Camera data the bias of the pitch estimation is removed.

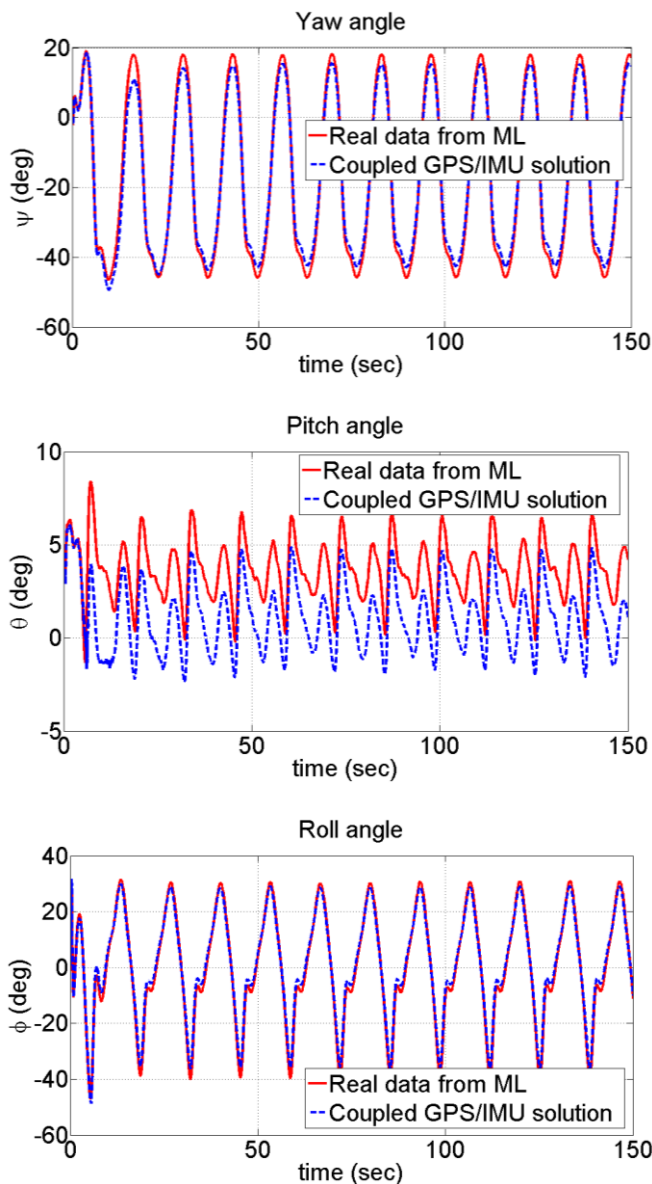


Figure 15. The result of the GPS/IMU fusion with respect to the ground truth; with red solid line the ground truth and with blue dashed line the result of the EKF; The bias in the pitch value can be seen in the middle figure

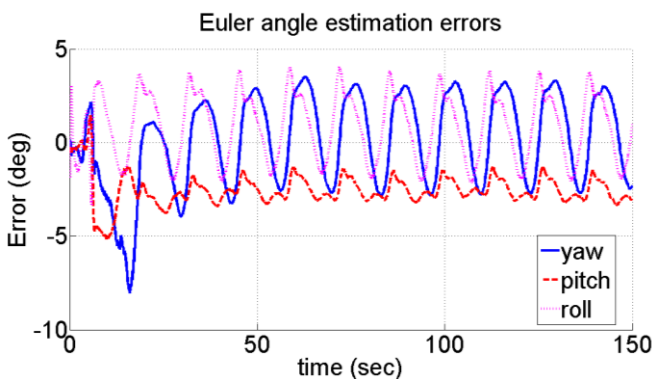


Figure 16. The error of the GPS/IMU fusion with respect to the ground truth

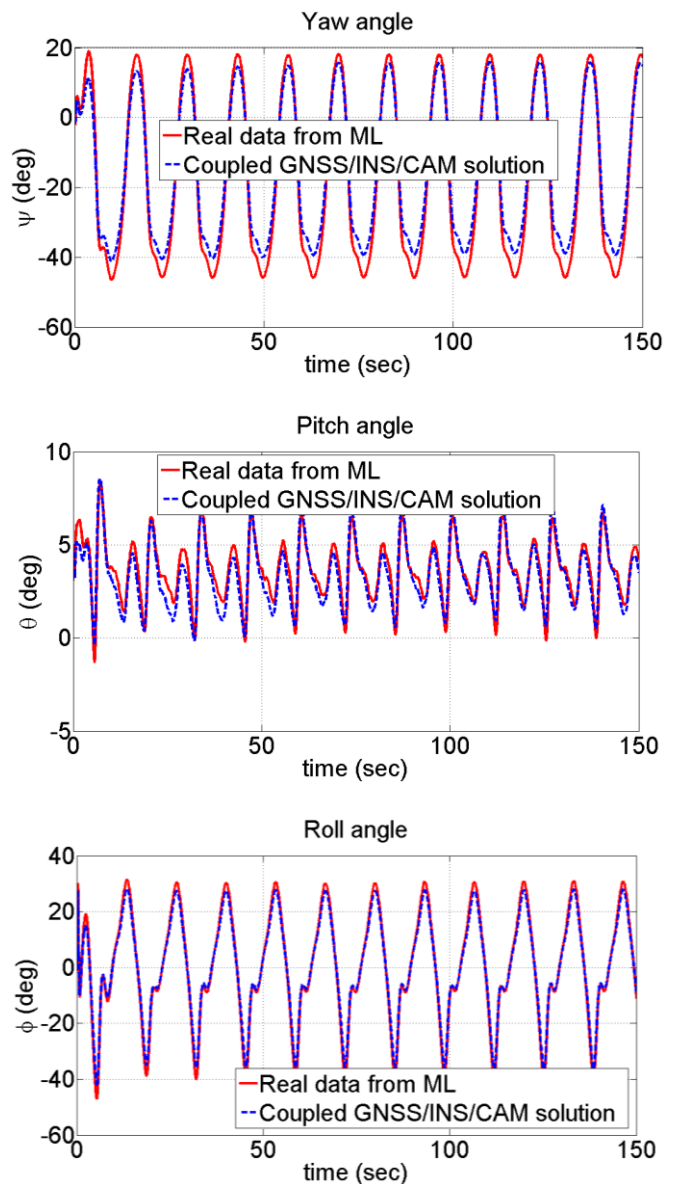


Figure 17. The result of the GPS/IMU/Camera fusion with the homography with respect to the ground truth; with red solid line the ground truth and with blue dashed line the result of the EKF; The pitch bias is eliminated

The comparison of the homography and the five point algorithm shows that the homography is indeed less affected by the noise as it was stated in III.E. The yaw angle error is less for the homography and the other two angles are at the same level. (Figure 19.)

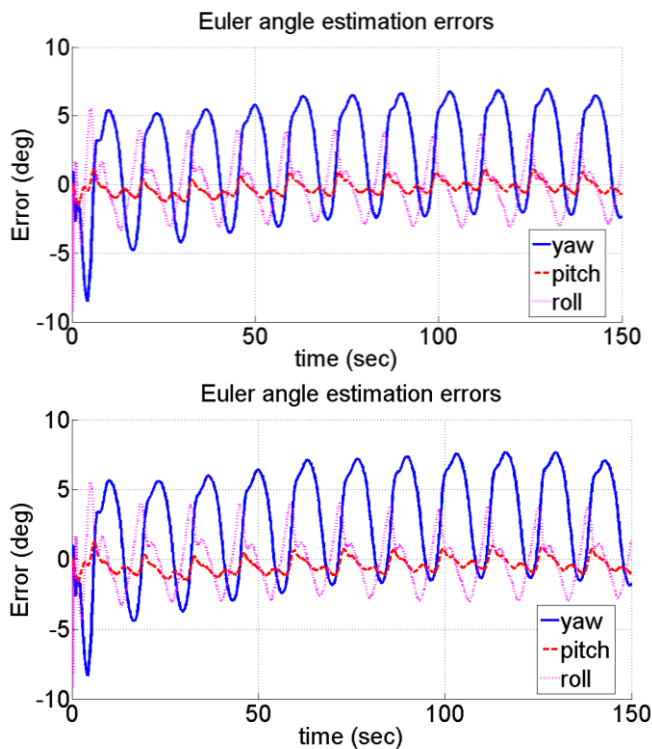


Figure 18. The Euler angle error of the GPS/IMU/Camera fusion with respect to the ground truth; top the results of the homography, bottom the results of the five point algorithm; the trends are similar

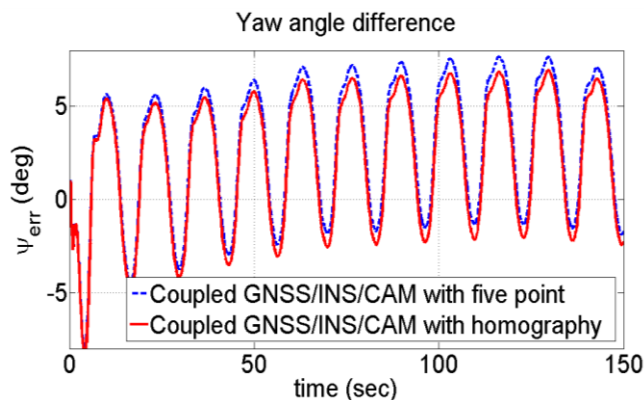


Figure 19. The yaw error of the GPS/IMU/Camera fusion with respect to the ground truth

V. CONCLUSIONS

In this paper the error analysis of four image processing algorithms targeting the reconstruction of camera orientation change is introduced. It is shown how the change of the spatial or temporal resolution as well as random noise affects these algorithms. It can be stated that the homography algorithm can be used in those situations where the computational power is restricted. If the precision is important than either the five point algorithm and the homography can be used keeping in mind the effect of translation and the pixelization.

Another contribution of the paper is the presentation of our first results with a coupled GPS/IMU/Camera based attitude estimator algorithm. It is shown that the inclusion of image based attitude changes can remove the bias error from pitch estimation of the GPS/IMU algorithm and so improve precision.

The next steps in the development will be the activation of mode 2 which means measurement correction based solely on camera orientation change and test and tuning based-on real flight data.

The analysis of image processing algorithms can be broadened to test others, for example a more advanced RANSAC variant. Other opportunity is the investigation of the effect of outliers in the paired feature points.

ACKNOWLEDGMENT

The authors thank Levente Hajder for his consultations on pose estimation algorithms and Borbála Pencz for her dedicated work on data processing.

The ONR Grant (Number: N62909-10-1-7081) is greatly acknowledged. The support of the grants TÁMOP-4.2.1.B-11/2/KMR-2011-0002 and TÁMOP-4.2.2/B-10/1-2010-0014 is gratefully acknowledged.

The financial support provided jointly by the Hungarian State, the European Union and the European Social Fund through the grant TÁMOP 4.2.4.A/1-11-1-2012-0001 is gratefully acknowledged.

REFERENCES

- [1] G. L. Dillingham, "Unmanned Aircraft Systems: Measuring Progress and Addressing Potential Privacy Concerns Would Facilitate Integration into the National Airspace System," U.S. Government Accountability Office, Washington, DC, Rep. GAO-12-981, Sep. 18, 2012.
- [2] T. Hutchings, "Architecting UAV sense & avoid systems," in *2007 Institution of Engineering and Technology Conference on Autonomous Systems*, 2007, pp. 1–8.
- [3] L. Forlenza et al., "Flight Performance Analysis of an Image Processing Algorithm for Integrated Sense-and-Avoid Systems," *Int. J. Aerosp. Eng.*, vol. 2012, no. 3, pp. 1–8, 2012.

- [4] T. Zsedrovits et al., "Collision avoidance for UAV using visual detection," in *Proc. of 2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, 2011, pp. 2173–2176.
- [5] B. Vanek et al., "Performance analysis of a vision only Sense and Avoid system for small UAVs," presented at the AIAA Guidance, Navigation, and Control Conference, Reston, Virginia, 2011.
- [6] T. Zsedrovits et al., "Visual Detection and Implementation Aspects of a UAV See and Avoid System," presented at the 20th European Conference on Circuit Theory and Design (ECCTD 2011), Linköping, Sweden, 2011.
- [7] B. Vanek et al., "Performance analysis of a vision only sense and avoid system for small UAVs" presented at the 9th European Workshop on Advanced Control and Diagnosis, Budapest, 2011.
- [8] A. Zarandy et al., "A five-camera vision system for UAV visual attitude calculation and collision warning," *Comput. Vis. Syst. Lect. Notes Comput. Sci.*, vol. 7963, pp. 11–20, 2013.
- [9] T. Zsedrovits et al., "Towards Real-Time Visual and IMU Data Fusion," presented at the AIAA Guidance, Navigation, and Control Conference and Exhibit, 2014.
- [10] C.-C. Chu et al., "Performance comparison of tight and loose INS-Camera integration," in *Proc. of Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*, 2011, p. 3516.
- [11] T. Chu et al., "Monocular camera/IMU/GNSS integration for ground vehicle navigation in challenging GNSS environments," *Sensors (Basel)*, vol. 12, no. 3, pp. 3162–85, Jan. 2012.
- [12] H. Stewénius et al., "Recent developments on direct relative orientation," *ISPRS J. Photogramm. Remote Sens.*, vol. 60, no. 4, pp. 284–294, Jun. 2006.
- [13] P. H. S. Torr and A. Zisserman, "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry," *Comput. Vis. Image Underst.*, vol. 78, no. 1, pp. 138–156, Apr. 2000.
- [14] S. Choi et al., "Performance Evaluation of RANSAC Family," in *Proc. of the British Machine Vision Conference 2009*, 2009, pp. 81.1–81.12.
- [15] Y. C. Paw, "Synthesis and validation of flight control for UAV," PhD Dissertation, University of Minnesota, 2009.
- [16] R. W. Beard and T. W. McLain, *Small unmanned aircraft: Theory and practice*. Princeton: Princeton University Press, 2012.
- [17] B. Vanek et al., "Safety Critical Platform for Mini UAS Insertion into the Common Airspace," presented at the AIAA Guidance, Navigation, and Control Conference, Reston, Virginia, 2014.
- [18] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [19] W. K. Pratt, *Digital Image Processing, PIKS Inside*. John Wiley & Sons Inc, 2001.
- [20] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–77, Jun. 2004.
- [21] R. I. Hartley, "Chirality," *Int. J. Comput. Vis.*, vol. 26, no. 1, pp. 41–61, 1998.
- [22] G. L. Mariottini and D. Prattichizzo, "EGT for multiple view geometry and visual servoing: robotics vision with pinhole and panoramic cameras," *Robot. Autom. Mag. IEEE*, vol. 12, no. 4, pp. 26–39, 2005.
- [23] J.-Y. Bouguet, "Complete camera calibration toolbox for matlab," 2004. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.
- [24] H. Stewénius, "Calibrated Fivepoint Solver," 2010. [Online]. Available: <http://www.vis.uky.edu/~stewe/FIVEPOINT/>.
- [25] S. Gleason and D. Gebre-Egziabher, *GNSS applications and methods*. Artech House, 2009.