

Research Article

Exploiting Attitude Sensing in Vision-Based Navigation for an Airship

Luiz G. B. Mirisola and Jorge Dias

Institute of Systems and Robotics, University of Coimbra, Pólo II, 3030 Coimbra, Portugal

Correspondence should be addressed to Luiz G. B. Mirisola, luiz.mirisola@gmail.com

Received 28 November 2008; Revised 14 March 2009; Accepted 23 June 2009

Recommended by Giovanni Muscato

An Attitude Heading Reference System (AHRS) is used to compensate for rotational motion, facilitating vision-based navigation above smooth terrain by generating virtual images to simulate pure translation movement. The AHRS combines inertial and earth field magnetic sensors to provide absolute orientation measurements, and our recently developed calibration routine determines the rotation between the frames of reference of the AHRS and the monocular camera. In this way, the rotation is compensated, and the remaining translational motion is recovered by directly finding a rigid transformation to register corresponding scene coordinates. With a horizontal ground plane, the pure translation model performs more accurately than image-only approaches, and this is evidenced by recovering the trajectory of our airship UAV and comparing with GPS data. Visual odometry is also fused with the GPS, and ground plane maps are generated from the estimated vehicle poses and used to evaluate the results. Finally, loop closure is detected by looking for a previous image of the same area, and an open source SLAM package based in 3D graph optimization is employed to correct the visual odometry drift. The accuracy of the height estimation is also evaluated against ground truth in a controlled environment.

Copyright © 2009 L. G. B. Mirisola and J. Dias. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

In this paper, computer vision techniques are combined with orientation measurements from an inertial and magnetic sensor package, and this combination is exploited to determine the relative pose of successive views taken by a mobile observer, such as an Unmanned Aerial Vehicle (UAV), moving above leveled terrain. The paper includes trajectory recovery for the remotely piloted unmanned airship of Figure 1, which carries a camera coupled with an Attitude Heading Reference System (AHRS), and a Global Positioning System (GPS) receiver. This visual odometry technique could be applied for any moving camera which images leveled terrain, including aerial vehicles, and it can be used alone if GPS is temporarily unavailable or be fused with GPS position fixes to obtain an improved trajectory and more coherent maps of the ground surface. After finding other relative pose constraints to close the loop, a graph-based 3D Simultaneous Localization and Mapping (SLAM) technique is applied to minimize the drift from visual odometry. Although

only relative height can be recovered from images and an initial height measurement is necessary to recover metric scale, the visual odometry and SLAM approaches recover the height component of the trajectory without additional measurements. The initial height measurement is obtained by imaging an object of known dimensions in the ground.

The benefits obtained when the orientation measurements allow the rotational motion to be compensated, and images to be reprojected onto a stabilized reference frame are explored in Section 3. Specifically, for a sequence of images of a planar area, the transformation that relates corresponding pixel coordinates in two images of a 3D plane is a planar homography. If the rotation is compensated, pixel correspondences can be projected in the ground plane, yielding sets of corresponding scene points, which are registered by finding a rigid transformation to register them directly in scene coordinates. This is an instance of the well-known Procrustes problem [1].

A modern AHRS outputs georeferenced orientation measurements, using accelerometers which measure the

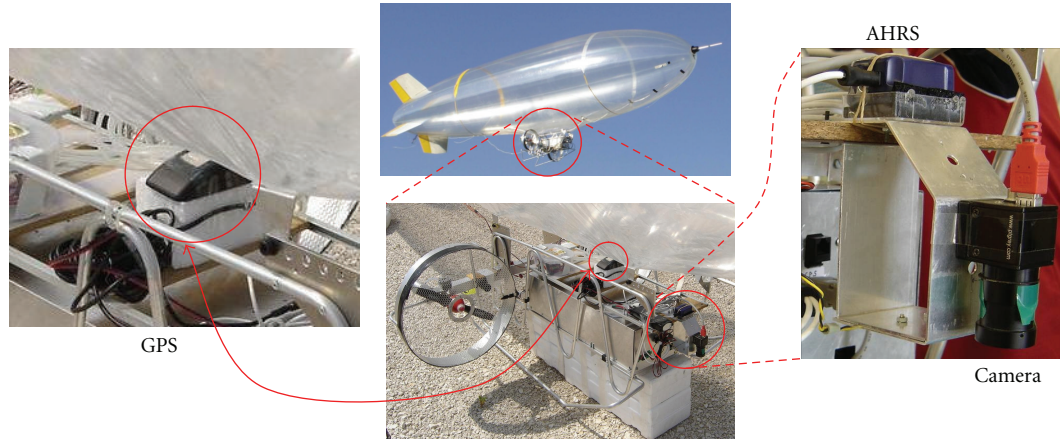


FIGURE 1: Our unmanned airship with the AHRS, camera, and the GPS receiver mounted in its gondola.

direction of gravity and magnetometers which measure the earth magnetic field. The AHRS firmware fuses and filters information from its internal sensors, freeing the main CPU for higher level tasks and generating outputs at a larger frequency than the typical frame rate of digital cameras.

The experiments described in Section 4 utilize a small micromachined AHRS that is fixed rigidly to a monocular digital camera, providing a georeferenced estimate of the camera orientation. To generate this estimate the rigid transformation between the camera and AHRS frames must be known. This transformation is estimated by our recently developed calibration procedure [2, 3], eliminating the need for precise mechanical assembly. This has already been used to improve robustness on image segmentation and 3D structure recovery from stereo [4, 5] or independent motion segmentation [6]. The coordinate frames and calibration processes used are described in Section 2, which also reviews relative pose recovery using only images of a plane.

1.1. Related Work. In [7], a stereovision-only approach is used to build a 3D map of the environment from stereo images taken by a remotely controlled airship, keeping estimates of the camera pose and the position of automatically detected landmarks on the ground. The landmarks are found by interest point algorithms applied on the aerial images. It was not their purpose to integrate inertial measurements.

Again with a UAV carrying a stereo camera, the trajectory can be recovered by registering successive sets of triangulated 3D points calculated for each stereo frame. Trajectories of hundred of meters have been recovered [8], although the UAV height is limited to a few meters due to stereo baseline size.

Image mosaicing was performed for an unmanned submarine navigating above flat horizontal sea bottom, using only images from a monocular camera as input for the calculation of relative poses [9]. The most recent vehicle orientation estimate was used to reproject the images onto a stabilized plane, avoiding using direct measures from inertial sensors. The vehicle pose is estimated, and a mosaic of the sea bottom is generated, which in turn is used

for navigation. Although it involved elaborate optimization steps, the registration converges only if the vehicle has restricted movement and shows minimal variation in roll and pitch angles. These results indicate a limitation for vision-only approaches.

A UAV trajectory can also be estimated by fusing GPS and on-board inertial data and considering a dynamic vehicle model. Given such accurate vehicle poses, images taken from a high-flying airplane are reprojected onto the ground plane thus achieving one-pixel accuracy with no need for image-based registration techniques [10].

Combined inertial and vision data were used to keep pose estimates in an underwater environment, navigating a robot submarine over a large area [11], with no access to a beacon-based localization system. Relative pose measurements from the images were used to avoid divergence of the tracked vehicle pose, and an image mosaic was generated as a byproduct.

In the context of an aerial vehicle, even without utilizing the available GPS data, inertial measurements and observations of artificial landmarks on images can be fused together to provide a full 6-DOF pose estimate, performing localization and mapping, and incorporating recent advances on filtering. Inertial sensors and barometric altitude sensors can also compensate for inaccurate GPS altitude measurements or satellite drop outs [12]. Similar results were also obtained using ground vehicles [13].

The trajectory of an aerial robot can be recovered from images of a planar, horizontal surface using interest point matching, with the help of an altimeter to find the scale [14]. When the the planar homography model is used various geometric constraints have been proposed to recover the right motion among the four solutions of the homography matrix decomposition [15]. This has been already performed for an airship by using clustering and blob-based interest point matching algorithms, building an image mosaic which in turn is used for navigation. The relative pose estimation involved homographies calculated from images of a planar area taken by a UAV [15] and image sequences taken by various UAVs [16].

The visual odometry approach proposed in this paper uses only orientation estimates from an AHRS and corresponding pixels from monocular camera images (obtained by the SURF [17] interest point matching algorithm), thus it is not limited by stereo baseline constraints and it does not depend on artificial landmarks. The orientation measurements are obtained from the AHRS, without considering any model of the vehicle dynamics, even if our small airship has large roll and pitch variations [18]. It may be used in very low heights when the GPS uncertainty in altitude is significant (e.g., during landing or taking off) or in large-altitude flights above a flat area. The camera movement is not restricted, provided that its orientation is measured and the ground plane is imaged. GPS is utilized only for comparison, and not on the trajectory recovery process itself, except in the experiment of Section 4.4 which combines GPS and visual odometry.

The reprojection of images into a virtual plane is well known [9, 10], but the homography model is still used in most works. The pure translation model is certainly an option and appears to be especially suitable to the estimation of the vertical motion. Moreover, the extraction of the translation vector up to scale is trivial, with a unique solution. In contrast, with the homography model, the recovered matrix must be algebraically decomposed into rotational and translational components, yielding four possible solutions, of which only one is the real relative pose [19]. Additionally, the optimization step included in homography estimation may be eliminated, decreasing the computational cost in realistic scenarios even if outlier removal becomes more difficult. Some preliminary results using aerial images and a pure translation model have already been obtained [20], but more comprehensive results are still missing in the literature.

2. Definitions and Reference Frames

The following notation will be used throughout this paper: upper case bold letters denote matrices (e.g., \mathbf{K}), lower case bold letters denote column vectors (e.g., \mathbf{t}), italic letters denote scalars (e.g., h), and letters in script font denote 3D points (e.g., \mathcal{X}).

2.1. Projective Camera Model. The standard pinhole camera model is used. The camera center is the origin of an Euclidean coordinate system called the camera frame and denoted as $\{\mathcal{C}\}_i$ frame, where i is a time index to indicate the camera movement. A point in space with homogeneous coordinates ${}^c\mathcal{X}$ in the camera frame is mapped to the homogeneous image pixel \mathbf{x} by the equation $\mathbf{x} = \mathbf{K}[\mathbf{I} \mid \mathbf{0}]\mathcal{X}$, where \mathbf{I} is the identity matrix, $\mathbf{0}$ the zero vector, and \mathbf{K} is the intrinsic parameter matrix, defined by

$$\mathbf{K} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where f represents the camera focal length in terms of pixel dimensions. The variables x_0 and y_0 are the principal point

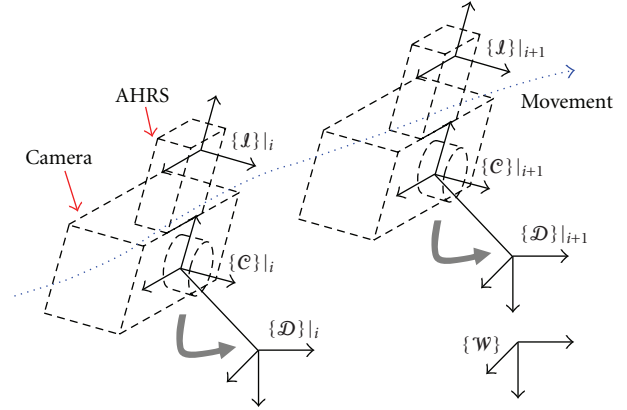


FIGURE 2: Moving observer and frames of reference.

coordinates in terms of pixel dimensions. The cameras are calibrated, that is, the matrix \mathbf{K} is determined, using the Camera Calibration Toolkit [21], which also determines the radial lens distortion coefficients. All images used in this paper were previously corrected for lens distortion.

2.2. Definitions of Reference Frames. The following other reference frames are defined, with i being a time index, as shown in Figure 2.

- (i) *Inertial Sensor Frame* $\{\mathcal{I}\}_i$. Defined by the measurement axes of the inertial sensors in the AHRS. The AHRS orientation output is the rotation between the $\{\mathcal{I}\}_i$ and the $\{\mathcal{W}\}$ frames.
- (ii) *World Frame* $\{\mathcal{W}\}$. A Latitude Longitude Altitude (LLA) frame.
- (iii) *Virtual Downwards Camera* $\{\mathcal{D}\}_i$. This is a georeferenced camera frame, which shares its origin with the $\{\mathcal{C}\}_i$ frame, but its optical axis points down, in the direction of gravity, and its other axes (i.e., the image plane) are aligned with the north and east directions. Figure 2(a) shows this frame on a different position than the $\{\mathcal{C}\}_i$ frame only to provide a clearer understanding of the drawing.

2.3. The Virtual Horizontal Plane Concept. The knowledge of the camera orientation provided by the orientation measurements allows the image to be reprojected on entities defined on an absolute georeferenced frame, such as a *virtual horizontal plane* (with normal parallel to gravity), at a distance f below the camera center, as shown in Figure 3(a). Projection rays from 3D points to the camera center intersect this plane, reprojecting the 3D point into the plane. This reprojected point corresponds to the image of the virtual camera $\{\mathcal{D}\}_i$, and it compensates differences due to heading and viewpoint changes. The camera height variation is not compensated, resulting in scale differences in the virtual images. Section 3.2 details how to perform this reprojected.

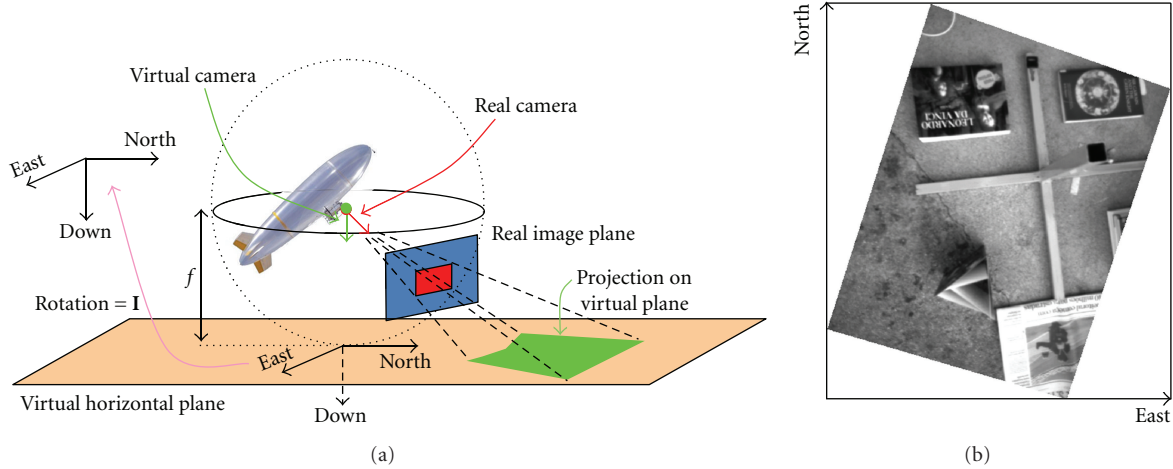


FIGURE 3: The virtual horizontal plane concept and an reprojected image example.

Figure 3(b) shows the reprojection of one image as an example. The original image is shown above and in the reprojected image shown below the cross in the ground is nearly aligned with the north and east axes.

2.4. Experimental Platforms and Calibration. The GPS used for comparison is a Garmin GPS35 without differential correction. The output includes the expected position error in the horizontal and vertical axes (eph and epv) for each position fix. The flight experiments used an MTi AHRS from Xsens [22]. The manufacturer states that its orientation output standard deviation is 1° if the sensor is static. The error should be larger for the moving UAV. The airship gondola has combustion engines and an aluminum structure, thus there is no magnetic interference which could hamper the AHRS compass.

The camera is a Point Grey Flea [23], shown in Figure 1 rigidly mounted together with the AHRS. The camera is first calibrated to determine its intrinsic parameter matrix and lens distortion parameters [21]. A subsequent off-line calibration routine [2, 3] uses checkboard images to find the rigid body rotation between the $\{\mathcal{C}\}$ and $\{\mathcal{I}\}$ frames, denoted ${}^{\mathcal{I}}\mathbf{R}_{\mathcal{C}}$. It is based on measurements of the gravity direction from both sensors. The camera observes vertical vanishing points from vertically placed chessboards, and the AHRS measures the gravity vector from its accelerometers. Timestamps supplied by the image acquisition library and by the sensors firmware are utilized to match the measurements of different sensors.

A Pentium IV 2.4 GHz computer was utilized in all computations reported in this paper.

2.5. Planar Surfaces and Homographies. Consider a 3D plane imaged by two identical cameras placed in different positions. Consider also a set of pixel correspondences belonging to that plane in the form of pairs of pixel coordinates $(\mathbf{x}, \mathbf{x}')$, where each pair corresponds to the projection of the same 3D point into each image. A homography represented by a 3×3 matrix \mathbf{H} relates these two sets of homogeneous

pixel coordinates such that $\mathbf{x}' = \mathbf{K} \cdot \mathbf{H} \cdot \mathbf{K}^{-1} \cdot \mathbf{x}$, and the homography is said to be *induced* by the 3D plane [24]. The homography can be recovered from pixel correspondences, and it is related to the 3D plane normal \mathbf{n} , the distance from the camera center to the plane d , and to the relative camera poses represented by the two camera projection matrices $\mathbf{P} = [\mathbf{I} \mid \mathbf{0}]$ and $\mathbf{P}' = [\mathbf{R} \mid \mathbf{t}]$, as shown in Figure 4(a), by

$$\lambda \mathbf{H} = \lambda \left(\mathbf{R} - \frac{\mathbf{t} \mathbf{n}^T}{d} \right), \quad (2)$$

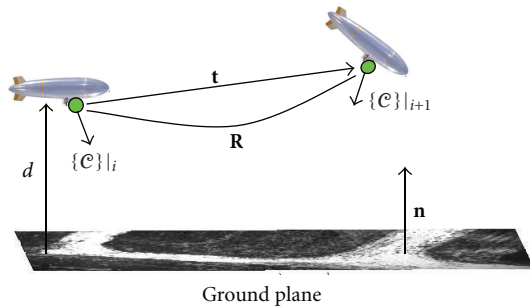
where the arbitrarily scaled matrix $\lambda \mathbf{H}$ is recovered first, and then the scale factor λ must be recovered. The scale λ is equal to the second largest singular value of $\lambda \mathbf{H}$, up to sign [19]. The correct sign of λ is recovered by imposing a positive depth constraint.

Then the normalized homography matrix \mathbf{H} can be decomposed into \mathbf{n} , the rotation matrix \mathbf{R} , and the translation vector \mathbf{t}/d [19]. The relative pose recovered has an inherent scale ambiguity, as the translation magnitude is not recovered, only the ratio \mathbf{t}/d . The recovered homography can be used to register an image pair by applying the recovered transformation \mathbf{H} on the first image as in the example of Figure 4(b).

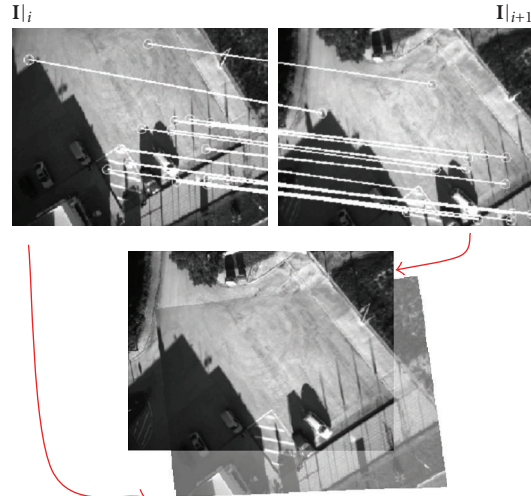
2.5.1. Pure Rotation Case. The *infinite homography* \mathbf{H}_∞ is the homography induced by the plane at infinity. It is also the homography between two images taken from two cameras *at the same position* (no translation, $\mathbf{t} = \mathbf{0}$), but rotated by a rotation represented by a matrix \mathbf{R} . The infinite homography can also be used to synthesize a virtual view from a nonexistent virtual camera, at a desired orientation, given the appropriate rotation matrix.

The infinite homography is calculated by a limiting process where d approaches infinity, or the translation \mathbf{t} tends to zero. In both cases the ratio \mathbf{t}/d tends to zero in (2):

$$\mathbf{H}_\infty \triangleq \lim_{\mathbf{t}/d \rightarrow 0} \mathbf{K} \mathbf{H} \mathbf{K}^{-1} = \lim_{\mathbf{t}/d \rightarrow 0} \mathbf{K} \left(\mathbf{R} + \frac{\mathbf{t}}{d} \mathbf{n}^T \right) \mathbf{K}^{-1} = \mathbf{K} \mathbf{R} \mathbf{K}^{-1}. \quad (3)$$



(a) A 3D plane is imaged by a moving camera



(b) Above, two aerial images with some pixel correspondences. Below, the same images registered with a homography

FIGURE 4: A 3D plane imaged by a moving camera induces a homography.

3. Trajectory Recovery under Pure Translation

In the context of a horizontal 3D plane imaged by a moving camera with measured orientation, this section explores the simpler motion model to achieve more accurate measurements of the camera height and to reconstruct the camera trajectory from an image sequence.

3.1. Facilitating Interest Point Matching with Reprojected Images. The first step of the processes described on this paper is to reproject each image onto the virtual horizontal plane, as shown in Section 2.3. In this way, the camera rotation is compensated and the relative pose between any pair of camera poses is reduced to a pure translation.

Another objective is to relax the demands on interest point detection, encoding and matching algorithms [25]. Actually, because the ratio of correct matches versus the total number of interest points detected is better with images taken from the same viewpoint, the interest point algorithm parameters may be tuned to detect less features, while still matching the same number of interest points correctly. Therefore, the interest point matching process can be faster (less interest points means less descriptor computation, and a smaller number of descriptors to match means faster matching), or more robust (more correct matches with the same number of detected interest points means greater probability of successful image registration), which is a tradeoff. Otherwise, it would be necessary that the feature encoding is invariant to heading and viewpoint differences in the original images [26].

For the low-altitude aerial dataset used in Section 4.2, the time spent matching interest point descriptors using the reprojected images was 41% of the time required to perform the matching with the original, nonreprojected images. The number of correctly matched interest points was 10% smaller with the reprojected images. The time spent to generate the

reprojected images must be discounted, but it was on average four times smaller than the speedup obtained in descriptor matching. Besides that, reprojecting the images may be a necessary task itself, for example, to generate the images drawn in Figure 9. In our previous work similar gains were reported with a small-scale dataset [25].

This reprojection—sometimes called *prewarping*—is already widely used to preprocess images taken from moving vehicles like submarines [9], and similar improvements in homography-based registration were reported in which viewpoint invariance is simulated by registering the images with an initial estimate of the homography [27]. An exhaustive evaluation of these improvements and tradeoffs is still missing in the literature but the existing evaluations of interest point algorithms [28, 29] demonstrate that their performance tends to degrade with changes in viewpoint.

All experiments in this paper used reprojected images, although only the reprojected coordinates of the matched interest points are used, thus the actual generation of reprojected images is not strictly necessary. Typically hundreds of corresponding points are found between a pair of consecutive images. The average number of valid correspondences (excluding outliers) in the airship flight experiment was 388. The original images and the homography model were used for comparison in Section 4.2.

3.2. The Infinite Homography. The infinite homography synthesizes a virtual view of a nonexistent virtual camera $\{\mathcal{D}\}_i$ from the real image taken by the camera $\{\mathcal{C}\}_i$. To accomplish this, the rotation that places the $\{\mathcal{C}\}_i$ frame into the $\{\mathcal{D}\}_i$ frame must be known. The translation between both frames is zero by definition.

For each image \mathbf{I}_i , a simultaneous AHRS orientation output ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{I}}|_i$ estimates the rotation between the $\{\mathcal{I}\}_i$ and $\{\mathcal{W}\}_i$ frames. Given the rotation matrix ${}^{\mathcal{J}}\mathbf{R}_{\mathcal{C}}$ from the camera-inertial calibration, the camera orientation in the

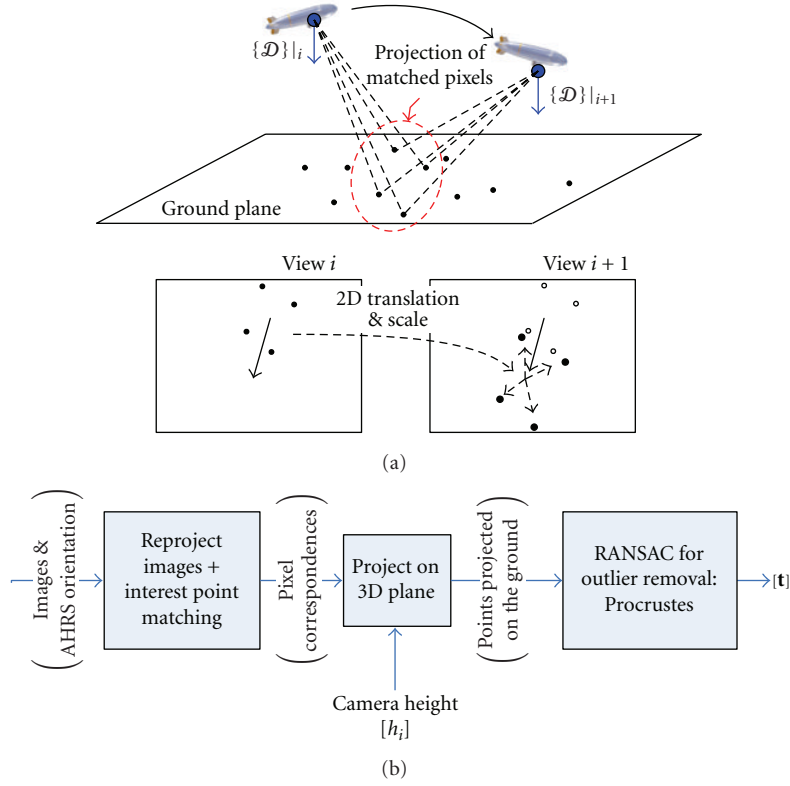


FIGURE 5: Finding the translation between successive camera poses by 3D scene registration.

world frame is calculated as the rotation ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{C}}|_i = {}^{\mathcal{W}}\mathbf{R}_{\mathcal{I}}|_i \cdot {}^{\mathcal{I}}\mathbf{R}_{\mathcal{C}}$.

The rotation ${}^{\mathcal{D}}\mathbf{R}_{\mathcal{W}}$ rotates from the $\{\mathcal{W}\}$ to the $\{\mathcal{D}\}|_i$ frame:

$${}^{\mathcal{D}}\mathbf{R}_{\mathcal{W}} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}. \quad (4)$$

Then the rotation between the $\{\mathcal{D}\}|_i$ frame and the $\{\mathcal{C}\}|_i$ frame is calculated as ${}^{\mathcal{D}}\mathbf{R}_{\mathcal{C}}|_i = {}^{\mathcal{D}}\mathbf{R}_{\mathcal{W}} \cdot {}^{\mathcal{W}}\mathbf{R}_{\mathcal{C}}|_i$.

Therefore the transformation used to reproject images into the virtual horizontal plane is

$${}^{\mathcal{D}}\mathbf{H}_{\mathcal{C}}|_i \triangleq \mathbf{K} \cdot {}^{\mathcal{D}}\mathbf{R}_{\mathcal{C}}|_i \cdot \mathbf{K}^{-1}. \quad (5)$$

3.3. Registering Sets of Scene Points (Procrustes). Suppose a sequence of aerial images of a horizontal ground patch, and that these images are reprojected on the virtual horizontal plane as shown in Section 3.2. The virtual cameras have horizontal image planes parallel to the ground plane. Given two successive views and pixel correspondences between them, the relative pose between the two camera positions must be recovered in the form of a 3D vector \mathbf{t} .

Each corresponding pixel is projected into the ground plane, generating a 3D point, as shown in Figure 5(a). Two sets of 3D points are generated for the two successive views,

and these sets are directly registered in scene coordinates. Indeed, as all points belong to the same ground plane, the registration is solved in 2D coordinates. Figure 5(b) shows a diagram of this process.

Each corresponding pixel pair $(\mathbf{x}, \mathbf{x}')$ is projected by (6) yielding a pair of points $(\mathcal{X}, \mathcal{X}')$:

$$(\mathcal{X}, \mathcal{X}') = \left(\begin{bmatrix} \frac{(x_x - n_x) \cdot h_i}{f} \\ -\frac{(x_y - n_y) \cdot h_i}{f} \\ h_i \end{bmatrix}, \begin{bmatrix} \frac{(x'_x - n_x) \cdot h_i}{f} \\ -\frac{(x'_y - n_y) \cdot h_i}{f} \\ h_i \end{bmatrix} \right), \quad (6)$$

where $\mathbf{x} = [x_x, x_y, 1]^T$, $\mathbf{x}' = [x'_x, x'_y, 1]^T$, in inhomogeneous form, h_i is the height over the ground plane of the first camera, and the minus sign in the second coordinate is only an adjustment between image and world axes, depending on the camera frame convention. If the position of the first camera is taken as the origin of the x and y coordinates (only relative pose must be determined here), the coordinates of \mathcal{X} are the correct coordinates of the projection of \mathbf{x} in the ground plane, in actual metric units. But \mathcal{X}' is not the correct projection because the second camera position was not considered—there is a difference of translation and scale between the two sets of corresponding \mathcal{X}' and \mathcal{X} points in the ground plane.

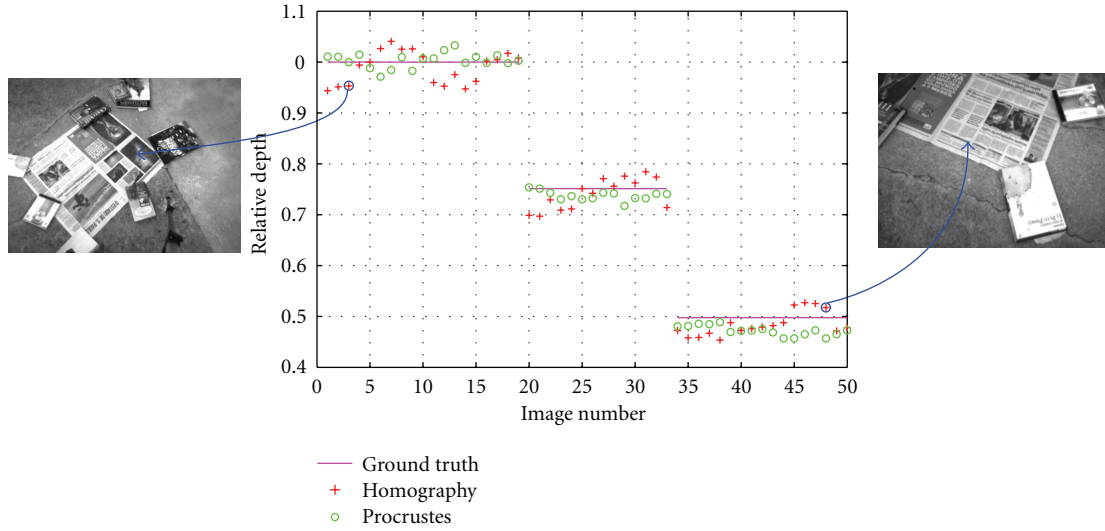


FIGURE 6: Relative heights to the ground for the tripod experiment, with two example images. Each unit on the abscissa corresponds to one image. Images were taken from 3 different heights.

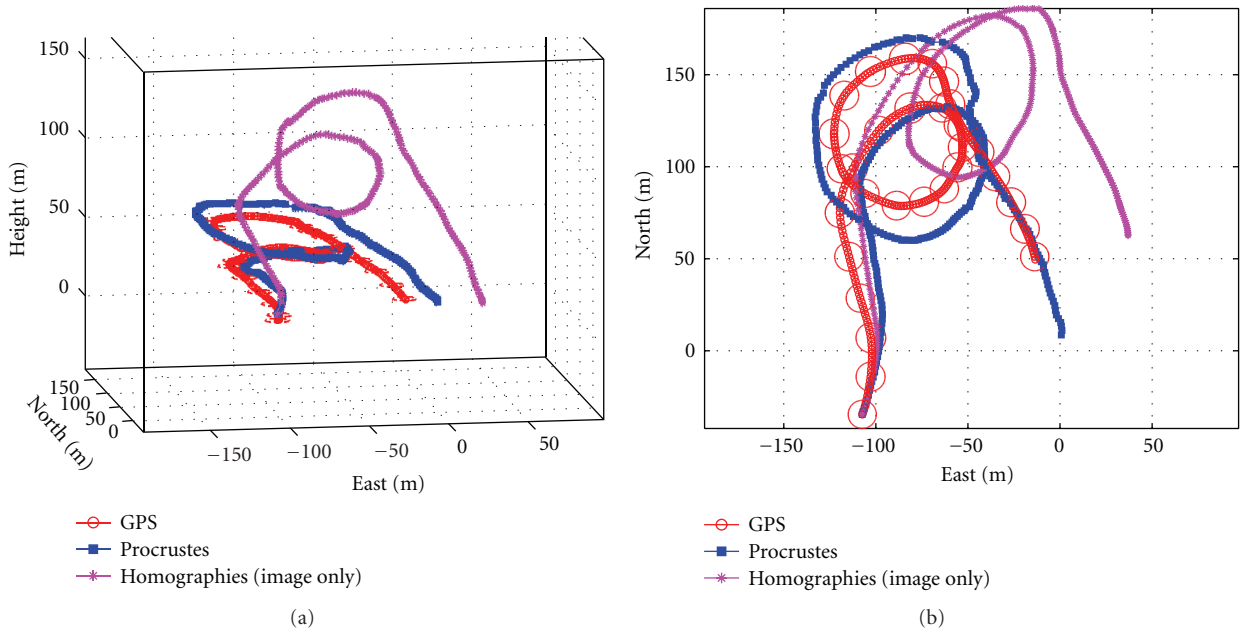


FIGURE 7: 3D (a) and 2D (b) plots comparing visual odometry with GPS position fixes. The circles indicate some of the GPS eph values (i.e., one standard deviation).

Given all $(\mathcal{X}, \mathcal{X}')$ pairs of corresponding points in the ground plane, the Procrustes routine shown in Appendix B finds the 2D translation and scale factor which register the two point sets, yielding the x and y components of \mathbf{t} and the scale factor s which represents the relative height between the first and second cameras. If the assumptions of having both image and ground planes parallel and horizontal are true, with outliers removed, and with equal confidence to all corresponding pixel coordinates, then it can be proved that the Procrustes solution is the best solution in a least squares sense. The z component of \mathbf{t} is recovered by the equation $s = h_{i+1}/h_i = (h_i + t_z)/h_i$.

4. Results

4.1. Tripod Experiment: Relative Height Recovery. In this experiment, the rigidly coupled AHRS-camera system of Figure 1 was mounted on a tripod and 50 images of the ground floor were taken from different viewpoints at three different heights. The tripod was moved manually but kept still while each image was being taken. Image examples are shown on Figure 6. The objective is to calculate the height of the camera for each view as a ratio of the height of the first view, and to compare the results obtained against hand-measured ground truth.

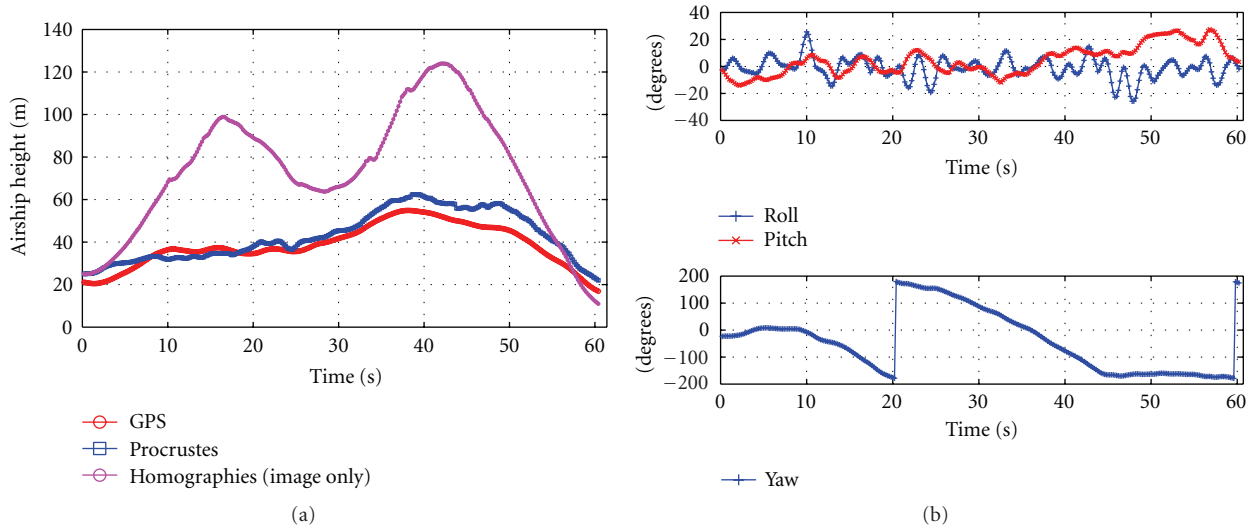


FIGURE 8: The airship height from GPS and visual odometry (a) and the attitude angles measured by the AHRS (b).

Figure 6 shows the relative height for all 50 images. Two arrows connect two highlighted points to their respective images. The tripod was set to three different heights, therefore the three horizontal lines are the ground truth. The circles are the relative heights found by the Procrustes routine. The crosses are the relative heights taken as the determinant of a homography, estimated with RANSAC, optimized to minimize the projection error on pixel correspondences, and scaled as in (2).

The results are summarized in Table 1, where the relative height unit is the height reference (1045 mm), and they show that the closed form Procrustes routine yields better accuracy than the homography model. There was no improvement in execution time, because although there is no an optimization step, outlier removal was much slower.

4.2. Trajectory Recovery for a UAV. The next experiments utilize images taken by our remotely controlled airship of Figure 1, carrying the AHRS-camera system and the GPS receiver, flying above a planar area next to Coimbra City Airfield. A satellite image of the flight area is given in Figure 10. The original datasets and videos recording the flights and complementing the results of this section and Section 4.4 are available on our website [30].

The letters written on the ground next to the runway were visible on the first image. The letters were measured on the image, and then the first airship height was calculated from the camera intrinsic parameters and the actual, hand-measured, size of the letters. The first height was found to be $h_1 = 25$ m.

The recovered trajectory is shown in Figure 7. Both 2D and 3D plots of the same data are provided. The GPS trajectory is indicated by small circles, with larger circles indicating the GPS *eph*, which was around 8 m in the horizontal axes. The GPS indicates a trajectory length of 543 m, and average speed of 9 m/s. A Kalman Filter with a discrete Wiener process acceleration model [31] filtered

the translation vectors for all methods and predicted the translation for the few image pairs for which the translation estimation was not successful and a measurement was missing. Appendix A shows details of the prediction equations for this Kalman Filter.

The squares in Figure 7 show the trajectory reconstructed by using the 3D translation vectors estimated by the Procrustes approach as the input of the Kalman Filter. The stars indicate the trajectory recovered by the homography model using only the original camera images, and not using AHRS data, reprojected images, or the Procrustes routine. As only the ratio \mathbf{t}/d is recovered by the homographies, the scale is recovered by multiplying the recovered vector by the currently estimated airship height. The airship position including its height is estimated by the Kalman Filter using these scaled 3D translation vectors as inputs. The airship height for each of the recovered trajectories and the attitude angles measured by the AHRS are shown separately in Figure 8. Tables 2 and 3 show the errors in the visual odometry, taking the GPS as a reference, and the execution times.

Another image sequence with a recovered trajectory is shown in Figure 9. The images are reprojected on the ground plane forming a map by using (6) to find the metric coordinates of the projection of its four corners in the ground plane and drawing the image on the canvas accordingly. The better alignment of the larger road and of the smaller lines in the grass field indicates that the map of Figure 9(a), which utilizes the airship poses recovered by the visual odometry, is better registered than the map of Figure 9(b), which utilizes the GPS position fixes. Both figures utilize the same image orientation data.

4.3. Analyzing the Effect of Orientation Error. If a pair of identical cameras placed at different positions has the same orientation, a relative height value μ may be calculated for each corresponding pixel pair $(\mathbf{x}_1, \mathbf{x}_2)$ as a ratio of image

TABLE 1: Results for relative height with a horizontal ground plane.

	Rel. height error		RANSAC	Average time (s)	
	RMS	Std		Optimiz.	Total
Homography	0.033	0.030	0.40	0.29	0.69
Procrustes	0.019	0.017	0.74	—	0.74

TABLE 2: Comparison of visual odometry with the GPS reference. All values are given in meters.

Unit: meters	3D Position error			2D Position error			Error in the length of \mathbf{t}		
	Avg.	Max.	Final	Avg.	Max.	Final	RMS	Avg.	Max.
Homography	57.5	98.2	52.3	43.1	67.6	52.0	0.88	-0.63	1.24
Visual odometry: Procrustes	16.5	43.6	43.6	15.3	43.3	43.3	0.73	-0.46	0.59

distances with the epipole \mathbf{e} (also called Focus of Expansion), in the form $\mu = \|\mathbf{x}_1 - \mathbf{e}\|/\|\mathbf{x}_2 - \mathbf{e}\|$ [32]. The relative height of an image pair is the ratio of the height of the second camera over the height of the first one. The reprojected images simulate virtual cameras with parallel and horizontal image planes, which image a horizontal plane. Thus all relative height values should be equal for all points belonging to the ground plane.

Figure 11 shows, for one example image of the Coimbra City Airfield, the relative height values for each matched pixel (correspondences classified as outliers are *not* shown). On the right, the same data is shown as a 3D plot with a larger scale for ease of visualization. The color scale and the z axis both indicate the same μ values. This example has noticeable orientation error, although the translation vector still could be estimated.

In Figure 11(a) all corresponding pixels are in a relatively narrow band close to the center of the image. There were other correctly matched pixels in other sections of the image but, as the motion model does not consider the noncompensated rotation, they were classified as outliers due to errors in the orientation estimation. This effect is more significant in areas far from the nadir point. Errors in the calibration of the rotation between the camera and inertial sensor frames should increase this effect. If the orientation error is too large, too many pixel correspondences are discarded and it is not possible to estimate the translation reliably.

Even among the inlets the measured height change varies between 2.3% and 1.6% of the first image height. The building in the bottom of the image is out of the horizontal plane and thus there are a few points that do not follow the general tendency.

These empirical observations can be verified by analyzing how an error in the camera orientation estimate affects a pixel coordinate projected in the ground plane. Suppose two camera projection matrices, $\mathbf{P}_1 = \mathbf{K}[\mathbf{R}_\theta \mid \mathbf{0}]$ and $\mathbf{P}_2 = \mathbf{K}[\mathbf{R}_\theta \mid \mathbf{t}]$, where \mathbf{R}_θ is a rotation matrix which represents a rotation of θ degrees over the x axis. Here \mathbf{R}_θ will represent an error on the camera orientation estimate—if the rotation \mathbf{R}_θ is ignored when the pixel coordinates are projected into the ground plane, then which effect does it have in the

coordinates of the resulting points in the ground plane? This effect should be analogous to the errors shown in Figure 11, as ground plane coordinates are a projection of virtual image coordinates, and the following form was found to result in simpler equations, easier to analyze.

Take a pixel coordinate $\mathbf{x} = [x, y, 1]^T$, and the 3D points which are projected into it in both cameras, as expressed by $\mathbf{x} = \mathbf{P}_1 \mathcal{X}_1$ and $\mathbf{x} = \mathbf{P}_2 \mathcal{X}_2$. Invert the projection equations expressing \mathcal{X}_1 and \mathcal{X}_2 as a function of \mathbf{x} . Subtracting both equations should result in a difference due to the translation (the z component is ignored as the points should be in the same plane):

$$\mathcal{X}_2 - \mathcal{X}_1 = \begin{bmatrix} \overbrace{\frac{(x - x_0) \cdot t_z}{f}}^{\text{vertical motion}} - \overbrace{t_x}^{\text{horiz. motion}} \\ (y - y_0) \cdot t_z - f \cdot t_y \\ -\sin(\theta) \cdot (y - y_0) + f \cdot \cos(\theta) \end{bmatrix}. \quad (7)$$

The rotation over the x axis affects only the y component. The x component, as expected, has terms depending on the horizontal and vertical motions, which are indicated, and does not depend on the rotation— θ does not appear in the first line. If $\theta = 0$ the y component is reduced to a form similar to the first component, but if $\theta \neq 0$ the change in the denominator will cause an error which increases with the distance from the pixel to the principal point and with $\sin \theta$, besides a smaller perturbation due to the $\cos \theta$ term.

This explains the behavior observed in Figure 11. The farther a corresponding pixel is from the principal point, the larger the error in relative height is. Moreover, this error should be more pronounced in a specific direction, depending on the direction of the error in the orientation estimate, and this causes the narrow band of inliers visible in Figure 11—the band represents the direction less affected by the orientation error.

The resilience to errors in the rotation estimate can be verified in simulation. Take the same projection matrices \mathbf{P}_1 and \mathbf{P}_2 , and generate 150 random points belonging to the field of view of camera \mathbf{P}_1 and to the ground plane which is 100 m below the camera. The points which also belong to the field of view of camera \mathbf{P}_2 are the simulated correspondences.

TABLE 3: Average execution times of visual odometry techniques, in seconds.

Unit: seconds	RANSAC		Optimization		Total
	Avg.	Std.	Avg.	Std.	
Homography	0.05	0.02	0.43	0.19	0.48
Visual odometry: Procrustes	0.18	0.08	—	—	0.18

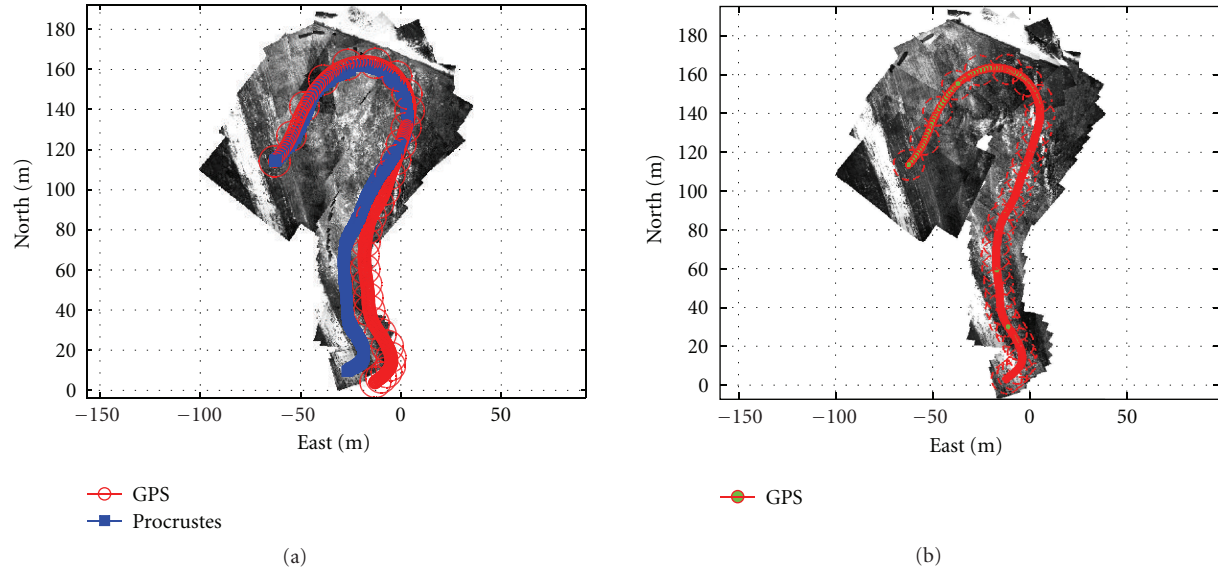


FIGURE 9: Maps formed by reprojecting the images on the ground plane, with trajectories superimposed. The circles indicate some of the GPS eph values (i.e., one standard deviation).



FIGURE 10: A satellite image of the flight area, the Coimbra City Airfield.

These points are projected to image coordinates in both cameras, a Gaussian error with standard deviation of 1 pixel is added to each pixel coordinates, and then the pure translation registration process is performed resulting in a translation vector to be compared with the true one.

Figure 12 shows the simulation results for various vertical and horizontal displacement values for the translation vector, and for different values of the orientation error θ . Each point on each plot is the average error of 8 configurations where the direction of the translation varied at 45° steps covering the 360° range. The error in the recovered translation vector is less than 2% of its magnitude even if the orientation error is as large as 20° , except for the smallest displacement, where the relative error is large because the translation is too small and the noise in pixel coordinates is dominant.

This analysis does not apply to loop-closing constraints as the orientation error may be very different in this case—here it is assumed that the orientation error \mathbf{R}_θ does not significantly change between two successive image acquisitions.

4.4. Combining GPS and Visual Odometry. The translation recovered by the visual odometry was fused with GPS position fixes by the Kalman Filter described in Appendix A. The GPS measures the airship position with standard deviation given by the eph and epv values supplied for each position fix, and translation vectors from the visual odometry are interpreted as a velocity measurement between two successive poses, with a constant covariance smaller in the vertical axis than in the horizontal ones. The GPS absolute position fixes keep the estimated airship position

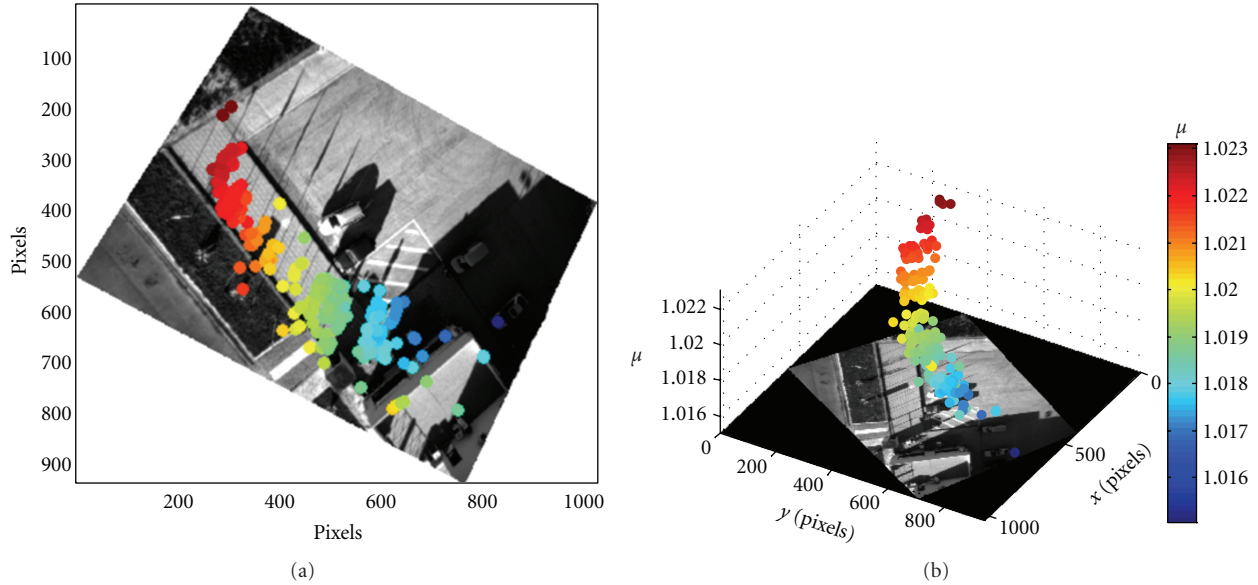
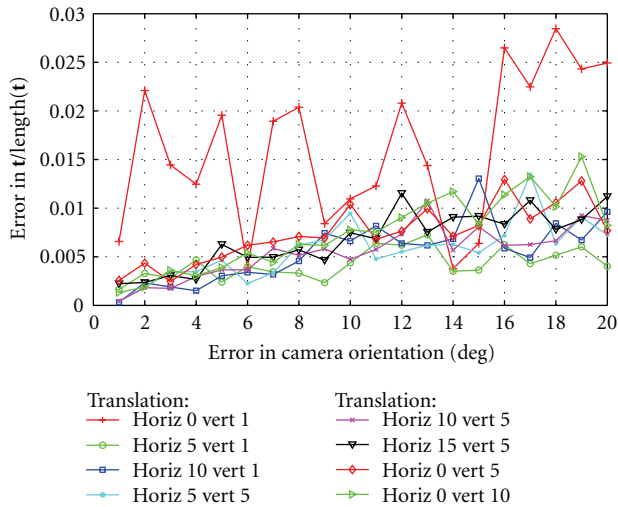
FIGURE 11: The μ values for individual pixel correspondences.

FIGURE 12: Relative error of the recovered translation vector in simulation.

from diverging, while the visual odometry measurements improve the trajectory locally. The fused trajectory is shown in Figure 13(a) next to the original GPS trajectory. The latter is shown alone in Figure 13(b). The maps in Figures 13(a) and 13(b) were generated from the same images and camera orientation values, the only difference being in the camera poses. The fused trajectory is still heavily dependent on the GPS, and its corresponding map is far from perfect, but some details such as the letters and the airfield runway are better registered with the fused trajectory (a) than with GPS alone (b). This dataset contains 1000 images (only 68 are drawn). This trajectory is too large to be recovered by visual odometry alone, therefore the map corresponding to this is not shown.

4.5. Closing the Loop with Graph-Based Optimization. When the same area is imaged a second time by the on-board camera, the relative pose between these two views is retrieved and used to “close the loop.” Then a graph is built where the nodes represent the camera 3D poses and the edges the relative pose constraints, both the sequential and the loop-closing ones. As the graph is initialized with the trajectory recovered by visual odometry, at first there is no error in the sequential constraints—but the loop-closing constraints will not be satisfied due to the drift in the visual odometry. By optimizing in the position of the nodes to minimize the error on each relative pose constraints, the drift can be minimized. Figure 14 shows the same trajectory of Figure 7 where the loop-closing constraints are indicated by the green arrows. Each arrow represents a relative pose constraint between the node at its tail and the node connected to its tip by a colored line. Therefore the colored lines represent the error relative to this constraint, which should be minimized.

To detect the loop-closing constraints, the principal point of each image along the trajectory is projected in the ground plane by (6) and stored. Then, for each image, the projection of its principal point in the ground plane is compared with the database, looking for a previous image which imaged the same area. The image corresponding to the closest principal point in the database is chosen as a matching candidate, not considering the last 5 seconds of the flight, and not considering principal points which are too far—often there is no candidate at all. Then interest point matching between this image pair is attempted. If it is successful the relative pose is recovered and a new edge is added to the graph.

To optimize the constraint graph, the open-source package TORO [33] was utilized with the following settings and modifications. (1) As the relative pose is a translation vector, and its magnitude depends on the estimated camera height, the covariances of all constraints were set to be proportional to the camera height. The standard deviation of the vertical

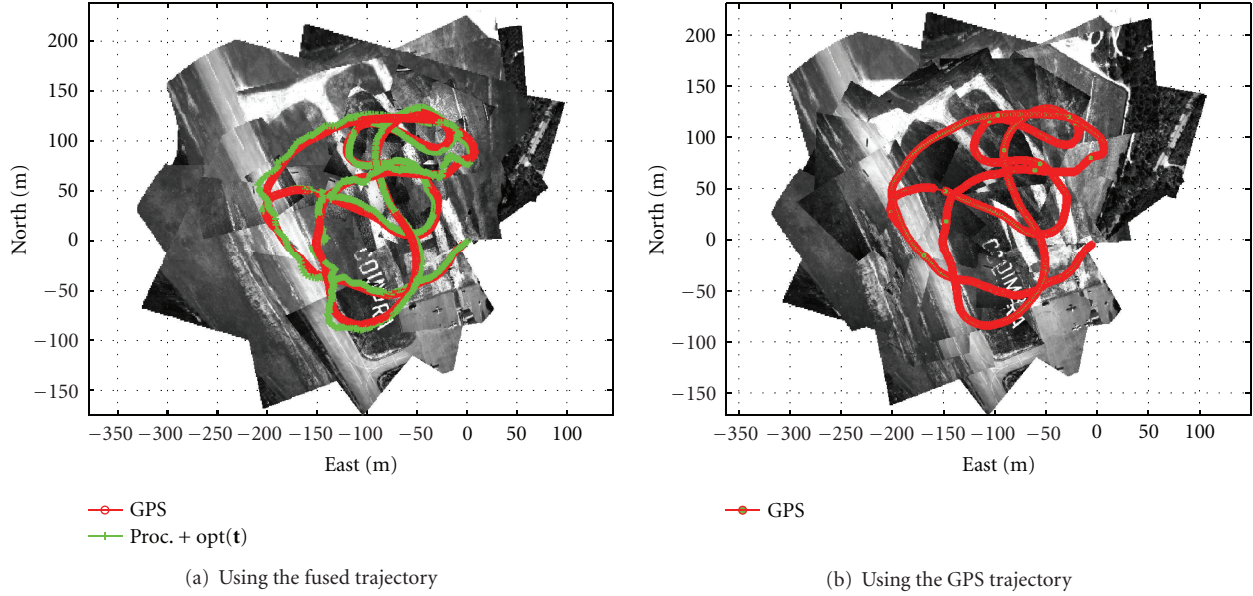


FIGURE 13: Data from GPS and visual odometry fused by a Kalman filter. The images are projected on the ground plane, forming a map.

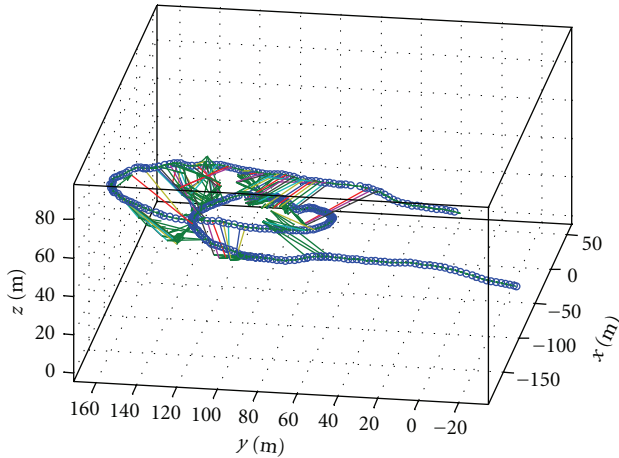


FIGURE 14: The airship trajectory with the loop-closing constraints indicated by the arrows. The lines represent the error which is minimized.

component was set to twice the relative value found in the tripod experiment (Table 1) to account for the dynamic AHRS scenario. (2) By the same reason, after each iteration each translation vector is recalculated taking into account its new height. To perform this adjustment the vector \mathbf{t}/h for each constraint is stored in the beginning, thus it is just multiplied by the new height. (3) The covariances of the loop-closing constraints were multiplied by four, decreasing their weight, to smoother the trajectory as there may be many camera poses not connected with loop-closing constraints.

Figure 15 shows the airship trajectory recovered by the visual odometry before and after the correction by graph optimization. The GPS trajectory is also provided for comparison. A few wrong loop-closing constraints distorted

the north-east side of both loops, but the next constraints corrected the trajectory again, approximating it to the true trajectory. The visual odometry drift after the loops was largely decreased. Table 4 compares both trajectories with GPS. It took 0.17 seconds to execute 100 iterations of the graph optimization for the whole graph.

5. Conclusion

In this paper, a pure translation model is used to recover the UAV trajectory using AHRS orientation estimates and aerial images of the horizontal ground plane. This model is also compared with the more common homography model against ground truth, by determining the relative heights in the tripod experiment. The GPS position fixes and the vehicle poses recovered by visual odometry were used to project the images on the ground plane, and the resulting map is more coherently registered in the short term if the airship poses are obtained from visual odometry. GPS and visual odometry data were also fused by a Kalman Filter, and the map generated with the fused trajectory is more accurate than the map generated with the GPS alone, even if the trajectory is too large to be recovered by visual odometry alone. Finally, a graph optimization SLAM package was used to minimize the drift in visual odometry.

In our previous work, using an older and less accurate AHRS, a projective model and an optimization step were used to improve the recovered trajectory [34]. Analyzing these old results, it was determined that the projective model achieved better results due to an indirect error checking issue. At some images, with too large orientation error, the optimization diverged and therefore the Kalman Filter ignored the measurement and executed only its prediction step. When direct registration was used, no numeric divergence is possible and a wrong measurement was accepted,

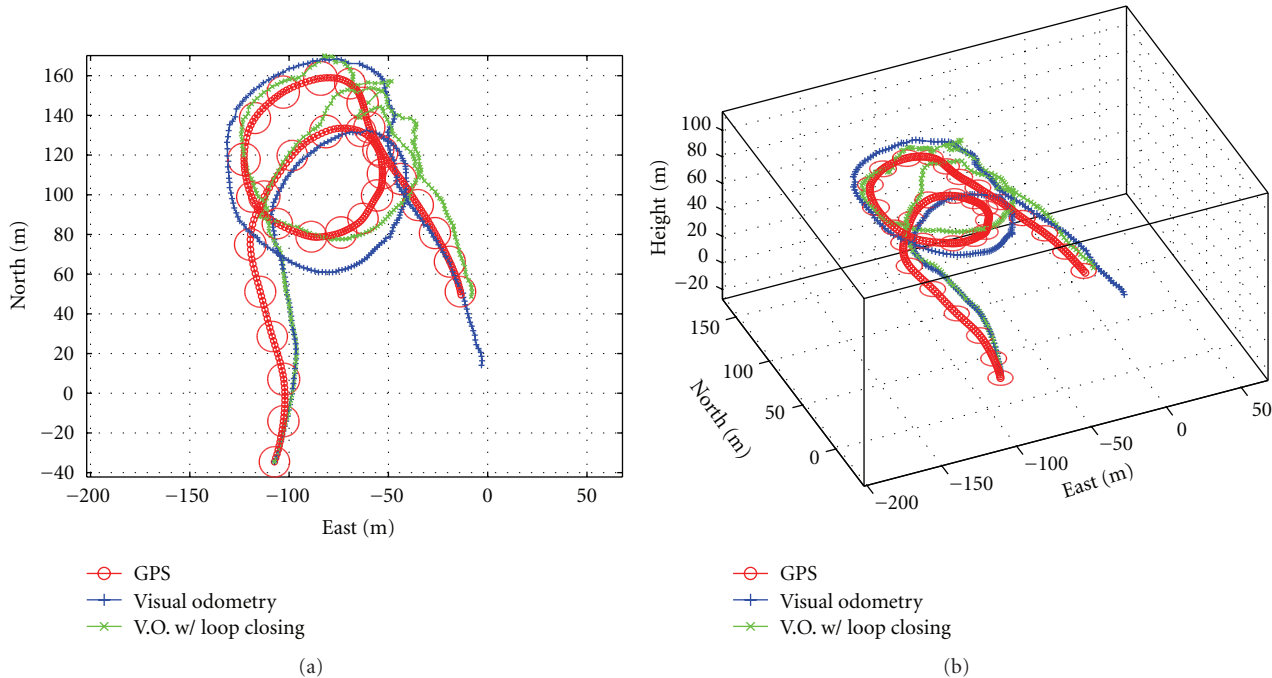


FIGURE 15: The airship trajectory recovered by visual odometry before and after applying the graph optimization package.

TABLE 4: Comparison of visual odometry before and after loop-closing, with the GPS as reference. All values are given in meters.

Unit: meters	3D position error			2D position error			Error in the length of \mathbf{t}		
	Avg.	Max.	Final	Avg.	Max.	Final	RMS	Avg.	Max.
Visual odometry: Procrustes only	16.5	43.6	43.6	15.3	43.3	43.3	0.73	-0.46	0.59
Visual odometry: after loop-closing	13.5	25.2	5.8	11.7	23.5	5.2	1.08	-0.57	1.1

increasing the drift too much. Now, given the current results, with a more accurate AHRS and a larger image framerate, the orientation estimates appear to be accurate enough to reliably apply the noniterative Procrustes registration (which theoretically should yield the optimum solution if all assumptions were true) allowing for much faster computation.

The pure translation model has performed better in the recovery of the vertical motion component than the image only approach. The vertical component is more critical because errors in the height estimation propagate not only in the vertical axis, but also as an error in the scale of the horizontal components. The effect of error accumulation in vertical motion is visible in Figure 7. The trajectories recovered by the two forms of visual odometry reproduce approximately the curves of the loop, but have errors in scale or shape due to errors in height estimation.

As the GPS uncertainty is larger in the altitude axis than in the horizontal axes, height estimation is again more critical, particularly for low-altitude flights where the GPS uncertainty is more significant due to the small distances involved. Moreover, while in high-altitude flights the ground plane can often be safely assumed as horizontal, during land-

ing and taking off the restriction to horizontal ground planes is more likely to be actually satisfied. Additionally, reasonable pose estimates were obtained using only the orientation estimates directly output by the relatively inexpensive and inaccurate AHRS utilized, and under relatively large roll and pitch variations.

Scale ambiguity is inherent to relative pose retrieval from images of planes, when using homographies [16] or even under pure translation motion. This fact increases the unavoidable drift of the visual odometry. Moreover, the 3D graph optimization has to take into account that the magnitude of the translation between two camera poses depends on the camera height, and update the constraints accordingly when a camera pose is updated, otherwise it cannot correct the trajectory length. Trajectories of hundred of meters with significant vertical motion were recovered, but the actual limit of the technique depends on the reliability of the AHRS orientation estimates and of the detection of loop-closing constraints under real flight conditions. The experimental and simulated results indicate that a modern AHRS can estimate the orientation with enough accuracy to obtain relatively accurate translation estimates.

Future work can explore more elaborate error models, for example, estimating an uncertainty for each pixel coordinate by propagating the uncertainty in the camera 6D pose. Procrustes problem variants with diverse uncertainty models have been solved, although for some cases there are only noniterative solutions [1]. Moreover, other Procrustes variants or the homology projective model used previously [34] could extend the approach to nonhorizontal ground planes. Other methods could be employed to detect loop-closing constraints more reliably. For example, databases of interest point descriptors can be used to retrieve previous images of the same area even if the estimated trajectory has drifted too far away from the true position.

The Procrustes registration and Kalman Filter are implemented as fast algebraic routines; the graph optimization could be executed in parallel to update the camera poses after a loop is detected, or may be intercalated by executing a few iterations after each image frame is processed. These operations could be implemented to be executed in real time. Therefore the only remaining time consuming operation is interest point detection and matching, including the detection of loop-closure and outlier detection. It has been shown that the reprojected images can be used to facilitate and accelerate the matching process, although other approaches to image matching, including featureless approaches, could be explored in future work.

Appendices

A. The Discrete Wiener Process Acceleration Model for the Kalman Filter

The Kalman Filters used to filter the airship trajectory and to fuse the visual odometry with GPS position fixes utilize the discrete Wiener process acceleration model [31]. The process noise, which is assumed to be a zero-mean white noise sequence, represents the acceleration increment during the sample period k . The filter state \mathbf{X} consists in the airship position, velocity, and acceleration:

$$\text{(filter state) } \mathbf{X} = \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \\ \ddot{\mathbf{x}} \end{bmatrix}, \quad (\text{A.1})$$

where \mathbf{x} is the 3D airship position. The state equation is

$$\text{(state equation) } \mathbf{X}(k+1) = \mathbf{F}\mathbf{X}(k) + \mathbf{\Gamma}v(k), \quad (\text{A.2})$$

where, with \mathbf{I} and $\mathbf{0}$ representing the identity and zero matrix of appropriate size, and T representing the length of the sample period ($T = 0.2$ seconds in the experiments reported in this paper):

$$\mathbf{F} = \begin{bmatrix} \mathbf{I} & T\mathbf{I} & \frac{T^2}{2}\mathbf{I} \\ \mathbf{0} & \mathbf{I} & T\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad (\text{A.3})$$

$$\mathbf{\Gamma} = \begin{bmatrix} \frac{T^2}{2}\mathbf{I} \\ T\mathbf{I} \\ \mathbf{I} \end{bmatrix}.$$

Therefore the prediction step is performed by the following equations:

(prediction of state)

$$\mathbf{X}(k+1 | k) = \mathbf{F}\mathbf{X}(k | k),$$

(prediction of covariance)

$$\mathbf{P}(k+1 | k) = \mathbf{F}\mathbf{P}(k | k)\mathbf{F}' + \mathbf{Q},$$

$$\mathbf{Q} = \begin{bmatrix} \frac{T^4}{4}\mathbf{I} & \frac{T^3}{2}\mathbf{I} & \frac{T^2}{2}\mathbf{I} \\ \frac{T^3}{2}\mathbf{I} & T^2\mathbf{I} & T\mathbf{I} \\ \frac{T^2}{2}\mathbf{I} & T\mathbf{I} & \mathbf{I} \end{bmatrix} \cdot \sigma_v^2. \quad (\text{A.4})$$

The value of σ_v should be of the order of magnitude of the maximum acceleration increment over the sample period ($\sigma_v = 0.35 \text{ m/s}^2$ in this paper). The update equations are the standard Kalman Filter equations.

B. The Procrustes Procedure

The similarity Procrustes problem, named after a greek mythological character, consists in finding a transformation to register two sets of points in an Euclidean space, with known point correspondences. More formally, the problem which is considered in this paper is that given two sets of n points in \mathbb{R}^d , in the form of $n \times d$ matrices \mathbf{X} and \mathbf{Y} , where the i th line in both matrices corresponds to the same point \mathcal{P}_i , the transformation parameters s , \mathbf{t} , and \mathbf{R} such that $\mathbf{Y} = s\mathbf{X}\mathbf{R} + \mathbf{1}\mathbf{t}^T$ must be determined. The notation $\mathbf{1}$ represents a vector of ones.

The derivation and proof of the solution can be found in many places including [1, 35]. The latter reference offers an extensive treatment of many variations of the problem. The steps to calculate the transformation are the following.

- (1) Compute $\mathbf{C} = \mathbf{X}^T\mathbf{J}\mathbf{Y}$, where \mathbf{J} is the matrix $\mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}^T$.
- (2) Compute the SVD of \mathbf{C} , and obtain the following matrices in the form $\mathbf{C} = \mathbf{P}\mathbf{\Phi}\mathbf{Q}^T$.
- (3) The optimal reflection/rotation matrix is $\mathbf{R} = \mathbf{Q}\mathbf{P}^T$.
- (4) The optimal scaling factor is $s = \frac{\text{tr}(\mathbf{X}^T\mathbf{J}\mathbf{Y})}{\text{tr}(\mathbf{Y}^T\mathbf{J}\mathbf{Y})}$.
- (5) The optimal translation vector when expressed in the frame of \mathbf{X} is $\mathbf{t} = n^{-1}(\mathbf{X} - s\mathbf{Y}\mathbf{R})^T\mathbf{1}$.

Acknowledgments

This work was supported by the Portuguese Foundation for Science and Technology, Grant BD/19209/2004, by EC project BACS (FP6-IST-027140), and by Air Liquide Portugal, which supplied helium gas for the field experiments.

References

- [1] J. C. Gower and G. B. Dijkstra, *Procrustes Problems*, Oxford Statistical Science Series, Oxford University Press, Oxford, UK, 2004.
- [2] J. Lobo and J. Dias, "Relative pose calibration between visual and inertial sensors," *International Journal of Robotics Research*, vol. 26, no. 6, pp. 561–575, 2007.
- [3] J. Lobo, InerVis Toolbox for Matlab, 2006, http://www2.deec.uc.pt/~jlobo/InerVis_WebIndex.
- [4] J. Lobo and J. Dias, "Vision and inertial sensor cooperation using gravity as a vertical reference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1597–1608, 2003.
- [5] L. G. B. Mirisola, J. Lobo, and J. Dias, "Stereo vision 3D map registration for airships using vision-inertial sensing," in *Proceedings of the 12th IASTED International Conference on Robotics and Applications (RA '06)*, pp. 102–107, Honolulu, Hawaii, USA, August 2006.
- [6] J. Lobo, J. F. Ferreira, and J. Dias, "Bioinspired visuo-vestibular artificial perception system for independent motion segmentation," in *Proceedings of the 2nd International Cognitive Vision Workshop (ICVW '06)*, Graz, Austria, May 2006.
- [7] E. Hygounenc, I.-K. Jung, P. Soueres, and S. Lacroix, "The autonomous blimp project at LAAS/CNRS: achievements in flight control and terrain mapping," *International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 473–512, 2004.
- [8] J. Kelly, S. Saripalli, and G. S. Sukhatme, "Combined visual and inertial navigation for an unmanned aerial vehicle," in *Proceedings of the 6th International Conference on Field and Service Robotics (FSR '07)*, Chamonix, France, July 2007.
- [9] N. R. E. Gracias, *Mosaic-based visual navigation for autonomous underwater vehicles*, Ph.D. thesis, Universidade Técnica de Lisboa-Instituto Superior Técnico, Lisbon, Portugal, December 2002.
- [10] A. Brown and D. Sullivan, "Precision kinematic alignment using a low-cost GPS/INS system," in *Proceedings of the ION GPS*, Portland, Ore, USA, September 2002.
- [11] R. M. Eustice, *Large-area visually augmented navigation for autonomous underwater vehicles*, Ph.D. thesis, Massachusetts Institute of Technology/Woods Hole Oceanographic Institution—Joint Program in Applied Ocean Science & Engineering, June 2005.
- [12] J. Kim, *Autonomous navigation for airborne applications*, Ph.D. thesis, The University of Sydney, Sydney, Australia, May 2004.
- [13] D. Strelow and S. Singh, "Motion estimation from image and inertial measurements," *International Journal of Robotics Research*, vol. 23, no. 12, pp. 1157–1195, 2004.
- [14] G. Conte and P. Doherty, "An integrated UAV navigation system based on aerial image matching," in *Proceedings of the IEEE Aerospace Conference (AC '08)*, Big Sky, Mont, USA, March 2008.
- [15] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "Improving vision-based planar motion estimation for unmanned aerial vehicles through online mosaicing," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06)*, pp. 2860–2865, Orlando, Fla, USA, May 2006.
- [16] L. Merino, J. Wiklund, F. Caballero, et al., "Vision-based multi-UAV position estimation," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 53–62, 2006.
- [17] H. Bay, T. Tuytelaars, and L. van Gool, "SURF: speeded up robust features," in *Proceedings of the 9th European Conference on Computer Vision (ECCV '06)*, vol. 3951 of *Lecture Notes in Computer Science*, pp. 404–417, Graz, Austria, May 2006.
- [18] E. Carneiro de Paiva, J. R. Azinheira, J. G. Ramos Jr., A. Moutinho, and S. S. Bueno, "Project AURORA: infrastructure and flight control experiments for a robotic airship," *Journal of Field Robotics*, vol. 23, no. 3-4, pp. 201–222, 2006.
- [19] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3D Vision*, Springer, New York, NY, USA, 2004.
- [20] E. Michaelsen, M. Kirchhof, and U. Stilla, "Sensor pose inference from airborne videos by decomposing homography estimates," in *Proceedings of the 20th of the International Society for Photogrammetry and Remote Sensing Congress*, Istanbul, Turkey, July 2004.
- [21] J. Bouguet, "Camera Calibration Toolbox for Matlab," 2006, http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.
- [22] XSens Tech, 2007, <http://www.xsens.com/>.
- [23] Point Grey Inc., 2007, <http://www.ptgrey.com/>.
- [24] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2000.
- [25] L. G. B. Mirisola and J. Dias, "Exploiting inertial sensing in mosaicing and visual navigation," in *Proceedings of the 6th IFAC Symposium on Intelligent Autonomous Vehicles (IAV '07)*, Toulouse, France, September 2007.
- [26] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, 2004.
- [27] F. Caballero, L. Merino, J. Ferraz, and A. Ollero, "Homography based kalman filter for mosaic building. Applications to UAV position estimation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 2004–2009, Rome, Italy, April 2007.
- [28] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.
- [29] K. Mikolajczyk, T. Tuytelaars, C. Schmid, et al., "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, 2005.
- [30] L. G. B. Mirisola and J. Dias, Datasets and Complementary results to "Exploiting Attitude sensing in vision based navigation for an Airship", 2008, <http://paloma.isr.uc.pt/~lgm>.
- [31] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, New York, NY, USA, 2001.
- [32] J. Arnsperg, K. Henriksen, and F. Bergholm, "Relating scene depth to image ratios," in *Proceedings of the 8th International Conference on Computer Analysis of Images and Patterns (CAIP '99)*, pp. 516–525, Ljubljana, Slovenia, September 1999.
- [33] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard, "Efficient estimation of accurate maximum likelihood maps in 3D," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 3472–3478, San Diego, Calif, USA, October-November 2007.

- [34] L. G. B. Mirisola, J. Dias, and A. T. De Almeida, "Trajectory recovery and 3D mapping from rotation-compensated imagery for an airship," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 1908–1913, San Diego, Calif, USA, October–November 2007.
- [35] I. Borg and P. Groenen, *Modern Multidimensional Scaling, Theory and Application*, Springer, New York, NY, USA, 1997.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

