

Performance Analysis of Camera Rotation Estimation Algorithms for UAS Sense and Avoid

Tamas Zsedrovits[†], Peter Bauer*, Mate Nemeth^{**}, Borbala Jani Matyasne Pencz*, Akos Zarandy^{**}, Balint Vanek*, Jozsef Bokor*

[†]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—Several camera rotation estimator algorithms are tested in simulations and on real flight videos in this paper. 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. Simulation results show that many times even the simplest algorithm can perform at an acceptable level of precision for the sensor fusion.

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

I. INTRODUCTION

In the last decade Unmanned Aircraft Systems (UAS) – beforehand Unmanned Aerial Vehicles (UAVs) – technology has evolved considerably. Besides the military applications now there is a great opportunity to use UAS in commercial applications as well [1]. More and more companies start to develop applications and services based on the UAS platform with low cost, mid-size UAVs. On the other hand, UAS applications are very limited today because of limited access to national airspace (NAS).

With the aim of integrating UAVs into NAS, their reliability needs to be increased as well as their capabilities need to be extended further, their ease of use needs to be improved and their cost have to be decreased. Two of the most important features are the collision avoidance or sense-and-avoid (SAA) capability and hardware redundancy [2]. The SAA has to be run on-board even if the connection between the aircraft and the control station is lost or some of the sensors fail.

As the goal is to develop a cost effective system the size and the energy consumption is limited. Thus a visual SAA system is favourable against conventional RADAR based solutions [3]. The main advantages of the EO based SAA systems are that they are lightweight and have affordable price. The drawbacks are the relatively high computational cost of the processing

algorithms and the restricted weather conditions and range. As the examples show, despite the drawbacks these systems can be a good choice for small UAS [4], [5].

It is also essential to have redundancy in multiple levels to handle critical situations: 1) the number of the similar sensor modules has to be increased and the system has to be able to fuse the information from different sensor modalities. In this case the use of our camera can be broadened to localisation task besides its main function in collision avoidance. Furthermore, with a Camera-IMU fusion better accuracy can be reached in the ego motion as shown in [6]. It means that our ego motion based SAA algorithm can be speed-up which provides even higher separation distance or the avoidance of aircrafts with higher speed.

Our work was inspired by the results of Chu et al. In [7] performance comparison of two types (tight and loose), Camera-IMU integration is through simulations. The authors say that tight coupling is better in terms of accuracy but it is less stable due to the linearization methods of the Kalman filters. It means that in low cost systems loose integration can perform better.

From the same group a multi sensor fusion (Camera, IMU, GPS) for automobiles with real drive test is introduced [8]. The Camera-IMU fusion outperforms the conventional IMU-GPS systems. The real time capability of the system is not solved yet. The main challenge here is the real time implementation of the image processing algorithm.

Many kind of video processing algorithms are suitable for UAV attitude estimation from a homography based calculation to the more difficult 5 point algorithm. Most of the time the original authors of these methods give some test cases, in which the accuracy and complexity is compared to previous works [9] [10]. Additionally, there are review papers which assemble and compare different algorithms from some perspective [11].

In our previous work some parts of an error analysis for these algorithms for Camera-IMU-GPS fusion were introduced in simulations [12]. In this paper the error analysis of four algorithms is shown. The analysis is done with realistic flight paths generated by the HIL simulator [13]. 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 and also preliminary results on flight videos 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 Camera-IMU-GPS fusion algorithm, and in Section V measurements on flight videos are introduced. Finally, the conclusions are drawn in Section VI.

II. ALGORITHMS

The coordinate frames and the transformation from the image to the real world is introduced in this section. Also the camera rotation estimation algorithms are described. Four kind of feature point based pose estimation algorithm are chosen for the test, which give a good cross-section of the pose estimators used. First, a homography based algorithm is tested, as the simplest solution for the problem with small computational need. Second, the 8 point algorithm as a typical method in the computer vision community. The 5 point algorithm is the third with better stability and accuracy, but higher computational need. The fourth one is an iterative, stochastic solution, the MLESAC.

A. Coordinate systems

In our case the conventional North-East-Down (NED) frame from [14] is used as an inertial (non-moving, non-rotating) frame (earth frame). This approximation is correct if our mid-size UAS travels short distances (less than 2 km).

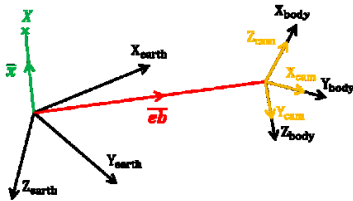


Fig. 1. The earth, the body and the camera coordinate systems in this specific scenario where the origins of body and camera systems coincide

The body and camera frames are also used in the calculations. In our case, the axes of the camera system are parallel with the axes of the body system, but the camera system is rotated (Fig. 1).

In Fig. 1 X is a feature point in the earth coordinate system characterized by vector \bar{x}_{earth} (the $\bar{(\)}_{\text{earth}}$ means

a vector with coordinates in earth coordinate system). $\bar{e}_{\text{b}}_{\text{earth}}$ gives the position of the body frame (here also the camera frame) relative to earth. The coordinates of point X in the camera frame can be calculated as follows: $\bar{x}_{\text{cam}} = \overline{\text{CB}} \overline{\text{BE}} (\bar{x}_{\text{earth}} - \bar{e}_{\text{b}}_{\text{earth}})$ Here, $\overline{\text{F}_2\text{F}_1}$ defines a transformation matrix from frame F_1 to F_2 .

B. Inertial Measurement Unit model

In the measurements and simulations a custom built IMU is used. It consists of the conventional components which are necessary for outdoor navigation, such as accelerometer, rate gyro, differential and absolute pressure sensor, magnetometer and a GPS unit [15].

C. Camera measurements

The camera is described as a special case of a projective camera [16]. The projection matrix $\overline{\text{P}}$ consists of the intrinsic and extrinsic parameters of the camera. The decomposition to the rotation matrix $\overline{\text{R}}$ and the translation vector \bar{t} is as follows:

$\overline{\text{P}} = \overline{\text{K}} [\overline{\text{R}} | \bar{t}] \overline{\text{R}}$ and \bar{t} are the extrinsic parameters, describing the attitude and position of the camera in the real world. $\overline{\text{K}}$ contains the intrinsic parameters in the

following way:

$$\overline{\text{K}} = \begin{bmatrix} f_x & 0 & p_1 \\ 0 & f_y & p_2 \\ 0 & 0 & 1 \end{bmatrix},$$

where f is the focal length in pixels (it can be different along the x and the y axes) and \bar{p} is the camera principal point.

Furthermore the camera is characterized with its resolution, because in this study the effect of spatial resolution change is investigated. The resolution is given in CPAR, which is the angular resolution of the central pixel. This way not only the camera sensor, but the optics, which is attached to the actual camera is taken into account. This way the results with different cameras can be compared easier. The CPAR is calculated in the following way:

$$\text{CPAR} = \tan^{-1} \frac{1}{f}, \text{ where } f \text{ is the focal length.}$$

D. Feature extraction and matching

A modified Harris corner detector is run on the image flow [17]. Two constraints are added to the conventional Harris algorithm: 1) we are looking for features only below the horizon, in the ground region and 2) from a given region only the strongest corner is selected. With these two constraints the number of feature locations is limited and also degenerate point pairs are neglected in most cases.

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

A basic scene homography based algorithm is used as one of the simplest solution for the problem. We assumed that the translation of the camera is negligible compared to the camera rotation thus only the latter is calculated. This problem is also known as inhomogeneous DLT problem and used for panoramic mosaicking. It can be described with the following equation:

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

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

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 elements of $\bar{\mathbf{h}}$ 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 $\bar{\mathbf{A}}$ 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 [16].

F. 8 point algorithm

The normalised 8 point algorithm is evaluated [16], where from the feature pairs the fundamental matrix \mathbf{F} is computed, which is defined by the epipolar constraint in the following way:

$$\bar{\mathbf{x}}'^T \bar{\mathbf{F}} \bar{\mathbf{x}} = 0$$

With the calibration of the camera the essential matrix $\bar{\mathbf{E}}$ can be directly computed transforming $\bar{\mathbf{F}}$ with the intrinsic matrices of the two consecutive cameras:

$$\bar{\mathbf{E}} = \bar{\mathbf{K}}'^T \bar{\mathbf{F}} \bar{\mathbf{K}}$$

Here we have the same camera during the image sequence, thus $\bar{\mathbf{K}}' = \bar{\mathbf{K}}$.

G. MLESAC

The MLESAC algorithm is a variant of the RANSAC, which is an iterative, stochastic method [10]. Here, again the fundamental matrix is computed, but it is more robust than the 8 point algorithm, because it uses probability features. The MLESAC is well balanced in the terms of accuracy and complexity as it is shown in [11], moreover the implementation is available online.

H. 5 point algorithm

The essential matrix can be computed also with a

more complex algorithm if we have calibrated camera and at least five feature point pairs, because it has got only five degrees of freedom. An efficient and numerically stable 5 point solver is given in [9] and [18]. From the test cases written in these papers it turns out that the 5 point algorithm could be more stable in the pure translational and pure rotational cases than other algorithms.

I. Camera pose from $\bar{\mathbf{E}}$

The three epipolar constraint based algorithm (5 point, 8 point, MLESAC) gives us the $\bar{\mathbf{E}}$ matrix from the feature point pairs.

From $\bar{\mathbf{E}}$ the two camera matrices can be calculated in canonical form ($\bar{\mathbf{P}} = [\bar{\mathbf{I}} | \bar{\mathbf{0}}]$ and $\bar{\mathbf{P}}' = [\bar{\mathbf{R}} | \bar{\mathbf{t}}]$) and $\bar{\mathbf{E}}$ can be decomposed as $\bar{\mathbf{E}} = [\bar{\mathbf{t}}]_{\times} \bar{\mathbf{R}}$, where $[\bar{\mathbf{t}}]_{\times}$ is a skew symmetric form of translation vector $\bar{\mathbf{t}}$ representing vector cross product. For the calculation $\bar{\mathbf{E}}$ has to be decomposed with SVD as follows:

$$\bar{\mathbf{E}} = \bar{\mathbf{U}} \text{diag}(1,1,0) \bar{\mathbf{V}}^T$$

From that four solutions can be constructed for the second camera. Only one of them satisfy the chirality constraint [19] that is in only one arrangement are the projected feature points in front of both cameras [16] for example: $\bar{\mathbf{P}}' = [\bar{\mathbf{U}} \bar{\mathbf{W}}^T \bar{\mathbf{V}}^T | \bar{\mathbf{u}}_3]$, where $\bar{\mathbf{u}}_3$ is the 3rd column of $\bar{\mathbf{U}}$ and

$$\bar{\mathbf{W}} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

J. UAV attitude change reconstruction from camera rotation change

The rotation of the camera ($\bar{\mathbf{R}}$) in between the two consecutive images can be calculated from the corresponding features, assuming canonical cameras. In the following deduction calibrated cameras and normalized image coordinates are assumed.

The original $\bar{\mathbf{x}}_{cam}$ not normalized feature point location can be transformed into the first camera frame in the following way:

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

In the second camera's frame the same $\bar{\mathbf{x}}_{cam}$ vector can be given using the $\bar{\mathbf{P}}'$ camera matrix with the following formula:

$$\bar{\mathbf{x}}' = \bar{\mathbf{P}}' \begin{bmatrix} \bar{\mathbf{x}}_{cam} \\ 1 \end{bmatrix} = [\bar{\mathbf{R}} \quad \bar{\mathbf{t}}] \begin{bmatrix} \bar{\mathbf{x}}_{cam} \\ 1 \end{bmatrix} = \bar{\mathbf{R}} \bar{\mathbf{x}}_{cam} + \bar{\mathbf{t}}$$

$\bar{\mathbf{x}}'$ is the image of point X in the second camera, which means after the rotation and translation of the aircraft body frame. This way $\bar{\mathbf{x}}'$ can be also constructed by considering the changed $\bar{\mathbf{B}}\bar{\mathbf{E}}'$ matrix and $\bar{\mathbf{e}}\bar{\mathbf{b}}'_{earth}$ vector: $\bar{\mathbf{x}}' = \bar{\mathbf{x}}_{cam}' = \bar{\mathbf{C}}\bar{\mathbf{B}} \bar{\mathbf{B}}\bar{\mathbf{E}}' (\bar{\mathbf{x}}_{earth} - \bar{\mathbf{e}}\bar{\mathbf{b}}'_{earth})$.

From the two representations of $\bar{\mathbf{x}}'$ and the original

expression for \bar{x}_{cam} by considering $\overline{\overline{BE}}' = \overline{\overline{\Delta BE}}$ and $\overline{\overline{eb}}'_{earth} = \overline{\overline{eb}}_{earth} + \overline{\overline{\Delta eb}}_{earth}$ the attitude change is: $\overline{\overline{\Delta}} = \overline{\overline{CB}}^T \cdot \overline{\overline{R}} \cdot \overline{\overline{CB}}$. The detailed deduction is in [6].

III. ESTIMATORS PERFORMANCE IN SIMULATIONS

The methods and results of the performance analysis of the pose estimation algorithms are introduced in this section. Using synthesized data, we have analysed the reachable precision with different algorithms. The algorithms are tested with different spatial and temporal resolutions in a MATLAB based simulation environment. As it is shown, the 5 point algorithm is the most precise. If the feature point coordinates are given with double precision, the error is in the range of the numerical error. If the point coordinates are discretized, like in a real camera image, the improvement of the error of the four algorithms calculated with different cameras are in a linear relation with the degree per pixel value of the actual cameras. It can be also seen that the performance of the homography based algorithm is close to the 5 point algorithm.

A. Simulation environment

For the performance comparison a MATLAB EGT toolbox based simulation environment is used [20]. The EGT served as a very simple rendering engine, calculating only the camera images and the feature point locations on realistic flight paths generated in our hardware-in-the-loop simulator [15]. Two kind of paths are used: 1) a sinusoidal trajectory and 2) a zig-zag path, both with nearly constant altitude. During the measurements it turned out that the tested algorithms show the same behaviour in both cases that is why only examples from the sinusoidal path are shown here.

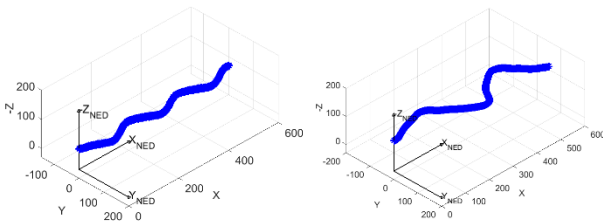


Fig. 2. Sinusoidal (left) and Zig-zag (right) path in the earth frame

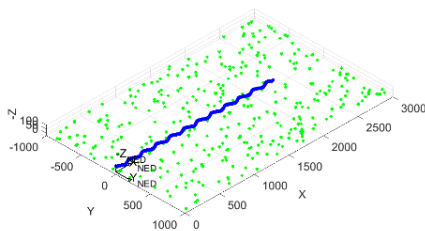


Fig. 3. Camera trajectory (with blue) and feature points (with green) in earth frame

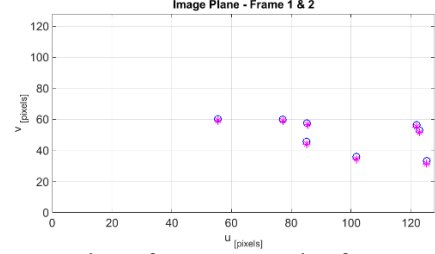


Fig. 4. Feature points of two consecutive frames on the image plane; with blue circles feature points of frame 1 and with magenta stars the feature points for frame 2; the camera resolution is 752×480

For the simulations 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 range of visibility is 800m, thus the camera can see only feature points which are closer than that. The dense feature point cloud can be clear away on the images near the horizon level this way. This simulates well the real images where feature points near the horizon cannot be extracted because the blurring atmospheric effects on distant objects.

In order to simulate the camera projection the camera matrix from the calibration of one of our small cameras is used. This small camera is part of the SAA system which is developed for the UAV. The camera was calibrated with the Camera Calibration Toolbox in MATLAB [21]. The resolution is 752×480 pixel and the Field of View (FOV) is ~63°×~43°. 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 Tab. 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

Tab. 1. Resolution and CPAR of cameras

The performance was measured in different spatial and temporal resolutions. In our SAA system the small cameras are running with maximum 56Hz sampling rate, which was approximated with 50Hz. In a different system due to less processing power or bigger resolution it can be lower. Thus the effect of the temporal resolution change was investigated in ten steps from 20 ms sampling time (50Hz) to 200 ms (5Hz).

For the spatial effects, the six different CPAR and for each case four cases are tested. First, each algorithms

are run on feature points with absolute precision. It means that the coordinates are not rounded after the projection. Then simulations with subpixel feature point resolution is run, because many feature point extraction algorithm support subpixel resolution. In this case, to the exact feature point coordinates a normal distribution noise with 0 mean and 0.5 pixel standard deviation is added. The luminance fluctuation and the nature of the point spread function in the real case is mimicked this way. Finally, the feature point coordinates are rounded to the nearest integer coordinate values to examine the effects of pixelization.

Standard implementations of the aforementioned algorithms are used. The 8 point algorithm and the MLESAC is implemented in the EGT toolbox [20] and the implementation of the 5 point algorithm is from its authors' website [22]. The homography algorithm was implemented in house based on [16].

B. Performance measures

During the measurements the rotation is calculated in between the two consecutive frames. From that the Euler angles are calculated and compared to the ground truth. To characterize the performance of each algorithm the absolute error of the three Euler angles are used.

$e_i = \sqrt{(\alpha_i - \alpha_i^{\text{calc}})^2}$, where α_i is the ground truth angle for the i^{th} frame (roll, pitch or yaw) and α_i^{calc} is the calculated angle. Furthermore, for each run also the mean, the median and the corrected standard deviation of the absolute error are calculated.

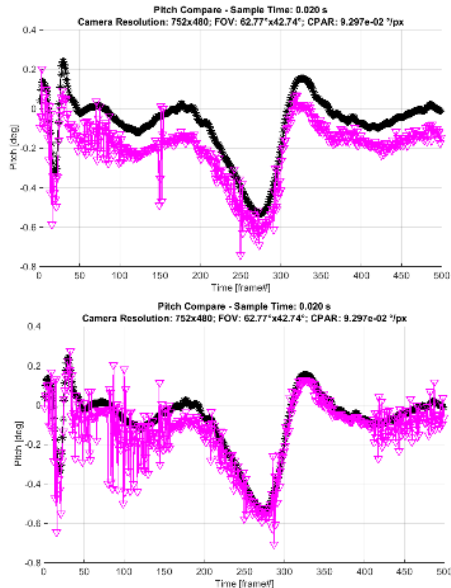


Fig. 5. Comparison of the pitch angle for homography on sinusoidal path; with black stars the ground truth, with magenta triangles the homography results; left without correction, right with correction

C. Homography algorithm correction

The homography cannot estimate the effect of the translation, that is why a simple correction step is introduced based on the sampling time, the measured velocity and the altitude. The translation has the biggest effect on the pitch, it has a smaller effect on the yaw and the error is distributed proportionally to the roll. Thus the correction term is the following:

$$\text{pitch}_{\text{correction}} = \frac{\cos(\text{roll}) + \sin(\text{roll})}{\cos(\text{roll})} \cdot f(\tau, \text{alt}, \bar{v})$$

$$\text{yaw}_{\text{correction}} = \frac{\cos(\text{roll}) - \sin(\text{roll})}{\cos(\text{roll})} \cdot f(\tau, \text{alt}, \bar{v})$$

The results are added to the pitch and yaw angles and $f(\tau, \text{alt}, \bar{v})$ is an empirical function based on the linear interpolation of the measured error term for different τ (sample time), altitude and velocity values. The effect of the correction of the pitch angle is shown on Fig. 5. It can be seen that the error is almost twice as much without the correction than with the correction.

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 algorithms can be checked and the effect of the temporal resolution change can be investigated independently.

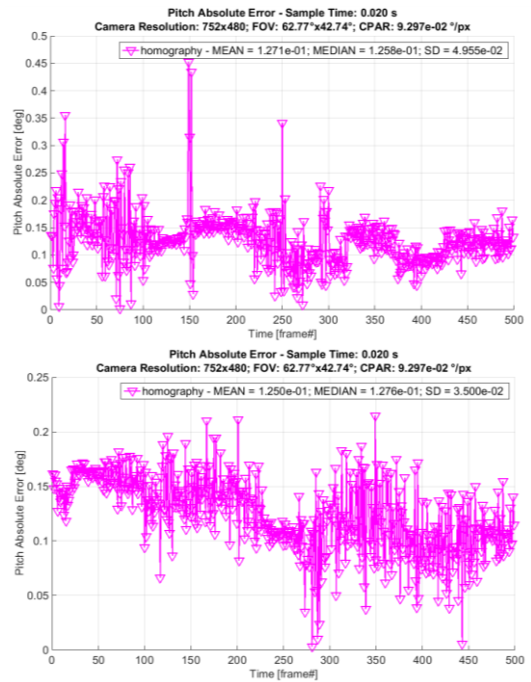


Fig. 6. Absolute error of pitch angle for homography on sinusoidal path; top without correction, bottom with correction

Fig. 7 shows the results of the simulations with absolute precise coordinates. The 5 point algorithm is in the range of the numerical precision in this case. The two other epipolar constraint based solutions perform

also well, the error is in the range of 0.1 pixel. And the performance of the homography is acceptable, as it is below 1 pixel.

As an example for the effect of translation, the pitch angle absolute mean error is shown in Fig. 8. The effect of the translation is the biggest on the pitch angle. Due to the bigger sample time the translation is bigger in between the two frames, which means bigger baseline separation. Theoretically, due to the bigger baseline separation, it 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 entirely true, the error is bigger as the step is bigger in between the frames except for the 5 point algorithm in some situations.

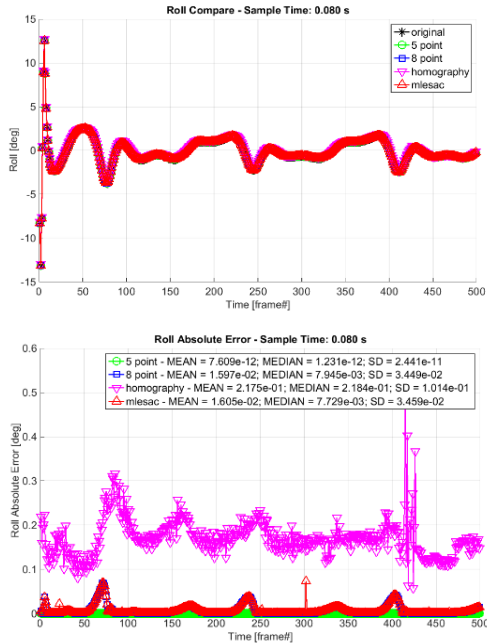


Fig. 7. Compare of the four different algorithm with absolute feature point precision on sinusoidal; top the roll angle, bottom the error of the roll angle; with black star the original, with green circle the 5 point, with blue square the 8 point, with magenta triangle the homography and with red triangle the MLESAC results

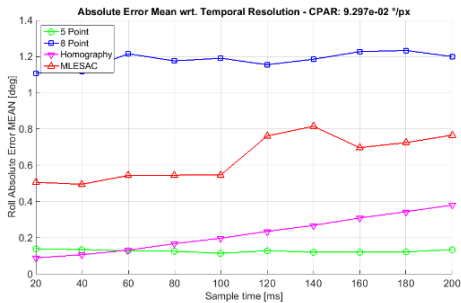


Fig. 8. Effect of the sample time change on the pitch angle error; on sinusoidal; the pitch angle is most affected by the translation effect

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. It is also important that the integral error would be smaller with the slower sampling frequency.

E. Results with subpixel precision

The subpixel resolution in the feature point coordinates is simulated with Gaussian random noise added to the precise coordinates.

Surprisingly there are random spikes in the 5 point algorithm error function in this case (Fig. 9). The explanation for this phenomenon can be some kind of numerical instability. Tab. 2 also shows that the 5 point algorithm is worse in this case than the others. On the other hand, as it is shown in Fig. 9 the mean error of the 5 point algorithm is smaller than of the 8 point and MLESAC with the noisy pixel data if the spatial resolution is small.

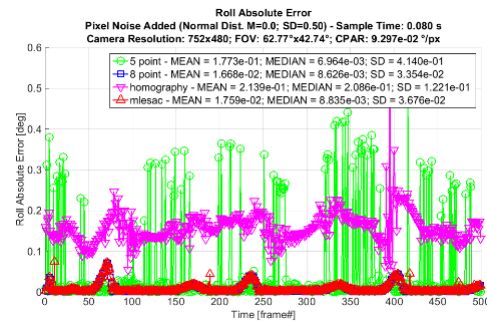


Fig. 9. Roll error with subpixel resolution on sinusoidal; the 5 point algorithm performance is worse than expected

Precision	5 point	8 point	Homography	MLESAC
Absolute	$3.171 \cdot 10^{-11}$	$2.087 \cdot 10^{-3}$	$5.065 \cdot 10^{-2}$	$1.323 \cdot 10^{-3}$
Subpixel	$1.234 \cdot 10^{-1}$	$1.080 \cdot 10^{-2}$	$7.150 \cdot 10^{-2}$	$1.959 \cdot 10^{-3}$
Pixelized	$9.371 \cdot 10^{-2}$	$5.476 \cdot 10^{-1}$	$1.169 \cdot 10^{-1}$	$3.240 \cdot 10^{-1}$

Tab. 2. Roll error of the four algorithms changing with different feature point precision for the CPAR=0.093°/px camera

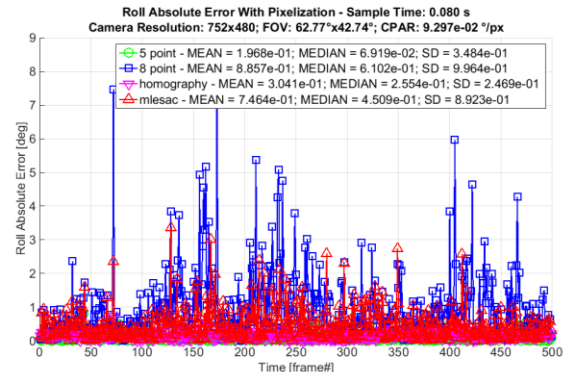


Fig. 10. Roll error with pixelization on sinusoidal path; the homography is almost as good as the 5 point algorithm

F. Results with integer coordinates

The best performing algorithm on the pixelized coordinates is the 5 point again, but most of the time the homography can keep up with its performance (see Fig. 10). This is important because the computational need of the homography algorithm is much less than the others.

From computational point of view, the five point algorithm is the most demanding, as it requires either the computation of the roots of a 10th degree polynomial or the computation of the Gröbner-basis of a 10*20 matrix. Though the latter is faster, it is still challenging in an embedded system since its computational need is very high.

The simplest is the homography, which requires the solution of a few straightforward equations and a singular value decomposition (SVD) on a 9*n matrix, where n is the number of co-ordinate pairs, which is most of the time is around 20. In a typical situation the difference of the computational demand of the homography is 2 orders of magnitude smaller than the computational demand of five point algorithm in the number of the multiplications.

Finally the next table summarizes the spatial resolution change in the pixelized case. The improvement of the error of the four algorithms calculated with different cameras are in a linear relation with the degree per pixel value of the actual cameras. It can be also seen that the performance of the homography based algorithm is close to the 5 point algorithm.

	double precision	pixelized (0.09°/px)	pixelized (0.05°/px)	pixelized (0.04°/px)
5p	9.1*10 ⁻¹¹	7.40*10 ⁻²	5.42*10 ⁻²	4.03*10 ⁻²
8p	2.83*10 ⁻³	4.99*10 ⁻¹	3.55*10 ⁻¹	2.82*10 ⁻¹
hom	1.97*10 ⁻²	7.41*10 ⁻²	5.64*10 ⁻²	5.09*10 ⁻²
MLESAC	3.79*10 ⁻³	2.63*10 ⁻¹	1.99*10 ⁻¹	1.61*10 ⁻¹

Tab. 3. Mean of the error produced by the different camera pose estimator algorithms

IV. HIL SIMULATION AND MEASUREMENT RESULTS

The coupled Camera-IMU-GPS attitude estimator system is introduced in this section. The HIL simulation includes the aircraft dynamical model in MATLAB Simulink completed with the RC transmitter, and on-board microcontroller. This is a closed loop system, where the hardware components of the real system with all the electronics can be tested. The sensory system of the aircraft is emulated in Simulink, and the sensor data is sent to the microcontroller, which can be placed also on board of the aircraft [13].

A. Coupled Camera-IMU-GPS attitude estimator

In this section the coupling of an IMU-GPS based aircraft attitude estimation algorithm (from [23]) with the camera-based rotation matrix increment estimate ($\bar{\Delta}$) 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 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 camera rotation.

The simulation 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 rotation matrix (here $\overline{\mathbf{BE}}$) instead of the error of Euler angles. The aircraft orientation in the second camera frame can now be represented in two different ways:

$$\begin{aligned} \overline{\mathbf{BE}}' &= \bar{\Delta} \overline{\mathbf{BE}}^{\text{camera}} \text{ from the camera} \\ \overline{\mathbf{BE}}' &= (\bar{\mathbf{I}} + [\delta\mathbf{E}]_{\times}) \overline{\mathbf{BE}}^{\text{IMU-GPS}} \text{ from the IMU-GPS} \end{aligned}$$

B. Measurement Results

In this section two measurement examples are shown. First the IMU-GPS solution and the Camera-IMU-GPS solution against the ground truth is plotted and the error of the Euler angles are shown (Fig. 11, Fig. 12 and Fig. 13). Then the results of the homography and 5 point algorithm run with the random noise case are shown (Fig. 14). In both the homography and the 5 point cases the sample time is 20ms, the CPAR is 0.093, and the sinusoidal path is used. For the homography only the errors are plotted, because the angle comparison is very similar to the 5 point algorithm.

In the following figures the IMU-GPS solution is compared to the Camera-IMU-GPS solution based on the HIL measurements.

The comparison of the IMU-GPS results with the Camera-IMU-GPS 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.

The comparison of the homography and the 5 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. (Fig. 15)

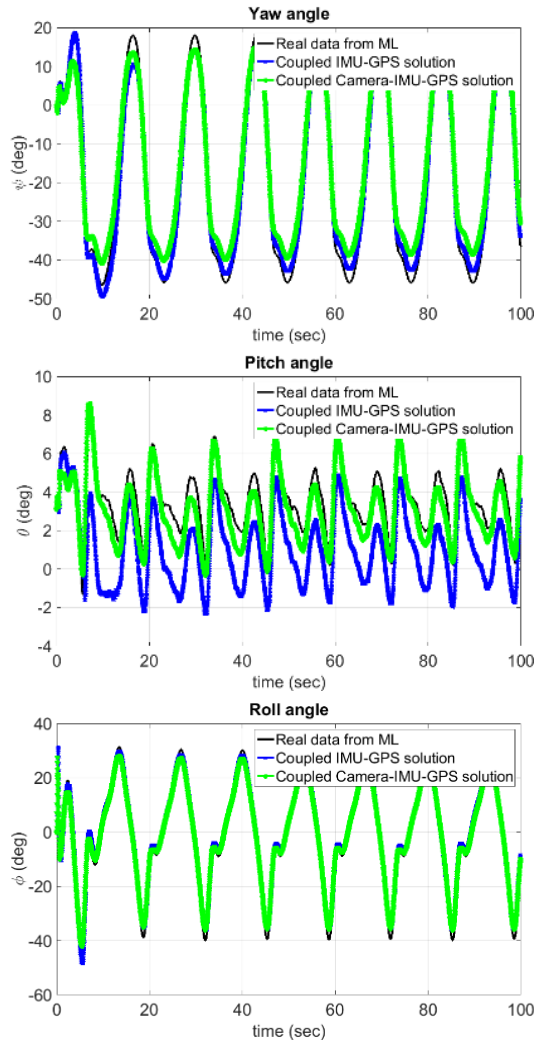


Fig. 11 The result of the fused data with respect to the ground truth; with black solid line the ground truth; with blue dashed line the result of the EKF; with green the Camera-IMU-GPS result; The bias in the pitch value can be seen in the middle figure

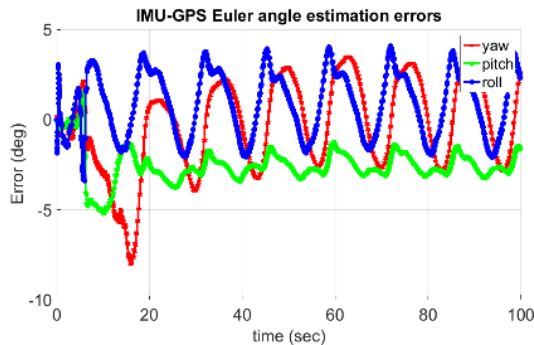


Fig. 12. The error of the IMU-GPS fusion with respect to the ground truth

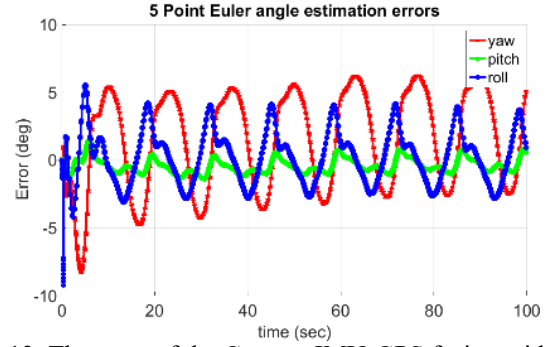


Fig. 13. The error of the Camera-IMU-GPS fusion with the 5 point algorithm with respect to the ground truth

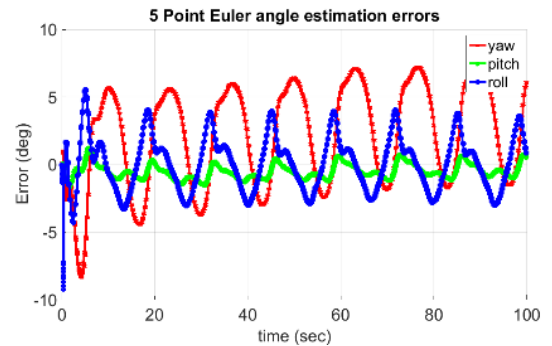
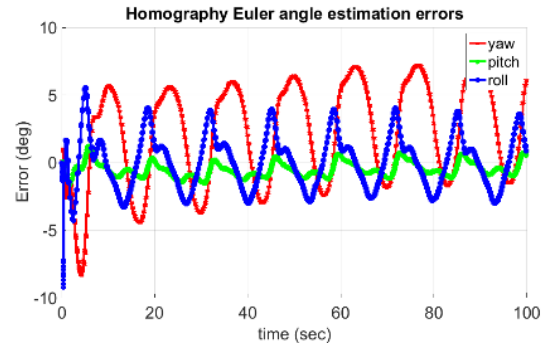


Fig. 14. The Euler angle error of the Camera-IMU-GPS fusion with respect to the ground truth; top the results of the homography, bottom the results of the 5 point algorithm; the trends are similar

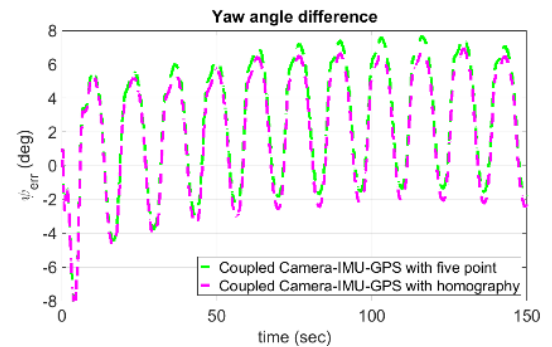


Fig. 15. The yaw error of the Camera-IMU-GPS fusion with respect to the ground truth

V. TESTS ON FLIGHT VIDEOS

The 5 point algorithm and the homography is also tested on real flight videos. In this section our flight

platform with the sensor-processor system for SAA is introduced in nutshells and an example measurement with preliminary results is shown. Although the work with the final integration of the previously introduced result to the system is not ready yet. Here the ground truth is not known precisely, the information from the image processing can only be compared to the coupled IMU-GPS estimator.

A. The aircraft

The airframe used in the flight tests is an upper wing, two engine foam aircraft with 1.85m length, 3.2m wingspan and about 10kg loaded weight.



Fig. 16. The aircraft called Orca, the camera system can be seen on the nose of the fuselage

B. Visual sensor-processor system

An essential part of the system is an image processing development platform, designed specifically to fit in the hull of a compact UAV. It functions as a test framework to evaluate the computational architectures implemented on the reconfigurable fabric. The framework consists of four elements: FPGA development board, with FMC-LPC connector; custom camera modules (up to four), interface card between the FPGA board and the cameras; SSD drive, to store the images.

According to the measurements, the power consumption of the whole system, with four camera modules running is 13.9 W, and it could be reduced even more with a custom FPGA board.

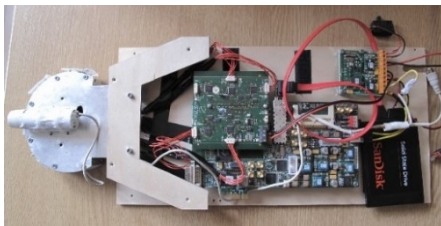


Fig. 17. The assembled development platform

In this case the system was used only to record the images and the sensorial data during the flight. The processing is done offline.

C. Rotation increments from flight videos

In the next example a measurement on real video sequence is shown. The video sequence is recorded with one camera from our camera system. On the consecutive frames 8 feature point pairs are selected by us in order

to minimize the noise caused by wrong matches. The refresh rate of the IMU is 100Hz and of the camera is 56Hz. The synchronisation and the write of IMU and camera data to the SSD are done by the FPGA.

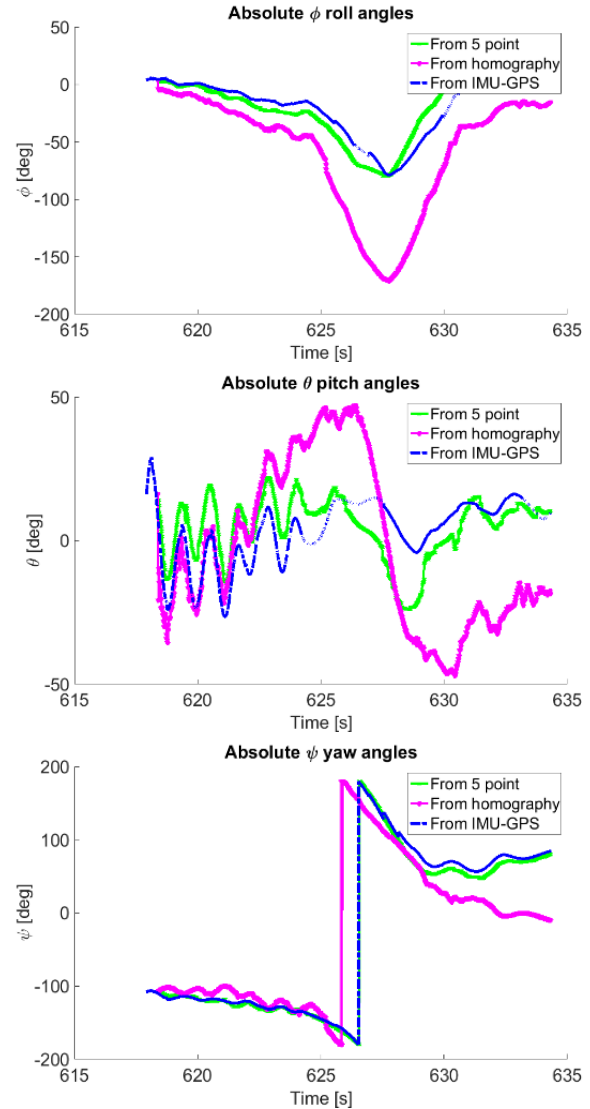


Fig. 18. Euler angles calculated on flight videos

From the matched points Euler angles are calculated with both the 5 point algorithm and the homography (Fig. 18). In this case the homography results seems worse than expected considering the simulations, but still captures the trends. Here the correction based on the velocity and altitude on the homography was not used.

As it was mentioned before there is no ground truth here. The rotation increments, which were calculated with the image processing algorithms are directly applied after each other. That causes the relatively big difference to the IMU-GPS data. The integration of the camera data in the real videos are still ongoing, and parallel to the test the FPGA implementation is started.

VI. CONCLUSIONS

In this paper the error analysis of four image processing algorithms targeting the reconstruction of camera orientation change is introduced. Also the correspondences in between the spatial or temporal resolution change and the performance is shown as well as how random noise affects these algorithms. According to the simulation results the homography algorithm can be used in those situations where the computational power is restricted. If the precision is important than either the 5 point algorithm and the homography can be used keeping in mind the effect of translation and the pixelization.

Results with a coupled Camera-IMU-GPS based attitude estimator algorithm are also introduced. It is shown that the inclusion of image based attitude changes can remove the bias error from pitch estimation of the IMU-GPS algorithm and so improve precision.

Finally the test on flight videos are introduced. These preliminary results show that the five point algorithm is close to the IMU-GPS solution even if the calculated rotation increments are applied after each other. The homography is not as stable as the five point algorithm in this case. Further measurements are needed to try the homography in the real environment.

ACKNOWLEDGMENT

The authors thank Levente Hajder for his consultations on pose estimation algorithms.

REFERENCES

- [1] P. Campoy et al., "Computer Vision Onboard UAVs for Civilian Tasks," *J. Intell. Robot. Syst.*, vol. 54, pp. 105–135, 2008.
- [2] 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.
- [3] T. Hutchings, "Architecting UAV sense & avoid systems," in 2007 Institution of Engineering and Technology Conference on Autonomous Systems, 2007, pp. 1–8.
- [4] B. Karhoff et al., "Eyes in the Domestic Sky: An Assessment of Sense and Avoid Technology for the Army's 'Warrior' Unmanned Aerial Vehicle," in Proc. of 2006 IEEE Systems and Information Engineering Design Symposium, 2006, pp. 36–42.
- [5] D. J. Lee et al., "See and avoidance behaviors for autonomous navigation," SPIE Opics East, Robot. Technol. Archit., vol. 5609, pp. 1–12, 2004.
- [6] T. Zsedrovits et al., "Towards Real-Time Visual and IMU Data Fusion," presented at the AIAA Guidance, Navigation, and Control Conference and Exhibit, 2014.

- [7] 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)*, 2001, p. 3516.
- [8] 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.
- [9] H. Stewénius et al., "Recent developments on direct relative orientation," *ISPRS J. Photogramm. Remote Sens.*, vol. 60, no. 4, pp. 284–294, Jun. 2006.
- [10] 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.
- [11] S. Choi et al., "Performance Evaluation of RANSAC Family," in *Proc. of the British Machine Vision Conference 2009*, 2009, pp. 81.1–81.12.
- [12] T. Zsedrovits et al., "Error Analysis of Algorithms for Camera Rotation Calculation in GPS/IMU/Camera Fusion for UAV Sense and Avoid Systems," Presented at the *International Conference on Unmanned Aircraft Systems*, Orlando, Florida, 2014.
- [13] Y. C. Paw, "Synthesis and validation of flight control for UAV," PhD Dissertation, University of Minnesota, 2009.
- [14] R. W. Beard and T. W. McLain, *Small unmanned aircraft: Theory and practice*. Princeton: Princeton University Press, 2012.
- [15] 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.
- [16] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [17] W. K. Pratt, *Digital Image Processing, PIKS Inside*. John Wiley & Sons Inc, 2001.
- [18] 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.
- [19] R. I. Hartley, "Chirality," *Int. J. Comput. Vis.*, vol. 26, no. 1, pp. 41–61, 1998.
- [20] 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.
- [21] J.-Y. Bouguet, "Complete camera calibration toolbox for matlab," 2004. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.
- [22] H. Stewénius, "Calibrated Fivepoint Solver," 2010. [Online]. Available: <http://www.vis.uky.edu/~stewe/FIVEPOINT/>.
- [23] S. Gleason and D. Gebre-Egziabher, *GNSS applications and methods*. Artech House, 2009.