Resilient Navigation with Multi-Sensor Data Fusion in GNSS-Denied Environment

Felix Wenk[1], Alexander Kläser[1], Stephan Palm[2]
[1] OHB Digital Connect GmbH, Communication and Sensing Solutions,
Manfred-Fuchs-Platz 2-4, 28357 Bremen, GERMANY
[2] Fraunhofer Institute for High Frequency Physics and Radar Techniques FHR,
Fraunhoferstraße 20, 53343 Wachtberg, GERMANY

felix.wenk@ohb.de

ABSTRACT

Usually, an airborne system (UAV or fighter) carrying sensors localizes itself using a combination of GNSS and INS. Nevertheless, particularly in environments near contested territory, GNSS may be jammed or even spoofed, forcing the aircraft to revert to dead reckoning using the INS only. Precise INS, however, are both large and expensive, limiting their applicability for UAVs which are meant primarily to carry 'payload sensors'.

This work focuses on cost-efficient approaches to enable the usage of sensors - such as SAR or multispectral line scanners - on aircrafts, in particular UAVs, which require accurate localization or at least ego-motion estimates to process the sensor data. This also is a prerequisite for resilient navigation in GNSS-denied environments.

To enable the use of sensors requiring localization, we are developing a system to localize a UAV or estimate its ego-motion in GNSS-denied environments from easily available commercial-of-the-shelf components, whose sensor data is fused to provide accurate UAV pose estimates. The idea is to estimate the ego-motion above ground by observing the ground passing by as the aircraft flies. To this end, we augmented the wing pod of an ultralight aircraft, which already carried a GNSS+INS system as well as a SAR payload sensor, with an experiment kit containing a MEMS-IMU as well as a setup of different cameras. This setup, which essentially constitutes a visual-inertial-odometry setup, includes a SWIR camera to achieve robustness against bad weather conditions, e.g., haze.

We developed a procedure to calibrate both in- and external camera parameters which achieves sub-pixel reprojection accuracy. The ego-motion estimation problem itself is split in two parts: Finding points of interest in the images and tracking them across images to establish point correspondences and computing the ego-motion from corresponding points. For the former, we look for points with a strong "cornerness" and find their corresponding points in other images by computing the optical flow between pairs of images. We then compute the ego-motion by solving for the aircraft poses and velocities at the image acquisition time points, which maximize the likelihood of the point measurements. Since the cameras' frame rates are much lower than the measurement frequency of the MEMS-IMU, we coalesce all IMU measurements into pose-and-velocity-deltas between two image acquisitions, which in turn are used as measurements in the maximum likelihood estimator.

To put our system to a test, we conducted a flight campaign, where we recorded data from our sensors over various types of ground, as well as the reference trajectory measured by the GNSS+INS system. The relative consecutive poses as estimated by our system deviate only centimeters from the reference trajectory. The MEMS-IMU data substantially increases robustness of our ego-motion estimate, and as a bonus, even using lower-resolution SWIR images, SAR image formation was satisfactory under benign flight patterns.

1 INTRODUCTION

Measurements of multi-spectral and multi-static sensors are important data sources to establish situational awareness. Typically, such sensors are carried by flying platforms, for instance, depending on the specific application, satellites, aircrafts or unmanned aerial vehicles (UAVs). Sensor carrying platforms need to localize themselves, both to navigate to and from the scenes to be observed and to make sense of the measurements recorded by the sensors. For instance, to form an image using the data recorded by a synthetic aperture radar (SAR), the ego-motion of the sensor and thus the platform at measurement time must be known. Aircrafts and UAVs usually localize themselves using a combination of an inertial navigation system (INS) and satellite positioning. A typical setup of an airplane carrying payload sensor as well as auxiliary navigation hardware in a pod below its left wing is displayed in Figure 1-1. The inside of the pod is shown in Figure 1-2 on the left, where you see the SAR pointing downward as well as the navigation hardware.

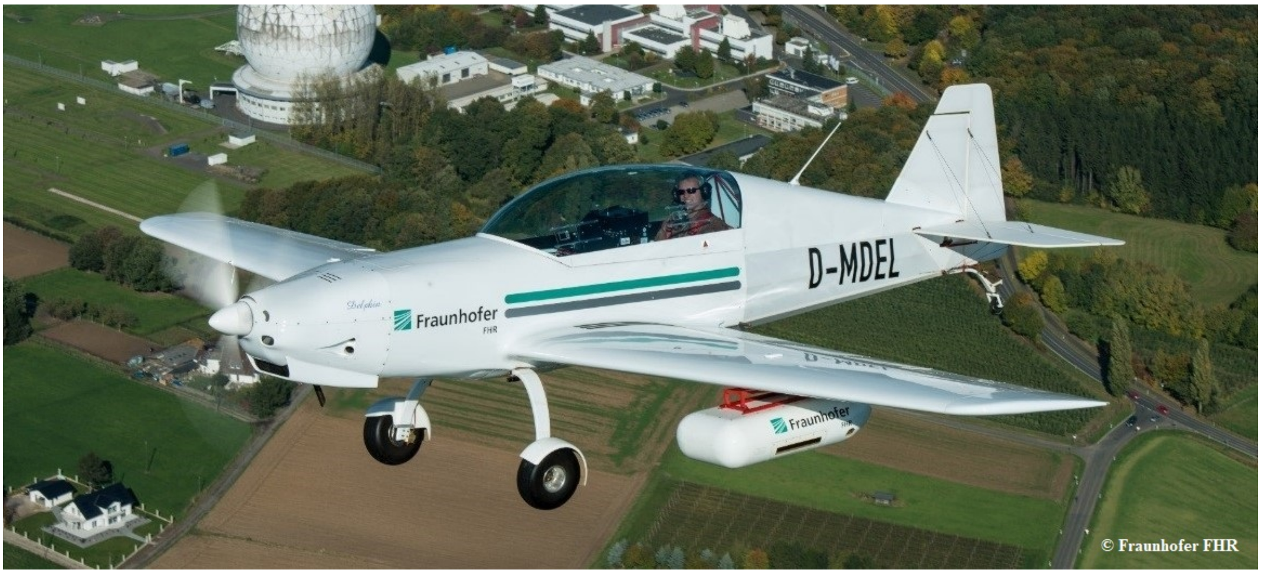

Figure 1-1: Plane carrying the payload sensor and auxiliary navigation electronics in a pod below its left wing. (© Fraunhofer FHR)

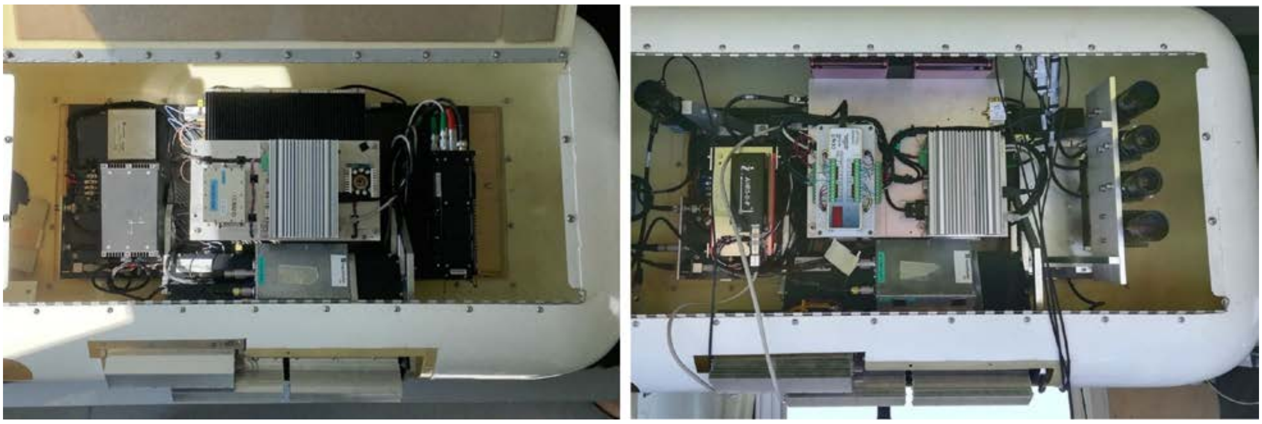

Figure 1-2: Peek inside the pod mounted below the airplane's wing. Left: Baseline with payload sensor (SAR) and auxiliary INS+GNSS. Right: Pod augmented with optical navigation hardware.

Satellite positioning, however, may be jammed or even spoofed, particularly over contested territory, forcing the airplane to revert to dead-reckoning using its inertial navigation system only. Depending on the precision of the INS, navigation without satellite positioning becomes impossible quickly, and the payload sensor's measurements are rendered useless, because without an accurate ego-motion estimate, which is the succession of the plane's poses, the measurements cannot be processed to form images. Additionally, precise INS are also heavy, large, and expensive, and thus unattractive to be used on, for instance, a smaller UAV.

The primary objective of this study was to demonstrate a different, cost-effective approach for estimating the ego-motion, such that the payload sensor's data, in this case SAR, can still be used even if satellite positioning is not available. For this, we substituted cameras for satellite positioning, i.e., by observing the ground passing by below as the plane flies, and to estimate the plane's ego-motion from these observations. This idea may later be extended to use geo-referenced images acquired prior to the flight, for instance by satellites, to achieve absolute positioning instead of visual-inertial odometry only.

This paper is organized as follows. Section 2 explores the experiment kit in more detail in subsection 2.1 before it describes our approach to calibrate the optical ego-motion estimation hardware in subsection 2.2. Section 3 covers the procedure we implemented to obtain an accurate ego-motion estimate from imagery of the ground. Section 4 covers the flight campaign as well as quantitative results. Section 4 also has a comparison of SAR images processed using data from the original navigation system with images using the ego-motion estimates derived from our visual-inertial odometry approach. At the end, section 5 concludes the paper along with an outlook on work in the near future.

2 EXPERIMENT KIT AND CALIBRATION

2.1 Experiment Kit

The experiment kit mounted in the wing pod is displayed in right half of Figure 1-2. In addition to the payload sensor and the original INS, it includes a MEMS IMU (Inertial Labs AHRS-II-P), four panchromatic cameras (The Imaging Source DMK 38uX304, 4096x3000 pixel), and a SWIR camera (Xenics BOBCAT, 640x512 pixel). Two of the panchromatic cameras each had a 75 mm lens and were about 60 cm apart, forming a small stereo pair. The other two panchromatic cameras had 35 mm lenses; one was used with only a quarter of the resolution to enable image acquisition at 30 Hz. The other panchromatic cameras acquired images at 10 Hz. At the target flight altitude of 300-400 m, each camera of the stereo pair had a ground sampling distance of about 1.5 cm, the other panchromatic cameras of 3 cm and 6 cm, respectively. The SWIR camera used a 75-mm-lens achieving 8 cm GSD at 300 m altitude with a frame rate of 20 Hz. All cameras were triggered using the PPS signal of the GPS receiver of the original navigation system.

2.2 Calibration

The camera calibration is divided into two parts: the internal and the external camera parameters. The internal parameters used in our system are the focal length, the position of the camera centre, as well as the radial distortion. The external parameters of each camera are its pose, which is its orientation and position, relative to the origin of the pod. The usual approach to determine such parameters is to set up a scene with known properties. Given the ideal values of the calibration parameters, it is then possible to compute the properties of the scene. The process of calibration is then to compute the scene properties using the best guess of the parameters, then to take actual images of the scene and optimize the parameters until the computed and observed properties coincide as well as possible.

The calibration scene for our system uses planar patterns [2] as displayed in Figure 2-1. The essential components of the calibration scene are calibration plates; Figure 2-1 shows four of them. Each calibration plate has in each of its corner a marker encoding the unique identifier of the plate.

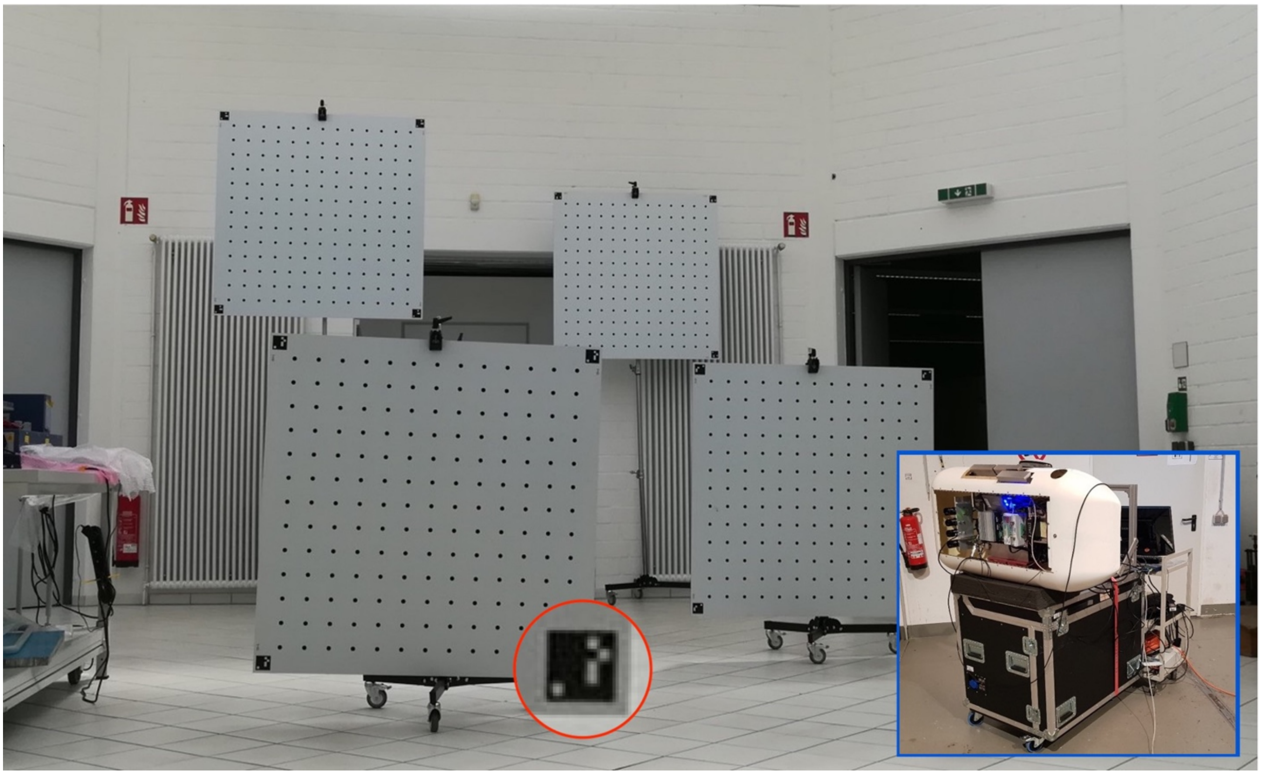

Figure 2-1: Calibration scene with calibration plates with a known grid of circles. Each plate has markers in its corners – example magnified in red circle – encoding the corresponding plate identifier. To take actual images, the pod is moved on a camera in front of the calibration plates.

Since the coordinates on the plates are known with respect to the corresponding plate's upper-left corner, we identify each scene point corresponding to a printed circle by ${}^{A}p_{Ak}$, which represents the coordinate in the 3-dimensional scene of the k'th point on plate A relative to the top-left corner of plate A. Using the pose – the external parameters – of the j'th camera relative to the pod, ${}^{Pod}T_{Cj}$, the camera matrix and the distortion coefficient – the internal camera parameters – of the j'th camera, K_j and k_j, we can compute the point ${}^{Ij}p_{Ak}$, where the k'th point on plate A appears in the picture I_j taken by camera C_j while the pod and plate A are at poses ${}^{W}T_{Pod}$ and ${}^{W}T_{A}$, respectively, using eq. (1).

$$ {}^{I_j}p_{A_k} = K_j \operatorname{distort}\left(k_j,\, {}^{\mathrm{Pod}}T_{C_j}^{-1}\, {}^{W}T_{\mathrm{Pod}}^{-1}\, {}^{W}T_A\, {}^{A}p_{A_k}\right) \qquad (1) $$

In eq. (1), we use homogeneous coordinates, e.g. a point in 3-dimensional space is represented by 4-element vectors, p = w [x y z 1]^{T}, with non-zero w and coordinate transforms are packed into 4x4 matrices. The distort function in eq. (2) projects a point on an image plane applying radial distortion using OpenCV's distortion model [1] with only the first coefficient.

$$ \operatorname{distort}(p,k) = \begin{bmatrix} sx/z \\ sy/z \\ 1 \end{bmatrix} \quad \text{with} \quad s = 1 + k\,((x/z)^2 + (y/z) \quad \text{and} \quad p = w\,[x\;y\;z\;1]^T \qquad (2) $$

To do the computation, the poses of the pod as well as the plates need to be known at the time when the respective picture is taken. Instead of measuring these poses somehow, we add them to the calibration problem. This makes the calibration computationally harder, however, since pictures of the calibration scene are taken while moving the pod on a cart in front of the calibration plates, which are not moved, this adds only (#plates + N) pose variables to the problem, where N is the number of pod poses from where pictures are taken. A typical picture of a calibration scene is shown in Figure 2-2. Since the poses of the plates do not have to be known in advance, we are essentially free to place the plates anywhere we want, however, we make sure to put the plates such they are not all co-planar and that the calibration images are taken from different orientations to avoid degeneracies. The differences between positions of the computed images of the calibration points and the detected images after the camera calibration completed is written next to the points. The differences are well below one pixel, usually below 0.5 pixel.

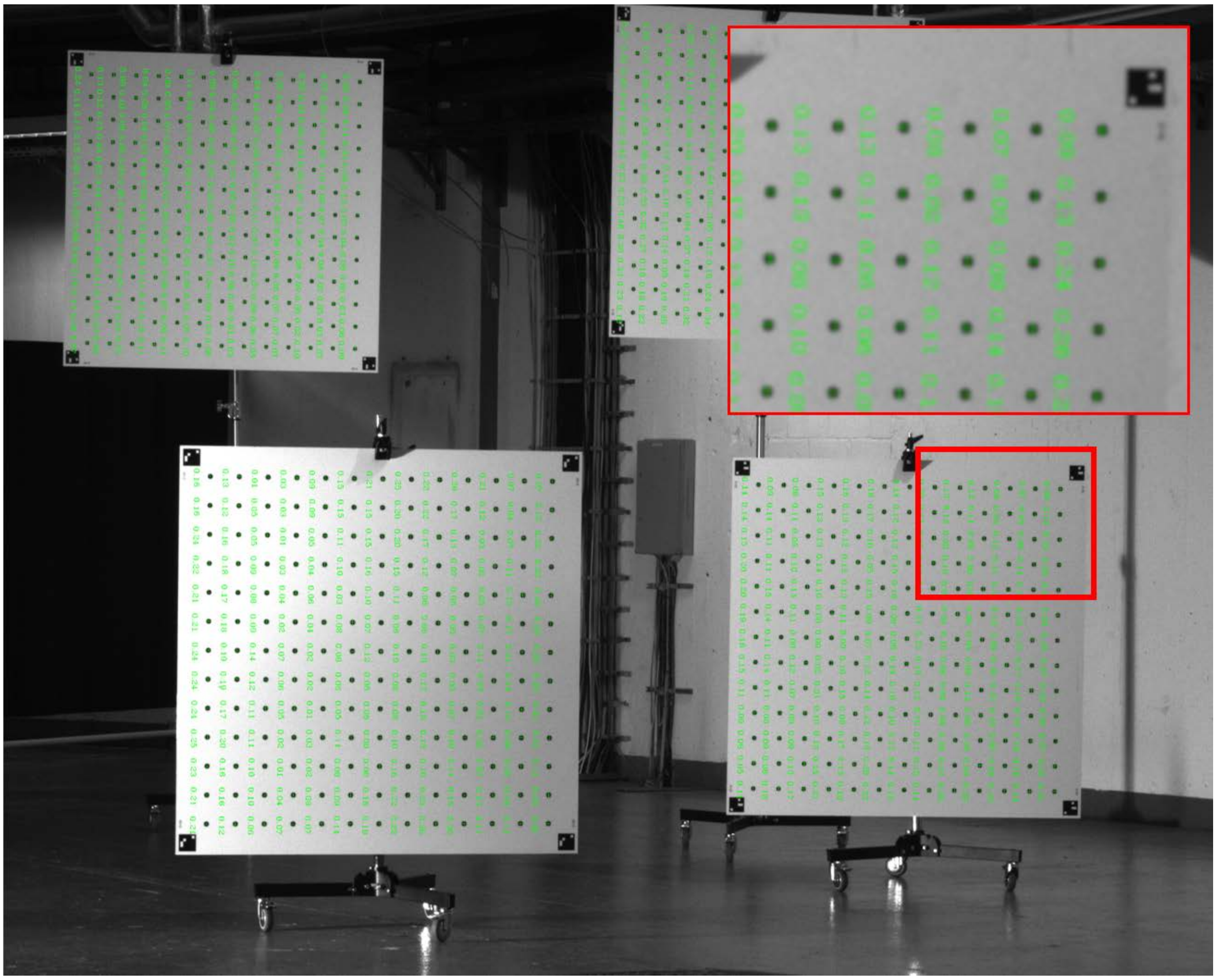

Figure 2-2: Picture taken during calibration. Difference between detected and computed images of the calibration points are written next to each point in green.

3 EGO-MOTION ESTIMATION

With the calibration in place, the cameras can now be used to determine the ego-motion of the platform carrying the sensors. The task is similar to the camera calibration task, i.e., the measurement model will almost be like eq. (1), except, of course, that there will be no calibration plates with points on them. Instead the cameras will take pictures of the scene below on ground. So we interpret the scene as a very dense set of points and “taking a picture with a camera” as computing the image under the camera’s projection function of each 3D scene point in the field of view of the camera. The first task of determining the ego-motion is to find out which images of points in the pictures taken by the cameras from different platform poses are images of the same 3D-point in the scene.

3.1 Corresponding Images of Scene Points

To establish the correspondences of images of 3D scene points across two different pictures, we compute the optical flow between those pictures. Essentially, the information contained in the optical flow between two pictures is where an image of a 3D scene point moved from one picture to the other. So with the coordinates of a point-image in the first picture and the optical flow between two pictures, the coordinates of the point-image of the same scene point in the second picture can be computed. An example of these displacements is drawn in Figure 3-1.

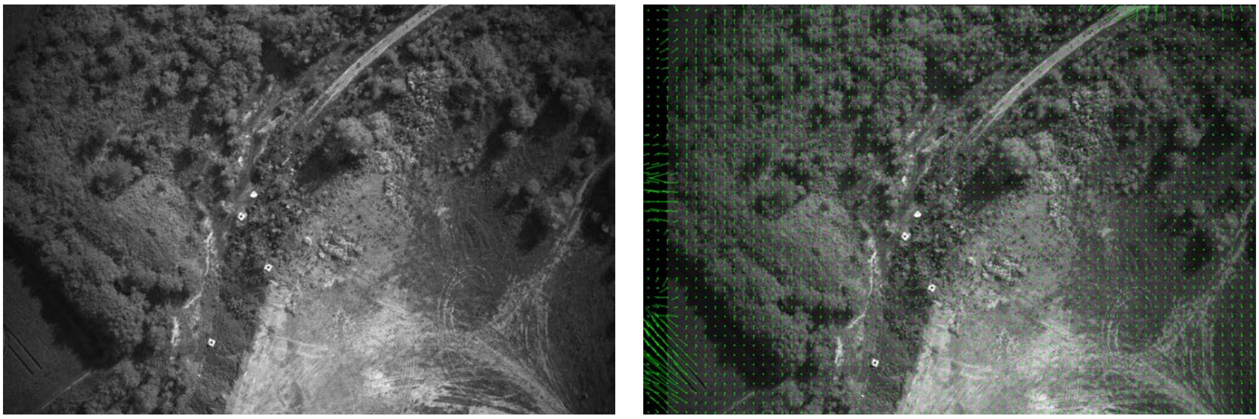

Figure 3-1: Optical flow between two pictures. Arrows over the right picture indicate where images of the same scene points are estimated to appear in the left image.

The estimation of the optical flow can fail for a number of reasons. The two pictures may be taken by different cameras from different poses, so the sets of scene points in the respective fields of view are different; for some point-images in the first picture, there simply are now corresponding images in the other picture. This happens near the boundaries of the pictures. Images of neighbouring scene points may also look alike such that it is impossible to tell one from the other. This happens in regions with very little structure, for instance in regions picturing a flat surface, such as a lake. To avoid picking spurious correspondences from these failure-cases for the ego-motion estimator, the boundaries of the pictures are ignored and only point-images whose immediate neighbourhood has a sufficiently high “cornerness” are picked. Additionally, a difference measure of the immediate neighbourhoods of corresponding point-images is computed and used to remove false correspondences.

Using the optical flow, we compute point correspondences for both pictures taken at the same time by different cameras as well as pictures taken by the same camera at consecutive image acquisition times. The former enables the triangulation of the corresponding scene points. The latter enables the computation of consecutive plane poses using the triangulated scene points, completing the data necessary to compute the ego-motion of the plane.

3.2 Ego-Motion: Successive Poses

From the image points and their correspondences, we estimate the successive poses, i.e. the ego-motion, of the plane. To do so, we remove the poses of the calibration plates from eq. (1), to obtain a measurement model in eq. (3) for arbitrary scene points, each of which is assigned an identifier ‘id’ according to its scene point correspondence. We also add the index k both to the image $I_{j,k}$ and the pose ${}^{W}T_{Pod,k}$ to denote the image of camera j taken at time t_k and the pose of the pod at the same time t_k, respectively.

$$ {}^{I_{j,k}}p_{\mathrm{id}} = K_j \,\mathrm{distort}\Big(k_j, {}^{\mathrm{Pod}}T_{C_j}^{-1}\, {}^{W}T_{\mathrm{Pod},k}^{-1}\, {}^{W}p_{\mathrm{id}}\Big) \tag{3} $$

Denoting the position where the image of the point with the corresponding id has been detected in $I_{j,k}$ by ${}^{j,k}z_{id}$, the (squared) projection error of an estimate of the pose ${}^{W}T_{Pod,k}$ with respect to that point is:

$$ Z_{(k)}^{(\mathrm{id})} = \sum_{j=0}^{m-1} \left\| {}^{j,k}z_{\mathrm{id}} - {}^{I_{j,k}}p_{\mathrm{id}} \right\|_{\Sigma_I}^{2} \mathbb{I}\{\mathrm{visible}(j,k,\mathrm{id})\} \tag{4} $$

In eq. (4) I{visible(j,k,id)} is 1 if the point with the identifier “id” is visible in the picture taken by camera j from pose k, otherwise it is 0. Σ_I is the covariance of the image coordinates to normalize the norm of the error, i.e. $||x||^2_{\Sigma}=x^T\Sigma^{-1}x$. Since the camera setup for our experiment includes a stereo pair, eq. (4) alone is theoretically sufficient to estimate the pod’s poses. From the images of a point taken at the same time by two different cameras we can triangulate the 3d-coordinates of the point in the scene relative to the cameras; from images of a few triangulated points taken at different times, e.g. t_k and t_{k+1}, we could then estimate the relative pod pose ${}^{Pod,k}T_{Pod,k+1}$ between two points in time. Defining the coordinate system of the scene, W, to be aligned with the first pose of the pod, i.e. W=Pod,0, this would theoretically yield the ego-motion we are looking for. Due to even small measurement and calibration errors and a very small distance between the cameras of the stereo pair, such an ego-motion estimate would be very inaccurate. This can be improved by exploiting that successive plane (and thus pod) poses are clearly correlated.

To make this correlation explicit, we augment the state of the plane at some time t_k to not only include it’s pose, ${}^{W}T_{Pod,k}$, but also its velocity, ${}^{W}v_k$. To correlate successive poses, we assume that the plane’s orientation stays unchanged except for a random, Gaussian-distributed angular velocity with mean zero, and that the plane’s velocity remains unchanged except for a random, Gaussian-distributed zero-mean acceleration. For the position and velocity of the plane at two consecutive points in time, t_k and t_{k+1}, this means that $D^{(k)}$ in eq. (5) is zero on average with variance Σ_v, where $\Delta t_k=t_{k+1}-t_k$.

$$ D^{(k)} = \left\| \begin{bmatrix} p_{k+1} - p_k - v_k \Delta t_k \\ v_{k+1} - v_k \end{bmatrix} \right\|_{\Sigma_v}^{2} \tag{5} $$

Σ_v is obtained by noticing that the deviation of velocity and position from a linear motion is due to integrating a Gaussian random acceleration with zero mean and some variance over Δt_k. In eq. (6) we treat each of the three spatial dimensions separately and expand the 2x2-covariance of $V(\Delta t_k)$ to the 6x6-matrix Σ_v by taking the Kronecker product with the 3x3-identity, i.e. $\otimes I_3$.

$$ \Sigma_v = \mathrm{E}\left[V(\Delta t_k)V(\Delta t_k)^T\right] \otimes \mathcal{I}_3, \quad V(\Delta t_k) = \int_0^{\Delta t_k} \begin{bmatrix} \Delta t_k - \tau \\ 1 \end{bmatrix} a(\tau)\, d\tau \tag{6} $$

Instead of using random variables with zero mean to model the motion between subsequent poses, the measurements of the MEMS IMU may be used. The IMU measures changes of the state of the plane: the angular velocity – the rate of change of orientation – and the linear acceleration – the rate of change of velocity and by integration position. We represent such a state change between t_{k-1} and t_k as ${}^{k-1}\Delta_k = \left[{}^{k-1}R_k^T \quad {}^{k-1}p_k^T \quad {}^{k-1}v_k^T\right]^T$, where R, p, and v are the change in orientation, position and velocity, respectively. With slight abuse of notation, R is a rotation matrix rotating vectors from the pose at t_k to the pose of t_{k-1}. Such a state change is built incrementally from IMU measurements, starting from the “change” between two identical states, which is just ${}^{k}\Delta_k$ = [I_3 0 0], using the model function g:

$$ {}^{i}\Delta_j = g\left({}^{i}\Delta_{j-1}, \omega_j, a_j\right) = \begin{bmatrix} {}^{i}R_j \\ {}^{i}p_j \\ {}^{i}v_j \end{bmatrix} = \begin{bmatrix} {}^{i}R_{j-1}\exp\left[\omega_j \Delta t\right]_\times \\ {}^{i}p_{j-1} + {}^{i}v_{j-1}\Delta t + \frac{1}{2}\,{}^{i}R_{j-1} a_j \Delta t^2 \\ {}^{i}v_{j-1} + {}^{i}R_{j-1} a_j \Delta t \end{bmatrix} \tag{7} $$

In eq. (7), a_j and ω_j the linear acceleration and angular velocity from the MEMS IMU, Δt the duration between two IMU samples. The IMU samples accumulated using eq. (7) between two consecutive states are approximately the difference between those states except for the lack of gravity, which is:

$$ \begin{bmatrix} {}^{k-1}R_k \\ {}^{k-1}p_k \\ {}^{k-1}v_k \end{bmatrix} = \begin{bmatrix} {}^{W}R_{k-1}^{-1}\,{}^{W}R_k \\ {}^{W}R_{k-1}^{-1}\left({}^{W}p_k - {}^{W}p_{k-1} - {}^{W}v_{k-1}(t_k - t_{k-1})\right) \\ {}^{W}R_{k-1}^{-1}\left({}^{W}v_k - {}^{W}v_{k-1}\right) \end{bmatrix} \tag{8} $$

In eq. (8) ${}^{W}R_k$, ${}^{W}p_k$, and ${}^{W}v_k$ are the state, i.e. orientation, position, and velocity at time t_k. Since the accelerometer of an IMU cannot measure gravity, e.g. an accelerometer in free fall measures zero acceleration although it is obviously accelerating with the gravitational acceleration, there is a “minus gravity” offset between the actual acceleration and the accelerometer measurement. This offset manifests itself in the difference between the actual state difference of eq. (8) and the integration of the IMU measurements in eq. (7):

$$ {}^{k-1}E_k = \begin{bmatrix} \log\left({}^{i}R_j^{-1}\,{}^{k-1}R_k\right) \\ {}^{k-1}p_k - {}^{i}p_j \\ {}^{k-1}v_k - {}^{i}v_j \end{bmatrix} = \begin{bmatrix} 0 \\ {}^{k-1}g(t_k - t_{k-1})^2/2 \\ {}^{k-1}g(t_k - t_{k-1}) \end{bmatrix} = {}^{k-1}G_k \tag{9} $$

In eq. (9), ${}^{k-1}g$ is the gravitational acceleration in the local reference frame at time t_{k-1}. ${}^{i}R_{j,}$ ${}^{i}p_{j,}$ ${}^{i}v_j$ are the orientation, position and velocity integrated from the measurements between times t_{k-1} and t_k. The middle equality only holds approximately, most importantly due to measurement noise. The corresponding normalized measurement error for the integrated IMU measurements between t_k and t_{k-1} is thus

$$ I_{(k)}^{(k-1)} = \left[{}^{k-1}E_k - {}^{k-1}G_k\right]_{\Sigma_\Delta} \tag{10} $$

The covariance Σ_Δ to normalize the measurement error is obtained by initializing it to zero and propagating it and the covariances of the angular velocity and acceleration measurements through the measurement model function, g, of eq. (7), by linearizing g at the respective values for Δ, ω and a.

Depending on how successive poses are to be linked – using the zero-mean-white-noise model or integrated IMU measurements – there are two different functions quantifying the error of an ego-motion estimate. Minimizing either of those provides the best estimate given the corresponding modelling assumptions and sensor data, respectively.

$$ \underset{{}^{W}R_k, {}^{W}p_k, {}^{W}v_k}{\arg\min} \sum_{k=0}^{N-1}\sum_{\mathrm{id}=0}^{M-1} Z_{(k)}^{(\mathrm{id})} + \sum_{k=0}^{N-2} D^{(k)} \quad \text{or} \quad \underset{{}^{W}R_k, {}^{W}p_k, {}^{W}v_k}{\arg\min} \sum_{k=0}^{N-1}\sum_{\mathrm{id}=0}^{M-1} Z_{(k)}^{(\mathrm{id})} + \sum_{k=1}^{N-1} I_{(k)}^{(k-1)} \tag{11} $$

4 FLIGHT CAMPAIGN AND RESULTS

To determine the ego-motion of plane carrying the experiment kit (see sect. 2.1 and the right half of Figure 1-2), we tested optimizing both the left and right cost functions of eqs. (11), e.g. without and with measurements from the MEMS IMU. We did that using data acquired during a flight campaign conducted over three days, during which the experiment kit was flown mounted on the ultralight plane displayed in Figure 1-1. The flight tracks passed different types of ground, e.g. agricultural fields as well as suburban areas. The acquired data includes, in addition to the camera images, angular velocity and acceleration measurements taken by the MEMS IMU as well as trajectory information from the original INS+GNSS solution, which we use as the reference to compare our ego-motion estimators to. The tracks themselves were straight-ahead flight with one exception, where the plane flew a succession of left and right turns to produce a trajectory which is potentially more challenging to estimate than a straight one.

One of the primary motivations for this work was to enable the usage of measurement of a payload sensor without GNSS, for instance the formation of images of measurements of a synthetic aperture radar (SAR). So in addition to the navigation reference and experiment hardware, the plane carried a SAR which was used during the flight campaign to record data while flying the straight-ahead tracks.

4.1 Robustness

As expected, the ego-motion estimator using MEMS IMU data turned out to be more robust than the estimator using the zero-mean-white-noise assumption for the acceleration and angular velocity. One example of the robustness difference is the translation sequence plotted in Figure 4-1. For the first half of the sequence, the position estimates (blue) of both the estimator not using IMU data (left) and the estimator using IMU data (right) yield plausible results. Both estimators track the reference fairly well until the estimate using only the zero-mean-white-noise assumption for the motion between two consecutive positions collapses. This happens when the estimator introduces a relatively large error, for instance due to wrong point-image to world-point correspondences, from which it does not recover. The estimator using IMU data to link consecutive poses is more robust against such errors, since, for instance, sudden orientation errors would be grossly incompatible with the angular velocity measurements.

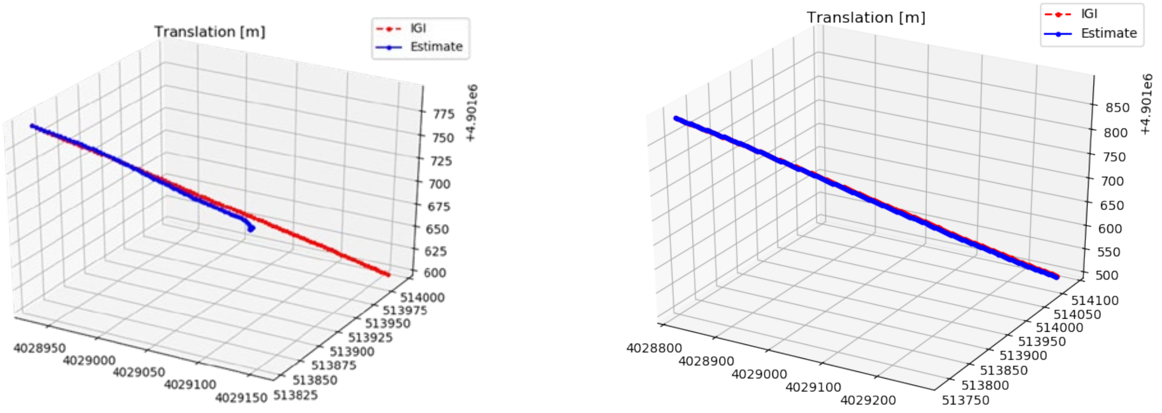

Figure 4-1: Translation component (blue) of ego-motion estimates obtained from the same data. Left: From camera images without IMU data using white-noise-acceleration. Right: From camera images with MEMS-IMU data. The translation from the IGI IMU+GNSS is drawn red for reference.

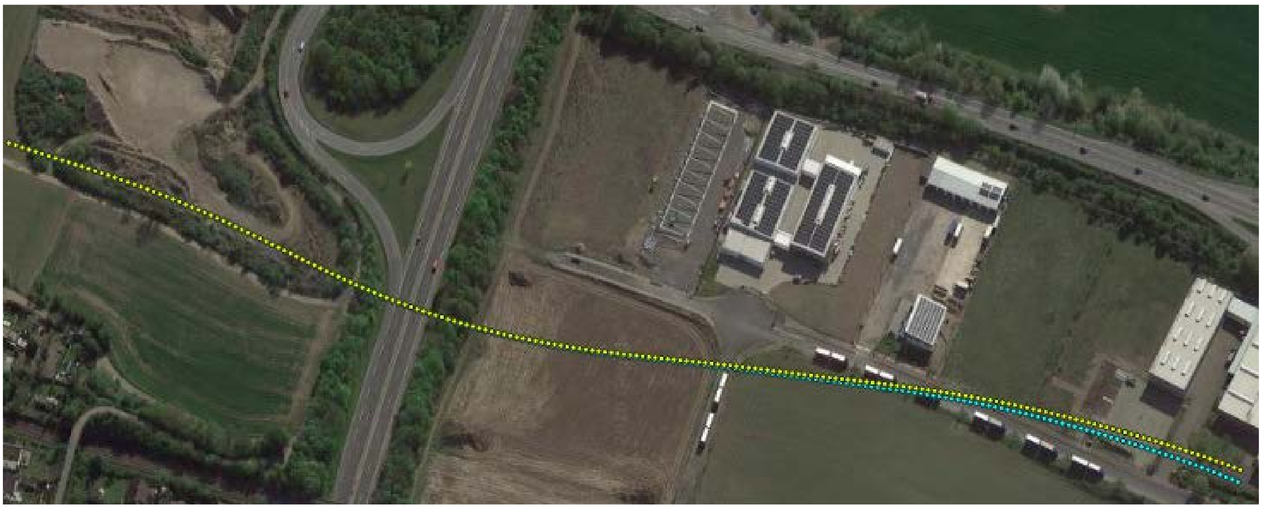

Figure 4-2: Translation components of the plane's trajectory of 20 seconds of flight along the track including turns plotted on a corresponding Google Satellite image. Blue: Positions taken from the reference data. Yellow: Positions of the ego-motion estimate obtained using camera imagery and MEMS IMU measurements.

4.2 Ego-Motion Accuracy

To judge the accuracy of the estimator, we compared ego-motion estimates against the output of the IGI IMU+GNSS, which we used as reference data. Figure 4-2 shows the position estimate of 20 seconds worth of flight drawn on to a Google Satellite image of the corresponding region. The estimate using cameras of our experiment kit and MEMS IMU data, plotted in yellow, follows the reference obtained by the IGI INS+GNSS fairly accurately but, since the sensor data provide no absolute position information, can be seen to slowly drift away from the reference positions at the end of the plot on the bottom right.

Figure 4-2 shows results qualitatively. To evaluate the estimator more quantitatively, for a short period of flight from the flight campaign, both the reference data (blue) as well as the corresponding ego-motion estimate (green) obtained from MEMS IMU measurements and SWIR camera images taken at 20 Hz are plotted in Figure 4-3.

The top row of Figure 4-3 shows the position estimate, the middle row the velocity estimate and the bottom row the estimate of the orientation. The latter agrees very well to the reference data taken by the IGI system, as do the latitude and longitude estimates. There are, however, periods during which the estimator introduces errors to the altitude, as can be seen in the top-left plot of Figure 4-3, from which the estimator does not recover.

To process sensor data of payload sensors, e.g. to form images of data of a SAR carried by the plane, the absolute position of the plane is not that important, however, the accuracy of the relative motion of the plane during the SAR data acquisition is crucial for the resolution of the resulting SAR images. The relative orientations and positions, i.e., their change between two image acquisition time points, recorded during the more challenging track including turns, which is also shown in Figure 4-2, are plotted in Figure 4-4. The plot of the reference data is drawn using a dashed line. As we have already seen in Figure 4-3, the orientation is very accurate; the plots for the reference data are only visible for the translation components in the bottom-left part of Figure 4-4. The differences between the reference data and the ego-motion estimate are plotted in the right half of Figure 4-4: The difference of the rotation is typically below 0.01 degrees, while the

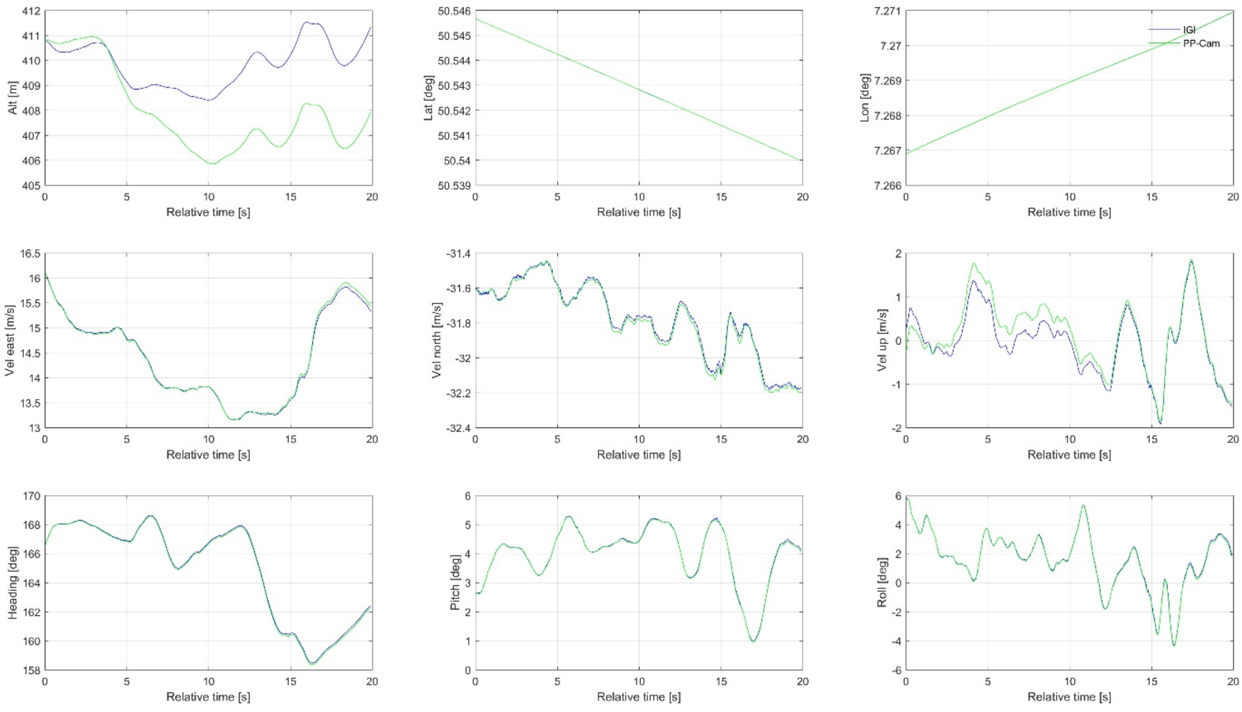

Figure 4-3: Excerpt of trajectory data of the flight campaign. Green: Estimate using MEMS IMU data and images of the SWIR camera captured at 20 Hz. Blue: Reference data of the IGI INS+GNSS.

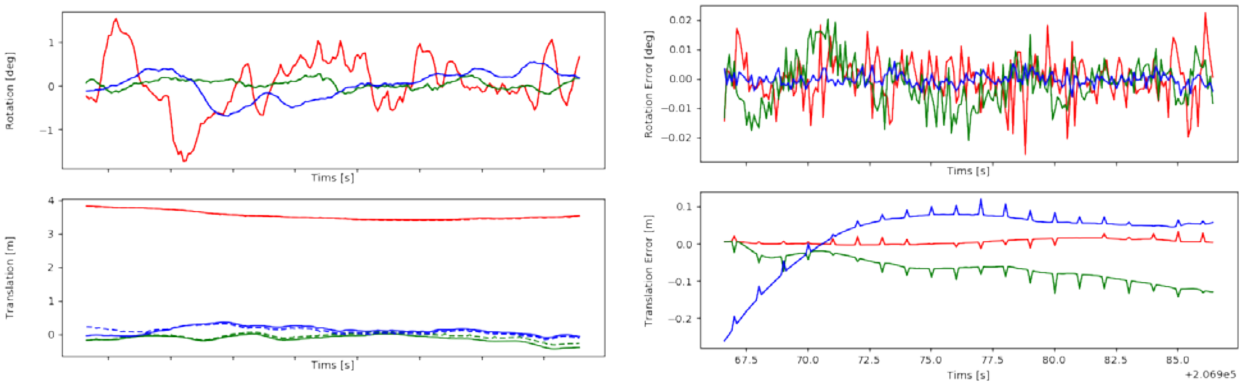

Figure 4-4: Left: Relative orientation and position around and along, respectively, the plane’s x (red), y (green), and z (blue) axes, using the INS+GNSS reference (dashed) and ego-motion estimate from MEMS IMU measurements and SWIR images at 10 Hz (solid). Right: Difference between the reference and the ego-motion estimate.

translation difference is mostly below 10 cm. The translation error also includes peculiar, regularly spaced spikes. These do not come from the data acquired by our ego-motion estimator. Instead, those sudden changes occur in the position data of the reference system and we suspect that they appear when the GNSS reference system integrates an absolute position measurement.

During the flight campaign, we evaluated the relative poses of four different tracks, named A02, B01, D01, and D03, with different camera configurations, including using one SWIR camera and VIS cameras including the stereo pair. The results are summarized in Table 4-1. As expected due to the lower resolution and the lack of a stereo pair, the translation results are worse for the SWIR camera (about 8 cm) compared to the VIS cameras (about 3 cm to 5 cm). Interestingly, the rotation errors appear to be better for the SWIR camera (about 0.01 degrees) compared to the results achieved by the VIS cameras (about 0.015 to 0.025 degrees). Our current guess as to why that is the case is that the estimator weighs the rotation information from the point correspondences less for the SWIR than for the VIS cameras. For the SWIR camera there are less correspondences and each pixel also covers a larger area on ground, while two of the three rotational degrees of freedom are already observable for absolute orientations using the gyro-/accelerometer pair and, obviously, for relative orientations by the gyrometer alone.

Table 4-1: Comparison of VIS and SWIR with 10 Hz images each over various tracks. Numbers are the root-mean-squared differences with respect to the reference data of the IGI INS+GNSS.

RMS \ Track	A02	B01 (21 s)	D01	D03
Rotation [deg] (SWIR)	0.013	0.008	0.010	0.013
Translation [m] (SWIR)	0.078	0.083	0.084	0.076
Rotation [deg] (VIS+Stereo)	0.014	0.027	0.020	0.026
Translation [m] (VIS+Stereo)	0.029	0.033	0.049	0.051

4.3 SAR Image Processing

The quality of the derived navigational data shall be further evaluated by the ability to process high resolution SAR images of the scene. The airborne FMCW SAR sensor used in the flight campaign operates at Ka-band at 35 GHz with a bandwidth of 1.5 GHz that results in a theoretical resolution of 10 cm. The raw data was processed with a time-domain backprojection approach on a georeferenced flat-plane earth model, see [3] and [4], with a pixel spacing of 3 cm. For comparison, the synchronized radar raw data was each processed using the calculated track data from the fiber optic gyroscope and the proposed ego-motion estimation by camera and SWIR data.

Results of the campaign are presented in Figure 4-5 that show an area in the vicinity of the German town Ochtendung and Figure 4-6 showing an area close to Miesenheim. The SAR images demonstrate that Ka-band data can be focused with high resolution by the proposed methods under benign conditions in linear flight mode. Regions with more challenging flight patterns like high drift or roll angles show slight defocusing effects compared to very high-quality IGI INS+GNSS reference data derived from laser gyros, see Figure 4-5 (b)-(d). Yet, objects in these areas are still recognizable and the deterioration is usually restricted to small areas.

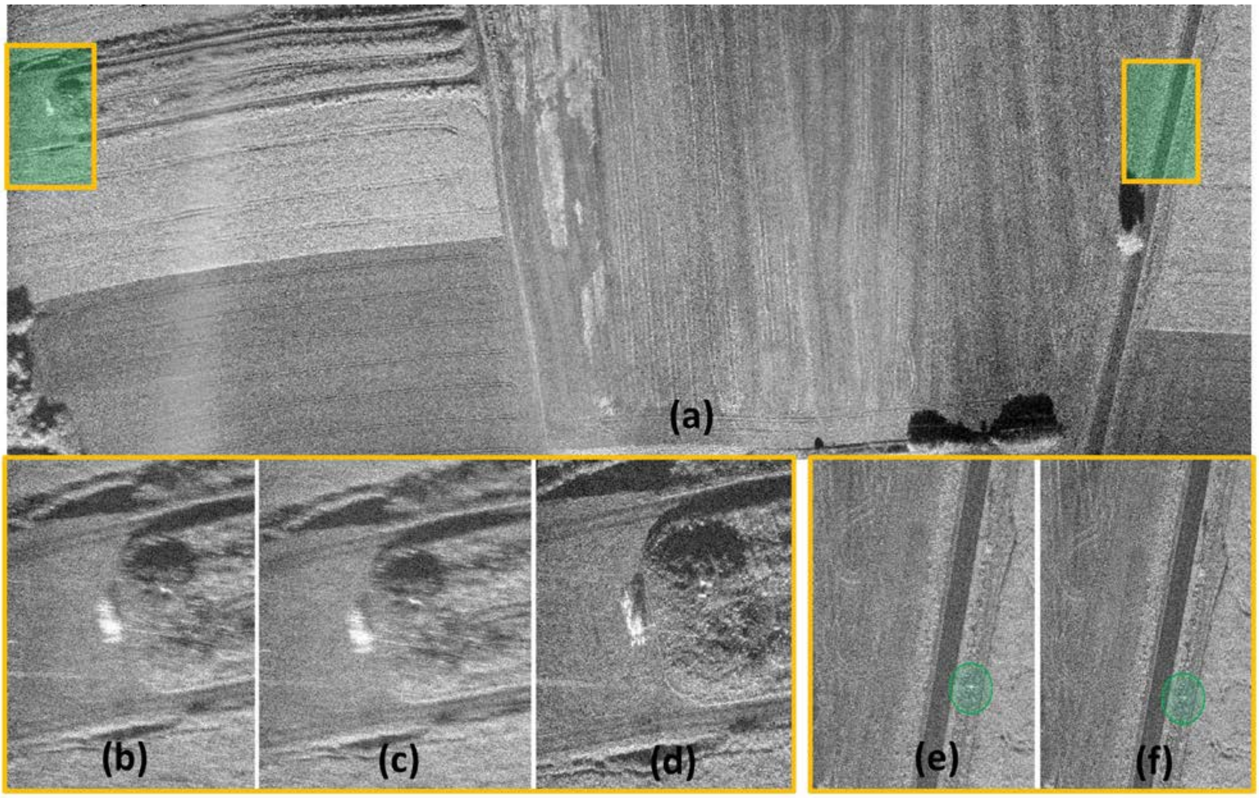

Figure 4-5: SAR image result by post-processing the radar raw data and the navigational data derived with different GPS/INS sensors. (a) Full scene processed with proposed camera ego motion estimation. (b) Zoom on smaller region with more challenging flight conditions. SAR image focused with camera tracking. (c) SAR image focused with SWIR tracking (d) SAR image focused with reference data (e) Zoom on field with corner reflector and road. Processed with camera tracking. (f) Processed with reference data.

The experiments demonstrate that objects such as roads, different field vegetation, buildings, parked vehicles, tracks in the field and even floodlight masts (see Figure 4-6 (b)-(c)) can be focused with only minor deterioration. The obtained resolution was further evaluated by corner reflectors positioned in the scene that serve as point targets. Analysis of the achieved ground resolution is presented in Table 4-2, which shows the mean point target response and resolution of 3 corner reflectors in the scenes. The maximum amplitude of the focused point target is 4-6 dB below the laser gyro standard while the resolution in azimuth is only slightly deteriorated from 10.9 cm to 14.5 cm or 17.8 cm, respectively.

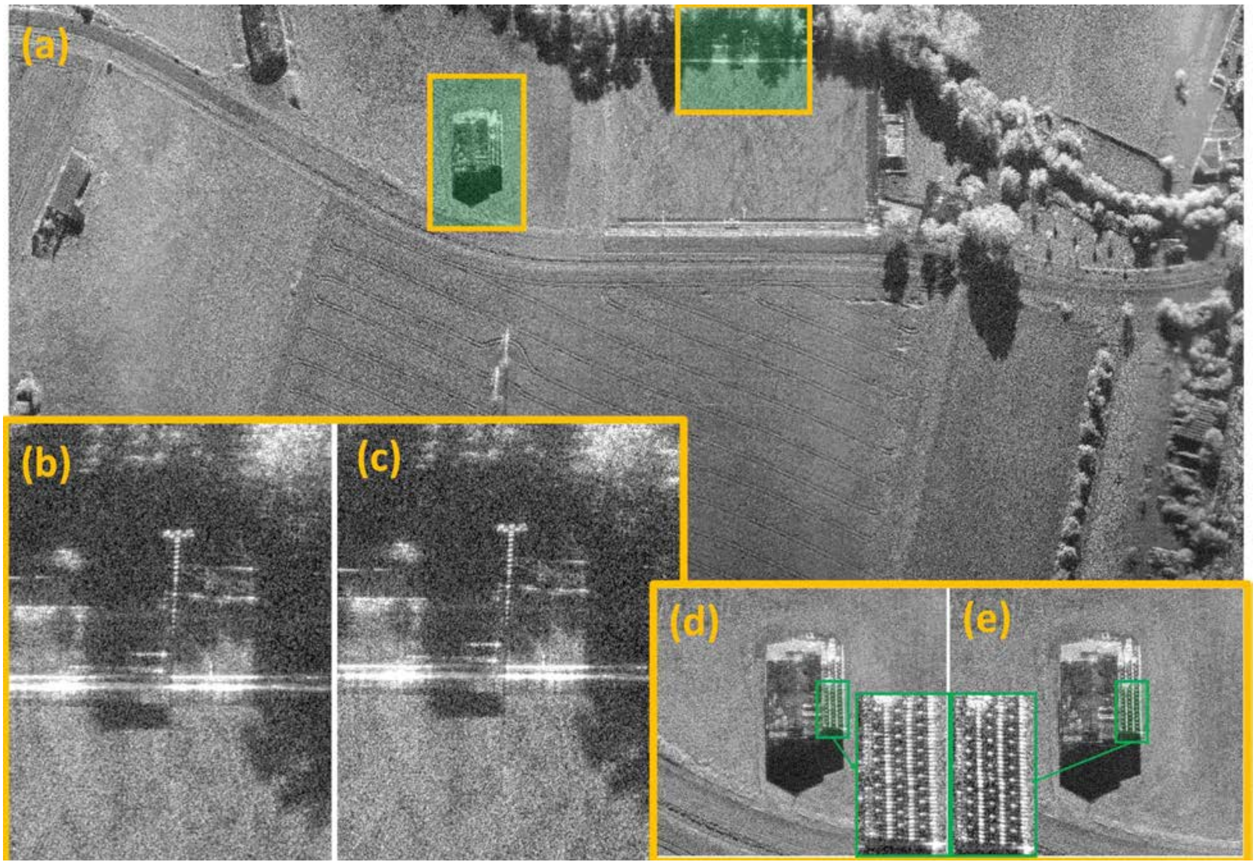

Figure 4-6: SAR image result by postprocessing the radar raw data and the navigational data derived with different GPS/INS sensors. (a) Full scene processed with proposed camera ego motion estimation. (b) Zoom on a floodlight in a stadium focused with camera tracking. (c) Focused with reference data. (d) Zoom on a building processed with camera tracking. (e) Processed with reference data.

Table 4-2: Comparison of maximum point target amplitude, range- and azimuth resolution with SAR data processed by proposed ego motion estimation and reference data.

	Max Point Target Amplitude [dB]	Resolution Range [cm]	Resolution Azimuth [cm]
Camera Ego Motion	13.23	11,4	17,8
SWIR Ego Motion	15,6	10,7	14,5
Reference Data	19,2	10,5	10,9

5 CONCLUSION

Our results clearly show that SAR image processing is possible using camera-based ego-motion estimates, even using infrared images, without GNSS and with completely automatic positioning feature extraction from images. Under benign flight conditions SAR images can be focused with high resolution.

Under more challenging flight conditions such as high drift or roll angles, the quality differences to SAR images formed using the ego-motion information from the INS+GNSS reference system become more noticeable. Additionally, the ego-motion estimate from camera images and MEMS IMU measurements drifts with respect to the GNSS reference, since, in contrast to GNSS, there is no absolute position reference in the images.

Drift may be reduced by adding loop closures to the camera measurements, i.e. re-recognizing ground features when they are passed a second time. This obviously works only if there actually are loops in the plane’s trajectory. Additionally, the extraction of points from images is not necessarily limited to images captured by the cameras on the plane, although those were the only ones we looked at in the scope of this work. Adding geo-referenced images, for instance images acquired by satellites, to the estimator would allow absolute positioning without GNSS and is a topic of future research.

ACKNOWLEDGMENTS

This work was funded by the Technical Center for Information Technology and Electronics WTD 81 (Part of German MoD).

REFERENCES

[1] OpenCV Camera Calibration and 3D Reconstruction, https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html

[2] Z. Zhang, A flexible new technique for camera calibration. IEEE Transactions on pattern analysis and machine intelligence 22.11, 2000

[3] Ribalta A (2011), Time-domain reconstruction algorithms for FMCW-SAR. IEEE Geoscience and Remote Sensing Letters, 8 (3): 396-400.

[4] Palm S, Sommer R, Janssen D, Tessmann A, Stilla U (2019), Airborne circular W-band SAR for multiple aspect urban site monitoring. IEEE Transactions on Geoscience and Remote Sensing, 57 (9): 6996-7016.