HAL
open science

Vision-Aided Inertial Navigation : low computational complexity algorithms with applications to Micro Aerial Vehicles

Chiara Troiani

To cite this version:

Chiara Troiani. Vision-Aided Inertial Navigation : low computational complexity algorithms with applications to Micro Aerial Vehicles. Robotics [cs.RO]. Université de Grenoble, 2014. English. NNT : 2014GRENM021 . tel-01548441

HAL Id: tel-01548441

https://theses.hal.science/tel-01548441v1

Submitted on 27 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques-informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Chiara TROIANI

Thèse dirigée par **Christian Laugier**
codirigée par **Agostino Martinelli**

préparée au sein du centre de recherche **INRIA Rhône-Alpes,**
du **Laboratoire d'Informatique de Grenoble**
dans **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Vision-Aided Inertial Navigation : low computational complexity algorithms with applications to Micro Aerial Vehicles

Thèse soutenue publiquement le **17 Mars 2014,**
devant le jury composé de :

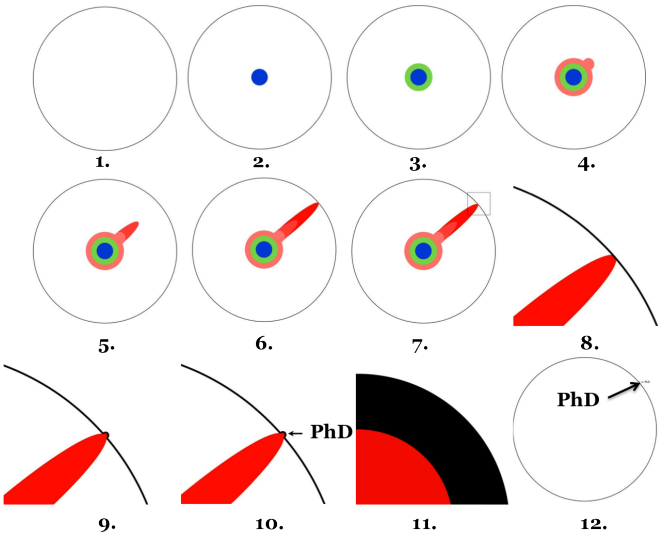
Simon LACROIX
Directeur de Recherche LAAS/CNRS, Toulouse, France, Rapporteur
Gianluca ANTONELLI
Professeur Universitá degli Studi di Cassino, Italie, Rapporteur
Christian LAUGIER
Directeur de Recherche INRIA Rhône-Alpes, Grenoble, France, Directeur de thèse
Agostino MARTINELLI
Chargé de Recherche INRIA Rhône-Alpes, Grenoble, France, Co-Directeur de thèse
Davide SCARAMUZZA
Professeur University of Zurich, Suisse, Examinateur (et Co-Encadrant)
Fabien BLANC-PAQUES
CTO Delta Drone, Grenoble, France, Examinateur

Université Joseph Fourier / Université Pierre Mendès France /
Université Stendhal / Université de Savoie / Grenoble INP

To my Mum and Dad,
because without their love and support
nothing would have been possible.

Acknowledgements

Matt Might, a Professor of Computer Science at University of Utah, described what a doctorate is in the following way:



1. Imagine a circle that contains all of human knowledge;
2. By the time you finish elementary school, you know a little;
3. By the time you finish high school, you know a bit more;
4. With a bachelor's degree, you gain a specialty;
5. A master's degree deepens that specialty;
6. Reading research papers takes you to the edge of human knowledge;
7. Once you're at the boundary, you focus;
8. You push at the boundary for a few years;
9. Until one day, the boundary gives way;
10. And, that dent you've made is called a PhD;
11. Of course, the world looks different to you now;
12. So, don't forget the bigger picture:
13. Keep pushing.

Going through a PhD is a long travel which passes over all those steps, each transition was an important moment, and in each of those moments I had the inestimable chance to be surrounded by people

who supported me, who coached me, who trusted me, who helped me. Now, sorry if those couple of pages will not have a "formal shape", but this is for me the best moment of my three years of PhD: the moment in which I can finally thank all the people who made this possible.

I would like to start thanking my supervisor, Agostino Martinelli, for the support, the patience, the advices, the interesting discussions, his epic stories and for allowing me to defend my thesis even if I didn't go yet on the top of the "Mont Blanc". Thanks to my co-supervisor Davide Scaramuzza for hosting me 8 months in his lab in Zurich, for all the discussions and the feedbacks and thanks because the 8 months spent at the AI Lab gave me also the big gift of a dear friend. Thanks to my thesis director Christian Laugier, for his guidance and for being an example of leader to follow under not only a working but also a human point of view. Thanks to Simon Lacroix and Gianluca Antonelli who accepted to review my thesis and to Fabien Blanc-Paques who allowed me to collaborate with his company and accepted to be member of my commitee.

Thanks to e-motion, because e-motion is not only an Inria's team, it is my Team, it is family, friends, work and amusement. Thanks to Myriam, because without Myriam there would not exist any e-motion, because each team needs its Myriam, but everybody who leaves e-motion realizes that there exist only one Myriam: ours! Thanks to SG (or "services généreux", as I have always called them, Gerard, Alain, Ben), because they are simply extraordinary. Thanks to Stephanie for... well, there would be too much to write, let's simply say: thanks for being always present even miles far away. Thanks to Alex and Matina for all the moments we spent together, the cooking, the hikings, the movies. Thanks to Jorge, because he survived "dans le bureau des filles" without going crazy and because now that this office became again "le bureau des filles", we really feel that there is someone missing. To Karla for all the times she knocked at my door with her delicious Mexican dishes, to Lota: my favourite helicopter pilot, to Dizan, Barbara, Manu, Amaury, Stephanie, Alessandro, Arturo, Procopio, Mike, Anne, Lukas, Juan, Juan David and more generally to all the e-motion team for all the moments shared, for all the coffee-breaks and especially for the support all of you gave me from the beginning until the end of this experience!

Thanks to the sFly team: I learned a lot from all of you! Thanks to my officemates in Zurich, Flavio, Mathias, Elias, Matia and Christian

for making my staying in Zurich unforgettable from a lot of points of view. Thanks to Volker, the officemate who each one would love to have in his office. Thanks to Daniele: “il piú bel regalo zurighese” and to the people of “anche a Zurigo un angolo di cielo puó essere rotondo” (Spalla, Bocci, Pippo, Prest, Spampi, Ale).

Thanks to my Professor, Costanzo Manes, for transmitting me his passion for robotics, for encouraging me to apply for a PhD and for recommending me to come here. Thanks to Paolo, for the constant on-line support and for the “insults” when I brake the batteries of my helicopter.

Thanks to my Family who gave me the stability and calm necessary to grew up and became what I am today. To my Dad, “sperando che un giorno riusciró ad assomigliargli almeno un po’ ”, and to my Mum, because “le somiglio giá tantissimo e ne sono fiera!”. To my sister because even if she will always be “la mia piccolina”, I had and I will always have a lot to learn from her. Thanks to Matteo, because he has always been and I know he will always be present. Thanks to Andrea for his “sixth sense” about knowing whenever I may need something and because “se c’é un fiocco é grazie a lui!”. Thanks to Nicole and Claude, because all the latest news I got about drones were coming from them and because “c’est beau de pouvoir se sentir en famille même à mille kilometres de chez moi”.

And last but absolutely not least: thanks to Mathias for being at my side from the beginning to the end of this experience acquiring a constantly growing role, for all his precious advices and his patience but especially for giving to this thesis a special additional significance.

Abstract

Accurate egomotion estimation is of utmost importance for any navigation system. Nowadays different sensors are adopted to localize and navigate in unknown environments such as GPS, range sensors, cameras, magnetic field sensors, inertial sensors (IMU). In order to have a robust egomotion estimation, the information of multiple sensors is fused. Although the improvements of technology in providing more accurate sensors, and the efforts of the mobile robotics community in the development of more performant navigation algorithms, there are still open challenges. Furthermore, the growing interest of the robotics community in micro robots and swarm of robots pushes towards the employment of low weight, low cost sensors and low computational complexity algorithms. In this context inertial sensors and monocular cameras, thanks to their complementary characteristics, low weight, low cost and widespread use, represent an interesting sensor suite.

This dissertation represents a contribution in the framework of vision-aided inertial navigation and tackles the problems of data association and pose estimation aiming for low computational complexity algorithms applied to MAVs.

For what concerns the data association, a novel method to estimate the relative motion between two consecutive camera views is proposed. It only requires the observation of a single feature in the scene and the knowledge of the angular rates from an IMU, under the assumption that the local camera motion lies in a plane perpendicular to the gravity vector. Two very efficient algorithms to remove the outliers of the feature-matching process are provided under the above-mentioned motion assumption. In order to generalize the approach to a 6DoF motion, two feature correspondences and gyroscopic data from IMU measurements are necessary. In this case, two algorithms are provided to remove wrong data associations in the feature-matching process. In the case of a monocular camera mounted on a quadrotor vehicle, motion priors from IMU are used to discard wrong estimations.

For what concerns the pose estimation problem, this thesis provides a closed form solution which gives the system pose from three natural features observed in a single camera image, once the roll and the pitch angles are obtained by

the inertial measurements under the planar ground assumption. Specifically, the system position and attitude can uniquely be determined by observing two point features but improved by exploiting the geometric constraints inherent to a virtual pattern formed by the three features.

In order to tackle the pose estimation problem in dark or featureless environments, a system equipped with a monocular camera, inertial sensors and a laser pointer is considered. The system moves in the surrounding of a planar surface and the laser pointer produces a laser spot on the abovementioned surface. The laser spot is observed by the monocular camera and represents the only point feature considered. Through an observability analysis it is demonstrated that the physical quantities which can be determined by exploiting the measurements provided by the aforementioned sensor suite during a short time interval are: the distance of the system from the planar surface; the component of the system speed that is orthogonal to the planar surface; the relative orientation of the system with respect to the planar surface; the orientation of the planar surface with respect to the gravity. A simple recursive method to perform the estimation of all the aforementioned observable quantities is provided. This method is based on a local decomposition of the original system, which separates the observable modes from the rest of the system.

All the contributions of this thesis are validated through experimental results using both simulated and real data. Thanks to their low computational complexity, the proposed algorithms are very suitable for real time implementation on systems with limited on-board computation resources. The considered sensor suite is mounted on a quadrotor vehicle but the contributions of this dissertations can be applied to any mobile device.

Résumé

L'estimation précise du mouvement 3D d'une caméra relativement à une scène rigide est essentielle pour tous les systèmes de navigation visuels. Aujourd'hui différents types de capteurs sont adoptés pour se localiser et naviguer dans des environnements inconnus : GPS, capteurs de distance, caméras, capteurs magnétiques, centrales inertielles (IMU, Inertial Measurement Unit). Afin d'avoir une estimation robuste, les mesures de plusieurs capteurs sont fusionnées. Même si le progrès technologique permet d'avoir des capteurs de plus en plus précis, et si la communauté de robotique mobile développe algorithmes de navigation de plus en plus performantes, il y a encore des défis ouverts. De plus, l'intérêt croissant des la communauté de robotique pour les micro robots et essaim de robots pousse vers l'emploi de capteurs à bas poids, bas coût et vers l'étude d'algorithmes à faible complexité. Dans ce contexte, capteurs inertiels et caméras monoculaires, grâce à leurs caractéristiques complémentaires, faible poids, bas coût et utilisation généralisée, représentent une combinaison de capteur intéressante.

Cette thèse présente une contribution dans le cadre de la navigation inertielle assistée par vision et aborde les problèmes de fusion de données et estimation de la pose, en visant des algorithmes à faible complexité appliqués à des micro-véhicules aériens.

Pour ce qui concerne l'association de données, une nouvelle méthode pour estimer le mouvement relatif entre deux vues de caméra consécutifs est proposée. Celle-ci ne nécessite l'observation que d'un seul point caractéristique de la scène et la connaissance des vitesses angulaires fournies par la centrale inertielle, sous l'hypothèse que le mouvement de la caméra appartient localement à un plan perpendiculaire à la direction de la gravité. Deux algorithmes très efficaces pour éliminer les fausses associations de données (outliers) sont proposés sur la base de cette hypothèse de mouvement. Afin de généraliser l'approche pour des mouvements à six degrés de liberté, deux points caracteristiques et les données gyroscopiques correspondantes sont nécessaires. Dans ce cas, deux algorithmes sont proposés pour éliminer les outliers. Nous montrons que dans le cas d'une caméra monoculaire montée sur un quadrotor, les informations de mouvement fournies par l'IMU peuvent être utilisées pour éliminer de mauvaises estimations.

Pour ce qui concerne le problème d'estimation de la pose, cette thèse fournit une solution analytique pour exprimer la pose du système à partir de l'observation de trois points caractéristiques naturels dans une seule image, une fois que le roulis et le tangage sont obtenus par les données inertielles sous l'hypothèse de terrain plan. Plus spécifiquement, la position et l'attitude du système peuvent être uniquement déterminées à partir de l'observation de deux points caractéristiques, mais améliorées en exploitant les contraintes géométriques inhérentes à un pattern virtuel formé par les trois points caractéristiques.

Afin d'aborder le problème d'estimation de la pose dans des environnements sombres ou manquant de points caractéristiques, un système équipé d'une caméra monoculaire, d'une centrale inertielle et d'un pointeur laser est considéré. Le système se déplace dans l'entourant d'une surface plane et le pointeur laser produit un petit point sur la surface. Le point laser est observé par la caméra monoculaire et représente le seul point caractéristique considéré. Grâce à une analyse d'observabilité il est démontré que les grandeurs physiques qui peuvent être déterminées par l'exploitation des mesures fourni par ce systeme de capteurs pendant un court intervalle de temps sont : la distance entre le système et la surface plane ; la composante de la vitesse du systme qui est orthogonale à la surface ; l'orientation relative du système par rapport à la surface et l'orientation de la surface par rapport à la gravité. Une méthode récursive simple a été proposée pour l'estimation de toutes ces quantités observables. Cette méthode est basée sur une décomposition locale du système d'origine, qui sépare les modes observables du reste du système.

Toutes les contributions de cette thèse sont validées par des expérimentations à l'aide des données réelles et simulées. Grace à leur faible complexité de calcul, les algorithmes proposés sont très appropriés pour la mise en oeuvre en temps réel sur des systèmes ayant des ressources de calcul limitées. La suite de capteur considérée est monté sur un quadrotor, mais les contributions de cette thèse peuvent être appliquées à n'importe quel appareil mobile.

Contents

Contents ix

List of Figures xii

1 Introduction 1
- 1.1 Context . . . 1
  - 1.1.1 Why quadrotors? . . . 2
  - 1.1.2 Brief quadrotor's history . . . 3
  - 1.1.3 Applications . . . 4
  - 1.1.4 The quadrotor concept . . . 4
- 1.2 Motivation and objectives . . . 6
  - 1.2.1 Minimalist perception . . . 9
- 1.3 Contributions . . . 10
- 1.4 Thesis outline . . . 12

2 Visual and Inertial sensors 13
- 2.1 A biological overview . . . 14
  - 2.1.1 The sense of sight . . . 14
  - 2.1.2 The perception of gravity . . . 15
- 2.2 Inertial Measurement Unit . . . 17
  - 2.2.1 Accelerometers . . . 18
  - 2.2.2 Gyroscopes . . . 19
- 2.3 Camera . . . 20
  - 2.3.1 Pinhole Camera Model . . . 21
- 2.4 Camera Calibration . . . 23
- 2.5 Camera-IMU Calibration . . . 24

3 Data association 27
- 3.1 Feature extraction and matching . . . 28
  - 3.1.1 Feature Detection . . . 28
  - 3.1.2 Feature Tracking . . . 29

3.1.3 Feature Matching 30
3.2 Outlier detection 31
3.2.1 Related works 32
3.2.2 Epipolar Geometry 33
3.2.3 1-point algorithm 34
3.2.3.1 Parametrization of the camera motion 34
3.2.3.2 1-point Ransac 37
3.2.3.3 Me-RE (Median + Reprojection Error) 37
3.2.3.4 Performance evaluation 37
3.2.3.5 Conclusions 42
3.2.4 2-point algorithm 45
3.2.4.1 Parametrization of the camera motion 45
3.2.4.2 Hough 47
3.2.4.3 2-point Ransac 48
3.2.4.4 Quadrotor motion model 49
3.2.4.5 Performance evaluation 51
3.2.4.6 Conclusions 55

4 Pose estimation **59**
4.1 Filtering approaches and closed form solutions 60
4.2 Virtual patterns 61
4.2.1 The System 62
4.2.2 The method 63
4.2.2.1 2p-Algorithm 63
4.2.2.2 3p-Algorithm 66
4.2.2.3 Scale factor initialization 67
4.2.2.4 Estimation of γ_1 and γ_2 71
4.2.3 Performance Evaluation 71
4.2.3.1 Simulations 71
4.2.3.2 Experimental Results 72
4.2.4 Conclusion 75
4.3 Virtual features 77
4.3.1 The System 78
4.3.2 Camera-laser module calibration 81
4.3.3 Observability Properties 82
4.3.4 Local Decomposition and Recursive Estimation 86
4.3.5 Performance Evaluation 88
4.3.5.1 Simulations 89
4.3.5.2 Preliminary experiments 91
4.3.5.3 Camera-laser module calibration 95
4.3.6 Conclusion 96

5 Conclusions **97**
5.1 Research Outlook 99

Bibliography **100**

List of publications **110**

List of Figures

1.1 MAVs classification. The vehicles in the pictures are: Perching Glider, MIT (a); Festo's Smartbird (b); AeroVironments Nano Hummingbird (c); Skybotix's CoaX (d); Parrot's AR.Drone (e). . 2

1.2 History of quadrotors with respect to their application fields. 1907: Paul Corny machine (both from 1907), Bréguet Giant Gyroplane. 1924: Ohemichen's quadrotor; 2000: ETH Zurich's *OS4* [16], [15], the CEA's *X4-flyer* [37] and the ANU's *X-4 Flyer* [79]; Nowadays there exists a lot of quadrotors. We list here few exemplars: AscTec Pelican [1], KMelRobotics' NanoQuad [3], Mikrokopter's quadrotor [4], Flyduino, Arducopter, Parrot AR.Drone, DeltaDrone quadrotor [2]. 3

1.3 Quadrotors in research projects and in companies. 5

1.4 Quadrotor notation. The four rotors are depicted in blue. w_i is the angular velocity of the i-th rotor, F_i and M_i are the vertical force and the moment respectively produced by the i-th rotor. The body-vehicle's frame B is shown in black, it is attached to the vehicle, and its origin is coincident with the vehicle's center of mass. In gray it is depicted the world reference frame W. 6

1.5 The quadrotor concept. The width of the arrows is proportional to the angular speed of the propellers. 7

1.6 Properties of some sensors commonly on-board a MAV. 8

1.7 Flight Assembled Architecture (a) at FRAC Centre in Orlans, France [25]. Robot Quadrotors Perform James Bond Theme (b) [53]. Figure (c) represents a motion capture system room. Figures (a),(c) courtesy of http://www.flyingmachinearena.org. 9

1.8 This table summarises the objective of this dissertations in terms of minimalist perception. 10

2.1 Complementary properties of cameras and IMU sensors. 14

2.2 Human eye. Image courtesy of http://www.biographixmedia.com. 15

2.3 Insect eye. Image courtesy of http://www.wikipedia.org. 16

2.4 Vestibular system. Image (a) courtesy of http://www.chrcentre.com.au, image (b) courtesy of http://biology.nicerweb.com. . 17
2.5 Halteres: small knobbed structures in some two-winged insects. They are flapped rapidly and function as gyroscopes, providing informations to the insect about his body rotation during flights. . 17
2.6 IMU applications Image courtesy of http://www.unmannedsystemstechnology.com. 18
2.7 Physical principle of a single-axis accelerometer. 19
2.8 Scheme of a 3D mechanical Gyroscope. Image courtesy of http://www.wikipedia.org. 20
2.9 Diagram of a CCD (a) and CMOS (b) sensor. 21
2.10 Matrix Vision mvBlueFOX-MLC usb camera. 22
2.11 Pinhole camera model for standard perspective cameras. 22
2.12 Chessboard images used for calibration. 24
2.13 Extracted corners on a chessboard. 25
2.14 Camera and IMU observing the vertical direction. Redrawn from [57]. 25
2.15 Camera poses relative to the calibration chessboard (a). Results of rotation estimation (b). 26

3.1 Comparison of feature detectors: properties and performances [88]. 30
3.2 Surf features matched across multiple frames overlaid on the first image. 31
3.3 Number of RANSAC iterations. 32
3.4 Epipolar constraint. $\mathbf{p_1}$, $\mathbf{p_2}$, T and P lie on the same plane (the *epipolar plane*). 33
3.5 Notation. 35
3.6 C_{p1} and C_{p2} are the reference frames attached to the vehicle's body frame but which z-axis is parallel to the gravity vector. They correspond to two consecutive camera views. C_{p0} corresponds to the reference frame C_{p1} rotated according to $dYaw$. 36
3.7 Synthetic scenario. The green line represents the trajectory and the red dots represent the simulated features. 39
3.8 Number of found inliers by Me-RE (red), 1-point RANSAC (cyan), 5-point RANSAC (black), true number of inliers(blue) for a perfect planar motion. 39
3.9 Number of found inliers by Me-RE (red), 1-point RANSAC (cyan), 5-point RANSAC (black), true number of inliers(blue) in presence of perturbations on the *Roll* and *Pitch* angles. 40

3.10 Number of found inliers by Me-RE (red), 1-point RANSAC (cyan), 5-point RANSAC (black), true number of inliers(blue) in presence of perturbations on the $dYaw$ angle. 40
3.11 Number of found inliers by Me-RE (red), 1-point RANSAC (cyan), 5-point RANSAC (black), true number of inliers(blue) for a non-perfect planar motion ($s_1 = 0.02 * sin(8 * w_c \cdot t)$). 41
3.12 Nano quadrotor from KMelRobotics: a 150g and 18cm sized platform equipped with an integrated Gumstix Overo board and MatrixVision VGA camera. 41
3.13 Plot of the real trajectory. The vehicle's body frame is depicted in black and the green line is the trajectory followed. 43
3.14 Number of found inliers by Me-RE (red), 1-point RANSAC (green), 5-point RANSAC (black), 8-point RANSAC (blue) along the trajectory depicted in Figure 3.13. 43
3.15 From the top to the bottom: $Roll$, $Pitch$ and $dYaw$ angles [deg] estimated with the IMU (red) versus $Roll$, $Pitch$ and $dYaw$ angles [deg] estimated with the Optitrack system (blue). The last plot shows the height of the vehicle above the ground (non perfect planarity of motion). 44
3.16 Computation time. 45
3.17 The reference frame C_0 and C_2 differ only for the translation vector T. $\rho = |T|$ and the angles α and β allow us to express the origin of the reference frame C_2 in the reference frame C_0. 46
3.18 Hough Space in α and β computed with real data. 48
3.19 Notation. 49
3.20 Motion constraints on a quadrotor relative to its orientation. $\Delta\phi > 0$ implies a movement along Y_{B_0} positive direction, $\Delta\theta < 0$ implies a movement along Y_{B_0} positive direction. 50
3.21 Synthetic scenario. The red line represents the trajectory and the blue dots represent the simulated features. The green dots are the features in the current camera view. 52
3.22 The IMU measurements are not affected by noise (ideal conditions). 53
3.23 The angles $\Delta\phi$, $\Delta\theta$ and $\Delta\psi$ are affected by noise. 53
3.24 Only the angles $\Delta\phi$ and $\Delta\theta$ are affected by noise. 54
3.25 Only the angle $\Delta\psi$ is affected by noise. 54
3.26 Real scenario. The vehicle body frame is represented in blue, while the red line represents the followed trajectory. 56
3.27 Number of inliers detected with the Hough approach (red), the 2-point RANSAC (cyan), the 5-point RANSAC (black) and the 8-point RANSAC (blue) along the trajectory depicted in Figure 3.26. 57

3.28 Computation time. 57
3.29 Errors between the relative rotations $\Delta\phi$ (err_R), $\Delta\theta$ (err_P), $\Delta\psi$ (err_Y) estimated with the IMU and estimated with the Optitrack. 58

4.1 Global frame. Two is the minimum number of point features which allows us to uniquely define a global reference frame. P_1 is the origin, the x_G-axis is parallel to the gravity and P_2 defines the x_G-axis 63
4.2 The 2p-algorithm. 64
4.3 The three reference frames adopted in our derivation. 64
4.4 The yaw angle ($-\alpha$) is the orientation of the X_N-axis in the global frame. 65
4.5 The triangle made by the 3 point features. 66
4.6 Flow chart of the proposed pose estimator 67
4.7 Estimated x, y, z (a), and $Roll$, $Pitch$, Yaw (b). The blue line indicate the ground truth, the green one the estimation with the 2p-Algorithm and the red one the estimation with the 3p-Algorithm 73
4.8 Mean error on the estimated states in our simulations. For the position the error is given in %. 74
4.9 AscTec Pelican quadcopter [1] equipped with a monocular camera. 74
4.10 Our Pelican quadcopter: a system overview 75
4.11 Scenario: The AR Marker and the 3 balls are used only with the aim to get a rough ground truth. The AR Marker provides the camera $6DOF$ pose in a global reference frame according to our conventions. 76
4.12 Estimated position (a), respectively x, y, z and estimated attitude (b), respectively $Roll$, $Pitch$, Yaw. The red lines represent the estimated values with the 3p-Algorithm, the blue ones represent a rough ground truth (from ARToolkit Marker). 76
4.13 Quadrotor equipped with a monocular camera, IMU and a laser pointer. The laser spot is on a planar surface and its position in the camera frame is obtained by the camera up to a scale factor. . 79
4.14 The original camera frame XYZ, the chosen camera frame $X'Y'Z'$ and the laser module at the position $[L_x,\ L_y,\ 0]$ and the direction $(\theta,\ \phi)$ (in the original frame) and position $[L,\ 0,\ 0]$ and the direction $(0,\ 0)$ (in the chosen camera frame). 80

4.15 Camera-Laser module calibration steps. Figure (a) is the camera image containing the laser spot projected onto a checkerboard. In green the reference frame attached to the checkerboard. Figure (b) represents three different camera positions used during the calibration process, the green grid represents the checkerboard and the red lines represent the reference frame attached to the checkerboard. 82
4.16 The Pelican quadcopter equipped with a monocular camera and a laser module. 83
4.17 A typical vehicle trajectory in our simulations. 90
4.18 Estimated α in absence (a) and in presence of bias (b) on the inertial data. Blue dots indicate ground true values while red discs indicate the estimated values. 91
4.19 Estimated P in absence (a) and in presence of bias (b) on the inertial data. Blue dots indicate ground true values while red discs indicate the estimated values. 92
4.20 Estimated R in absence (a) and in presence of bias (b) on the inertial data. Blue dots indicate ground true values while red discs indicate the estimated values. 92
4.21 Estimated v_0 in absence (a) and in presence of bias (b) on the inertial data. Blue dots indicate ground true values while red discs indicate the estimated values. 93
4.22 Estimated d in absence (a) and in presence of bias (b) on the inertial data. Blue dots indicate ground true values while red discs indicate the estimated values. 93
4.23 Estimated α in the experiment. Blue dots indicate ground true values while red discs indicate the estimated values. 94
4.24 Estimated P (a) and R (b) in the experiment. Blue dots indicate ground true values while red discs indicate the estimated values. . 94
4.25 Estimated v_o (a) and d (b) in the experiment. Blue dots indicate ground true values while red discs indicate the estimated values. . 95
4.26 The Pelican quadcopter equipped with a monocular camera and a laser module and passive markers. 95

Chapter 1

Introduction

Contents

1.1 Context **1**

1.1.1 Why quadrotors? 2

1.1.2 Brief quadrotor's history 3

1.1.3 Applications 4

1.1.4 The quadrotor concept 4

1.2 Motivation and objectives **6**

1.2.1 Minimalist perception 9

1.3 Contributions **10**

1.4 Thesis outline **12**

1.1 Context

In recent years, flying robotics has received significant attention from the robotics community. The ability to fly allows easily avoiding obstacles and quickly having an excellent birds eye view. These navigation facilities make flying robots the ideal platform to solve many tasks like exploration, mapping, reconnaissance for search and rescue, environment monitoring, security surveillance, inspection etc. In the framework of flying robotics, micro aerial vehicles (MAV) have a further advantage. Due to the small size they can also be used in narrow out- and indoor environment and they represent only a limited risk for the environment and people living in it.

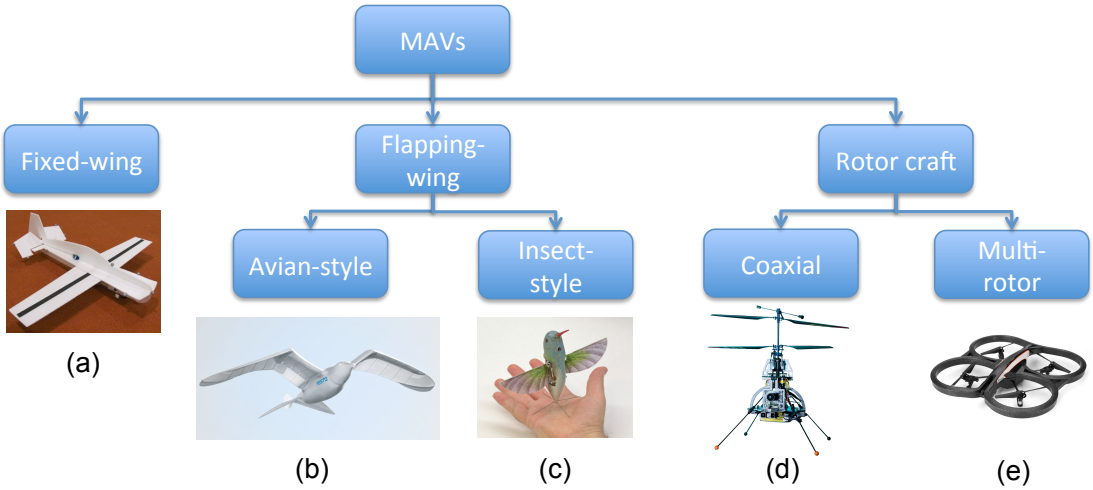


Figure 1.1: MAVs classification. The vehicles in the pictures are: Perching Glider, MIT (a); Festo's Smartbird (b); AeroVironments Nano Hummingbird (c); Skybotix's CoaX (d); Parrot's AR.Drone (e).

1.1.1 Why quadrotors?

Micro aerial vehicles can be classified into: Fixed-wing, Flapping-wing and rotorcrafts [52] (Figure 1.1). Fixed-wing vehicles are not very agile in three-dimensional complex environment. Flapping-wing vehicles can be divided into avian-style and insect-style vehicles [61]. The development of the former is strongly limited by the lack of knowledge about fluid-structure coupling and aeroelasticity. Insect-style flapping wing MAVs and rotor-craft can perform stationary flight and forward flight, which represents a significant advantage in terms of maneuverability. Nevertheless, insect-style flapping wing vehicles present an increasing complexity and it has not yet been demonstrated whether they can be considered more convenient than rotor crafts. The most common rotor craft configurations for MAVs are the Coaxial rotorcraft and the Quadrotor aircraft. The former is well represented by the Skybotix Coax [5], a vehicle equipped with two co-axial, counter-rotating rotors and a stabilizer bar and the quadrotor vehicles. Quadrotors, thanks to their fast reaction to external disturbances, light weight and low crash impact, inherent safety, compactness, simple mechanical design, easier maneuverability and controllability and ability to carry small payloads, are nowadays the best option.

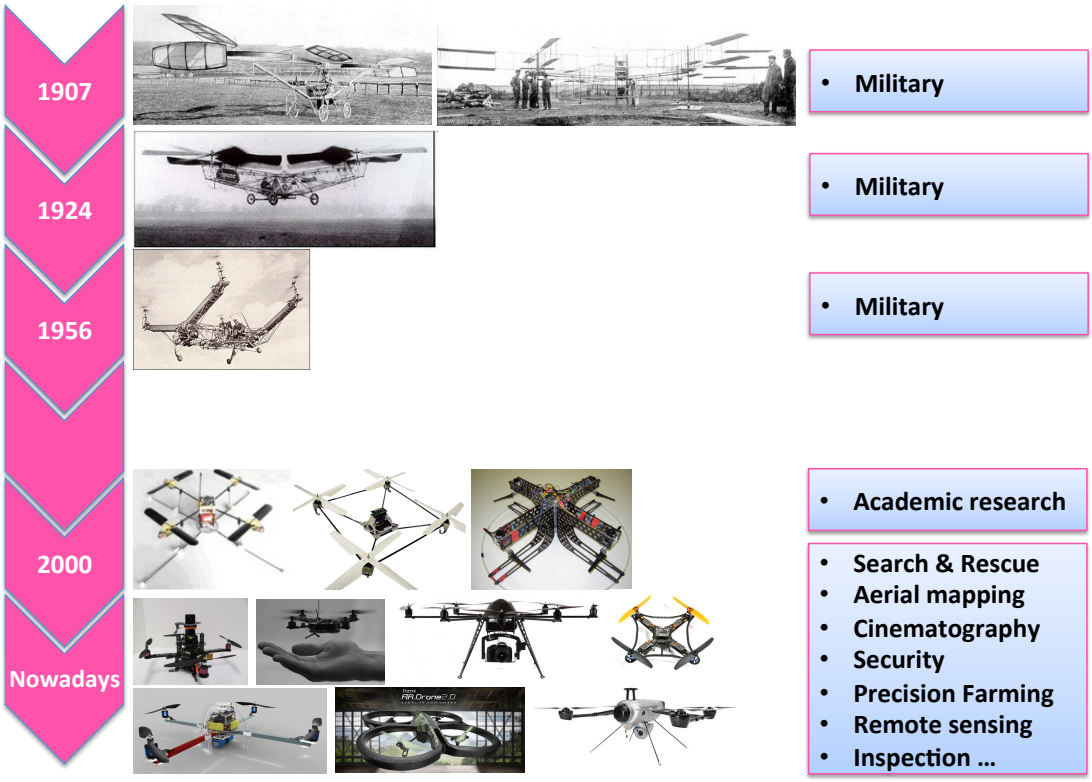


Figure 1.2: History of quadrotors with respect to their application fields. 1907: Paul Corny machine (both from 1907), Bréguet Giant Gyroplane. 1924: Ohemichen's quadrotor; 2000: ETH Zurich's *OS4* [16], [15], the CEA's *X4-flyer* [37] and the ANU's *X-4 Flyer* [79]; Nowadays there exists a lot of quadrotors. We list here few exemplars: AscTec Pelican [1], KMelRobotics' NanoQuad [3], Mikrokopter's quadrotor [4], Flyduino, Arducopter, Parrot AR.Drone, DeltaDrone quadrotor [2].

1.1.2 Brief quadrotor's history

At the beginning of the 20th century the French scientist and academician Charles Richet built a small, unmanned helicopter. Although the vehicle was not successful, it inspired two of his students, Louis and Jaques Bréguet, which in 1907, under his supervision, built the first quad-rotor (Figure 1.2) [55]. At the same time, the French engineer Paul Corny, designed an other quadrotor vehicle (Figure 1.2). Both the machines were reported to have carried a pilot off the ground but both of them lacked in stability and didn't have a proper control architecture.

In 1920, Étienne Ohemichen, an employee of the French Peugeot car company, built a quadrotor machine, with eight additional rotors for control and propulsion.

Nevertheless, his machine was underpowered and needed a hydrogen balloon for the stabilization. After various attempts, in 1924 Ohemichen demonstrated that a vertical flight machine could be stable and somehow maneuverable, and his vehicle was considered the earliest mention of a complete quadrotor hovering vehicle in history (Figure 1.2).

In 1956, Marc Adam Kaplan designed the most successful of the early designs of the rotor-craft (the Convertawings Model A quadrotor). With this machine Kaplan proved the quadrotor concept and realized the first four-rotor helicopter able to perform successful forward flight. This helicopter was intended to be the prototype for a line of much larger civil and military quadrotor helicopters, but due to a lack of orders for commercial or military versions, the project was terminated.

At the beginning of the third millennium, quadrotors drew the attention of academic researchers in order to address the problems faced by small-scale UAVs. The ETH Zurich's *OS4* [16], [15], the CEA's *X4-flyer* [37] and the ANU's *X-4 Flyer* [79] represents the first quadrotor research platforms.

1.1.3 Applications

Estimates from UAV Market Research (2011) reveals that the UAV market is estimated to exceed US$60 bilion in the next three years, and this forecast does not take into account for the thousands of MAVs already fielded.

As illustrated in Section 1.1.2, the development of quadrotors has been boosted by the military (Figure 1.2). Nowadays, considering the high number and variety of research projects involving quadrotors and of companies already selling ready-to-flight products, their application field is widely extended (Figure 1.3). Quadrotor vehicles result suitable for search and rescue operations [69], [84], powerline inspection [39], [81], and building inspection, crop dusting, precision farming, remote sensing, security related tasks, aerial mapping, aerial photography, aerial delivery and cinematography tasks.

As stated by Vijay Kumar in [52]: "*While fixed-base industrial robots were the main focus in the first two decades of robotics, and mobile robots enabled most of the significant advances during the next two decades, it is likely that UAVs, and particularly micro-UAVs, will provide a major impetus for the next phase of education, research, and development*".

1.1.4 The quadrotor concept

Quadrotors are VTOL (Vertical Take-Off and Landing) MAVs, lifted and propelled by four rotors in cross configuration (Figure 1.4). They present two pairs of identical fixed-pitch propellers, driven in opposite direction (two clockwise and

	Name	Details	Applications	Website
Research projects	sFly	Swarm of Micro Flying Robots.	Search and Rescue.	www.sfly.org
	AIRobots	Innovative Aerial Service Robots for remote inspection by contact.	Building inspection, Sample picking, Aerial remote manipulation.	airobots.ing.unibo.it
	ARCAS	Aerial Robotics Cooperative Assembly System.	Joint transportation, Precise placement and assembly.	www.arcas-project.eu
	ALCEDO	Student project at ETH Zurich.	Support in rescue of avalanche victims.	www.alcedo.ethz.ch
Companies	Ascending Technologies	Munich, Germany.	Aerial Imaging, Research platforms, Art shows (in collaboration with Ars Electronica Solutions).	www.asctec.de
	Delta Drone	Grenoble, France.	Inspection, Environmental analysis, Cartography, Search and rescue.	www.deltadrone.fr
	KMelRobotics		Aerial Imaging, Research platforms, Art shows.	kmelrobotics.com
	Microdrones	Germany.	Aerial photography, Live broadcasting, Inspection, Surveillance.	www.microdrones.com

Figure 1.3: Quadrotors in research projects and in companies.

two counter-clockwise). This allows to control lift and torque avoiding the need of a tail rotor. Control of vehicle motion is performed by varying the angular velocity of one or more propellers, thereby changing its torque load and thrust/lift characteristics (Figure 1.5). Taking-off and landing are performed by increasing or decreasing respectively the speed of the four rotors simultaneously. Rotation about the vertical-axis is obtained by augmenting the angular speed of two opposite propellers while decreasing the speed of the remaining two. Translational motion is strongly coupled to the vehicle attitude. In order to achieve a lateral movement, the quadrotor must adjust its pitch or roll by augmenting the angular speed of one rotor and decreasing the one of its diametrically opposite rotor. Due to its four actuators and its six degrees of freedom motion, the quadrotor is an under-actuated and highly dynamically unstable system.

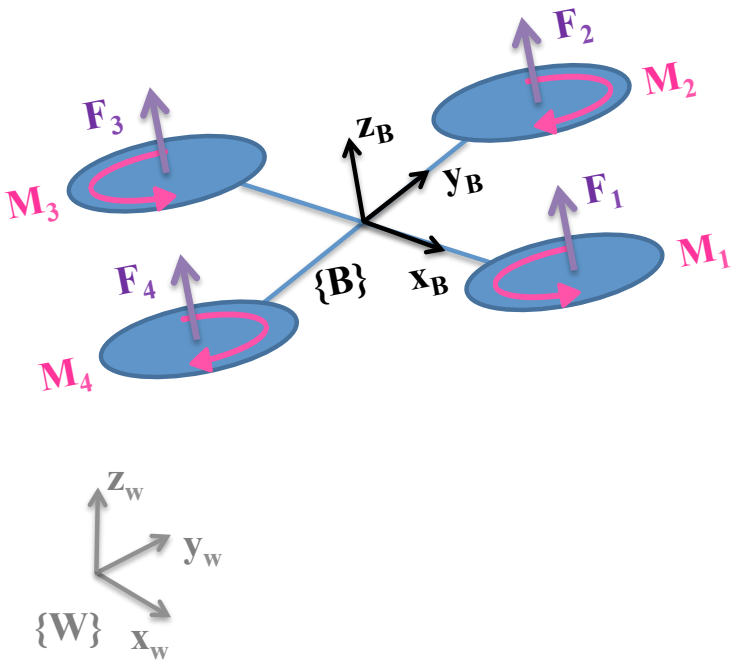


Figure 1.4: Quadrotor notation. The four rotors are depicted in blue. w_i is the angular velocity of the i-th rotor, F_i and M_i are the vertical force and the moment respectively produced by the i-th rotor. The body-vehicle's frame B is shown in black, it is attached to the vehicle, and its origin is coincident with the vehicle's center of mass. In gray it is depicted the world reference frame W.

1.2 Motivation and objectives

A crucial problem on an airborne vehicle is the stabilization and control in attitude and position, i.e. in six degrees of freedom. Most of the controlling approaches for MAVs present a cascade control structure [15]. The inner loop is devoted to the attitude control and the outer one to the position control. Due to the vehicle's high dynamics, the attitude controller must run at higher frequency than the position one.

With an attitude controller it is possible to perform a stationary flight at a fixed altitude, but it is not possible to compensate for drifts in the horizontal plane. The attitude controller relies on inertial measurement units, while the choice of the sensors related to the position controller is strictly task dependent and represents still an open problem for MAVs.

The most popular approaches to solve this problem are mainly based on the fusion of the data provided by an IMU and a GPS [7], [104], or IMU, GPS

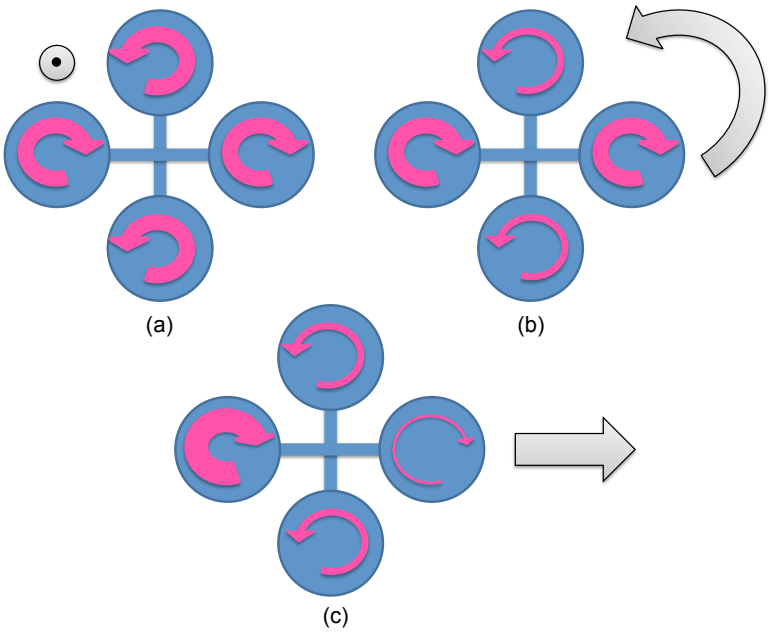


Figure 1.5: The quadrotor concept. The width of the arrows is proportional to the angular speed of the propellers.

and camera [98]. However, these approaches require a reliable GPS signal. Laser range finders have also been adopted in this framework [8],[12],[11]. Nevertheless, range finders sensors have two drawbacks: they have a restricted perception area (limited distance and field of view) and they are still too heavy for MAVs.

Figure 1.6 shows different sensors that can be mounted on a micro aerial vehicle with their advantages and drawbacks.

Vijay Kumar's group at GRASP Lab (General Robotics, Automation, Sensing and Perception), University of Pennsylvania and Raffaello D'Andrea's group at IDSC (Institute for Dynamic Systems and Control), ETH Zurich, achieved impressive results with quadrotors catching the attention of not only all the robotics community but also of the media and the general public (Figure 1.7). Aggressive maneuvers [96], dancing quadrotors [87], acrobatics [18], quadrotors throwing and catching a ball [80], constructions with quadrotor teams [102], [56] represent outstanding works on quadrotor control and trajectory tracking, but they are feasible thanks to the high frequency vehicle position feedback provided by a motion capture system [6]. This means that the vehicles' workspace must be equipped with high resolution (up to 16 Megapixels) and high frame rate (up to 1000 fps) cameras as shown in Figure 1.7c. Motion capture systems represent a perfect testbed to develop control algorithms, to test state estimation algorithms and to

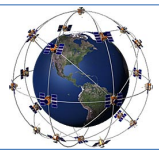
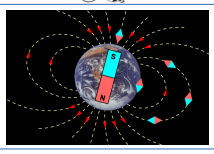
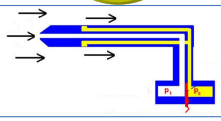
		Advantages	Drawbacks
Inertial sensors		• Linear accelerations; • Angular velocities.	• Biased and noisy measurements; • Large uncertain for slow motions.
GPS		• Absolute position; (outdoor).	• Bad or no reception in indoor or urban environments.
Magnetic field sensors		• Earth's magnetic field direction (outdoor).	• Disturbed by electronic devices nearby
Barometric / Pressure sensors		• Absolute altitude.	• Not reliable indoor; • Affected by weather conditions.
Airspeed sensors		• Vehicle's airspeed.	• Not suitable for rotorcrafts.
Cameras		• Vast information; • Visual feedback.	• Affected by light changes; • Textured environments required.
Laser rangefinders		• Distance to objects.	• Heavy for MAVs • Expensive • 2D information

Figure 1.6: Properties of some sensors commonly on-board a MAV.

simulate GPS signal in indoor environments, but they have the big drawbacks of being not portable and very expensive. Motion capture systems cannot therefore be considered a solution for MAVs autonomous navigation.

The limitations of Global Positioning and motion capture systems highlight the importance of on-board perception not relying on external infrastructures. From the other hand, the limited sensing payload and the limited on-board computation of MAVs represent a bottleneck for the developing of autonomous nav-

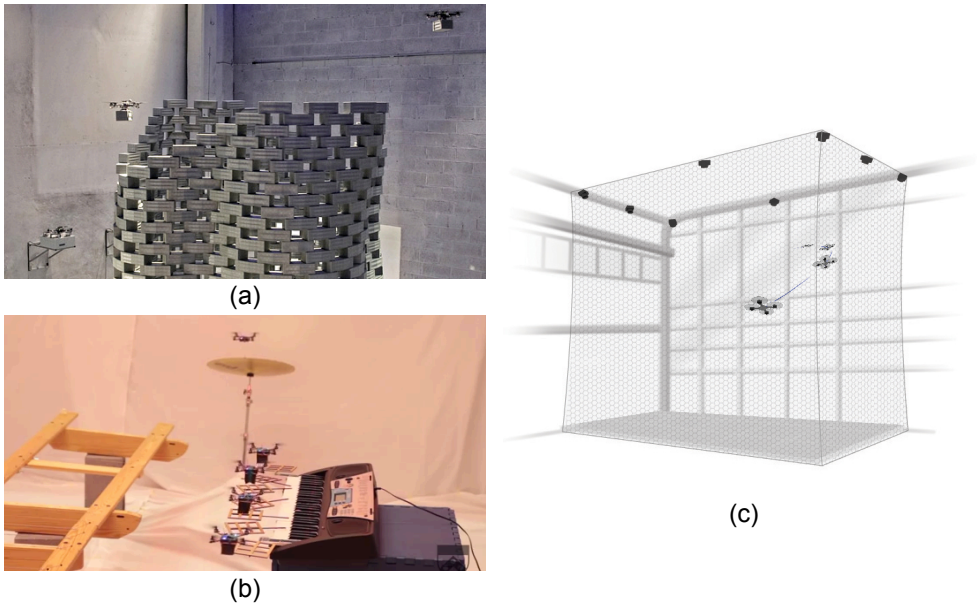


Figure 1.7: Flight Assembled Architecture (a) at FRAC Centre in Orlans, France [25]. Robot Quadrotors Perform James Bond Theme (b) [53]. Figure (c) represents a motion capture system room. Figures (a),(c) courtesy of http://www.flyingmachinearena.org.

igation algorithms.

A viable option for GPS denied environments is obtained by fusing visual and inertial data. This option has become very popular for Micro Aerial Vehicle (MAV) navigation due to the low cost, power consumption and weight.

1.2.1 Minimalist perception

The trend of robotics miniaturization boosts researchers towards minimalist design, investigation of the least complex solutions to a given class of tasks and selection of the simplest set of sensors. According to this perspective, throughout this dissertation we decided to tackle the problems of data association and pose estimation in the framework of MAVs navigation.

The considered sensor suite is essentially composed of a monocular camera rigidly attached to an inertial measurement unit and mounted on a micro quadrotor. The choice of the sensors is related to their complementary properties (see Figure 2.1), their low power consumption, low cost and low weight.

Once chosen our minimal set of sensors, we want to establish the minimal amount of information necessary to perform the data association and pose esti-

Task	Minimal sensor suite	Minimal amount of information	Hypothesis	Algorithm
Outlier rejection	• Camera • IMU	• 1 feature • Angular rates (IMU)	Local planar motion	1-point algorithm
	• Camera • IMU	• 2 features • Angular rates (IMU)	None (6 DoF)	2-point algorithm
Pose estimation	• Camera • IMU	• 3 features • IMU	Planar ground assumption	Virtual patterns
	• Camera • IMU • Laser module	• 1 feature (laser spot) • IMU	Planar surface with unknown orientation.	Virtual features

Figure 1.8: This table summarises the objective of this dissertations in terms of minimalist perception.

mation tasks, and to derive minimal complexity algorithms (Figure 1.8).

1.3 Contributions

The topics addressed by this thesis are the data association and the pose estimation of a camera-IMU system in the framework of MAVs navigation. The previous section highlighted the importance of on-board perception and the limitations inherent to small flying platforms such as limited payload and resources. In order to find a compromise between the aforementioned specifics, we propose minimal complexity algorithms which exploits system properties of typical navigation constraints in indoor or city-like environments.

The contributions of this dissertation are summarized below:

1. Two low computational complexity methods to perform the outlier detection task.

    1.1 *1-point algorithm.* Relying on the assumption that the local vehicle motion lies in a plane perpendicular to the gravity vector, we provide an efficient algorithm which only requires the observation of a single feature in the scene and the knowledge of the angular rates from an IMU. It's efficiency and low computational complexity make it suitable for real-time implementations.

1.2 *2-point algorithm.* Algorithm which requires the observation of one more feature with respect to the previous one, but it relaxes the hypothesis on the vehicle motion, being therefore suitable to tackle the outlier detection problem in the case of a 6DoF motion. Additionally, we show that if the monocular camera is rigidly attached to a quadrotor vehicle, motion priors from IMU can be exploited to discard wrong estimations in the framework of a 2-point-RANSAC based approach.

2. Two low computational complexity algorithms to face the pose estimation problem.

   2.1 *Virtual patterns.* Many localization algorithms utilize artificial landmarks, such as for example ultrasonic beacons, bar-code reflectors, visual pattern, but they are not reliable in a landmark-free environment. Nevertheless, the geometry of a known visual pattern encodes useful information. Starting from this consideration, and taking advantage of the so called "planar ground assumption" (common scenario in indoor or city-like environments), we propose an algorithm which exploits the geometric information encoded in the angles of a virtual triangle made by three natural features belonging to the ground plane. The algorithm is based on a closed solution which provides the vehicle pose from a single camera image, once the roll and pitch angles are obtained by the inertial measurements.

   2.2 *Virtual features.* The feature extraction and matching task is computationally very expensive and fails in dark or featureless environments. In order to significantly reduce the computational burden and to make the feature matching task more robust with respect to outliers, we introduce a virtual feature by equipping the vehicle with a laser pointer. This problem differs from the classical vision and IMU data fusion one, because the feature is moving jointly with the vehicle. To the best of our knowledge, this problem has never been considered so far.

      - *Observability analysis.* An observability analysis is performed to identify the physical quantities (called "observable modes") that can be estimated by using the information provided by the aforementioned sensor suite.
      - *Local decomposition.* To estimate the observable modes, we perform a local decomposition of the original system and we apply a simple recursive method (Extended Kalman Filter) to the observable sub-system.

1.4 Thesis outline

The rest of the thesis is structured as follows.

Chapter 2 - *Vision and Inertial sensors* - provides a brief overview of visual and inertial sensing from both a biological and technological point of view. The calibration techniques used during the experiments are here described.

Chapter 3 - *Data association* - introduces the first two contributions of this dissertation. In the first section we provide an overview of the data association problem for the feature matching process. The feature detection, tracking and matching problems are introduced. In the second section we describe the data association problem and we provide an overview about the state of the art. Two low computational complexity methods to perform the outlier detection task between two different views of a monocular camera rigidly attached to an inertial measurement unit are presented.

Chapter 4 - *Pose estimation* - introduces the last two contributions of this dissertation. In the first section we provide an overview of the visual-aided inertial pose estimation problem, with an emphasis on aerial navigation. Two low computational complexity algorithms to face the pose estimation problem are presented.

Chapter 5 - *Conclusions* - summarizes the contributions of this dissertation and provides perspectives for future developments.

Chapter 2

Visual and Inertial sensors

Contents

2.1 A biological overview **14**

2.1.1 The sense of sight 14

2.1.2 The perception of gravity 15

2.2 Inertial Measurement Unit **17**

2.2.1 Accelerometers 18

2.2.2 Gyroscopes 19

2.3 Camera **20**

2.3.1 Pinhole Camera Model 21

2.4 Camera Calibration **23**

2.5 Camera-IMU Calibration **24**

In this chapter we give a brief overview of visual and inertial sensing from both a biological and technological point of view. The calibration techniques used during the experiments are here described.

In recent years, the fusion of vision and inertial sensing has received great attention by the mobile robotics community. These sensors require no external infrastructure and this is a key advantage for robots operating in unknown environments where GPS signals are shadowed. Additionally, these sensors have very interesting complementarities and together provide rich information to build a system capable of vision-aided inertial navigation and mapping.

	CAMERA	IMU
Slow motion	• Good feature tracking	• Large measurement uncertainty
Fast motion	• Tracking less accurate (motion blur, effect of camera sampling rate) • Higher frame rate means increase in bandwidth and a drop of real time performances	• Lower relative uncertainty. • Precise measurements of large speed and accelerations.
Impossible to distinguish	• A near object with low relative speed from a far object with higher relative speed.	• A change in inclination from body acceleration.

Figure 2.1: Complementary properties of cameras and IMU sensors.

2.1 A biological overview

A special issue of the *International Journal of Robotics Research* was recently been devoted to the problem of fusing vision and inertial data [27]. In [22], a tutorial introduction to the vision and inertial sensing is presented. This work provides a biological point of view and it illustrates how vision and inertial sensors have useful complementarities allowing them to cover the respective limitations and deficiencies.

2.1.1 The sense of sight

Vision is one of the most important senses and is common to almost all living creatures. In an essay on the differences between human and animal vision, [29], the authors stated that "we must never make the mistake of thinking that only we see the world as it really is." The evolution of the organs dedicated to visual perception followed the different needs of animals and adapted to different circumstances. Eyesight helps fulfilling the most basic living activities: hunt, exploration, protection. Nature teaches us that the way how to perceive the environment, to interpret it, and therefore the technology behind the development of new sensing sensors, must be application dependent.

The organs devoted to visual perception are the eyes. They perform the conversion of light into electro-chemical impulses. There are more than 40 different types of eye in nature. The simplest eye, the one belonging to microorganisms, only detects if the environment is dark or bright.

The human eye (Figure 2.2) has a more complex structure which allows us to collect the light in the environment, regulate its intensity through a diaphragm

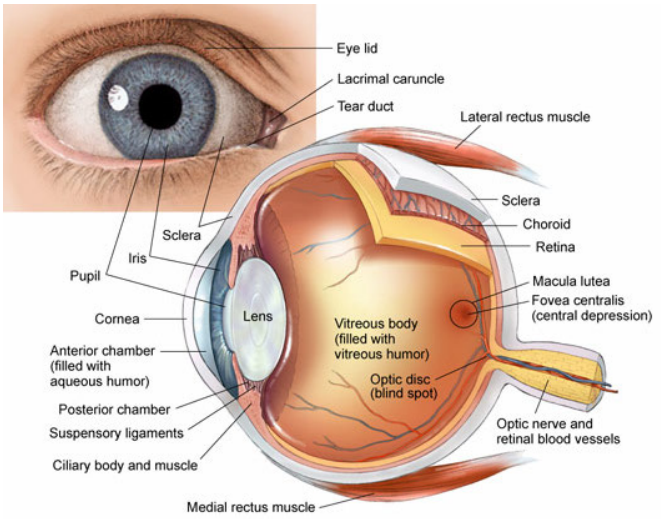


Figure 2.2: Human eye. Image courtesy of http://www.biographixmedia.com.

and focus it by deforming a lens in order to form an image on the retina. The retina is a layered structure with several layers of neurons interconnected by synapses. The neurons sensitive to light are the photoreceptor cells. They can be distinguished into rods and cones. The cones are located in the fovea (the part of the retina where the light is focused by the lens) and they are sensitive to colors. The rods are more sensitive to light than cones, they are located around the fovea and they are responsible for night vision and peripheral vision. The image focused on the retina is transformed into electrical signals which are transmitted to the brain through the optic nerve.

Insects have compound eyes (Figure 2.3), consisting of thousands of photoreceptor units (ommatidium) located on a convex surface. The perceived image derives from the combination of the inputs of each individual eye unit. One of the benefits of compound eyes with respect to simple eyes is the large field of view. In some cases they are also able to detect the polarization of light.

2.1.2 The perception of gravity

All living organisms must have a perception of the environment in which they live, of the forces that are acting on them and of their motion. An important role is assumed by the perception of gravity and of the orientation of the body with respect to it. In most mammals, the sensing system devoted to the perception of gravity (therefore movement and sense of balance) is called *vestibular system* (Figure 2.4a) and it presents a similar structure of an Inertial Measure-

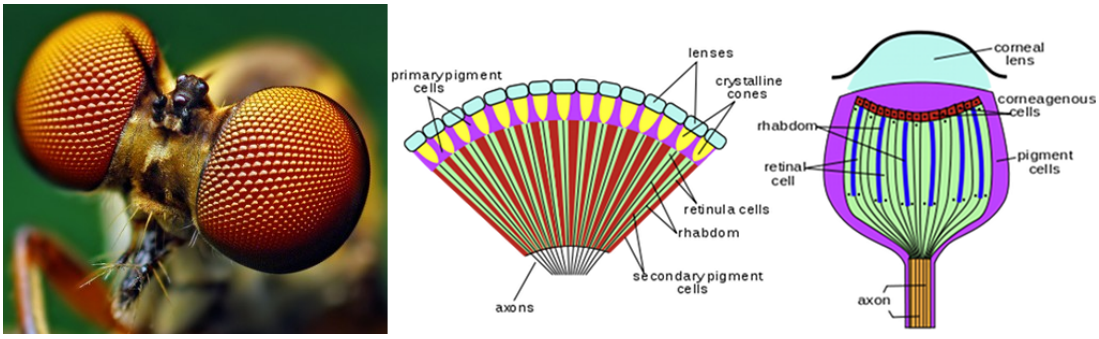


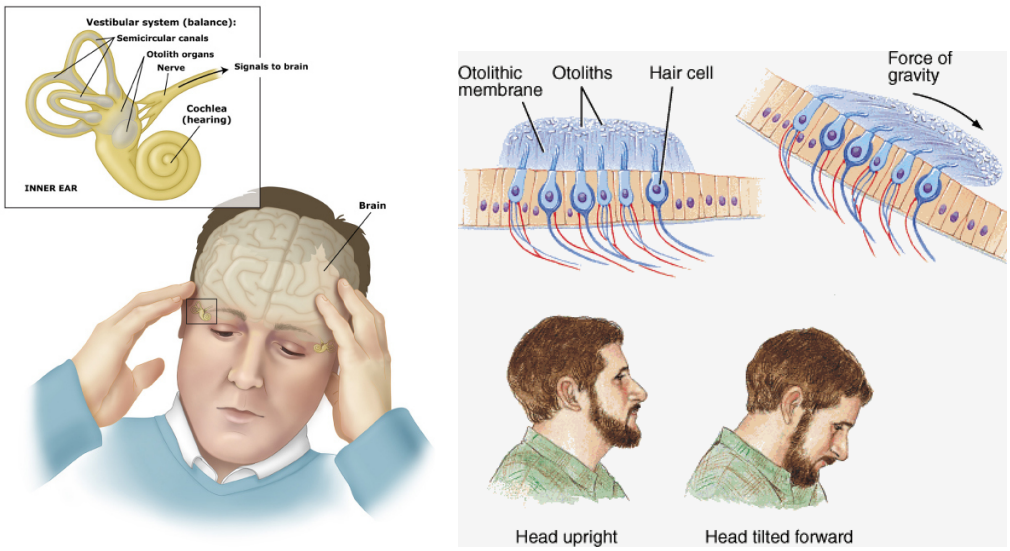
Figure 2.3: Insect eye. Image courtesy of http://www.wikipedia.org.

ment Unit. The human vestibular system is located within the inner ear and it is composed of three main parts: the vestibule, the cochlea, the circular canals. In the vestibule there are the utricle and the saccule, two otoliths organs devoted to the measurement of inertial and gravitational forces (Figure 2.4b). Both of them contain otoliths which acts as proof mass. The otolithic membrane acts as the spring and the damper and hair celles, embedded in the membrane are responsible for the displacement detection. They provide therefore information about the linear motion and the angular position of the head. In order to distinguish the gravity from body motions, animals uses other cues such as vision. It seems also that they can divide the acceleration signal by frequency. Lower frequency components are associated to pose and high frequency ones to acceleration.

The angular velocity of the head is measured by the three semicircular canals which are oriented in three orthogonal planes [35]. Each channel is filled with a viscous fluid. As a result of any rotation, the fluid pushes against one of the extremities of the channel, where is located the ampulla. The latter senses the corresponding force.

The brain processes all the signals provided by the three channels and provides an estimation of the head instantaneous angular velocity, in order to ensure gaze stability (process known as *vestibulo-ocular reflex*).

Insects have similar ways to detect a change in speed and direction. Flies and other insects are equipped with mechanoreceptors (antennas) which contain Johnston's organ responsible of the antenna displacement detection due to pressure, gravity, or sound. Flies have also a device very similar to a gryroscope (halteres) (Figure 2.5) very important in gaze stabilization.



(a) (b)

Figure 2.4: Vestibular system. Image (a) courtesy of http://www.chrcentre.com.au, image (b) courtesy of http://biology.nicerweb.com.

Figure 2.5: Halteres: small knobbed structures in some two-winged insects. They are flapped rapidly and function as gyroscopes, providing informations to the insect about his body rotation during flights.

2.2 Inertial Measurement Unit

An Inertial Measurement Unit (IMU) is an electronic device usually composed by a three-axis accelerometer and a three-axis gyroscope. They are called *Inertial*

Figure 2.6: IMU applications Image courtesy of http://www.unmannedsystemstechnology.com.

sensors because their working principle is based on the resistance to a change in momentum (property of inertia). Accelerometers sense the translational body's acceleration (they measure spatial derivative order 2) and the gyroscopes sense the rate of change of the body's orientation (spacial derivative order 1). According to the principles of inertial navigation systems (INS) [54], linear position, linear velocity and angular position are estimated by integration. Inertial sensors do not require any external infrastructure but the gravity field. The development of low cost and low weight inertial sensors, called MEMS (Micro Electro Mechanical Systems) inertial sensors, have lead to the extension of their field of applications. Initially mainly used for aerospace applications, those sensors are nowadays incorporated into a lot of different mass-produced devices such as vehicles, cellphones, gaming consoles, sports training devices, digital cameras, laptops. Their applications spread from inertial navigation (Figure 2.6) to earthquake detection, seismic reflection profiling, volcanoes monitoring and magma motion detection, medical applications, vehicle security, health monitoring, digital camera orientation, image stabilization, laptop drop detection.

2.2.1 Accelerometers

An accelerometer senses the acceleration of a mass at rest in the frame of reference of the accelerometer device. An accelerometer at rest on the earth surface measures an acceleration of $9.81m/s^2$ corresponding to the gravity acceleration. On the contrary, an accelerometer in free fall measures zero acceleration.

The physical principle exploited by an accelerometer is quite simple (Figure 2.7). A mass, m, usually called *seismic mass* or *proof mass* is supported by a spring c. A viscous damper, b, provides damping proportional to the relative velocity of the test mass and the sensor body and it is necessary for a quick stabilization of the system. When the sensor is subjected to acceleration, the mass

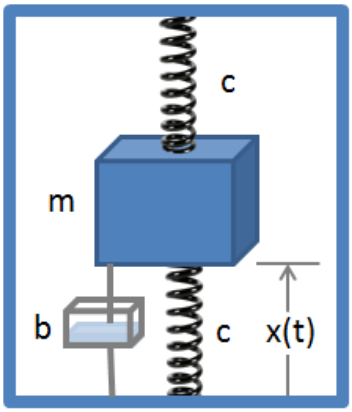


Figure 2.7: Physical principle of a single-axis accelerometer.

is displaced and the displacement is measured in order to obtain the acceleration. Equation 2.1 expresses the dynamics of the system.

$$\ddot{x}(t) + 2\xi\omega_n\dot{x}(t) + \omega_n^2 x(t) = -\ddot{y}(t) \tag{2.1}$$

The acceleration of the sensor body, $\ddot{y}(t)$, is converted to spatial displacement with a natural frequency ω_n and a damping ratio ξ. The mechanical motion is then converted into an electrical signal.

It is not possible to distinguish if the sensor is accelerating or if it is subject to some components of gravity acceleration. To resolve this ambiguity it is necessary to make strong assumptions or to fuse the sensor information with measurements from other sensors.

Nowadays there are three main types of MEMS accelerometers: capacitive, piezo-electric and piezo-resistive. The piezo-electric devices have a large dynamic range but they are not suitable for inertial navigation systems due to the the lack of DC response. In piezo-resistive accelerometers, the test mass displacement is measured by a piezo-resistor which changes its value. They are preferred in high shock applications. In capacitive accelerometers, the mass displacement is measured by a changing capacitor. They are more performant in low frequency range and they can be used in servo mode to achieve high stability and linearity. A detailed overview about MEMS has been given in [103] and [58].

2.2.2 Gyroscopes

A gyroscope is an heading sensor, used to measure or maintain orientation and it has benn invented by Léon Foucault in 1852. Gyroscopes can be essentially

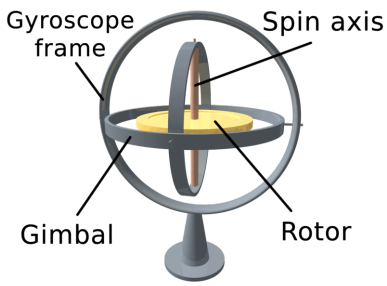


Figure 2.8: Scheme of a 3D mechanical Gyroscope. Image courtesy of http://www.wikipedia.org.

classified into two main categories: mechanical gyroscopes and optical gyroscopes.

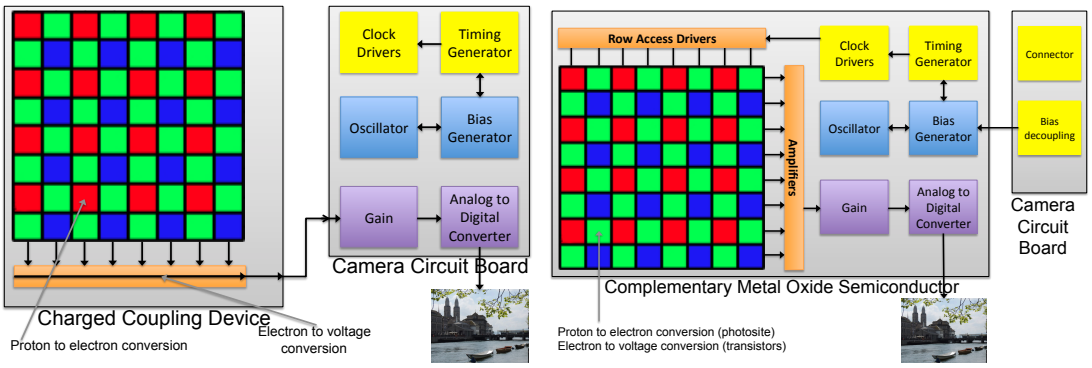
Mechanical gyroscopes (Figure 2.8) are based on the principal of conservation of angular momentum. A classical mechanical gyroscope consists of a spinning wheal or disk on an axle. Once that the device is spinning, it tends to resist to changes of orientation.

Vibrating structure gyroscopes (VSG), called also Coriolis Vibratory Gyroscopes (CVG), do not present any rotating disk. The basic physical principle is that a vibrating object tends to continue vibrating in the same plane as its support rotates. The Coriolis effect induced by rotation is measured.

Optical gyroscopes appeared for the first time at the beginning of 1980 and their primary application was related to aircraft. Instead of moving mechanical parts, optical gyroscopes rely on two monochromatic light beams, or lasers, emitted from the same source and they are based on the *Sagnac effect*, named after French physicist Georges Sagnac in 1913. Two laser beams are sent travelling through an optical fiber, one in clockwise direction and the other one in counterclockwise direction. According to the Sagnac effect, the beam travelling in the direction of rotation has a higher frequency. The difference in frequency of the two beams is proportional to the angular velocity of the cylinder. Optical gyroscopes are not sensible to vibrations, accelerations, shocks, and provide very precise rotational rate information.

2.3 Camera

Vision is a very powerful sense and this explains the great attention devoted in the last decades to the design and development of technological devices able to convert light into digital images and image processing algorithm able to extract task-dependent useful information from camera measurements. Vision sensors are nowadays very light weight, low cost, low power devices and they become



(a) (b)

Figure 2.9: Diagram of a CCD (a) and CMOS (b) sensor.

very popular in robotic applications.

The process related to the conversion of photons falling on an imaging sensor into digital values is known as *digital image formation*. Light rays, emitted from a light source and reflected from different surfaces belonging to the scene, pass through the camera's lenses and reach the image sensor. Those rays are integrated for the duration of the exposure and then sent to a set of amplifiers. Nowadays there are two main kinds of image sensors: CCD (Charged Coupled Device) and CMOS (Complementary Metal Oxide on Silicon).

A CCD is an array of light-sensitive analog devices (pixels) (Figure 2.9a). Each pixel, once hit by light, releases electrical charge which is read pixel by pixel from the chip. The camera circuitry converts voltage values into digital data.

A CMOS imaging chip is, as CCD, an array of light-sensitive analog devices (pixels) (Figure 2.9b). The signals provided by each pixel are measured and amplified in parallel thanks to extra-circuitry placed along the side of each pixel.

A CMOS sensor presents advantages with respect to a CCD one. The CMOS chip is simpler, requires less power (one-hundredth less than a CCD chip). This is a very important characteristic if the camera is mounted on board to systems with limited autonomy.

For our experiments we choose therefore a monocular camera based on CMOS technology.

2.3.1 Pinhole Camera Model

A camera model, in order to characterize the transformation between $3D$ scene point coordinates and $2D$ image pixel coordinates, must take into account the transformation between the camera and the world, and the size and position with

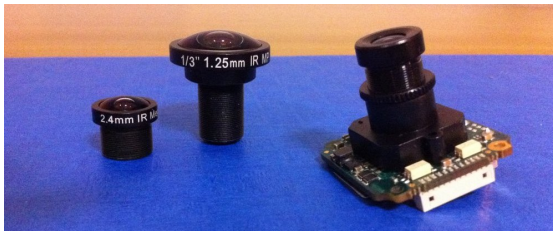


Figure 2.10: Matrix Vision mvBlueFOX-MLC usb camera.

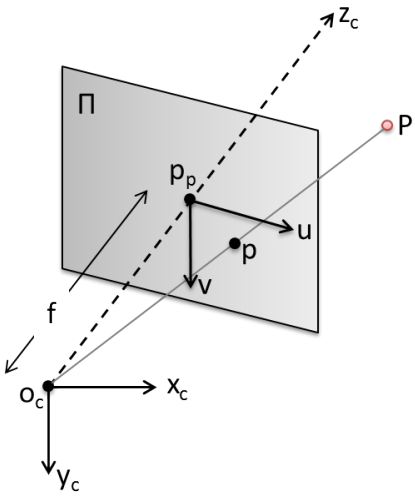


Figure 2.11: Pinhole camera model for standard perspective cameras.

respect to the optical center of the image sensor (pixelization).

Following the convention of the pinhole camera, the *image plane* Π is located between the *optical center* and the scene (Figure 2.11). The axis perpendicular to the image plane and passing through the optical center is named *optical axis* and the intersection of it with the image plane is called *principal point* p_p. The distance f between the image plane and the optical center is called *focal length*.

Let $\{C\}$ be the camera reference frame, with the origin in the *optical center* o_c (or center of projection) and with the z-axis parallel to the optical axis. Let u and v be the two axes that identify a $2D$ reference frame belonging to the image plane, with the center coincident with the principal point p_p. Let $P = [x, y, z]$ and $p = [u, v]$ be the $3D$ coordinate of a scene point in the camera reference frame and its corresponding $2D$ pixel coordinate on the image plane respectively.

By using homogeneous coordinates for P and p, $\tilde{p} = [u, v, 1]'$ and $\tilde{P} =$

$[x, y, z, 1]'$ the projection equation becomes:

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} fk_u & 0 & u_0 & 0 \\ 0 & fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \tag{2.2}$$

where u_0 and v_0 are the coordinates of the principal point, k_u and k_v are the inverse of the pixel size along u and v respectively, measured in $pixel \cdot m^{-1}$. λ is equal to the third coordinate of P in the camera reference frame. A monocular camera provides therefore the position of a point in the scene, up to a scale factor λ. In order to recover λ we need a stereo pair, multiple images, or the fusion of monocular and inertial information.

If we want to express the coordinates of a point P in the world reference frame, we have to take into account the transformation between the last one and the camera reference frame. In this case, being $\tilde{P}_w = [x_w, y_w, z_w, 1]$ the homogeneous coordinate of a scene point P in the world reference frame, the projection equation becomes:

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} [R|T] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}. \tag{2.3}$$

Where $\alpha_u = fk_u$ and $\alpha_v = fk_v$ represent the focal length in pixels. The matrices R and T encode the relative rotation and translation between the camera and the world reference frame and are called *camera extrinsic parameters.*

The matrix

$$A = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

is called *intrinsic parameter matrix* and its elements are called *camera intrinsic parameters.*

2.4 Camera Calibration

Camera calibration consists in the estimation of the intrinsic and extrinsic parameters of the camera model in order to map the image points into the corresponding scene points with the highest precision possible. The idea behind the camera calibration process is that by knowing the $3D$ coordinates $\tilde{P}$ of some particular scene points and their corresponding pixel coordinates in the image plane $\tilde{p}$, it is possible to estimate the unknown parameters that characterize the camera model.

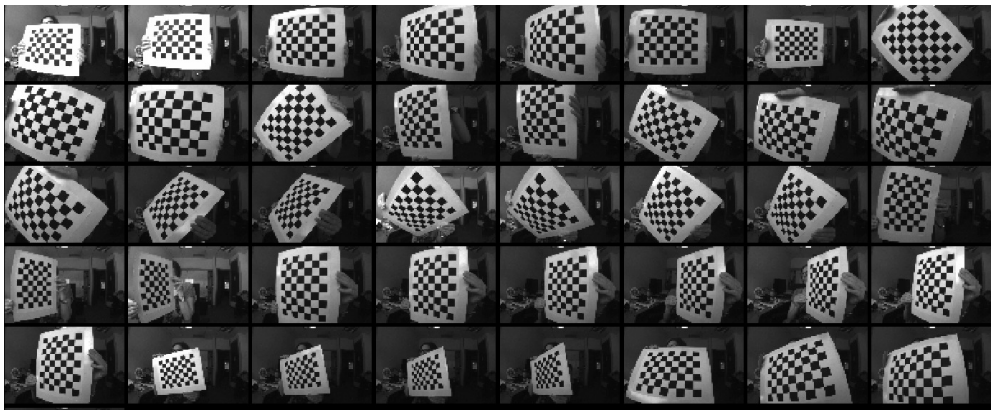


Figure 2.12: Chessboard images used for calibration.

One of the first camera calibration algorithm was proposed in 1987 by [95] and it consists in a two-stage technique to compute the position and the orientation first and the internal parameters later, by knowing corresponding $2D$ pixel coordinates and $3D$ scene point coordinates. In 1998 [106] introduced the use of a planar grid as calibration object. In order to facilitate the corner extraction process, the most used planar grid is a chessboard-like pattern (Figure 2.13).

This method requires several images of the chessboard taken from different position and orientation (Figure 2.12). It is important that the pattern covers the bigger portion possible of the camera field of view and the accuracy of the result increases with the number of images used.

The intrinsic and extrinsic camera parameters (including radial and tangential distortion) are identified by knowing the coordinates of the corners on the pattern and their corresponding pixel coordinates in each image and solving a least square minimization plus a nonlinear refinement. This method has been implemented in an open source Matlab toolbox [17] and it is the method we used for our experiments.

2.5 Camera-IMU Calibration

For a robust fusion of the information provided from a monocular camera and an Inertial Measurement Unit, we need to know the relative orientation and translation between the reference frames associated to the two sensors (inter-sensor calibration).

There exist many methods to calibrate a camera-IMU system [45] [71] offline or online [101].

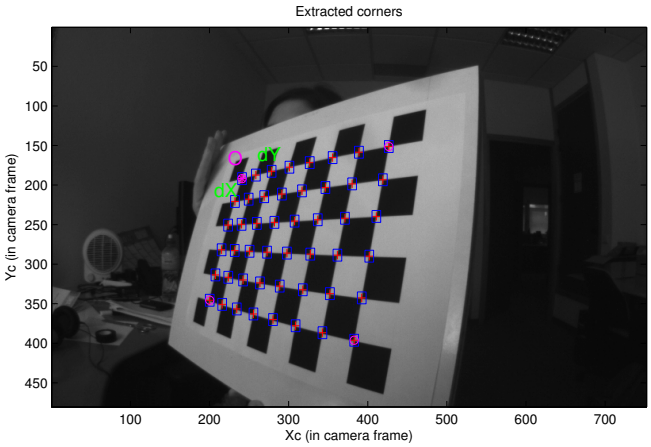


Figure 2.13: Extracted corners on a chessboard.

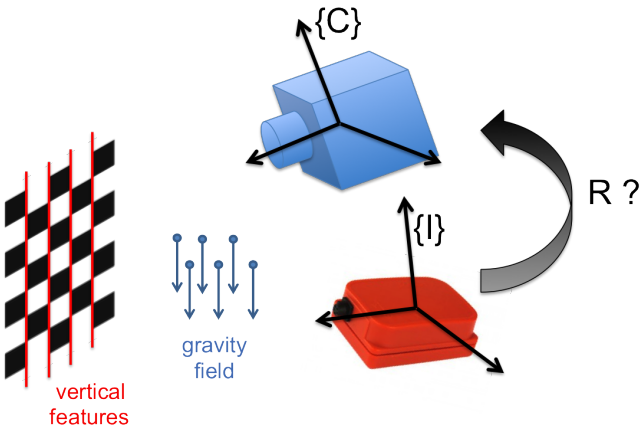
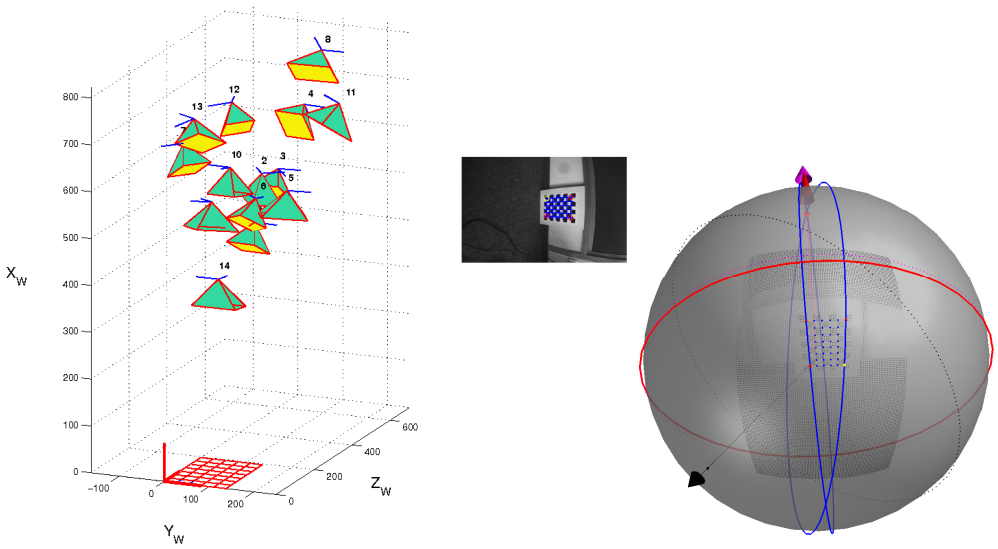


Figure 2.14: Camera and IMU observing the vertical direction. Redrawn from [57].

To perform the Camera-IMU calibration we used the *InerVis Toolbox* for Matlab from [57].

In order to estimate the rotation between the two reference frames, the two sensors must observe the vertical direction (Figure 2.14).

Once the camera is calibrated, it is possible to recover the extrinsic parameters (rotation and translation) with respect to a reference frame attached on the



(a) (b)

Figure 2.15: Camera poses relative to the calibration chessboard (a). Results of rotation estimation (b).

chessboard. When the system is static, the accelerometers senses only the gravity vector and they can therefore provide the gravity direction in the IMU reference frame. By using a chessboard, positioned in a way that the vertical lines of the chessboard are parallel to the gravity vector, it is possible to determine the vertical direction in the camera reference frame (orthogonal Procrustes method for 3D attitude estimation). The relative accelerometer and camera measurement are recorded from different static poses of the system. Being v_i^I and v_i^C the estimation of the vertical direction extracted from IMU and camera measurements respectively, the rotation between the two sensor corresponds to the quaternion q that maximises:

$$\sum_{i=1}^{n} (q\, v_i^I\, q^*) \cdot v_i^C. \tag{2.5}$$

Chapter 3

Data association

Contents

3.1 Feature extraction and matching 28

3.1.1 Feature Detection 28

3.1.2 Feature Tracking 29

3.1.3 Feature Matching 30

3.2 Outlier detection 31

3.2.1 Related works 32

3.2.2 Epipolar Geometry 33

3.2.3 1-point algorithm 34

3.2.4 2-point algorithm 45

This chapter introduces the first two contributions of this dissertation. In the first section we provide an overview of the data association problem related to the feature matching process. The feature detection, tracking and matching problems are introduced. In the second section we describe the data association problem and we give an overview about the state of the art. Two low computational complexity methods to perform the outlier detection task between two different views of a monocular camera rigidly attached to an inertial measurement unit are presented. The first one only requires the observation of a single feature in the scene and the knowledge of the angular rates provided by an inertial measurement unit, under the assumption that the local camera motion lies on a plane perpendicular to the gravity vector. In the second proposed algorithm we relax the hypothesis on the camera motion. The observation consists of two features in the scene (instead of only one) and of angular rates from inertial measurements. We show that if the camera is on-board a quadrotor vehicle, motion priors from

inertial measurements can be used to discard wrong data association. Both the methods are evaluated on synthetic and real data.

3.1 Feature extraction and matching

Image features represent the informative content of a raw image. Feature extraction plays therefore an important role in the creation of compact and robust environmental models for map building and localization.

Features are repeatable and salient structures extracted from images and mathematically formalized, that characterize the environment. They are classified into *low-level feature* and *high-level feature*. The former are geometrical primitives like points, lines, corners, blobs, polygons while the latter are objects. Raw images contain an high amount of data, but a low level of distinctiveness. The feature extraction process performs an abstraction of the raw image reducing the volume of data, but augmenting the level of distinctiveness.

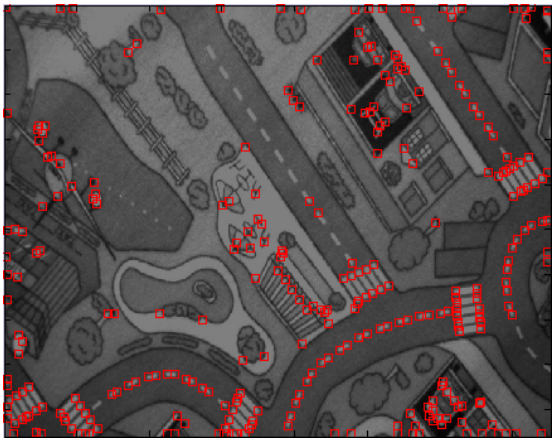
Once that the features are extracted in the first image, their relative correspondences must be identified in the consecutive images. There are two main approaches to select features and their correspondences:

- Extracting features in one image and using local search techniques to track them in the following images;
- Extracting features in all the images independently and match them according to the similarity of their descriptors.

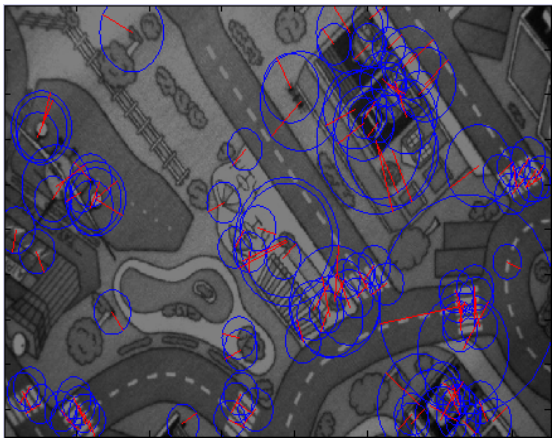
3.1.1 Feature Detection

A local feature is an image pattern that can be distinguished from the neighbor pixels thanks to the difference in intensity, texture and color. In the framework of visual odometry, great importance is given to corners and blobs because of the precision with which they can be localized. A corner is defined as the intersection between two or more edges, while blobs are image patterns with distinctive intensity, texture and color. A good feature detector must have the following properties:

- *Repetability*: a large number of features should be detected in consecutive images;
- *Localization accuracy*: both in scale and position;
- *Computational efficiency*: it must be suitable for real time applications;



(a) Harris corners.



(b) Surf features.

- *Robustness*: to noise, blur, artifacts;
- *Distinctiveness*: in order to reduce wrong data associations;
- *Invariance*: to photometric changes (affine intensity) and geometric (2D rotation, scale, affine transformation) changes.

The choice of a feature detector is not only function of its properties but also of the environment, of the task, of the computational constraints and of the motion baseline. Corners are faster to compute than blobs, easier to localize in position but less distinctive, difficult to localize in scale, more difficult to redetect after large changes in viewpoint and scale, but they can be a good choice in urban and indoor environments.

A performance evaluation of feature detectors and descriptors can be found in [88], [70].

3.1.2 Feature Tracking

Detecting features in one image and tracking them by using local search techniques is suitable for small scale environments in which the motion baseline and the appearance deformation is small. The local search techniques mainly used for tracking features are [88]:

- *Sum of Absolute Differences* (SAD)
- *Sum of Squared Differences* (SSD)
- *Normalized Cross Correlation* (NCC)

	Corner Detector	Blob Detector	Rotation invariant	Scale invariant	Affine invariant	Repeatability	Localization accuracy	Robustness	Efficiency
Harris	x		x			+++	+++	++	++
Shi-Tomasi	x		x			+++	+++	++	++
Harris-Laplacian	x	x	x	x		+++	+++	++	+
Harris-Affine	x	x	x	x	x	+++	+++	++	++
SUSAN	x		x			++	++	++	+++
FAST	x		x			++	++	++	++++
SIFT		x	x	x	x	+++	++	+++	+
MSER		x	x	x	x	+++	+	+++	+++
SURF		X	x	x	x	++	++	++	++

Figure 3.1: Comparison of feature detectors: properties and performances [88].

3.1.3 Feature Matching

The feature matching task consists of searching the correspondences for features in different images.

The easiest way to search for correspondences is to compare the feature descriptors of all the features detected in the first image with the descriptors of all the features detected in the next image by using a similarity measurement. The features in the second image with the closer descriptors are selected as the correspondences. However it can happen that one feature in the first image is associated with more than one feature in the second image. To disambiguate the solution, a *mutual consistency check* is performed. Every feature in the second image is paired with features in the second image. The pairs showing consistent preferred match are chosen as image correspondences.

The main disadvantage of this approach is that its computational complexity is quadratic in the number of features. It is therefore not suitable for real-time applications if the number of features is high. A faster image matching technique consists in searching the correspondences not in the whole image, but in regions where the features are expected to be. Those regions can be computed by using a motion model (assuming constant velocity [26]) or by using additional sensors like IMU, lasers, GPS or wheel odometry [62].

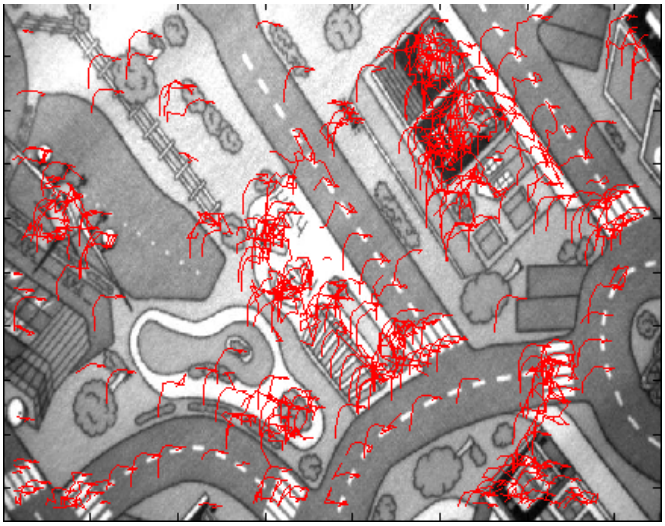


Figure 3.2: Surf features matched across multiple frames overlaid on the first image.

3.2 Outlier detection

The sets of feature correspondences are usually contaminated by outliers, i.e. wrong data associations. It is of outmost importance to remove them because they can negatively affect the accuracy of the esimated motion.

The standard method for model estimation from a set of data affected by outliers is RANSAC (RANdom SAmple Consensus) [32]. It consists of randomly selecting a set of data points, computing the corresponding model hypothesis, and verifying this hypothesis on all the other data points. The solution is the hypothesis with the highest consensus. The number of iterations (N) necessary to guarantee a robust outlier removal is [32]:

$$N = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^s)} \tag{3.1}$$

where s is the number of data points from which the model can be computed, ε is the percentage of outliers in the dataset, p is the probability of success requested. Figure 3.3 shows the number of iterations (N) with respect to the number of points necessary to estimate the model (s). The values are computed for $p = 0.99$ and $\epsilon = 0.5$. Note that N is exponential in the number of data points s; this means that it is extremely important to look for minimal parametrizations of the model, in order to reduce the number of iterations, which is of utmost importance for vehicles equipped with a computationally-limited embedded computer.

Number of points (s)	1	2	3	5	8
Number of Iterations (N)	7	16	35	145	1177

Figure 3.3: Number of RANSAC iterations.

3.2.1 Related works

When the camera is calibrated, its six degrees of freedom (DoF) motion can be inferred from a minimum of five-point correspondences, and the first solution to this problem was given in 1913 by Kruppa [50]. Several five-point minimal solvers were proposed later in [31],[77],[90], but an efficient implementation, based on [77], was found only in 2003 by Nister [75] and later revised in [89]. Before that, the six- [78], seven- or eight- solvers were commonly used. However, the five-point solver has the advantage that it works also for planar scenes. A more detailed analysis of the state of the art can be found in [85].

Despite the five-point algorithm represents the minimal solver for 5DoF motion of calibrated cameras, in the last few decades there have been several attempts to exploit different cues to reduce the number of motion parameters. In [33], the authors proposed a three-point minimal solver for the case of two known camera-orientation angles. For instance, this can be used when the camera is rigidly attached to a gravity sensor (in fact, the gravity vector fixes two camera-orientation angles). Later, the work in [73] improved on [33] by showing that the three-point minimal solver can be used in a four-point (three-plus-one) RANSAC scheme. The three-plus-one stands for the fact that an additional far scene point (ideally, a point at infinity) is used to fix the two orientation angles. Using their four-point RANSAC, they also showed a successful 6 DoF VO. A two-point minimal solver for 6-DoF Visual Odometry was proposed in [49], which uses the full rotation matrix from an IMU rigidly attached to the camera. In the case of planar motion, the motion model complexity is reduced to 3 DoF and can be parameterized with two points as described in [76]. For wheeled vehicles, the work in [86, 82] showed that the motion can be locally described as planar and circular, and, therefore, the motion model complexity is reduced to 2 DoF, leading to a one-point minimal solver. Additionally, it was shown that, by using a simple histogram voting technique, outliers can be found in as little as a single iteration. A performance evaluation of five-, two-, and one-point RANSAC algorithms for Visual Odometry was finally presented in [83].

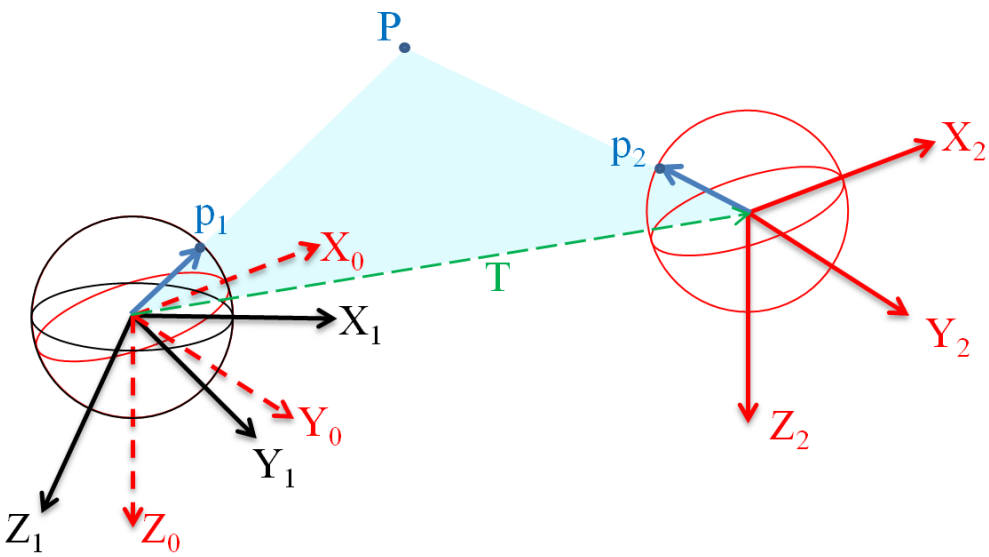


Figure 3.4: Epipolar constraint. $\mathbf{p_1}$, $\mathbf{p_2}$, T and P lie on the same plane (the *epipolar plane*).

3.2.2 Epipolar Geometry

Before going on, we would like to recall some definitions about epipolar geometry. When a camera is calibrated, it is always possible to project the feature coordinates onto a unit sphere. This allows us to make our approach independent of the camera model.

Let $\mathbf{p_1} = (x_1, y_1, z_1)$ and $\mathbf{p_2} = (x_2, y_2, z_2)$ be the image coordinates of a point feature seen from two camera positions and back projected onto the unit sphere (i.e., $\| \mathbf{p_1} \| = \| \mathbf{p_2} \| = 1$) (Figure 3.4).

The image coordinates of point features relative to two different unknown camera positions must satisfy the *epipolar constraint* (Figure 3.4) [38].

$$\mathbf{p_2}^T \mathbf{E} \mathbf{p_1} = 0 \tag{3.2}$$

where $\mathbf{E}$ is the *essential matrix*, defined as $\mathbf{E} = [\mathbf{T}]_\times \mathbf{R}$. $\mathbf{R}$ and $\mathbf{T} = [T_x, T_y, T_z]^T$ describe the relative rotation and translation between the two camera positions, and $[\mathbf{T}]_X$ is the skew symmetric matrix:

$$[\mathbf{T}]_\times = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \tag{3.3}$$

According to equation (3.2), the essential matrix can be computed given a set

of image coordinate points. **E** can then be decomposed into **R** and **T** [38].

The minimum number of feature correspondences needed to estimate the essential matrix is function of the degrees of freedom of the camera's motion. In the case of a monocular camera performing a 6DoF motion (three for the rotation and three for the translation), considered the impossibility to recover the scale factor, a minimum of five correspondences is needed.

3.2.3 1-point algorithm

In this section we propose a novel method to estimate the relative motion between two consecutive camera views, which only requires the observation of a single feature in the scene and the knowledge of the angular rates from an inertial measurement unit, under the assumption that the local camera motion lies in a plane perpendicular to the gravity vector [93]. Using this 1-point motion parametrization, we provide two very efficient algorithms to remove the outliers of the feature-matching process. Thanks to their inherent efficiency, the proposed algorithms are very suitable for computationally-limited robots. We test the proposed approaches on both synthetic and real data, using video footage from a small flying quadrotor. We show that our methods outperform standard RANSAC-based implementations by up to two orders of magnitude in speed, while being able to identify the majority of the inliers.

3.2.3.1 Parametrization of the camera motion

We consider a micro aerial vehicle equipped with a monocular camera and an IMU. The transformation between the camera reference frame $\{C\}$ and the vehicle's body frame $\{B\}$ (that for aerial vehicles is coincident with the IMU frame) can be computed using [57]. Without loss of generality, we can assume that these two frames are coincident.

According to aerospace conventions [24], the X_B-axis of an aerial vehicle commonly defines the forward direction, the Z_B-axis is downward, and the Y_B-axis follows the right-hand rule. We assume the same convention for our vehicle (Figure 3.5). We use the $Z-Y-X$ Euler angles to model the rotation of the vehicle in the World frame. To go from the World frame to the Body frame, we first rotate about z_W axis by the angle Yaw, then rotate about the intermediate y-axis by the angle $Pitch$, and finally rotate about the X_B-axis by the angle $Roll$.

We define as well a coordinate frame $\{C_p\}$ attached to the vehicle, with the same origin as the vehicle's Body Frame but with its z-axis aligned to the gravity vector ($\mathbf{g}$). The $Roll$ and $Pitch$ angles and the relative rotation about Z_{C_p}-axis ($dYaw$) of the vehicle are provided by the IMU fusing the integration of the high frequency gyroscopic measurements with the gravity direction obtained by the

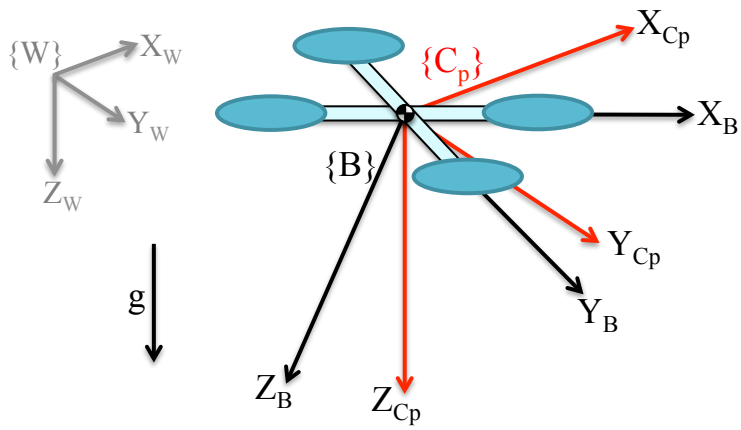


Figure 3.5: Notation.

accelerometers. If the system is in motion, the resulting estimation allows us to safely recover the short term relative orientation of the vehicle, that is only affected by a slowly changing drift term.

Considering that the camera is rigidly attached to the vehicle, two camera orientation angles are known (they correspond to the *Roll* and *Pitch* angles provided by the IMU).

If $R_x(\gamma)$, $R_y(\gamma)$, $R_z(\gamma)$ are the orthonormal rotation matrices for rotation of γ about the x-, y- and z-axes, the matrices

$$\begin{aligned} {}^{Cp1}R_{B_1} &= (R_x(Roll_1) \cdot R_y(Pitch_1))^T \\ {}^{Cp2}R_{B_2} &= (R_x(Roll_2) \cdot R_y(Pitch_2))^T \end{aligned} \tag{3.4}$$

allow us to virtually rotate the two camera frames into two new frames $\{C_{p_1}\}$ and $\{C_{p_2}\}$ (Figure 3.6). $Pitch_i$ and $Roll_i$, $(i = 1, 2)$ are the angles provided by the IMU relative to two consecutive camera frames.

The two new image planes are parallel to the ground ($z_{C_{p_1}} \parallel z_{C_{p_2}} \parallel g$).

If the vehicle undergoes perfect planar motion, the essential matrix depends only on 2 parameters. Integrating the gyroscopic data within the time interval relative to two consecutive camera frames (i.e. the camera framerate), we can obtain the relative rotation of the two frames about Z_{C_p}-axis. We define a third reference frame C_{p_0}, that corresponds to the reference frame C_{p_1} rotated according to $dYaw$, in order to have the same orientation of C_{p2} (Figure 3.6)). The matrix that describes this rotation is the following:

$${}^{Cp0}R_{Cp1} = (R_z(dYaw))^T \tag{3.5}$$

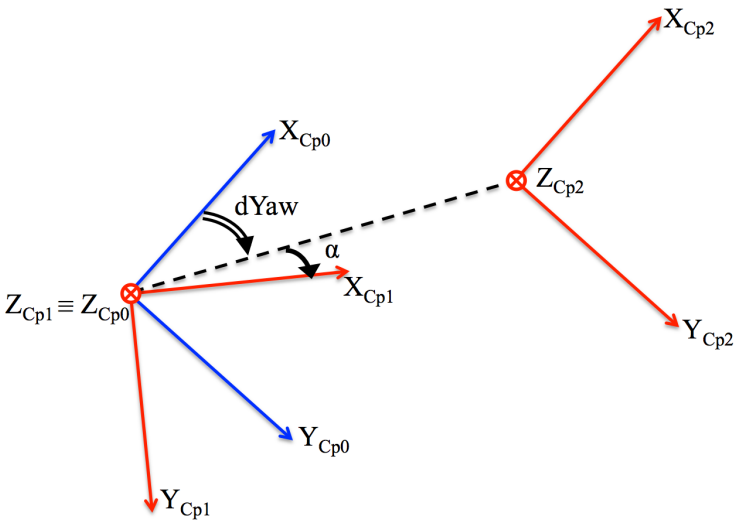


Figure 3.6: C_{p1} and C_{p2} are the reference frames attached to the vehicle's body frame but which z-axis is parallel to the gravity vector. They correspond to two consecutive camera views. C_{p0} corresponds to the reference frame C_{p1} rotated according to $dYaw$.

To recap we can express the image coordinates into the new reference frames according to:

$$\begin{aligned} \mathbf{p_{C_{P_0}}} &= {}^{Cp0}R_{Cp1} \cdot {}^{Cp1}R_{B_1} \cdot \mathbf{p_1} \\ \mathbf{p_{C_{P_2}}} &= {}^{Cp2}R_{B_2} \cdot \mathbf{p_2} \end{aligned} \tag{3.6}$$

At this point the transformation between $\{C_{p0}\}$ and $\{C_{p2}\}$ is a pure translation:

$$\begin{aligned} \mathbf{T} &= \rho[cos(\alpha) \quad -sin(\alpha) \quad 0]^T \\ \mathbf{R} &= I_3 \end{aligned} \tag{3.7}$$

and it depends only on α and on ρ (the scale factor). The essential matrix results therefore notably simplified:

$$E = [\mathbf{T}]_\times \mathbf{R} = \rho \begin{bmatrix} 0 & 0 & -sin(\alpha) \\ 0 & 0 & -cos(\alpha) \\ sin(\alpha) & cos(\alpha) & 0 \end{bmatrix} \tag{3.8}$$

At this point, being $\mathbf{p_{C_{P_0}}} = [x_0 \quad y_0 \quad z_0]^T$ and $\mathbf{p_{C_{P_2}}} = [x_2 \quad y_2 \quad z_2]^T$, we impose the epipolar constraint according to (3.2) and we obtain the homogeneous

equation that must be satisfied by all the point correspondences.

$$(x_0 z_2 - z_0 x_2) sin(\alpha) + (y_0 z_2 - z_0 y_2) cos(\alpha) = 0 \tag{3.9}$$

where $\mathbf{p_0} = [x_0 \quad y_0 \quad z_0]^T$ and $\mathbf{p_2} = [x_2 \quad y_2 \quad z_2]^T$ are the directions (or unit-sphere coordinates) of a matched feature in $\{C_{p0}\}$ and $\{C_{p2}\}$ respectively. Equation 3.9 depends only on one parameter (α). This means that the relative vehicle motion can be estimated using only a single image feature correspondence.

At this point we can recover the angle α from 3.9:

$$\alpha = tan^{-1}\left(\frac{z_0 y_2 - y_0 z_2}{x_0 z_2 - z_0 x_2}\right) \tag{3.10}$$

3.2.3.2 1-point Ransac

One feature correspondence is randomly selected from the set of all the matched features. The motion hypothesis is computed according to (3.7). Without loss of generality we can set $\rho = 1$. Inliers are, by definition, the correspondences which satisfy the model hypothesis within a defined threshold. The number of inliers in each iteration is computed using the reprojection error. We used an error threshold of 0.5 pixels. The minimum number of iterations to guarantee a good outlier detection, considering $p = 0.99$ and $\varepsilon = 0.5$ is 7 (according to (3.1)).

3.2.3.3 Me-RE (Median + Reprojection Error)

The angle α is computed from all the feature correspondences according to (3.10). A distribution $\{\alpha_i\}$ with $i = 1, 2, \ldots, N_f$ is obtained, where N_f is the number of correspondences between the two consecutive camera images.

The best angle α^* is computed as the median of the afore-mentioned distribution $\alpha^* = \text{median}\{\alpha_i\}$.

The inliers are then detected by using the reprojection error. Unlike the 1-point RANSAC, this algorithm is not iterative. Its computational complexity is linear in N_f.

3.2.3.4 Performance evaluation

We evaluated the performance of the proposed approaches on both synthetic and real data. We compare our 1-point RANSAC and Me-RE methods with the 5-point RANSAC [75] in simulations, and with the 5-point RANSAC [75] and the 8-point RANSAC [59] in experiments on real data.

Experiments on synthetic data We simulated different trajectories of a quadrotor moving in indoor scenarios (Figure 3.7). The simulations have been performed using the *Robotics and Machine Vision Toolbox* for Matlab [24].

To make our simulations as close as possible to the experiments, we simulated a quadrotor vehicle moving in indoor environment, equipped with a downlooking monocular camera. We randomly generated 1600 features on the ground plane (Figure 3.7). Note that no assumptions are made on the feature's depth.

We simulated a perspective camera with the same parameters of the one we used for the experiments and added a Gaussian noise with zero mean and standard deviation of 0.5 pixels to each image point. The vehicle was flying at the fix height of $2m$ above the ground. We generated a circular trajectory (easily repeatable in our flying arena) with a diameter of $1.5m$. The period for one rotation is $10s$. The camera framerate is $15Hz$, its resolution is 752 x 480. For the 1-point RANSAC and the Me-RE, we set a threshold of 0.5 pixels. For the 5-point ransac we set a minimum number of trials of 145 iterations, and a threshold of 0.5 pixels as well.

In Figure 3.8 we present the results obtained along the aforementioned trajectory in the case of perfect planar motion (the helicopter is flying always at the same height above the ground, and the *Roll* and *Pitch* angles are not affected by noise).

Figure 3.9 represents the results when the *Roll* and *Pitch* angles are affected by a Gaussian Noise with standard deviation of 0.3 degrees.

We evaluated as well the case in which the measure of the angle $dYaw$ is affected by a Gaussian Noise with standard deviation of 0.3 degrees. The relative results are shown in Figure 3.10

We finally evaluated the case of non perfect planar motion introducing a sinusoidal noise (frequency 4 rad/s and with amplitude of $0.02m$) on the z_W-component of motion of the vehicle. Figure 3.11 represents the relative results.

We can observe that the *Median + Reprojection Error* (Me-RE) performs always better than the 1-point RANSAC, and requires no iterations (its computational complexity is linear in the number of features).

In the case of perfect planar motion (Figure 3.8), the Me-RE algorithm finds more inliers than the 5-point RANSAC. The latter algorithm requires at least 145 iterations according to Figure 3.3 to insure a good performance.

When the variables *Roll*, *Pitch* and $dYaw$ are affected by errors (Figures 3.9 and 3.10), the performance of our algorithms drops, but they can still find almost the 50% of inliers.

As expected, if the vehicle's motion is not perfectly planar (Figure 3.11), the performances of the 1-point RANSAC and the Me-RE get worse. The oscillations that we can see in the plots are related to the fact that when the vehicle is approaching the ground, less features are in the field of view of its on-board camera.

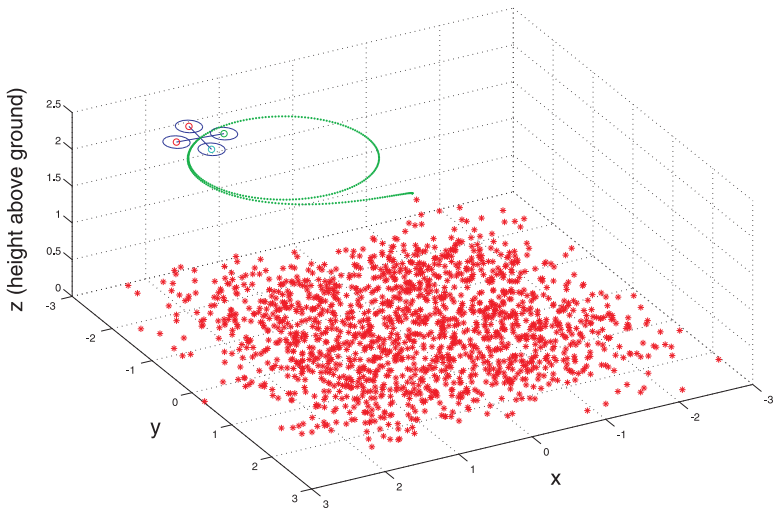


Figure 3.7: Synthetic scenario. The green line represents the trajectory and the red dots represent the simulated features.

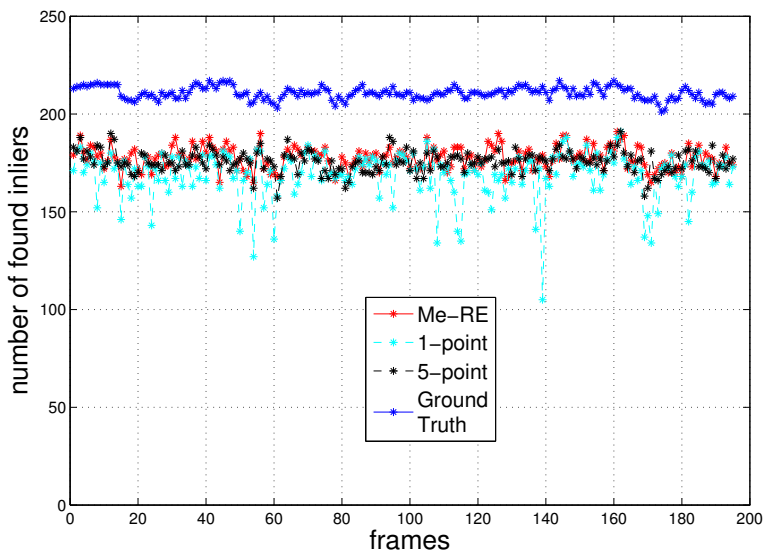


Figure 3.8: Number of found inliers by Me-RE (red), 1-point RANSAC (cyan), 5-point RANSAC (black), true number of inliers(blue) for a perfect planar motion.

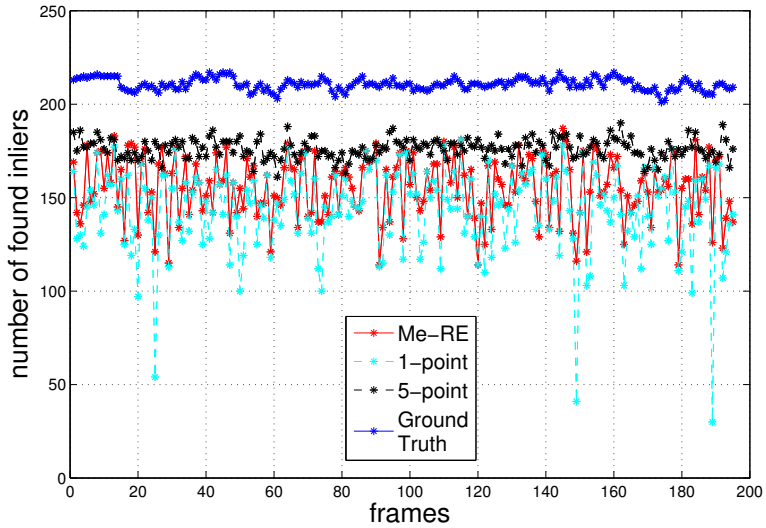


Figure 3.9: Number of found inliers by Me-RE (red), 1-point RANSAC (cyan), 5-point RANSAC (black), true number of inliers(blue) in presence of perturbations on the *Roll* and *Pitch* angles.

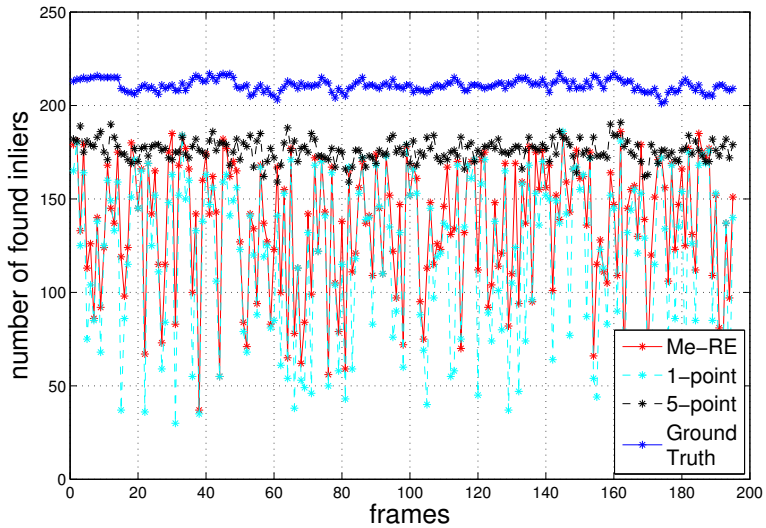


Figure 3.10: Number of found inliers by Me-RE (red), 1-point RANSAC (cyan), 5-point RANSAC (black), true number of inliers(blue) in presence of perturbations on the $dYaw$ angle.

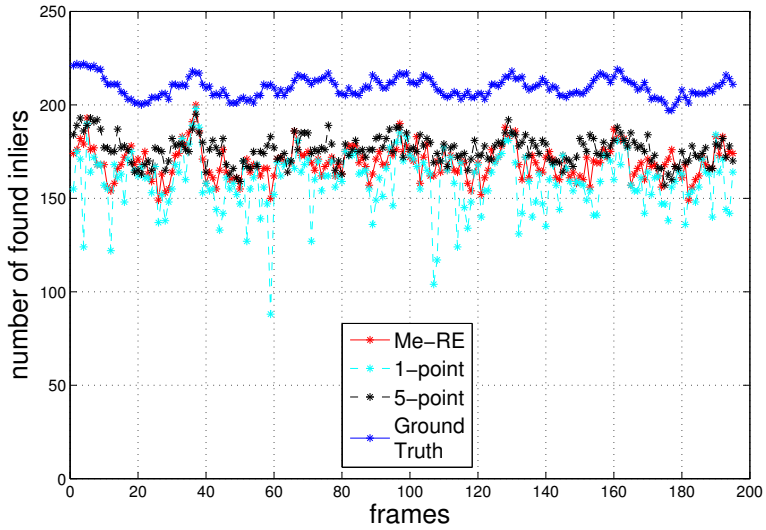


Figure 3.11: Number of found inliers by Me-RE (red), 1-point RANSAC (cyan), 5-point RANSAC (black), true number of inliers(blue) for a non-perfect planar motion ($s_1 = 0.02 * sin(8 * w_c \cdot t)$).

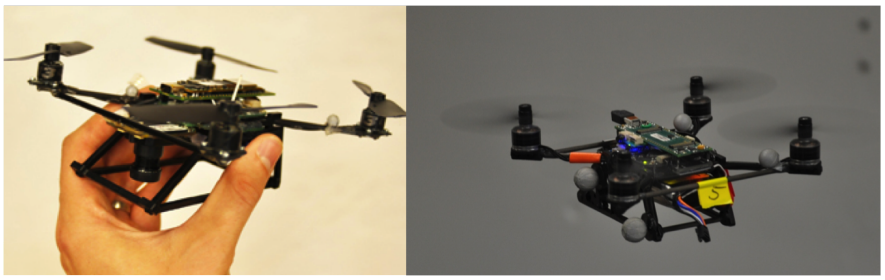


Figure 3.12: Nano quadrotor from KMelRobotics: a 150g and 18cm sized platform equipped with an integrated Gumstix Overo board and MatrixVision VGA camera.

Experiments on real data We tested our method on a nano quadrotor (Figure 3.12) [3] equipped with a MicroStrain 3DM-GX3 IMU (250 Hz) and a Matrix Vision mvBlueFOX-MLC200w camera (FOV: 112 deg).

The monocular camera has been calibrated using the *Camera Calibration Toolbox for Matlab* [17]. The extrinsic calibration between the IMU and the camera has been performed using the *Inertial Measurement Unit and Camera Calibration Toolbox* [57]. The dataset was recorded in our flying arena and ground

truth data have been recorded using an Optitrack motion capture system with submillimeter accuracy.

The trajectory has been generated using the TeleKyb Framework [36] (Figure 3.13). The vehicle followed a circular trajectory ($1.5m$ of diameter, period of $10s$) with fixed height above the ground of $1.5m$. We computed SURF features (Speeded Up Robust Feature). The feature detection and matching tasks has been performed using the *Machine Vision Toolbox* from [24].

To evaluate the performance of our methods, we compared the number of inliers found by the 1-point RANSAC and Me-RE methods with the number of inliers found by the 5-point RANSAC and the 8-point RANSAC methods. Figure 3.14 presents the result of this comparison.

We observe that in the interval $[380 : 490]$ the Me-RE algorithm has a very good performance (it finds even more inliers than the 5-points RANSAC). On the contrary the performance drops in the intervals $[350 : 380]$ and $[490 : 540]$. The last plot in Figure 3.15 shows the height of the vehicle above the ground during the trajectory. We can notice that in the interval $[380 : 490]$ the motion of the vehicle along the z-World axis is smoother than in the other intervals, therefore it affects less the performance of the 1-point and of the Me-RE methods.

Figure 3.16 shows the computation time of the compared algorithms, implemented in Matlab and run on an *Intel Core i7-3740QM Processor*. According to our experiments, the 5-point RANSAC takes about 67 times longer than the 8-point. The reason of this is that for each candidate point set, the 5-point RANSAC returns up to ten motion solutions and this involves both Singular Value Decomposition (SVD) and Groebner-basis decompositions. Instead, the 8-point RANSAC only returns 1 solution and has only one SVD, no Groebner-basis decomposition.

The Me-RE algorithm is not considered as a complete alternative to the 5-point RANSAC. However, thanks to its negligible computation time (Figure 3.16), it can be run at each frame. If the resulting number of inliers will be below a defined threshold, it will be more suitable to switch to the 5-point algorithm.

3.2.3.5 Conclusions

In this section we presented two algorithms (1-point RANSAC and *Median + Reprojection Error*) to perform outlier detection on computationally constrained micro aerial vehicles. The algorithms operate with the aid of an on-board IMU and assume that the vehicle's motion is locally planar. Both the algorithms rely on the reprojection error to look for inliers once that the essential matrix has been estimated, but the 1-point RANSAC needs at least 7 iterations to provide a satisfying solution, whereas the Me-RE's computational complexity is linear in the number of features and the performance is better. The Me-RE algorithm

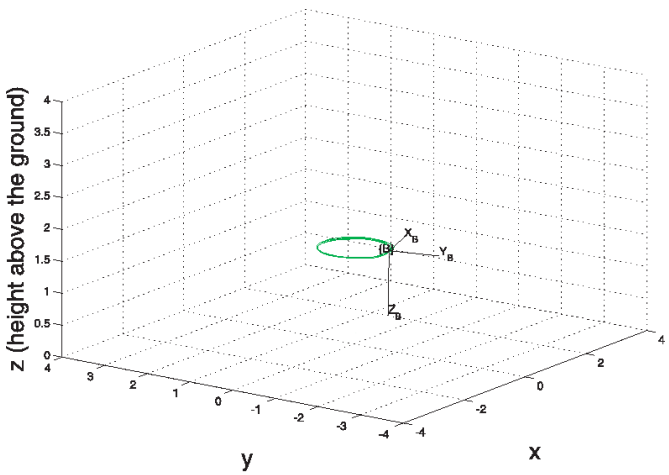


Figure 3.13: Plot of the real trajectory. The vehicle's body frame is depicted in black and the green line is the trajectory followed.

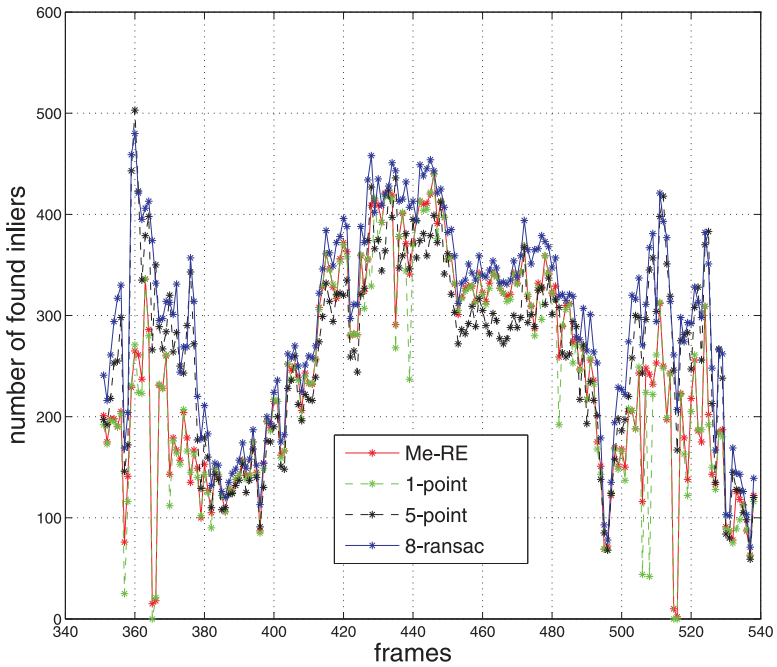


Figure 3.14: Number of found inliers by Me-RE (red), 1-point RANSAC (green), 5-point RANSAC (black), 8-point RANSAC (blue) along the trajectory depicted in Figure 3.13.

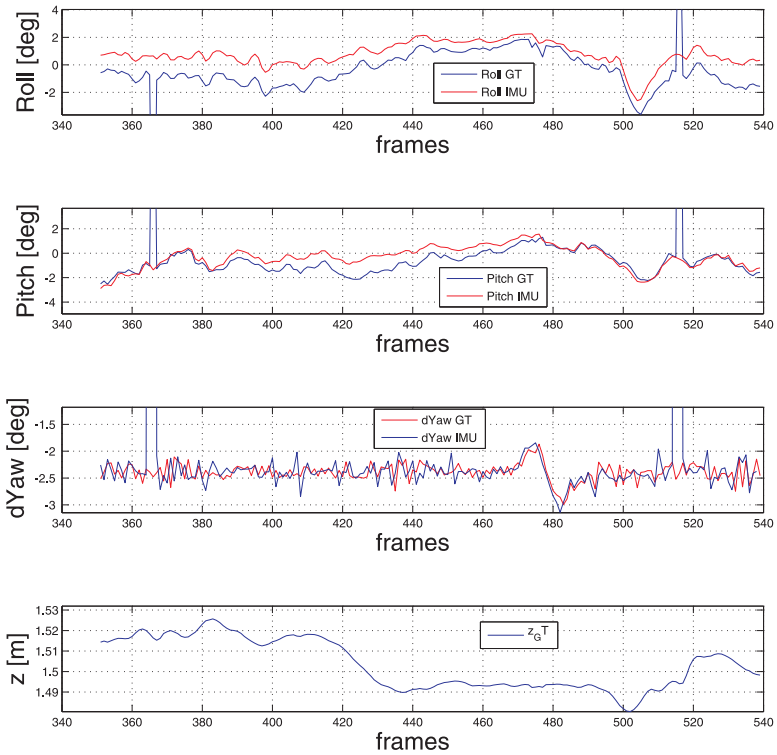


Figure 3.15: From the top to the bottom: *Roll*, *Pitch* and $dYaw$ angles [deg] estimated with the IMU (red) versus *Roll*, *Pitch* and $dYaw$ angles [deg] estimated with the Optitrack system (blue). The last plot shows the height of the vehicle above the ground (non perfect planarity of motion).

Algorithm	Me-Re	1-point	5-points	8-points
Time [s]	0.0028	0.0190	2.6869	0.0396

Figure 3.16: Computation time.

can therefore be a good replacement of the 5-point RANSAC when the motion of the vehicle is smooth and the camera framerate is high. The motion can then be refined applying standard methods [89], [38] to the remaining inliers. Considering that $\alpha*$ is estimated as the median of the distribution of the α computed from all the feature correspondences (3.10) , the standard deviation of this distribution can be considered as an index of reliability of the Me-RE algorithm.

3.2.4 2-point algorithm

In this section we present a novel method to perform the outlier rejection task between two different views of a camera rigidly attached to an Inertial Measurement Unit (IMU). Only two feature correspondences and gyroscopic data from IMU measurerments are used to compute the motion hypothesis [94]. By exploiting this 2-point motion parametrization, we propose two algorithms to remove wrong data associations in the feature-matching process for case of a 6DoF motion. We show that in the case of a monocular camera mounted on a quadrotor vehicle, motion priors from IMU can be used to discard wrong estimations in the framework of a 2-point-RANSAC based approach. The proposed methods are evaluated on both synthetic and real data.

3.2.4.1 Parametrization of the camera motion

Let us consider a camera rigidly attached to an Inertial Measurement Unit (IMU) consisting of three orthogonal accelerometers and three orthogonal gyroscopes. The transformation between the camera reference frame $\{C\}$ and the IMU frame $\{I\}$ can be computed using [57]. Without loss of generality, we can therefore assume that these two frames are coincident ($\{I\} \equiv \{C\}$). The $\Delta\phi$, $\Delta\theta$ and $\Delta\psi$ angles characterizing the relative rotation between two consecutive camera frames can be calculated by integrating the high frequency gyroscopic measurements, provided by the IMU. This measurement is affected only by a slowly-changing drift term and can safely be recovered if the system is in motion.

If $R_x(\Delta)$, $R_y(\Delta)$, $R_z(\Delta)$ are the orthonormal rotation matrices for rotations of Δ about the x-, y- and z-axes, the matrix

$$^{C_0}R_{C_1} = (R_x(\Delta\phi) \cdot R_y(\Delta\theta) \cdot R_z(\Delta\psi))^T \tag{3.11}$$

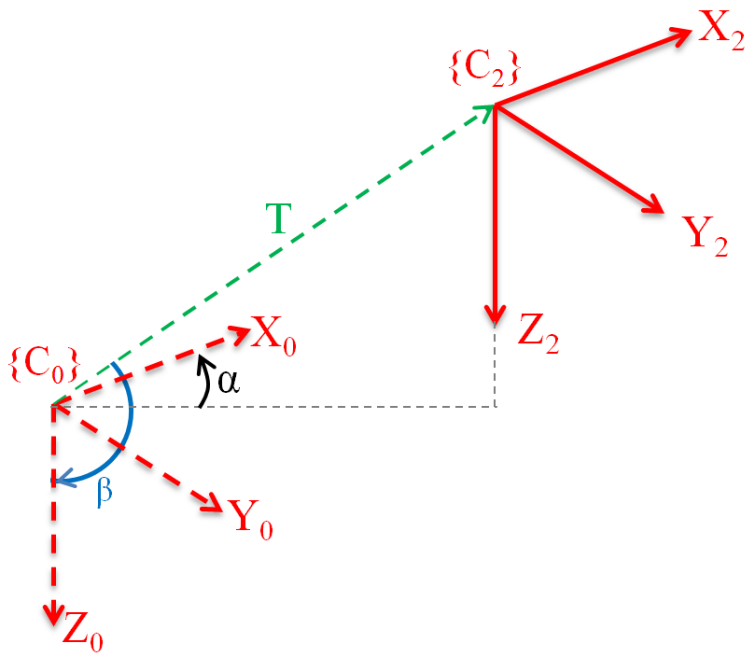


Figure 3.17: The reference frame C_0 and C_2 differ only for the translation vector T. $\rho = |T|$ and the angles α and β allow us to express the origin of the reference frame C_2 in the reference frame C_0.

allows us to virtually rotate the first camera frame $\{C_1\}$ into a new frame $\{C_0\}$ (Figure 3.4) having the same orientation of the second one $\{C_2\}$.

The matrix ${}^{C_0}R_{C_1}$ allows us to express the image coordinates relative to C_1 into the new reference frame C_0:

$$\mathbf{p_0} = {}^{C_0}R_{C_1} \cdot \mathbf{p_1}. \tag{3.12}$$

At this point, the transformation between $\{C_0\}$ and $\{C_2\}$ is a pure translation

$$\begin{aligned} \mathbf{T} &= \rho[s(\beta) \cdot c(\alpha) \quad -s(\beta) \cdot s(\alpha) \quad c(\beta)]^T \\ \mathbf{R} &= I_3, \end{aligned} \tag{3.13}$$

which depends only on the angles α and β and on the scale factor ρ. The essential matrix results therefore simplified:

$$E = [\mathbf{T}]_\times \mathbf{R} = \rho \begin{bmatrix} 0 & -c(\beta) & -s(\beta) \cdot s(\alpha) \\ c(\beta) & 0 & -s(\beta) \cdot c(\alpha) \\ s(\beta) \cdot s(\alpha) & s(\beta) \cdot c(\alpha) & 0 \end{bmatrix}. \tag{3.14}$$

With $s(\cdot)$ and $c(\cdot)$ we denote the $\sin(\cdot)$ and $\cos(\cdot)$ respectively. At this point, being $\mathbf{p_0} = [x_0 \quad y_0 \quad z_0]^T$ and $\mathbf{p_2} = [x_2 \quad y_2 \quad z_2]^T$, the coordinates of a feature matched between two different camera frames and backprojected onto the unit sphere, we impose the epipolar constraint according to (3.2) and we obtain the homogeneous equation that must be satisfied by all the point correspondences.

$$\begin{aligned} x_2(y_0c(\beta) + z_0s(\alpha)s(\beta)) - y_2(x_0c(\beta) - z_0c(\alpha)s(\beta)) + \\ -z_2(y_0c(\alpha)s(\beta) + x_0s(\alpha)s(\beta)) = 0. \end{aligned} \tag{3.15}$$

Equation (3.15) depends on two parameters (α and β). This means that the relative vehicle motion can be estimated using only two image feature correspondences that we will identify as $\mathbf{p_A}$ and $\mathbf{p_B}$, where $\mathbf{p_{i_j}} = [x_{i_j} \quad y_{i_j} \quad z_{i_j}]^T$ with $i = A, B$ and $j = 0, 2$ indicate the direction of the feature i in the reference frame j.

At this point, we can recover the angles α and β solving (3.15) for the features $\mathbf{p_A}$ and $\mathbf{p_B}$:

$$\begin{aligned} \alpha &= -tan^{-1}\left(\frac{c_4c_2 - c_1c_5}{c_4c_3 - c_1c_6}\right), \\ \beta &= -tan^{-1}\left(\frac{c_1}{c_2c(\alpha) + c_3s(\alpha)}\right), \end{aligned} \tag{3.16}$$

where

$$\begin{aligned} c_1 &= x_{A_2}y_{A_0} - x_{A_0}y_{A_2}, \\ c_2 &= -y_{A_0}z_{A_2} + y_{A_2}z_{A_0}, \\ c_3 &= -x_{A_0}z_{A_2} + x_{A_2}z_{A_0}, \\ c_4 &= x_{B_2}y_{B_0} - x_{B_0}y_{B_2}, \\ c_5 &= -y_{B_0}z_{B_2} + y_{B_2}z_{B_0}, \\ c_6 &= -x_{B_0}z_{B_2} + x_{B_2}z_{B_0}. \end{aligned} \tag{3.17}$$

Finally, without loss of generality, we can set the scale factor ρ to 1 and estimate the essential matrix according to (3.14).

3.2.4.2 Hough

The angles α and β are computed according to (3.16) from all the feature pairs matched between two consecutive frames and distant from each other more than a defined threshold (see Section 3.2.4.5). A distribution $\{\alpha_i, \beta_i\}$ with $i = 1, 2, \ldots, N$ is obtained, where N is a function of the position of the features in the environment.

To estimate the best angles α^* and β^*, we build a Hough Space (Figure 3.18) which bins the values of $\{\alpha_i, \beta_i\}$ into a grid of equally spaced containers. Considering that the angle β is defined in the interval $[0, \pi]$ and that the angle α is defined in the interval $[0, 2\pi]$, we set 360 bins for the variable α and 180 bins for the variable β. The number of bins of the Hough Space encodes the resolution

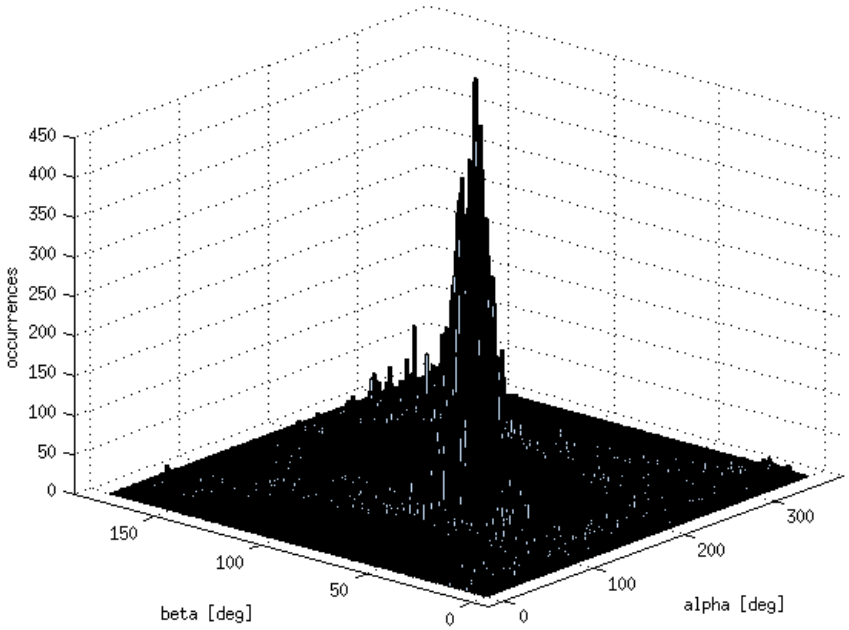


Figure 3.18: Hough Space in α and β computed with real data.

of the estimation.

The angles α^* and β^* are therefore computed as

$$< \alpha^*, \beta^* >= \text{argmax}\{H\},$$

where H is the Hough Space.

The factors that influence the distribution are the error on the estimation of the relative rotation, the image noise, and the percentage of outliers in the data. The closer we are to ideal conditions (no noise on the IMU measurements), the narrower will be the distribution. The wider is the distribution, the more uncertain is the motion estimate.

To detect the outliers, we calculate the reprojection error relative to the estimated motion model.

The camera motion estimation can be then refined processing the remaining subset of inliers with standard algorithms [89], [38].

3.2.4.3 2-point Ransac

Using (3.13) we compute the motion hypothesis that consists of the translation vector $\mathbf{T}$ and the rotation matrix $\mathbf{R} = \mathbf{I}_3$ by randomly selecting two features from

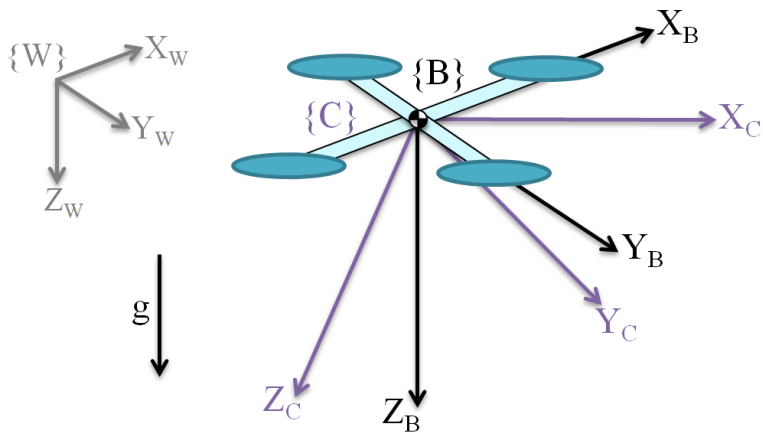


Figure 3.19: Notation.

the correspondence set. To have a good estimation, we check that the distance between the selected features is below a defined threshold (see Section 3.2.4.5). If it is not the case, we randomly select another pair of features. Constraints on the motion of the camera can be exploited to discard wrong estimations. The inliers are then computed using the reprojection error. The hypothesis that shows the highest consensus is considered to be the solution.

3.2.4.4 Quadrotor motion model

We consider a quadrotor equipped with a monocular camera and an IMU.

The vehicle body-fixed coordinate frame $\{B\}$ has its Z_B-axis pointing downward (following aerospace conventions [24]). The X_B-axis defines the forward direction and the Y_B-axis follows the right-hand rule.

Without loss of generality we can consider the IMU reference frame $\{I\}$ coinciding with the vehicle body frame $\{B\}$.

The modelization of the vehicle rotation in the World frame $\{W\}$ follows the $Z-Y-X$ Euler angles convention: being ϕ, θ, ψ respectively the *Roll*, *Pitch* and *Yaw* angles of the vehicle, to go from the World frame to the Body frame, we first rotate about z_W axis by the angle ψ, then rotate about the intermediate y-axis by the angle θ, and finally rotate about the X_B-axis by the angle ϕ.

The transformation between the camera reference frame $\{C\}$ and the IMU frame $\{I\}$ can be computed using [57]. Without loss of generality, we can therefore assume that also these two frames are coincident ($\{I\} \equiv \{C\} \equiv \{B\}$).

A quadrotor has 6DoF, but its translational and angular velocity are strongly coupled to its attitude due to dynamic constraints. If we consider a coordinate

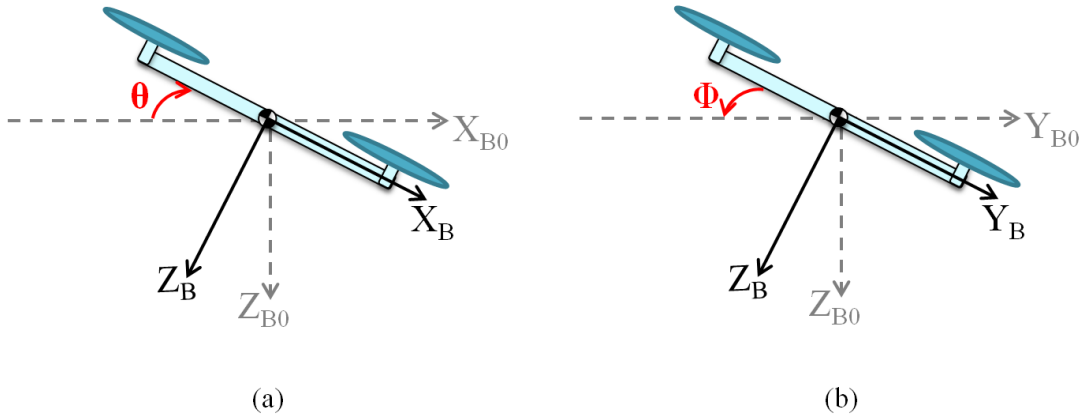


Figure 3.20: Motion constraints on a quadrotor relative to its orientation. $\Delta\phi > 0$ implies a movement along Y_{B_0} positive direction, $\Delta\theta < 0$ implies a movement along Y_{B_0} positive direction.

frame $\{B_0\}$ with the origin coincident with the one of the vehicle's body frame $\{B\}$ and the X_{B_0} and Y_{B_0} axes parallel to the ground, we observe that, in order to move in the X_{B_0} direction, the vehicle must rotate about the y-axes axis (*Pitch* angle), while, in order to move in the Y_{B_0} direction, it must rotate about the x-axis (*Roll* angle) (Figure 3.20).

These motion constraints allow us to discard wrong estimations in a RANSAC based outlier detection approach. By looking at the relation between the x and y component of the estimated translation vector and the $\Delta\phi$, $\Delta\theta$ angles provided by the IMU measurements (the same used in (3.11)), we are able to check the consistency of the motion hypothesis. If the estimated motion satisfies the condition

$$\begin{array}{c} ((|\Delta\phi| > \epsilon)\&(\Delta\phi \cdot T_y > 0)) \quad \| \\ ((|\Delta\theta| > \epsilon)\&(\Delta\theta \cdot T_x < 0)) \quad \| \\ ((|\Delta\phi| < \epsilon)\&(|\Delta\theta| < \epsilon)), \end{array} \tag{3.18}$$

we count the number of inliers (the number of correspondences that satisfy the motion hypothesis according to a predefined threshold) by using the reprojection error, otherwise we select another feature pair. The condition in (3.18) is satisfied if the x and y components of the motion hypothesis are coherent with the orientation of the vehicle. If both the angles $\Delta\phi$ and $\Delta\theta$ are below the threshold ϵ, we cannot infer nothing about the motion and we proceed in the evaluation of the model hypothesis using the reprojection error.

The value of the threshold ϵ (see 3.2.4.5) is a function of the vehicle dynamics and of the controller used.

Using (3.1) and considering $p = 0.99$ and $\varepsilon = 0.5$, we calculate the minimum number of iterations necessary to guarantee a good performance to our algorithm and we set it to 16.

3.2.4.5 Performance evaluation

To evaluate the performance of our algorithms, we run simulations and experiments on real data. We compared the proposed approaches with the 5-point RANSAC [75] on synthetic data, and with the 5-point RANSAC [75] and the 8-point RANSAC [59] on real data.

Experiments on synthetic data We built a synthetic scenario for our simulation by using the *Robotics and Machine Vision Toolbox* for Matlab [24]. We simulated a quadrotor equipped with a downlooking monocular camera and an IMU, moving in an indoor environment (Figure 3.21). Random features were generated without any assumption on the structure of the environment.

The on-board downlooking monocular camera was simulated as a perspective camera with the same intrinsic parameters of the camera that we used in the experiments. A white gaussian noise with a standard deviation of 0.5 pixels was added to each extracted feature.

We generated a trajectory consisting of a take-off and of a constant-height maneuver. The camera framerate is $15Hz$, its resolution is 752 x 480. For the reprojection error in the 2-point RANSAC and in the Hough algorithm, we set a threshold of 0.5 pixels. For the 5-point RANSAC, we set the minimum number to trials to 145 iterations, and the threshold to 0.5 pixels for the reprojection error.

Figure 3.22 shows the results of a simulation run along the trajectory depicted in Figure 3.21, in the ideal case of no noisy IMU measurements. The helicopter takes off and performs a constant height maneuver.

In Figure 3.23, we present the results related to simulations where the quantities $\Delta\phi$, $\Delta\theta$ and $\Delta\psi$ are affected by a Gaussian noise with standard deviation of 0.3 degrees. Those errors do not affect the performance of the 5-point algorithm (that does not use IMU readings to compute the motion hypothesis). In this case, the Hough and the 2-point RANSAC approaches can still detect more than half of the inliers. The motion hypothesis can then be computed on the obtained set of correspondences by using standard approaches [89], [38].

In Figure 3.24, we present the results related to simulations where the quantities $\Delta\phi$ and $\Delta\theta$ are affected by a Gaussian noise with standard deviation of 0.3 degrees and in Figure 3.25 only the angle $\Delta\psi$ is affected by a Gaussian noise with standard deviation of 0.3 degrees. These two plots show that errors on rotations about the camera optical axis (that in our case coincides with rotations about

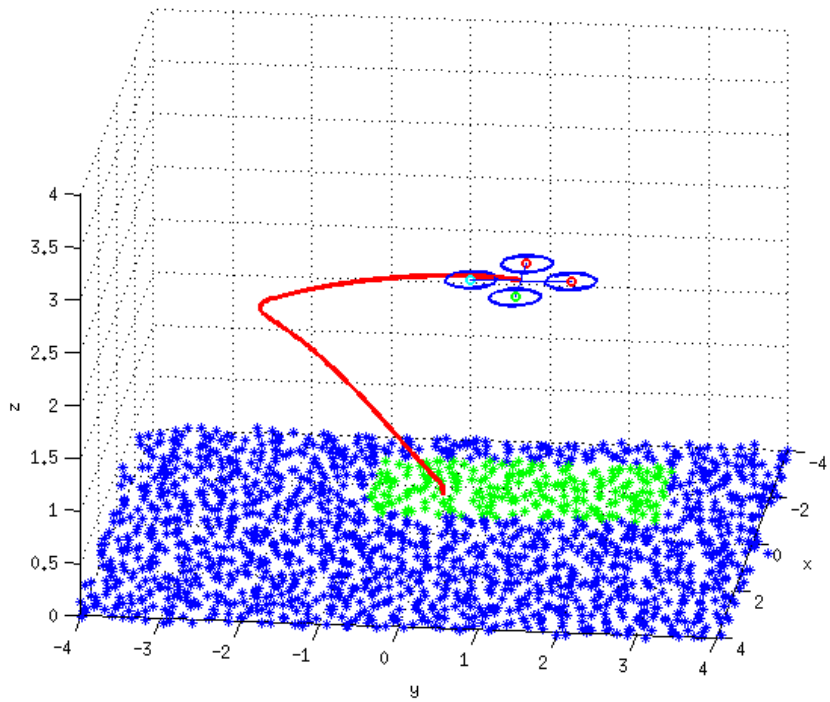


Figure 3.21: Synthetic scenario. The red line represents the trajectory and the blue dots represent the simulated features. The green dots are the features in the current camera view.

the vehicle Z_B axis, i.e. errors on $\Delta\psi$) affects more the performances of both the algorithms than errors on $\Delta\phi$ and $\Delta\theta$.

Experiments on real data The proposed approaches are tested on our nano quadrotor (Figure 3.12) [3] equipped with a MicroStrain 3DM-GX3 IMU (250 Hz) and a Matrix Vision mvBlueFOX-MLC200w camera (FOV: 112 deg and a resolution of 752 x 480).

The monocular camera calibration has been performed using the *Camera Calibration Toolbox for Matlab* [17].

To extrinsically calibrate the IMU and the camera, we used the *Inertial Measurement Unit and Camera Calibration Toolbox* [57].

To validate the performance of our methods, we flew the quadrotor in our flying arena, equipped with an Optitrack motion capture system with submillimeter accuracy. The trajectory consisted of a take-off and a constant-height maneuver above the ground, as shown in Figure 3.26 and was generated using

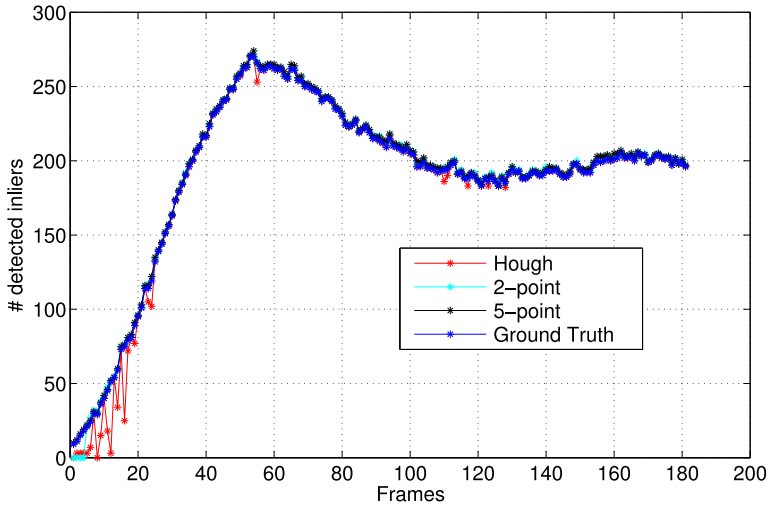


Figure 3.22: The IMU measurements are not affected by noise (ideal conditions).

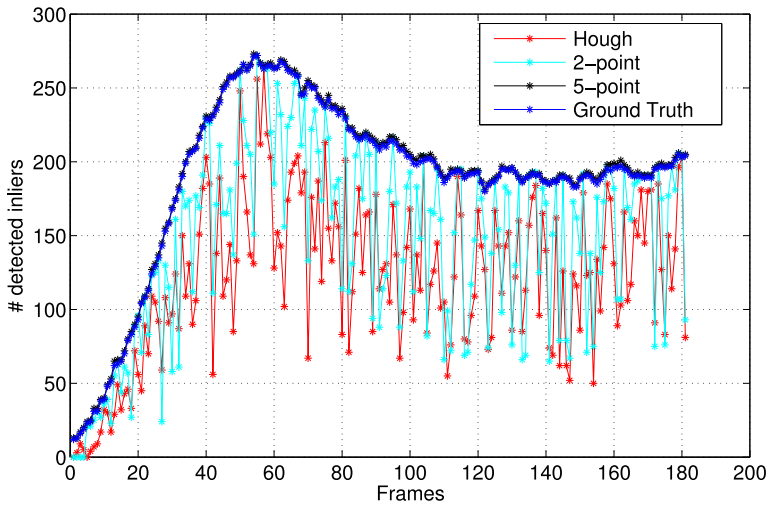


Figure 3.23: The angles $\Delta\phi$, $\Delta\theta$ and $\Delta\psi$ are affected by noise.

the TeleKyb Framework [36]. We recorded a dataset composed of camera images, IMU measurements and ground truth data provided by the Optitrack.

We processed our dataset with SURF features, matching them in consecutive camera frames. We run the 8-point RANSAC method on each correspondences set to have an additional term of comparison.

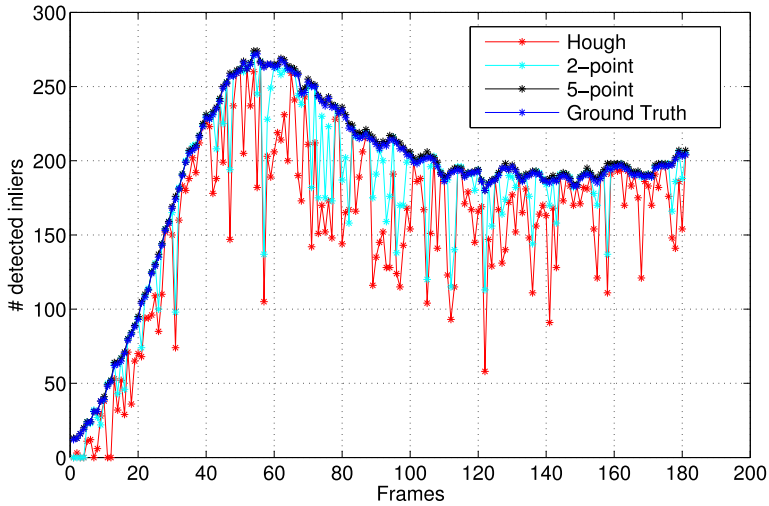


Figure 3.24: Only the angles $\Delta\phi$ and $\Delta\theta$ are affected by noise.

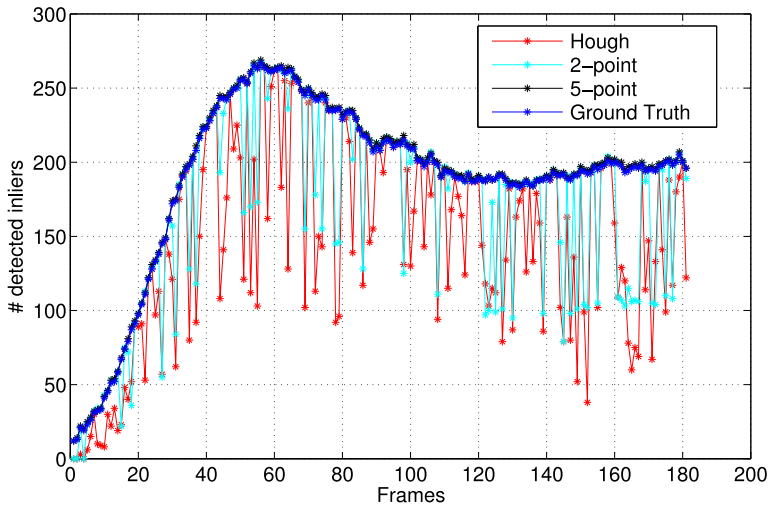


Figure 3.25: Only the angle $\Delta\psi$ is affected by noise.

To evaluate the performance of our methods, we compared the number of inliers detected using the Hough and the 2-point RANSAC methods with 5-point and an 8-point RANSAC. For the 2-point RANSAC we set $\epsilon = 0.1$ deg. The results of this comparison are shown in Figure 3.27.

Figure 3.29 shows the error characterizing the estimated relative rotation be-

tween two consecutive camera frames obtained by IMU measurements and the ground truth values.

Looking at both Figure 3.27 and Figure 3.29, we can notice that the smaller are the errors on the angles estimations, the higher is the number of inliers detected by the Hough and the 2-point RANSAC method.

Our algorithms and the algorithms that we used for the comparison, are implemented in Matlab and run on an *Intel Core i7-3740QM Processor*. We summarize their computation time in Figure 3.28. We can notice that the computation time of the 5-point RANSAC is almost 67 times the computation time of the 8-point RANSAC. This is due to the fact that the 5-points returns up to 10 motion solutions for each candidate set. Singular Value Decomposition (SVD) and Groebner-basis decompositions are involved and this explains the high computation time.

The computation time of the Hough algorithm is function of the number of feature pairs used to compute the distribution in Figure (3.18). In our experiments, we choose all the feature pairs distant more than a defined threshold one to each other. We experimentally set this threshold to 30 degrees on the unit sphere.

3.2.4.6 Conclusions

In this section, we proposed two algorithms (Hough and 2-point RANSAC) to address the outlier rejection task systems equipped with a monocular camera rigidly attached to an IMU. We used a quadorotor micro aerial vehicle as platform to demonstrate the validity of our results. We show that the relations between the vehicle's translational and angular velocity and its attitude can be exploited in order to discard wrong estimations in the framework of a RANSAC-based approach.

Both methods rely on on-board IMU measurements to calculate the relative rotation between two consecutive camera frames and to the reprojection error to detect the inliers. The two algorithms differ in the way to compute the motion hypothesis.

The computation time of the Hough algorithm (Figure 3.28) is function of the number of feature pairs used to compute the distribution in Figure (3.18). Smart policies for the choice of the pairs of features to use (based for example on the feature positions in the image plane and not only on their relative position) can be used in order to reduce the computational complexity of the approach.

Experimental results show that the 2-point RANSAC algorithm can be a good replacement of the 5-point RANSAC. The motion hypothesis can always be refined by processing the found inliers with classic methods [89], [38].

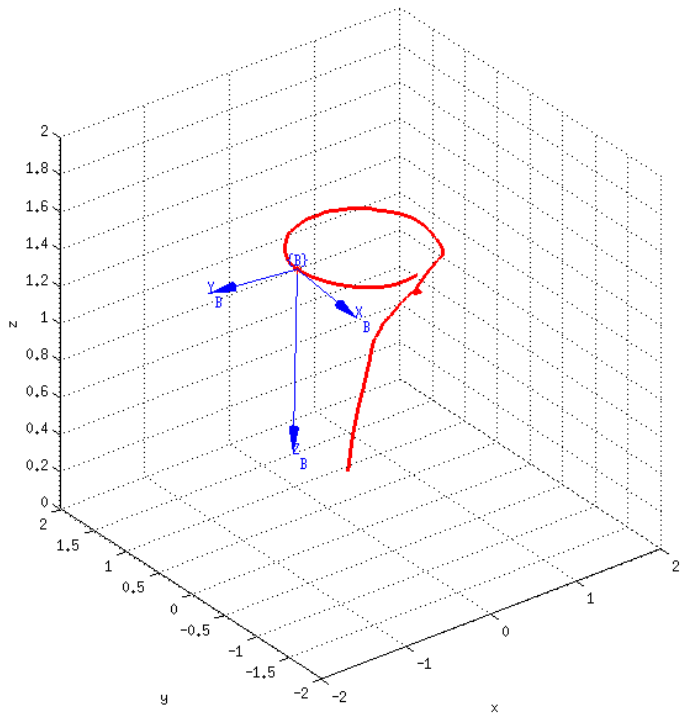


Figure 3.26: Real scenario. The vehicle body frame is represented in blue, while the red line represents the followed trajectory.

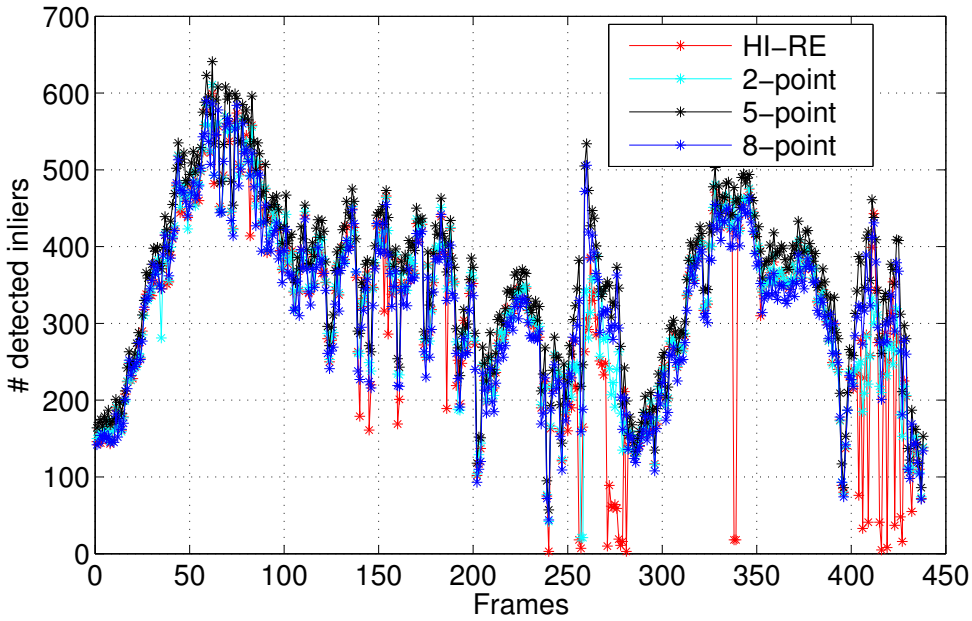


Figure 3.27: Number of inliers detected with the Hough approach (red), the 2-point RANSAC (cyan), the 5-point RANSAC (black) and the 8-point RANSAC (blue) along the trajectory depicted in Figure 3.26.

Algorithm	Hough	2-points	5-points	8-points
Time [s]	0.498	0.048	2.6869	0.0396

Figure 3.28: Computation time.

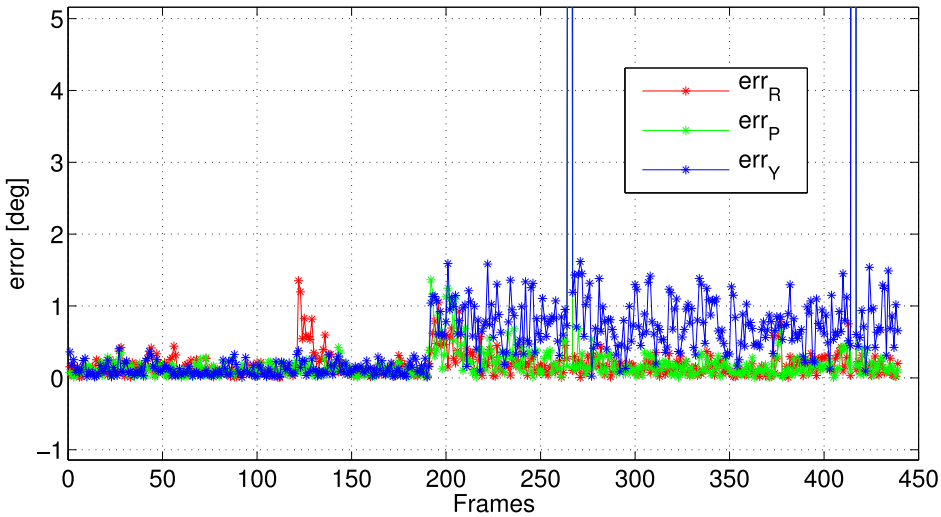


Figure 3.29: Errors between the relative rotations $\Delta\phi$ (err_R), $\Delta\theta$ (err_P), $\Delta\psi$ (err_Y) estimated with the IMU and estimated with the Optitrack.

Chapter 4

Pose estimation

Contents

4.1 Filtering approaches and closed form solutions 60

4.2 Virtual patterns 61

4.2.1 The System 62

4.2.2 The method 63

4.2.3 Performance Evaluation 71

4.2.4 Conclusion 75

4.3 Virtual features 77

4.3.1 The System 78

4.3.2 Camera-laser module calibration 81

4.3.3 Observability Properties 82

4.3.4 Local Decomposition and Recursive Estimation 86

4.3.5 Performance Evaluation 88

4.3.6 Conclusion 96

This chapter introduces the last two contributions of this dissertation. In the first section we provide an overview of the visual-aided inertial pose estimation problem, with an emphasis on aerial navigation. Two low computational complexity algorithms to face the pose estimation problem are presented. The first method requires inertial measurements from an IMU, and the observation of three features in the scene, under the planar ground assumption. It does not require any known pattern, but it exploits the geometrical constraints of a virtual one: three features form a triangle.

The latter contribution faces the problem of featureless or dark environments. An aerial vechicle, moving in the surrounding of a planar surface is considered. In order to reduce the computational burden required to perform the feature extraction and matching task, a virtual feature is introduced by equipping the vehicle with a laser pointer (in addition to a monocular camera and an IMU). The laser spot produced by the laser pointer on the planar surface is the unique point feature observed. We identified the physical quantities that can be determined with this setup, we analytically derived the link between those quantities and the sensor data, and we estimated them with an Extended Kalman Filter.

4.1 Filtering approaches and closed form solutions

The problem of fusing vision and inertial data has been extensively investigated in the past. In [23], a tutorial introduction to the vision and inertial sensing is presented. This work provides a biological point of view and it illustrates how vision and inertial sensors have useful complementarities allowing them to cover the respective limitations and deficiencies. Specifically, as it has been derived very recently in [65], the fusion of these sensors allows us to obtain the speed and the scale factor in closed form, allowing real time applications and robustness with respect to kidnapping.

In [10], inertial and visual sensors are used to perform egomotion estimation. The sensor fusion is obtained by an Extended Kalman Filter (EKF) and by an Unscented Kalman Filter (UKF). The approach proposed in [34] extends the previous one by also estimating the structure of the environment where the motion occurs. In particular, new landmarks are inserted on line into the estimated map. This approach has been validated by conducting experiments in a known environment where a ground truth was available. Also, in [97] an EKF has been adopted. In this case, the proposed algorithm estimates a state containing the robot speed, position and attitude, together with the inertial sensor biases and the location of the features of interest. In the framework of airbone SLAM, an EKF has been adopted in [46] to perform $3D-$SLAM by fusing inertial and vision measurements. It was observed that any inconsistent attitude update severely affects any SLAM solution. The authors proposed to separate attitude update from position and velocity update. Alternatively, they proposed to use additional velocity observations, such as air velocity observation. More recently, a vision based navigation approach in unknown and unstructured environments has been suggested [14].

Recent works investigate the observability properties of the vision-aided inertial navigation system [43], [45], [44], [63], [65], [66] and [72]. In particular, in

[65], the observable modes are expressed in closed-form in terms of the sensor measurements acquired during a short time-interval.

Visual UAV pose estimation in GPS-denied environments is still challenging. Many implementations rely on visual markers, such as patterns or blobs, located in known positions [105], [28], [21]. Those approaches have the drawback that can work only in structured environment. In [40] Visual-Inertial Attitude Estimation is performed using image line segments for the correction of accumulated errors in integrated gyro rates when an unmanned aerial vehicle operates in urban areas. The approach will not work in environments that do not present a strong regularity in structure.

In [101], [99] the authors developed a very robust Vision Based Navigation System for micro helicopters. Their pose estimator is based on a monocular VSLAM framework (PTAM, Parallel Tracking and Mapping [48]). This software was originally developed for augmented reality and improved with respect to robustness and computational complexity. The resulting algorithm can be used in order to make a monocular camera a real-time on-board sensor for pose estimates. This allowed the first aerial vehicle that uses on-board monocular vision as a main sensor to navigate through an unknown GPS-denied environment and independently of any external artificial aids [100], [99].

Natraj et al. [74] proposed a vision based approach, close to structured light, for roll, pitch and altitude estimation of UAV. They use a fisheye camera and a laser circle projector, assuming that the projected circle belongs to a planar surface. The latter must be orthogonal to the gravity vector in order to allow the estimation of the aforementioned quantities. The attitude estimation of the planar surface becomes crucial in order to extend the operational environment of UAVs. Shipboard operations, search and rescue cooperation between ground and aerial robots, low altitude manoeuvres, require to attenuate the position error and to track the platform attitude.

4.2 Virtual patterns

In this section we propose a new approach to perform MAV localization by only using the data provided by an Inertial Measurement Unit (IMU) and a monocular camera [91]. The goal of our investigation is to find a new pose estimator which minimizes the computational complexity. We focus our attention on the problem of relative localization, which makes possible the accomplishment of many important tasks (e.g. hovering, autonomous take off and landing). In this sense, we minimize the number of point features which are necessary to perform localization. While 2 point features is the minimum number which provides full observability, by adding an additional feature, the precision is significantly im-

proved, provided that the so-called planar ground assumption is honoured. This assumption has recently been exploited on visual odometry with a bundle adjustment based method [47]. The proposed method does not use any known pattern but only relies on three natural point features belonging to the same horizontal plane, which form therefore a virtual pattern (a triangle). It is based on a closed solution which provides the vehicle pose from a single camera image, once the roll and the pitch angles are obtained by the inertial measurements. The first step of the approach provides a first estimate of the roll and pitch (through the IMU data) and then the vehicle heading by only using two of the three point features and a single camera image. In particular, the heading is defined as the angle between the MAV and the segment made by the two considered point features. Then, the same procedure is repeated two additional times, i.e., by using the other two pairs of the three point features. In this way, three different heading angles are evaluated. On the other hand, these heading angles must satisfy two geometrical constraints, which are fixed by the angles characterizing the triangle made by the three point features. These angles are estimated in parallel by an independent Kalman Filter. The information contained in the geometrical constraints is then exploited by minimizing a suitable cost function. This minimization provides a new and very precise estimate of the roll and pitch and consequently of the yaw and the vehicle position.

4.2.1 The System

Let us consider an aerial vehicle equipped with a monocular camera and IMU sensors. We assume that the transformation among the camera frame and the IMU frame is known (we can assume that the vehicle frame coincides with the camera frame).

We assume that three reliable point-features are detected on the ground (i.e. they belong to the same horizontal plane). As we will see, two is the minimum number of features necessary to perform localization. Figure 4.1 shows our global frame G, which is defined by only using two features, P_1 and P_2. First, we define P_1 as the origin of the frame. The z_G-axis coincides with the gravity vector but with opposite direction. Finally, P_2 defines the x_G-axis [1].

Then, by applying the method in [65], the distance between these point features can be roughly determined by only using visual and inertial data (specifically, at least three consecutive images containing these points must be acquired).

[1]Note that the planar assumption is not necessary to define a global frame. It is sufficient that P_1 and P_2 do not lie on the same vertical axis (defined by the gravity). The X_G-axis can be defined assuming that P_2 belongs to the $x_G - z_G$-plane. In other words, P_2 has zero y_G coordinate.

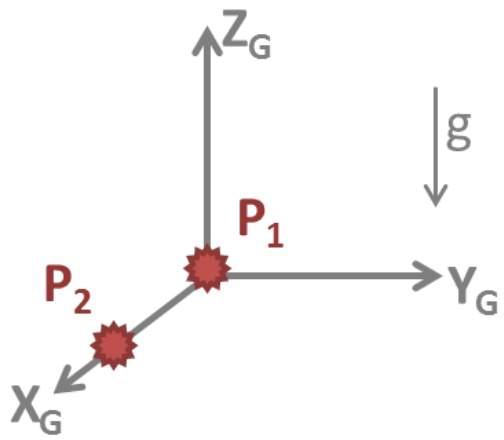


Figure 4.1: Global frame. Two is the minimum number of point features which allows us to uniquely define a global reference frame. P_1 is the origin, the x_G-axis is parallel to the gravity and P_2 defines the x_G-axis

4.2.2 The method

The first step of the method consists in estimating the Roll and the Pitch angles. This is performed by an Extended Kalman Filter (EKF) which estimates the gravity in the local frame by only using inertial data. Once the direction of the gravity vector in the local frame is estimated, the Roll and Pitch angles are obtained. The second step returns the Yaw angle and the position of the vehicle taking as input the Roll and Pitch angles and a single camera image (*3p-algorithm*, Section 4.2.2.2). The core of the *3p-algorithm* is the *2p-algorithm*, which is described in the next section.

4.2.2.1 2p-Algorithm

This algorithm needs only two point features in a single camera image, and the Roll and Pitch angles estimated from IMU measurements. Figure 4.2 represents a schematic of the algorithm.

For each feature, the camera provides its position in the local frame up to a scale factor. The knowledge of the absolute Roll and Pitch, allows us to express the position of the features in a new vehicle frame N, which Z_N-axis is parallel to the gravity vector. Figure 4.3 displays all the reference frames: the global frame G, the vehicle frame (represented by V) and the new vehicle frame N. Our goal is to determine the coordinates of the origin of the vehicle frame in the global frame and the orientation of the X_N-axis with respect to the x_G-axis (which corresponds

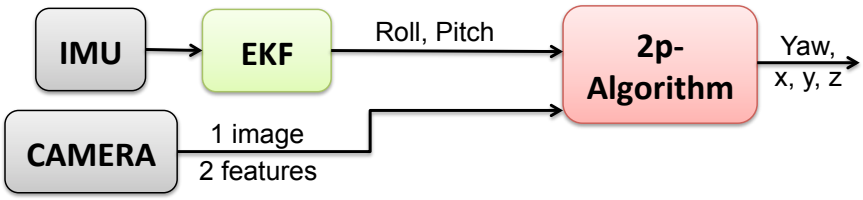


Figure 4.2: The 2p-algorithm.

to the Yaw angle of the vehicle in the global frame).

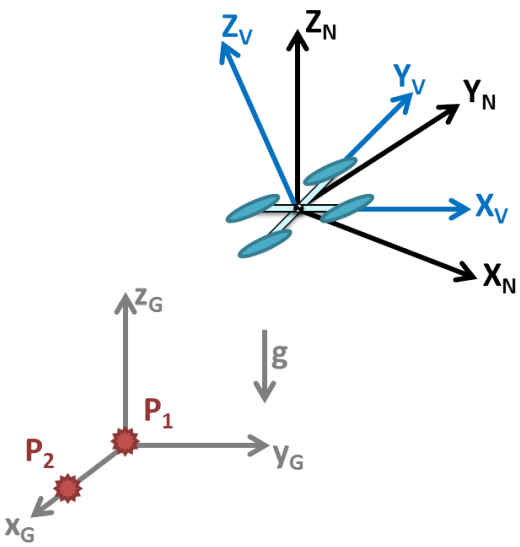


Figure 4.3: The three reference frames adopted in our derivation.

Let us denote with $[x_1, y_1, z_1]^T$ and $[x_2, y_2, z_2]^T$ the coordinates of P_1 and P_2 in the new local frame. The camera provides $\mu_1 = \frac{x_1}{z_1}$, $\nu_1 = \frac{y_1}{z_1}$, $\mu_2 = \frac{x_2}{z_2}$ and $\nu_2 = \frac{y_2}{z_2}$. Additionally, the camera also provides the sign of z_1 and z_2[1].

Since the Z_N-axis has the same orientation as the z_G-axis, and the two features P_1 and P_2 belongs to a plane perpendicular to the gravity vector, $z_1 = z_2 = -z$, where z is the position of the origin of the vehicle frame in the global frame. We

[1]For a camera with a field of view smaller than $180deg$ the $z-$component is always positive in the original camera frame.

obtain:

$$P_1 = -z \begin{bmatrix} \mu_1 \\ \nu_1 \\ 1 \end{bmatrix} \qquad P_2 = -z \begin{bmatrix} \mu_2 \\ \nu_2 \\ 1 \end{bmatrix} \tag{4.1}$$

Let us denote by D the distance between P_1 and P_2. We have:

$$z = \pm \frac{D}{\sqrt{\Delta\mu_{12}^2 + \Delta\nu_{12}^2}} \tag{4.2}$$

with $\Delta\mu_{12} \equiv \mu_2 - \mu_1$ and $\Delta\nu_{12} \equiv \nu_2 - \nu_1$. In other words, z can be easily obtained in terms of D. The previous equation provides z up to a sign. This ambiguity is solved considering that the camera provides the sign of z_1 and z_2. Then, we obtain $x_1 = -z\mu_1$, $y_1 = -z\nu_1$, $x_2 = -z\mu_2$ and $y_2 = -z\nu_2$. It is therefore easy to obtain $\alpha = \arctan 2(\Delta\nu_{12},\ \Delta\mu_{12})$ (Figure 4.4). Hence,

$$Yaw = -\alpha = -\text{atan}(\Delta\nu_{12}/\ \Delta\mu_{12}) \tag{4.3}$$

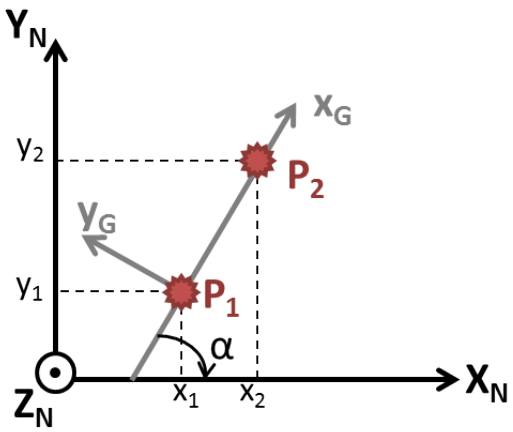


Figure 4.4: The yaw angle $(-\alpha)$ is the orientation of the X_N-axis in the global frame.

Finally we obtain the coordinates of the origin of the vehicle frame in the global frame,

$$\begin{aligned} x &= -\cos(\alpha)\ x_1 - \sin(\alpha)\ y_1 \\ y &= \sin(\alpha)\ x_1 - \cos(\alpha)\ y_1 \\ z &= \pm \tfrac{D}{\sqrt{\Delta\mu_{12}^2 + \Delta\nu_{12}^2}} \end{aligned} \tag{4.4}$$

Note that the position x, y, z is obtained in function of the distance D. Specifically, the position scales linearly with D. As previously said, a rough knowledge of this distance is provided by using the method in [65] and described in section 4.2.2.3. We remark that a precise knowledge of this distance is not required to accomplish tasks like hovering on a stable position.

4.2.2.2 3p-Algorithm

The three features form a triangle in the (x_G, y_G)-plane. For the sake of clarity, we start our analysis supposing that we know the angles characterizing the triangle (γ_1 and γ_2 in Figure 4.5). Then, we will show how we estimate on line these angles (Section 4.2.2.4).

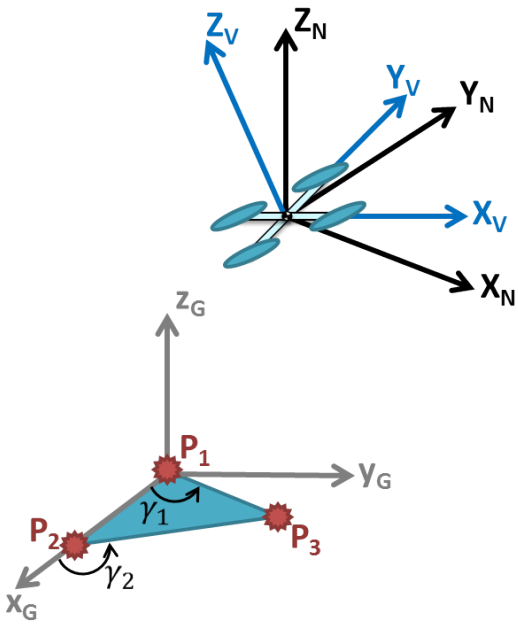


Figure 4.5: The triangle made by the 3 point features.

We run the $2p-$algorithm three times, respectively with the sets of features (P_1, P_2), (P_1, P_3) and (P_2, P_3) as input. We obtain three different angles. Yaw_{12} is the Yaw of the vehicle in the global frame given in (4.3) while the other expressions are:

$$\begin{aligned} Yaw_{12} &= -\text{atan}(\Delta\nu_{12}/\Delta\mu_{12}) \\ Yaw_{13} &= -\text{atan}(\Delta\nu_{13}/\Delta\mu_{13}) \\ Yaw_{23} &= -\text{atan}(\Delta\nu_{23}/\Delta\mu_{23}) \end{aligned} \tag{4.5}$$

The three above-mentioned angles must satisfy the following constraints:

$$\begin{aligned} \gamma_1 &= Yaw_{13} - Yaw_{12} \\ \gamma_2 &= Yaw_{23} - Yaw_{12} \end{aligned} \tag{4.6}$$

Let us denote the known values of these angles with γ_1^0 and γ_2^0. We correct the estimation of the roll and pitch angles by exploiting these constraints. We solved a nonlinear least-squares problem minimizing the following cost function:

$$c(\zeta) = [(Yaw_{13} - Yaw_{12} - \gamma_1^0)^2 + (Yaw_{23} - Yaw_{12} - \gamma_2^0)^2] \tag{4.7}$$

in which the variables Yaw_{ij} are nonlinear functions of $\zeta = [Roll, Pitch]^T$.

Once the least-squares algorithm finds the Roll and Pitch angles that minimize the cost function, we can estimate the Yaw angle and the coordinates x, y and z as described in $2p-$algorithm (Figure 4.6).

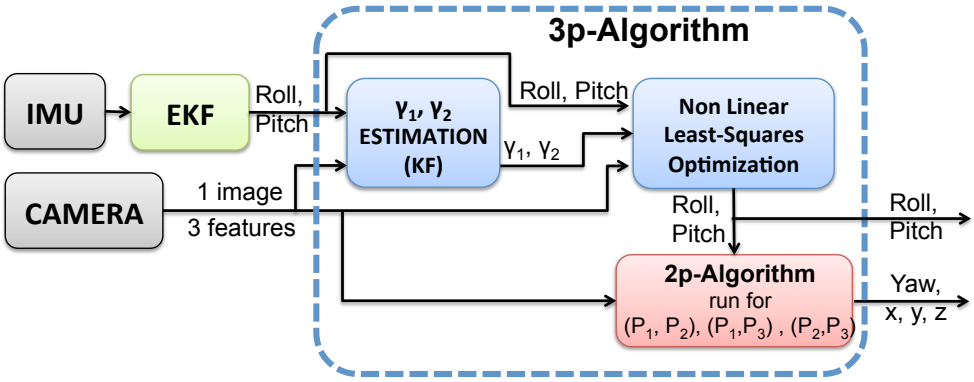


Figure 4.6: Flow chart of the proposed pose estimator

4.2.2.3 Scale factor initialization

Recent works on visual-inertial structure from motion have demonstrated its observability properties [43], [45], [44], [63], [65], [68], [66], and [72]. It has been proved that the states that can be determined by fusing inertial and visual information are: the system velocity, the absolute scale, the gravity vector in the local frame, and the biases that affects the inertial measurements. The work in [65] expresses all the observable modes at a given time T_{in} in closed-form and only in function of the visual and inertial measurements registered during the time interval $[T_{in}, T_{fin}]$.

In the following, we will adopt lower-case letters to denote vectors in the global frame (e.g. the gravity is $g = [0, 0, -g]^T$, where $g \simeq 9.8ms^{-1}$). Lower-case letters

will be used to denote vectors in the camera frame. Since this local frame is time dependent, we adopt the following notation: $W_t(\tau)$ will be the vector with global coordinates $w(\tau)$ in the local frame at time t. Additionally, we will denote with $C_{t_1}^{t_2}$ the matrix which characterizes the rotation occurred during the time interval $(t_1,\ t_2)$. We have: $W_{t_1}(\tau) = C_{t_1}^{t_2} W_{t_2}(\tau)$ and $(C_{t_1}^{t_2})^{-1} = C_{t_2}^{t_1}$. Finally, C^t will denote the rotation matrix between the global frame and the local frame at time t, i.e., $w(\tau) = C^t W_t(\tau)$.

The position r of the system is:

$$r(t) = r(T_{in}) + v(T_{in})\Delta t + \int_{T_{in}}^{t} \int_{T_{in}}^{\tau} a(\xi) d\xi d\tau \tag{4.8}$$

where $t \in [T_{in},\ T_{fin}]$.

Integrating by part we obtain:

$$r(t) = r(T_{in}) + v(T_{in})\Delta t + \int_{T_{in}}^{t} (t-\tau) a(\tau) d\tau \tag{4.9}$$

where $v \equiv \frac{dr}{dt}$, $a \equiv \frac{dv}{dt}$ and $\Delta t \equiv t - T_{in}$.

The accelerometers provide the acceleration in the local frame and they also perceive the gravitational acceleration. Their measurements are also corrupted by a constant term (B) usually called bias. We can therefore write the accelerometer measurement like this:

$$A_\tau(\tau) \equiv A_\tau^i(\tau) - G_\tau + B \tag{4.10}$$

where $A_\tau^i(\tau)$ is the inertial acceleration and G_τ is the gravity acceleration in the local frame (depending on time because the local frame can rotate). Rewriting equation (4.9) by highlighting the vector $A_\tau(\tau)$ provided by the accelerometer and neglecting the bias term B:

$$r(t) = r(T_{in}) + v(T_{in})\Delta t + g\frac{\Delta t^2}{2} + C^{T_{in}}\left[S_{T_{in}}(t)\right] \tag{4.11}$$

where:

$$S_{T_{in}}(t) \equiv \int_{T_{in}}^{t} (t-\tau) C_{T_{in}}^{\tau} A_\tau(\tau) d\tau;$$

The matrix $C_{T_{in}}^{\tau}$ can be obtained from the angular speed during the interval $[T_{in},\ \tau]$ provided by the gyroscopes [30].The vector $S_{T_{in}}(t)$ can be obtained by integrating the data provided by the gyroscopes and the accelerometers delivered during the interval $[T_{in},\ t]$.

The visual measurements related to the observation of N point-features are

recorded simultaneously with the inertial measurements. Let us denote the feature position in the physical world with p^i, $i = 1, \ ..., \ N$. $P_t^i(t)$ denotes their position at time t in the local frame at time t. We have:

$$p^i = r(t) + C^{T_{in}} C_{T_{in}}^t P_t^i(t) \tag{4.12}$$

Writing this equation for $t = T_{in}$ we obtain:

$$p^i - r(T_{in}) = C^{T_{in}} P_{T_{in}}^i(T_{in}) \tag{4.13}$$

By inserting the expression of $r(t)$ provided in (4.11) into equation (4.12), by using (4.13) and by pre multiplying by the rotation matrix $(C^{T_{in}})^{-1}$, we obtain:

$$C_{T_{in}}^t P_t^i(t) = P_{T_{in}}^i(T_{in}) - V_{T_{in}}(T_{in})\Delta t - G_{T_{in}} \frac{\Delta t^2}{2} - S_{T_{in}}(t) \tag{4.14}$$
$$i = 1, \ 2, \ ..., \ N$$

A single image processed at time t, provides the position of the N features up to a scale factor, which correspond to the the vectors $P_t^i(t)$. The data provided by the gyroscopes during the interval $(T_{in}, \ T_{fin})$ allow us to build the matrix $C_{T_{in}}^t$. At this point, having the vectors $P_t^i(t)$ up to a scale, allows us to also know the vectors $C_{T_{in}}^t P_t^i(t)$ up to a scale.

We assume that the camera provides n_i images of the same N point-features at consecutive image stamps: $t_1 = T_{in} < t_2 < ... < t_{n_i} = T_{fin}$. For the sake of simplicity, we adopt the following notation:

- $P_j^i \equiv C_{T_{in}}^{t_j} P_{t_j}^i(t_j)$, $i = 1, \ 2, \ ..., \ N$; $j = 1, \ 2, \ ..., \ n_i$
- $P^i \equiv P_{T_{in}}^i(T_{in})$, $i = 1, \ 2, \ ..., \ N$
- $V \equiv V_{T_{in}}(T_{in})$
- $G \equiv G_{T_{in}}$
- $S_j \equiv S_{T_{in}}(t_j)$, $j = 1, \ 2, \ ..., \ n_i$

The vectors P_j^i can be written as $P_j^i == \lambda_j^i \mu_j^i$. Without loss of generality we can set $T_{in} = 0$. Equation (4.14) can be written as follows:

$$P^i - Vt_j - G\frac{t_j^2}{2} - \lambda_j^i \mu_j^i = S_j \tag{4.15}$$

The corresponding linear system is:

$$\begin{cases} -G\frac{t_j^2}{2} - Vt_j + \lambda_1^1 \mu_1^1 - \lambda_j^1 \mu_j^1 & = S_j \\ \lambda_1^1 \mu_1^1 - \lambda_j^1 \mu_j^1 - \lambda_1^i \mu_1^i + \lambda_j^i \mu_j^i & = 0_3 \end{cases} \tag{4.16}$$

where $j = 2, ..., n_i,\ i = 2, ..., N$ and 0_3 is the 3×1 zero vector. This linear system consists of $3(n_i - 1)N$ equations in $Nn_i + 6$ unknowns. The two column vectors X and S and the matrix Ξ are defined as follows:

$$X \equiv [G^T,\ V^T,\ \lambda_1^1,\ ...,\ \lambda_1^N,\ ...,\ \lambda_{n_i}^1,\ ...,\ \lambda_{n_i}^N]^T$$

$$S \equiv [S_2^T,\ 0_3,\ ...,\ 0_3,\ S_3^T,\ 0_3,\ ...,\ 0_3,\ ...,\ S_{n_i}^T,\ 0_3,\ ...,\ 0_3]^T$$

$$\Xi \equiv \tag{4.17}$$

$$\left[\begin{array}{c|c|c|c|c|c|c|c|c|c|c}
T_2 & S_2 & \mu_1^1 & 0_3 & 0_3 & -\mu_2^1 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_{33} & 0_{33} & \mu_1^1 & -\mu_1^2 & 0_3 & -\mu_2^1 & \mu_2^2 & 0_3 & 0_3 & 0_3 & 0_3 \\
... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\
0_{33} & 0_{33} & \mu_1^1 & 0_3 & -\mu_1^N & -\mu_2^1 & 0_3 & \mu_2^N & 0_3 & 0_3 & 0_3 \\
... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\
... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\
T_{n_i} & S_{n_i} & \mu_1^1 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & -\mu_{n_i}^1 & 0_3 & 0_3 \\
0_{33} & 0_{33} & \mu_1^1 & -\mu_1^2 & 0_3 & 0_3 & 0_3 & 0_3 & -\mu_{n_i}^1 & \mu_{n_i}^2 & 0_3 \\
... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\
0_{33} & 0_{33} & \mu_1^1 & 0_3 & -\mu_1^N & 0_3 & 0_3 & 0_3 & -\mu_{n_i}^1 & 0_3 & \mu_{n_i}^N
\end{array}\right]$$

where $T_j \equiv -\frac{t_j^2}{2} I_3$, $S_j \equiv -t_j I_3$ and I_3 is the identity 3×3 matrix; 0_{33} is the 3×3 zero matrix. The linear system in (4.16) can be written in a compact format:

$$\Xi X = S \tag{4.18}$$

The linear system in 4.18 contains completely the sensor information. By adding the following equation to the system:

$$|\Pi X|^2 = g^2 \tag{4.19}$$

where $\Pi \equiv [I_3,\ 0_3\ ...\ 0_3]$, it is possible to exploit the information related to the fact that the magnitude of the gravitational acceleration is known.

The Visual-Inertial Structure from Motion problem consists in the determination of the vectors: P^i, $(i = 1,\ 2,\ ...,\ N)$, V, G and it can be solved by finding the vector X, which satisfies (4.18) and (4.19). The scale factors are the quantities λ_j^i for $i = 1,\ 2,\ ...,\ N$, $j = 1,\ 2,\ ...,\ n_i$ contained in the state vector X.

In our case to initialize the scale factor we need at least three consecutive images containing the two points P_1 and P_2. This is enough considering that we know the gravity magnitude and that we know in advance we will not occur in degenerative cases (none of the camera poses will be aligned along with the two

features, and the three camera poses and the two features will not belong to the same plane) [67].

4.2.2.4 Estimation of γ_1 and γ_2

In order to estimate the angles characterizing the triangle γ_1 and γ_2 (Figure 4.5), we run a Kalman filter. The state that we want to estimate is $\Gamma = [\gamma_1, \gamma_2]^T$. During the prediction step the filter does not update neither the state Γ nor its covariance matrix because the angles are constant in time. For the observation step we need the estimated Roll and Pitch (which allow us to virtually rotate the vehicle frame V into the the new frame N) and the observations of the three features in the current camera image $[x_i, y_i, z]^T = z[\mu_i, \nu_i, 1]^T$ for $i = 1, 2, 3$. At this point the sides of the triangle can be computed according to: $a = z\sqrt{\Delta\mu_{12}^2 + \Delta\nu_{12}^2}$, $b = z\sqrt{\Delta\mu_{13}^2 + \Delta\nu_{13}^2}$, $c = z\sqrt{\Delta\mu_{23}^2 + \Delta\nu_{23}^2}$.

Applying the law of cosine we can easily compute the two required angles:

$$\gamma_1 = \operatorname{acos}\left(\frac{a^2+b^2-c^2}{2ab}\right)$$
$$\gamma_2 = \pi - \operatorname{acos}\left(\frac{a^2+c^2-b^2}{2ac}\right)$$

Note that these angles are independent from z. γ_1 and γ_2 represent the observation of the state Γ of the Kalman Filter.

4.2.3 Performance Evaluation

4.2.3.1 Simulations

In order to evaluate the performance of the presented method, we simulated different $3D$ trajectories and scenarios.

The considered scenarios to test the 2p-Algorithm is shown in Figure 4.1. The features are $P_1 = [0, 0, 0]$, $P_2 = D * [1, 0, 0]$, where $D = 0.1m$. To compare the 2p-Algorithm with the 3p-Algorithm, we added a third feature $P_3 = D * [0.5, \sqrt{3}/2, 0]$ (Figure 4.5). The angles γ_1 and γ_2 are respectively $60deg$ and $120deg$.

The trajectories are generated with a quadrotor simulator that, given the initial conditions, the desired position and desired Yaw, performs a hovering task [20]. The initial vehicle position is $x = y = z = 0\ m$, the initial vehicle speed is $v_x = v_y = v_z = 0\ ms^{-1}$ in the global frame.

Starting from the performed trajectory, the true angular speed and the linear acceleration are computed each $0.01s$ We denote with $\boldsymbol{\Omega_i^{true}}$ and $\boldsymbol{A_{v\ i}^{true}}$ the true value of the body rates and linear accelerations at time stamp i. The IMU readings are generated as following: $\boldsymbol{\Omega_i} = N\left(\boldsymbol{\Omega_i^{true}} - \boldsymbol{\Omega_{bias}}, P_{\Omega_i}\right)$ and $\boldsymbol{A_i} = N\left(\boldsymbol{A_{v\ i}^{true}} - \boldsymbol{A_g} - \boldsymbol{A_{bias}}, P_{A_i}\right)$ where:

- N indicates the Normal distribution whose first entry is the mean value and the second one is the covariance matrix;
- P_{Ω_i} and P_{A_i} are the covariance matrices characterizing the accuracy of the IMU;
- $\boldsymbol{A_g}$ is the gravitational acceleration in the local frame and $\boldsymbol{A_{bias}}$ is the bias affecting the accelerometer's data;
- $\boldsymbol{\Omega_{bias}}$ is the bias affecting the gyroscope's data.

In all the simulations we set both the matrices P_{Ω_i} and P_{A_i} diagonal and in particular: $P_{\Omega_i} = \sigma^2_{gyro} I_3$ and $P_{A_i} = \sigma^2_{acc} I_3$, where I_3 is the identity 3×3 matrix. We considered several values for σ_{gyro} and σ_{acc}, in particular: $\sigma_{gyro} = 1\ deg\ s^{-1}$ and $\sigma_{acc} = 0.01\ ms^{-2}$.

The camera is simulated as follows. Knowing the true trajectory of the vehicle, and the position of the features in the global frame, the true bearing angles of the features in the camera frame are computed each $0.3s$. Then, the camera readings are generated by adding zero-mean Gaussian errors (whose variance is set to $(1\ deg)^2$) to the true values.

Figures 4.7.a show the results regarding the estimated x, y and z. Figures 4.7.b show the results regarding the estimated $Roll$, $Pitch$ and Yaw. In each figure we represent the ground truth values in blue, the values estimated with the 2p-Algorithm in green and the values estimated with the 3p-algorithm in red.

Figure 4.8 summarizes these results by providing the mean error on the estimated position and attitude.

4.2.3.2 Experimental Results

This section describes our experimental results. The robot platform is a *Pelican* from *Ascending Technologies* [1] equipped with an Intel Atom processor board (*1.6 GHz, 1 GB RAM*) (Figure 4.9).

Our sensor suite consists of an Inertial Measurement Unit (*3-Axis Gyro, 3-Axis Accelerometer*) belonging to the Flight Control Unit (FCU) AscTec Autopilot , and a monocular camera (*Matrix Vision mvBlueFOX*, FOV: 130 deg). The camera is calibrated using the Camera Calibration Toolbox for Matlab [17]. The calibration between the IMU and the camera has been performed using the Inertial Measurement Unit and Camera Calibration Toolbox in [57]. The IMU provides measurements update at a rate of $100Hz$, while the camera framerate is $10Hz$.

The Low Level Processor (LLP) of our Pelican is flashed with the *2012 LLP Firmware* [1] and performs attitude data fusion and attitude control. We flashed

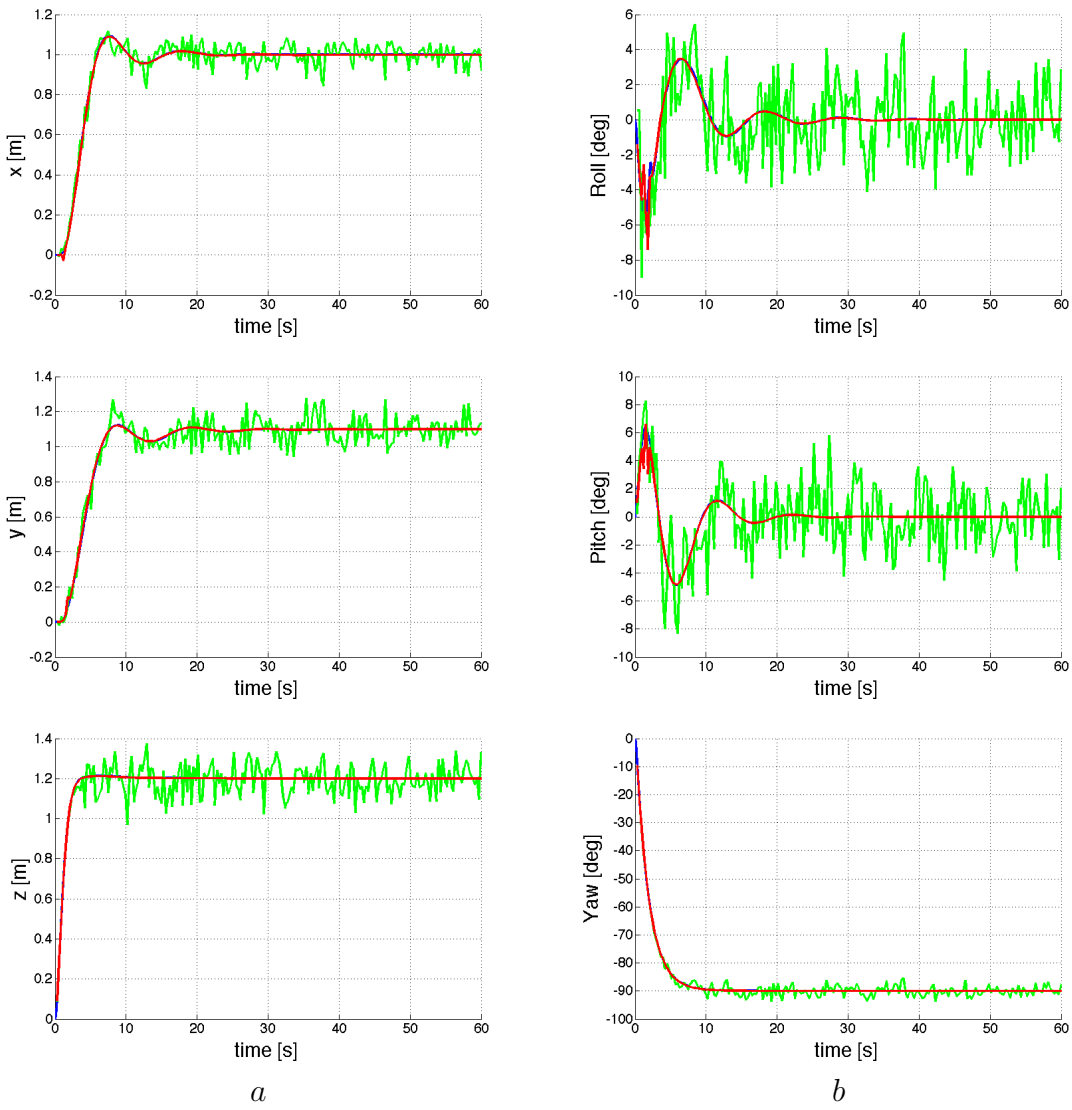


Figure 4.7: Estimated x, y, z (a), and *Roll*, *Pitch*, *Yaw* (b). The blue line indicate the ground truth, the green one the estimation with the 2p-Algorithm and the red one the estimation with the 3p-Algorithm

	x	y	z	Roll	Pitch	Yaw
3p-Algorithm	0.26 %	0.24 %	0.08 %	0.07 deg	0.04 deg	0.01 deg
2p-Algorithm	4.08 %	5.41 %	5.23 %	1.63 deg	1.72 deg	1.36 deg

Figure 4.8: Mean error on the estimated states in our simulations. For the position the error is given in %.

Figure 4.9: AscTec Pelican quadcopter [1] equipped with a monocular camera.

the High Level Processor (HLP) with the asctec_hl_firmware [9]. The on-board computer runs linux 10.04 and ROS (Robot Operating System). We implemented our method using ROS as a middleware for communication and monitoring . The HLP communicates with the on-board computer through a FCU-ROS node. The communication between the camera and the on-board computer is achieved by a ROS node as well. The presented algorithms are running online and on-board at $10Hz$.

The scenario setup is shown in Figure 4.11. We used an ARToolKit Marker with the only aim of having a ground truth to evaluate the performance of our approach. The estimation of the camera pose provided by the marker is not used to perform the estimation. The marker is positioned such that it's reference frame is coincident with the configuration shown in Figure 4.5. The three features considered are the center of the three little balls in Figure 4.11. The use of three blob markers instead of natural features is only related to the need to get a ground truth. The information related to the pattern composed by the 3 features ($D = 0.25m$, $\gamma_1 = 60deg$, $\gamma 2 = 120deg$) is only used to evaluate the performance of our approach. The algorithm does not require any information about the

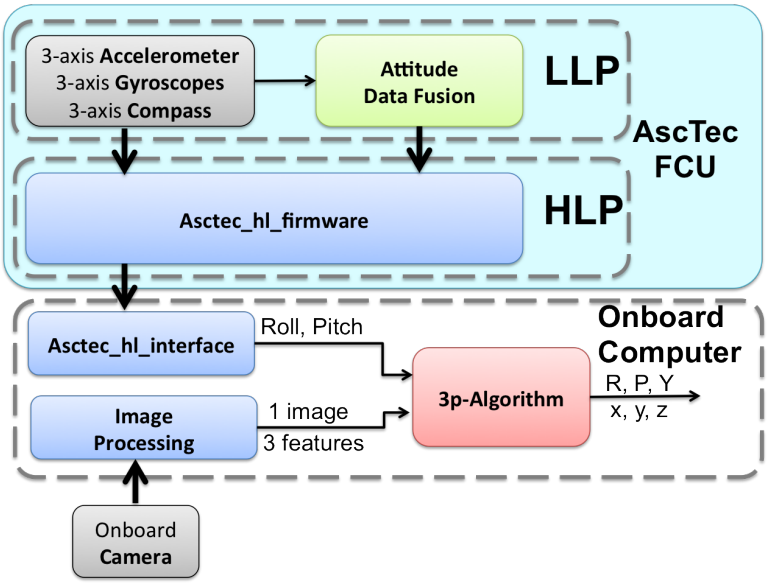


Figure 4.10: Our Pelican quadcopter: a system overview

features configuration.

Figures 4.12.a and 4.12.b show respectively the position and the attitude estimated by using the proposed approach and compared with the ground truth obtained with the ARToolkit marker. From Figure 4.12.a we see that the difference between our estimates and the ground truth values is of the order of $2cm$ for x and y and less than $0.5cm$ for z. From Figure 4.12.b we see that the difference between our estimates and the ground truth values is of the order of $2deg$ for $Pitch$ and less than $0.5deg$ for $Roll$ and Yaw.

We believe that the main source of error is due to the distortion of the lens, which is not fully compensated by the calibration. Note that this distortion also affects our ground truth. We plan to test our approach in an environment equipped with a Motion Capture System.

4.2.4 Conclusion

In this section we proposed a new approach to perform MAV localization by only using the data provided by an Inertial Measurement Unit and a monocular camera. The approach exploits the so-called planar ground assumption and only needs three natural point features. It is based on a closed solution which provides the vehicle pose from a single camera image, once the roll and the pitch angles are obtained by the inertial measurements. This makes the approach very simple in terms of computational complexity and robust since the closed form

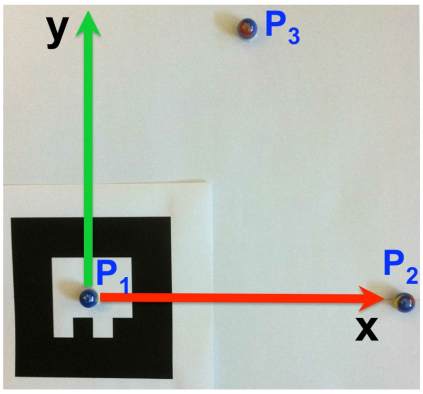


Figure 4.11: Scenario: The AR Marker and the 3 balls are used only with the aim to get a rough ground truth. The AR Marker provides the camera $6DOF$ pose in a global reference frame according to our conventions.

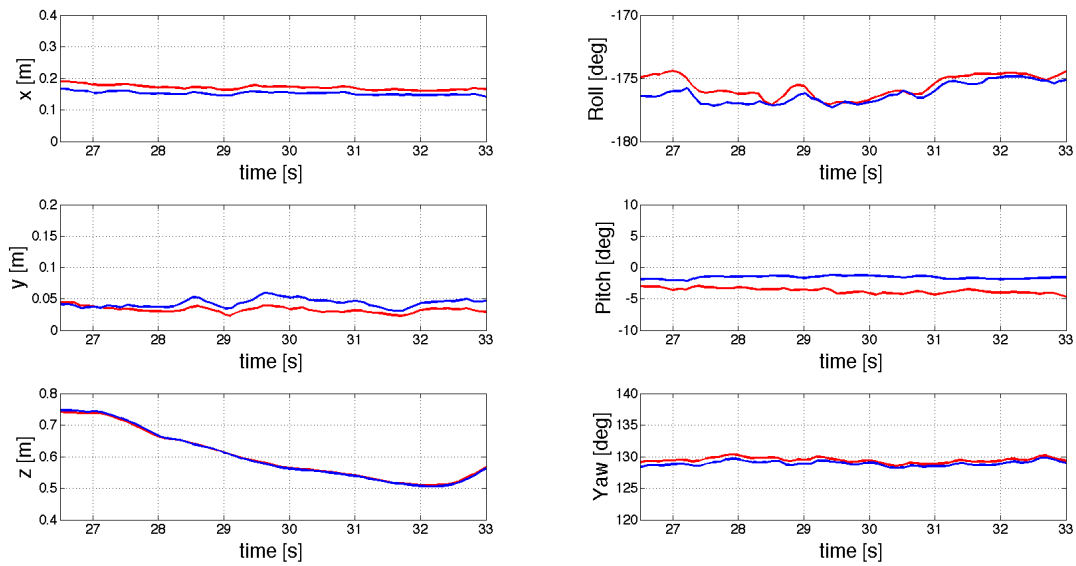


Figure 4.12: Estimated position (a), respectively x, y, z and estimated attitude (b), respectively $Roll$, $Pitch$, Yaw. The red lines represent the estimated values with the 3p-Algorithm, the blue ones represent a rough ground truth (from ARToolkit Marker).

solution makes unnecessary any initialization. We evaluated the performance of the proposed approach by using both synthetic and real data. We also described the results obtained by implementing the approach on our quadrotor in real-time and on-board.

The very low computational cost of the proposed approach makes it suitable for pose control in tasks like hovering, autonomous take off and landing.

4.3 Virtual features

In this section we consider a vehicle which accomplishes a $3D$-trajectory in the surrounding of a planar surface. The vehicle is equipped with a monocular camera and inertial sensors. This is the typical navigation problem in an indoor environment or in a city-like environment. All the approaches previously mentioned, require to extract natural features from the images provided by the camera and in particular to detect the same features in different images. The feature matching task becomes critical in outdoor environment because of possible illumination changes. In order to significantly reduce the computational burden required to perform these tasks and to make the feature matching more robust, we introduce a *virtual* feature by equipping our vehicle with a laser pointer [92]. The laser beam produces a laser spot on the planar surface. This laser spot is observed by the monocular camera and it is the unique point feature used by the proposed approach.

To the best of our knowledge, this problem has never been considered so far. Compared to classical vision and IMU data fusion problems, the feature is moving in the environment but we exploit the hypothesis that it moves on a planar surface. The first question which arises is to understand which are the observable modes, i.e. the physical quantities that can be determined by only using the inertial data and the camera observation of the laser spot during a short time-interval. The results provided in Section 4.3.3 address precisely this issue. Then, the second step we consider is to analytically determine the link between the observable modes and the sensor data. This is obtained by performing a local decomposition of the original system (Section 4.3.4). This decomposition separates the observable modes from the rest of the system and will allow us to introduce a simple recursive method to perform the estimation of all the observable modes (Section 4.3.4). The method is validated by using synthetic data (Section 4.3.5). Preliminary tests with real data are also provided and more complete experiments are in progress.

4.3.1 The System

Let us consider an aerial vehicle equipped with a monocular camera and IMU sensors. The vehicle is also equipped with a laser pointer. The configuration of the laser pointer in the camera reference frame is known. The vehicle moves in the surrounding of a planar surface and we assume that the laser spot produced by the laser beam belongs to this planar surface (see fig. 4.13). The position and the orientation of this planar surface are unknown. The camera observations consist in the position of the laser spot in the camera frame up to a scale factor. The IMU consists of three orthogonal accelerometers and three orthogonal gyroscopes. We assume that the monocular camera is intrinsically calibrated and that the transformations among the camera frame and the IMU frames are known (we can assume that the vehicle frame coincides with the camera frame). The IMU provides the vehicle angular speed and acceleration. Actually, regarding the acceleration, the one perceived by the accelerometer ($\boldsymbol{A}$) is not simply the vehicle acceleration ($\boldsymbol{A_v}$). It also contains the gravity acceleration ($\boldsymbol{G}$). In particular, we have $\boldsymbol{A} = \boldsymbol{A_v} - \boldsymbol{G}$ since, when the camera does not accelerate (i.e. $\boldsymbol{A_v}$ is zero) the accelerometer perceives an acceleration which is the same of an object accelerated upward in the absence of gravity.

We will use uppercase letters when the vectors are expressed in the local frame and lowercase letters when they are expressed in the global frame. Hence, regarding the gravity we have: $\boldsymbol{g} = [0,\ 0,\ -g]^T$, being $g \simeq 9.8\ ms^{-2}$.

Finally, we will adopt a quaternion to represent the vehicle orientation. Indeed, even if this representation is redundant, it is very powerful since the dynamics can be expressed in a very easy and compact notation [51].

Our system is characterized by the state $[\boldsymbol{r},\ \boldsymbol{v},\ q]^T$ where $\boldsymbol{r} = [r_x,\ r_y,\ r_z]^T$ is the $3D$ vehicle position, $\boldsymbol{v}$ is its time derivative, i.e. the vehicle speed in the global frame ($\boldsymbol{v} \equiv \frac{d\boldsymbol{r}}{dt}$), $q = q_t + iq_x + jq_y + kq_z$ is a unitary quaternion (i.e. satisfying $q_t^2 + q_x^2 + q_y^2 + q_z^2 = 1$) and characterizes the vehicle orientation. The analytical expression of the dynamics and the camera observations can be easily provided by expressing all the $3D$ vectors as imaginary quaternions. In practice, given a $3D$ vector $\boldsymbol{w} = [w_x,\ w_y,\ w_z]^T$ we associate with it the imaginary quaternion $w_q \equiv 0 + iw_x + jw_y + kw_z$. The dynamics of the state $[r_q,\ v_q,\ q]^T$ are:

$$\begin{cases} \dot{r}_q = v_q \\ \dot{v}_q = qA_{vq}q^* = qA_q q^* + g_q \\ \dot{q} = \dfrac{1}{2} q\Omega_q \end{cases} \tag{4.20}$$

being q^* the conjugate of q, $q^* = q_t - iq_x - jq_y - kq_z$ and $\boldsymbol{\Omega}$ the angular velocity.

We derive the expression of the camera observation consisting in the position of the laser spot in the camera frame up to a scale factor. The laser spot is on

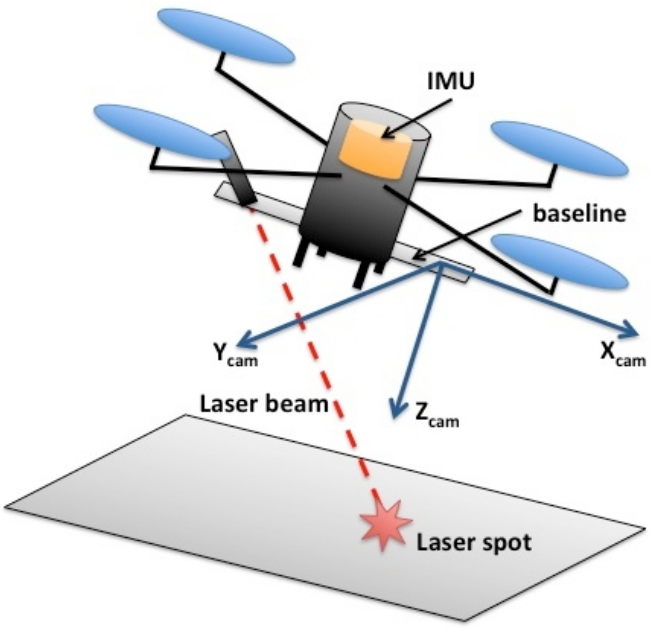


Figure 4.13: Quadrotor equipped with a monocular camera, IMU and a laser pointer. The laser spot is on a planar surface and its position in the camera frame is obtained by the camera up to a scale factor.

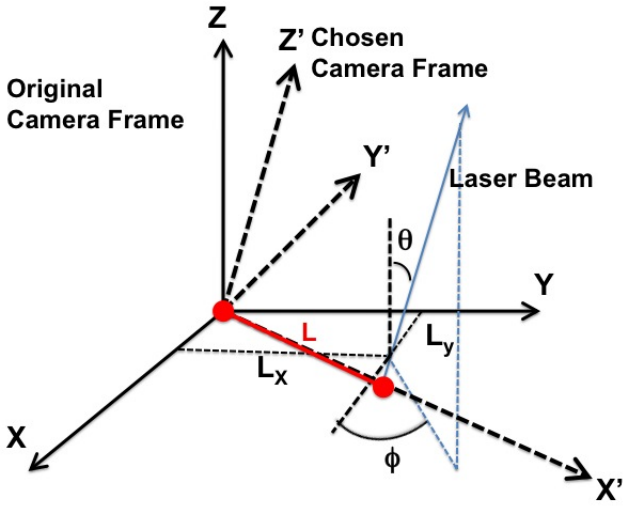


Figure 4.14: The original camera frame XYZ, the chosen camera frame $X'Y'Z'$ and the laser module at the position $[L_x,\ L_y,\ 0]$ and the direction $(\theta,\ \phi)$ (in the original frame) and position $[L,\ 0,\ 0]$ and the direction $(0,\ 0)$ (in the chosen camera frame).

a planar surface whose configuration is unknown. Without loss of generality, we choose the camera frame with the z-axis parallel to the laser pointer (see figure 4.14). In addition, the camera frame is such that the laser beam intercept the $xy-$plane in $[L,\ 0,\ 0]$. In 4.3.2 we introduce a simple and efficient method to determine the parameter L together with the rotation to transform vectors from the original camera frame into the chosen camera frame.

Finally, we characterize the planar surface in the global frame with the equation $z = ky$, where k is an unknown parameter.

In these settings, by carrying out analytical computation (which uses the basic quaternion rules) we obtain the analytical expression of the position $[X_s,\ Y_s,\ Z_s]$ of the laser spot in the camera reference frame. We have:

$$\begin{cases} X_s = L \\ Y_s = 0 \\ Z_s = \dfrac{r_z + 2q_zq_xL - 2q_ykq_xL - 2q_yq_tL - 2q_tLkq_z - kr_y}{2kq_zq_y - 2kq_tq_x - q_z^2 - q_t^2 + q_y^2 + q_x^2} \end{cases} \tag{4.21}$$

The camera provides the vector $[X_s,\ Y_s,\ Z_s]$ up to a scale factor. This is equivalent to the two ratios $\frac{X_s}{Z_s}$ and $\frac{Y_s}{Z_s}$. Hence, since the latter is identically zero, the camera

observation is given by $h_{cam} = \frac{X_s}{Z_s}$, which is:

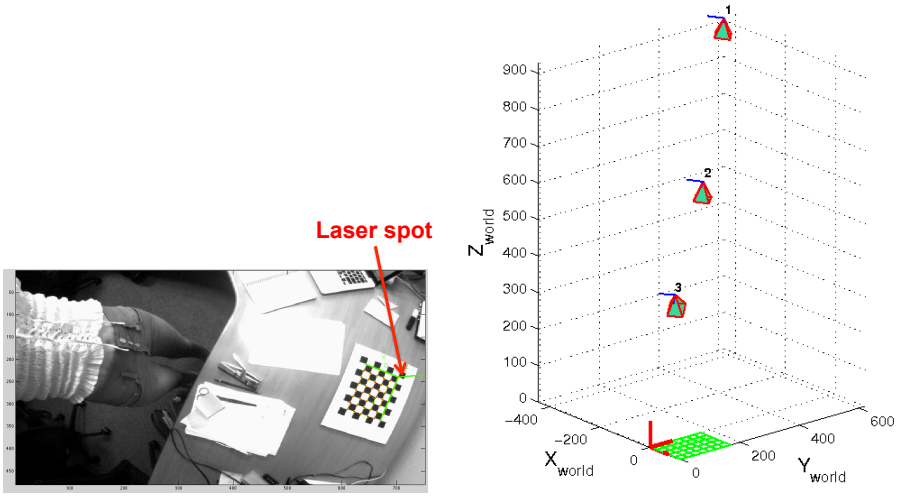
$$h_{cam} = \frac{L(2kq_zq_y - 2kq_tq_x - q_z^2 - q_t^2 + q_y^2 + q_x^2)}{r_z + 2q_zq_xL - 2q_ykq_xL - 2q_yq_tL - 2q_tLkq_z - kr_y} \tag{4.22}$$

4.3.2 Camera-laser module calibration

In figure 4.14, we display the position and the direction of the laser pointer in the original camera frame. The calibration consists in estimating the four parameters $L_x,\ L_y,\ \theta,\ \phi$. In other words, it consists in estimating the line made by the laser beam in the original camera frame. This line is determined starting from the position of the laser spot in the original camera frame for at least two spots. To have an accurate estimate, the two spots must be as far as possible one each other. The precision can be further improved by considering more than two spots (Figure 4.15b) and by finding the best line fit. In order to have the Cartesian coordinates of a single spot in the original camera frame, it suffices to project the spot on a checkerboard (Figure 4.15a). By using the Camera Calibration Toolbox for Matlab [17], it is possible to get the equation of the plane containing the checkerboard in the original camera frame and, known the direction of the spot from the camera measurement, the 3D position is finally obtained.

The chosen camera frame is obtained by rotating the original frame such that in the new frame the $z-$axis has the same orientation of the laser beam. Additionally, we also require that the laser beam intersects the new $x - axis$. In other words, we require that the laser beam intersects the new $xy-$plane in the point $[L,\ 0,\ 0]^T$, for a given L. We want to obtain the quaternion q which characterizes this rotation. This will allow us to express the vectors provided by the camera in the chosen frame. Note that, since the two frames only differ by a rotation (i.e., they share the same origin), we are allowed to express the vectors provided by the camera in the new frame, even if these vectors are defined up to a scale. Finally, in this section we want to determine the value of L. As we will see, both the quaternion q and the parameter L only depend on the calibration parameters: $L_x,\ L_y,\ \theta,\ \phi$.

We start by rotating the original frame of ϕ about its $z-$axis. The quaternion characterizing this rotation is $q_{z-axis}(\phi) = \cos\left(\frac{\phi}{2}\right) + k\sin\left(\frac{\phi}{2}\right)$. Then, we rotate the frame obtained with this rotation of θ about its $y-$axis. The quaternion characterizing this rotation is $q_{y-axis}(\theta) = \cos\left(\frac{\theta}{2}\right) + j\sin\left(\frac{\theta}{2}\right)$. Hence, the previous two rotations are characterized by the quaternion $q_{zy} \equiv q_{z-axis}(\phi)q_{y-axis}(\theta)$. The obtained frame has the $z-$axis aligned with the laser beam. On the other hand, the laser beam does not intersect necessarily the $x-$axis. To obtain this, we have to rotate again the frame about its current $z-$axis. Let us compute the intersection of the laser beam with the $xy-$plane. We compute this point in



(a) (b)

Figure 4.15: Camera-Laser module calibration steps. Figure (a) is the camera image containing the laser spot projected onto a checkerboard. In green the reference frame attached to the checkerboard. Figure (b) represents three different camera positions used during the calibration process, the green grid represents the checkerboard and the red lines represent the reference frame attached to the checkerboard.

the original frame. By a direct computation we obtain: $r^{inters} = [L_x - \tau_x^2 L_x - \tau_x\tau_y L_y,\ L_y - \tau_x\tau_y L_x - \tau_y^2 L_y,\ -\tau_x\tau_z L_x - \tau_y\tau_z L_y]^T$, where $\tau_x = \sin(\theta)\cos(\phi)$, $\tau_y = \sin(\theta)\sin(\phi)$, $\tau_z = \cos(\theta)$. We then compute this vector in the rotated frame by doing the quaternion product: $R_q^{inters} = q_{zy}^* r_q^{inters} q_{zy}$. Let us denote R^{inters} with $[L'_x,\ L'_y,\ 0]^T$. We finally have: $L = \sqrt{L'^2_x + L'^2_y}$ and $q = q_{zy} q_{z-axis}(\alpha)$, where $q_{z-axis}(\alpha) = \cos\left(\frac{\alpha}{2}\right) + k\sin\left(\frac{\alpha}{2}\right)$ and $\alpha = \arctan\left(\frac{L'_y}{L'_x}\right)$.

4.3.3 Observability Properties

We investigate the observability properties of the system whose dynamics are given in (4.20) and whose observation function is given in (4.22). We have also to consider the constraint $q^*q = 1$. This can be dealt as a further observation (system output):

$$h_{const}(r_q,\ v_q,\ q) = q^*q \tag{4.23}$$

Finally, we want to investigate whether the parameter k is identifiable or not. This is done by performing an observability analysis on the extended state $\boldsymbol{S} =$

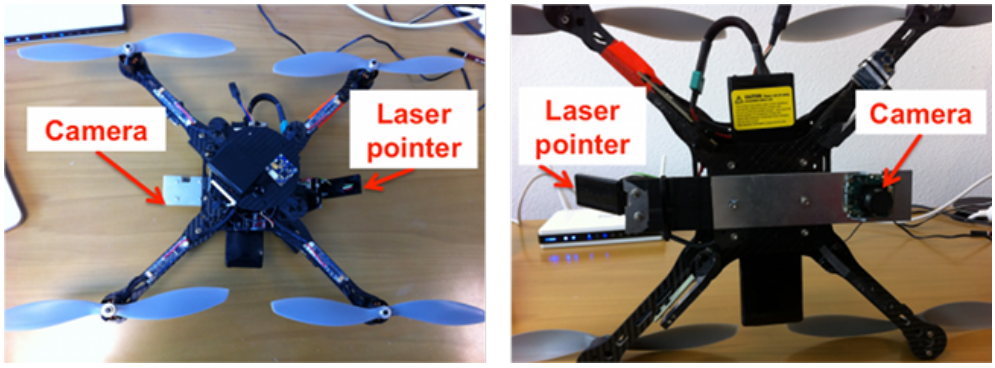


Figure 4.16: The Pelican quadcopter equipped with a monocular camera and a laser module.

$[r_q,\ v_q,\ q, k]^T$, whose dynamics are given in (4.20) and by the additional equation $\dot{k} = 0$.

We apply the method introduced in [64]. This will allow us to detect all the observable modes, i.e. all the physical quantities that can be determined by only using the information contained in the data provided by the IMU and the camera in a given time-interval.

The system is characterized by the state: $\boldsymbol{S} = [r_q\ v_q\ q,\ k]^T$, whose dimension is 11 (r_q and v_q are imaginary quaternions, i.e. they are characterized by 3 real numbers; q contains 4 real numbers; k is a real number). The dynamics are given in (4.20) together with the equation $\dot{k} = 0$ and the observations are given in (4.22) and (4.23). To compute the Lie derivatives, we need to express the dynamics as follows:

$$\dot{\boldsymbol{S}} = \boldsymbol{f}(\boldsymbol{S}, \boldsymbol{u}) = \boldsymbol{f_0}(\boldsymbol{S}) + \sum_{i=1}^{L} \boldsymbol{f_i}(\boldsymbol{S}) u_i \tag{4.24}$$

We have $L = 6$ and the six inputs are the three components of the acceleration, $\boldsymbol{A}$, and the three components of the angular speed, $\boldsymbol{\Omega}$. Namely: $u_1 = A_x$, $u_2 = A_y$, $u_3 = A_z$, $u_4 = \Omega_x$, $u_5 = \Omega_y$, $u_6 = \Omega_z$. The seven vector functions $\boldsymbol{f_0},\ \boldsymbol{f_1},\ \ldots, \boldsymbol{f_6}$ are:

$$\boldsymbol{f_0} = [v_x,\ v_y,\ v_z,\ 0,\ 0,\ -g,\ \boldsymbol{0_5}]^T$$

$$\boldsymbol{f_1} = [\boldsymbol{0_3},\ q_t^2 + q_x^2 - q_y^2 - q_z^2,\ 2q_tq_z + 2q_yq_x,\ -2q_tq_y + 2q_zq_x,\ \boldsymbol{0_5}]^T$$

$$\boldsymbol{f_2} = [\boldsymbol{0_3},\ -2q_tq_z + 2q_yq_x,\ q_t^2 + q_y^2 - q_z^2 - q_x^2,\ 2q_tq_x + 2q_zq_y,\ \boldsymbol{0_5}]^T$$

$$\boldsymbol{f_3} = [\boldsymbol{0_3},\ 2q_tq_y + 2q_zq_x,\ -2q_tq_x + 2q_zq_y,\ q_t^2 + q_z^2 - q_x^2 - q_y^2,\ \boldsymbol{0_5}]^T$$

$$\boldsymbol{f_4} = [\mathbf{0_6},\ -1/2q_x,\ 1/2q_t,\ 1/2q_z,\ -1/2q_y,\ 0]^T$$
$$\boldsymbol{f_5} = [\mathbf{0_6},\ -1/2q_y,\ -1/2q_z,\ 1/2q_t,\ 1/2q_x,\ 0]^T$$
$$\boldsymbol{f_6} = [\mathbf{0_6},\ -1/2q_z,\ 1/2q_y,\ -1/2q_x,\ 1/2q_t,\ 0]^T$$

where we denoted with $\mathbf{0_n}$ the vector line whose dimension is n and whose entries are all zeros.

We must compute the Lie derivatives of the two observation functions given in (4.22) and (4.23) with respect to all the vector fields. By a direct computation, performed by using the symbolic Matlab computational tool, we were able to find not more than 6 independent Lie derivatives[1].

In particular, according to the notation introduced in [64], the system has 5 continuous symmetries which are:

$$\boldsymbol{w_s^1} = [0,\ 1,\ k,\ \mathbf{0_8}]^T$$
$$\boldsymbol{w_s^2} = [1,\ \mathbf{0_{10}}]^T$$
$$\boldsymbol{w_s^3} = [\mathbf{0_4},\ 1,\ k,\ \mathbf{0_5}]^T$$
$$\boldsymbol{w_s^4} = [\mathbf{0_3},\ 1,\ \mathbf{0_7}]^T$$
$$\boldsymbol{w_s^5} = [\mathbf{0_6},\ q_z - kq_y,\ q_y + kq_z,\ -q_x + kq_t,\ -q_t - kq_x,\ 0]^T$$

The observable modes are all the solutions of the system of partial differential equations associated with the five symmetries. For instance, the equation associated with w_s^1 is $\frac{\partial}{\partial r_y} + k\frac{\partial}{\partial r_z} = 0$ (see [64] for more details). Since this system of partial differential equations consists of 5 equations on a manifold whose dimension is 11, the number of independent solutions is $6 = 11 - 5$ [42]. A possible choice of these solutions is:

$$\begin{aligned}
&r_z - kr_y \\
&v_z - kv_y \\
&2[c_k s_k(-q_t^2 + q_x^2 - q_y^2 + q_z^2) + (2c_k^2 - 1)(q_t q_x + q_y q_z)] \\
&4c_k s_k(q_t q_z + q_x q_y) + 2(2c_k^2 - 1)(q_t q_y - q_x q_z) \\
&k \\
&q^* q
\end{aligned}$$

where $c_k \equiv \cos\left(\frac{\arctan(k)}{2}\right)$ and $s_k \equiv \sin\left(\frac{atan(k)}{2}\right)$. By knowing the value of the first solution, $r_z - kr_y$ and the value of the fifth solution, k, we can determine the

[1] A possible choice of 6 independent Lie derivatives is: $L^0 h_{const}$, $L^0 h_{cam}$, $L^1_{\boldsymbol{f_0}} h_{cam}$, $L^1_{\boldsymbol{f_4}} h_{cam}$, $L^1_{\boldsymbol{f_5}} h_{cam}$, $L^2_{\boldsymbol{f_0},\boldsymbol{f_0}} h_{cam}$.

quantity $\frac{|r_z - kr_y|}{\sqrt{1+k^2}}$, which is the distance of the vehicle from the planar surface[1]. Hence, to better visualize the physical meaning, it is convenient to select the following 6 observable modes:

$$
\begin{aligned}
m_1 &= \frac{r_z - kr_y}{\sqrt{1+k^2}} \\
m_2 &= \frac{v_z - kv_y}{\sqrt{1+k^2}} \\
m_3 &= 2[c_k s_k(-q_t^2 + q_x^2 - q_y^2 + q_z^2) + (2c_k^2 - 1)(q_t q_x + q_y q_z)] \\
m_4 &= 4c_k s_k(q_t q_z + q_x q_y) + 2(2c_k^2 - 1)(q_t q_y - q_x q_z) \\
m_5 &= k \\
m_6 &= q^* q
\end{aligned}
$$

Finally, the physical meaning of also m_3 and m_4 becomes clear by referring to a new global frame $\tilde{x},\ \tilde{y},\ \tilde{z}$. This frame has the $\tilde{x}\tilde{y}$-plane coincident with the planar surface. In other words, this new global frame has the vertical axis coincident with the axis orthogonal to the planar surface. In this new frame, m_1 is the $\tilde{z}$-coordinate of the vehicle, m_2 is the component of the vehicle speed along the $\tilde{z}$-axis. m_3 and m_4 are related to the roll and pitch angles of the vehicle in the new frame. In particular, the roll angle is $\arctan\left(\frac{m_3}{\sqrt{1-m_3^2-m_4^2}}\right)$ and the pitch is $\arcsin(m_4)$. m_5 is related to the orientation of the $\tilde{x}\tilde{y}$-plane with respect to the gravity. In particular, the $\tilde{z}$-axis makes an angle $\arctan(k) = \arctan(m_5)$ with the gravity. m_6 is trivially the magnitude of the quaternion, which is 1 since it describes a rotation.

From now on, we will adopt the new frame to characterize the vehicle configuration. The state in this frame is $\tilde{\boldsymbol{S}} = [\tilde{r}_q\ \tilde{v}_q\ \tilde{q},\ k]^T$. The $\tilde{x},\ \tilde{y},\ \tilde{z}$-frame is obtained by rotating the x, y, z-frame about the x-axis of the angle $\arctan(k)$. Hence, it is characterized by the quaternion:

$$
p = \cos\left(\frac{\arctan(k)}{2}\right) + i\sin\left(\frac{\arctan(k)}{2}\right) \tag{4.25}
$$

Therefore, $q = p\tilde{q}$ or:

$$
\tilde{q} = p^* q \tag{4.26}
$$

By using the quaternion p it is also possible to obtain:

$$
\tilde{r}_q = p^* r_q p \qquad \tilde{v}_q = p^* v_q p \tag{4.27}
$$

[1] In other words, also $\frac{r_z - kr_y}{\sqrt{1+k^2}}$ is a solution of the system of partial differential equations.

By using (4.26) and (4.27), we obtain the expressions of the observable modes in the new coordinates. We have:

$$\begin{aligned} m_1 &= \tilde{r}_z \\ m_2 &= \tilde{v}_z \\ m_3 &= 2(\tilde{q}_t\tilde{q}_x + \tilde{q}_y\tilde{q}_z) \\ m_4 &= 2(\tilde{q}_t\tilde{q}_y - \tilde{q}_x\tilde{q}_z) \\ m_5 &= k \\ m_6 &= \tilde{q}^*\tilde{q} \end{aligned} \tag{4.28}$$

In particular m_1 is the $\tilde{z}$-coordinate of the vehicle, m_2 the component of the vehicle speed along the $\tilde{z}$-axis, m_3 and m_4 are related to the roll and pitch angles of the vehicle in this frame. In particular, the roll is $\arctan\left(\frac{m_3}{\sqrt{1-m_3^2-m_4^2}}\right)$ and the pitch is $\arcsin(m_4)$. m_5 is related to the orientation of the $\tilde{x}\tilde{y}$-plane with respect to the gravity. In particular, the $\tilde{z}$-axis makes an angle $\arctan(k) = \arctan(m_5)$ with the gravity. m_6 is trivially the magnitude of the quaternion $\tilde{q}$, which is 1 since it describes a rotation.

By summarizing the results of the observability analysis performed in this section we say that the information contained in the data provided by the IMU and the camera during a given time-interval, allows us to determine the six modes m_1, ..., m_6. For this reason, in the rest of the section, we will focus our attention only on these six quantities (actually, on the first five, since m_6 trivially expresses the constraint of having a unitary quaternion).

4.3.4 Local Decomposition and Recursive Estimation

The goal of this section is to provide a method able to estimate the observable modes in (Section 4.3.3). To achieve this goal, the first step is to determine the link between the observable modes and the sensor data. By adopting the terminology introduced in [41], we have to perform a local decomposition of our system. We remind the reader that the local decomposition is the extension of the Kalman canonical decomposition [19] to the case of a non linear system. It consists in writing the equations characterizing the dynamics and the observation only in terms of the observable modes. We also remind the reader that in the non linear case it is often impossible to characterize the system with a unique decomposition. The decomposition only holds in a local region of the space of states. This is the reason why it is called *local* decomposition. To cover the entire space of states more than one decomposition is required (see [41]). In the following, we will show that for our system the number of decompositions is two.

We first provide the dynamics of the state $\tilde{\boldsymbol{S}} = [\tilde{r}_q \ \tilde{v}_q \ \tilde{q}, \ k]^T$. We obtain:

$$\begin{cases} \dot{\tilde{r}}_q = \tilde{v}_q \\ \dot{\tilde{v}}_q = \tilde{q} A_q \tilde{q}^* + p^* g_q p \\ \dot{\tilde{q}} = \frac{1}{2} \tilde{q} \Omega_q \\ \dot{k} = 0 \end{cases} \tag{4.29}$$

A local decomposition for the dynamics is:

$$\begin{cases} \dot{m}_1 = m_2 \\ \dot{m}_2 = -m_4 A_x + m_3 A_y + \xi(m_3, \ m_4) A_z + g_z \\ \dot{m}_3 = \Omega_x \ \xi(m_3, \ m_4) + \Omega_z m_4 \\ \dot{m}_4 = \Omega_y \ \xi(m_3, \ m_4) - \Omega_z m_3 \\ \dot{g}_z = 0 \end{cases} \tag{4.30}$$

where g_z is the component of the gravity along the $\tilde{z}$-axis, i.e. $g_z = -g \cos(\arctan(k)) = \frac{-g}{\sqrt{1+k^2}} = \frac{-g}{\sqrt{1+m_5^2}}$; the function $\xi(m_3, \ m_4)$ depends on the original state and in particular changes its sign depending on the sign of $\tilde{q}_x^2 + \tilde{q}_y^2 - \frac{1}{2}$:

$$\xi(m_3, \ m_4) \equiv \begin{cases} \sqrt{1 - m_3^2 - m_4^2} \ \ if \ \ \tilde{q}_x^2 + \tilde{q}_y^2 < \dfrac{1}{2} \\ -\sqrt{1 - m_3^2 - m_4^2} \ \ if \ \ \tilde{q}_x^2 + \tilde{q}_y^2 > \dfrac{1}{2} \end{cases} \tag{4.31}$$

Hence, as previously said, we have two local decompositions for our original system. The validity of (4.30) can be checked by using (4.28) and (4.29). Note that deriving (4.30) is troublesome. In contrast, checking its validity is very simple since it only demands to perform differentiation.

To complete the local decomposition we need to express the camera observation function in (4.22) in terms of the observable modes. We obtain:

$$h_{cam} = \frac{L \xi(m_3, \ m_4)}{m_4 L - m_1} \tag{4.32}$$

The validity of (4.32) can be checked by using (4.22), (4.25), (4.26), (4.28) and (4.31).

The equations (4.30) and (4.32) represent a local decomposition for our system. They provide the analytical expression of the link between the observable modes and the sensor data. Specifically, equation (4.30) provides the link between the observable modes and the IMU data. Equation (4.32) provides the link between the observable modes and the data delivered by the monocular

camera. Having these equations allows us to perform the estimation of the state $[m_1,\ m_2,\ m_3,\ m_4,\ g_z]$. An efficient and simple approach is obtained by using an Extended Kalman Filter, EKF. To implement this filter it suffices to compute the Jacobian of the dynamics in (4.30) and the Jacobian of the observation function in (4.32) ([13]).

Let us consider the state $\boldsymbol{m} = [m_1,\ m_2,\ m_3,\ m_4,\ g_z]$. The basic ingredients to implement an EKF, which estimates $\boldsymbol{m}$, are the analytical expression of the Jacobians of the dynamics, and the observation [13].

The Jacobian of the observation is obtained by differentiating the expression of h_{cam} given in (4.32) with respect to $\boldsymbol{m}$, i.e.

$$H \equiv \frac{\partial h_{cam}}{\partial \boldsymbol{m}} = \frac{L\xi(m_3,\ m_4)}{(m_4 L - m_1)^2} \times$$

$$\times \begin{bmatrix} 1 & 0 & \frac{-m_3(m_4 L - m_1)}{1-m_3^2-m_4^2} & \frac{L m_3^2 - L + m_1 m_4}{1-m_3^2-m_4^2} & 0 \end{bmatrix}$$

where the function $\xi(m_3,\ m_4)$ is defined in (4.31). Regarding the Jacobian of the dynamics, we need first of all to discretize the equations in (4.30). Let us denote with δt the discretization time step. The Jacobian of the dynamics with respect to the state $\boldsymbol{m}$ is:

$$F_m = \begin{bmatrix} 1 & \delta t & 0 & 0 & 0 \\ 0 & 0 & \delta t\,(A_y - A_z \overline{m}_3) & -\delta t\,(A_x + A_z \overline{m}_4) & 0 \\ 0 & 0 & -\delta t \Omega_x \overline{m}_3 & \delta t(\Omega_z - \Omega_x \overline{m}_4) & 0 \\ 0 & 0 & -\delta t(\Omega_z + \Omega_y \overline{m}_3) & -\delta t \Omega_y \overline{m}_4 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where $\overline{m}_3 = \frac{m_3}{\xi(m_3,\ m_4)}$ and $\overline{m}_4 = \frac{m_4}{\xi(m_3,\ m_4)}$. The Jacobian of the dynamics with respect to the input $\boldsymbol{u} = [A_x,\ A_y,\ A_z,\ \Omega_x,\ \Omega_y,\ \Omega_z]^T$ is:

$$F_u = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -m_4 & m_3 & \xi & 0 & 0 & 0 \\ 0 & 0 & 0 & \xi & 0 & m_4 \\ 0 & 0 & 0 & 0 & \xi & -m_3 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where $\xi \equiv \xi(m_3,\ m_4)$.

4.3.5 Performance Evaluation

We evaluate the performance of the proposed strategy by using both synthetic and real data. The advantage of simulations is that the ground truth is perfectly known and this allows us a quantitative evaluation of the proposed strategy.

4.3.5.1 Simulations

We simulate many different trajectories in $3D$ and many different scenarios corresponding to different orientation of the planar surface. For all the simulations we use the proposed strategy to estimate the observable modes, i.e.:

- the distance of the vehicle from the planar surface ($d = |m_1|$);
- the component of the vehicle speed orthogonal to the planar surface ($v_o = m_2$);
- the roll (R) and the pitch (P) angles in the $\tilde{x},\ \tilde{y},\ \tilde{z}$-frame (i.e. the frame where the $\tilde{x},\ \tilde{y}$-plane coincides with the planar surface);
- the orientation of the plane with respect to the gravity (α). ...

Specifically, in all the simulations the values of the estimated $d,\ v_o,\ R,\ P$ and α will be compared with the ground truth values.

Simulated Trajectories The trajectories are generated by randomly generating the linear and angular acceleration of the vehicle at 100 Hz. In particular, at each time step, the three components of the linear acceleration and the angular speed are generated as Gaussian independent variables whose mean values will be denoted respectively with μ_a and μ_ω and whose variances will be denoted respectively with σ_a^2 and σ_ω^2. We set the parameters to be close to a real case: $\mu_a = 0\ ms^{-2}$, $\sigma_a = 1\ ms^{-2}$, $\mu_\omega = 0\ deg\ s^{-1}$ and $\sigma_\omega = 10\ deg\ s^{-1}$. The initial vehicle position is at $\tilde{x} = 0,\ \tilde{y} = 0,\ \tilde{z} = 1m$. The initial vehicle speed is $[1,\ 0,\ 0]ms^{-1}$ in the $\tilde{x},\ \tilde{y},\ \tilde{z}$-frame.

Simulated Sensors Starting from the performed trajectory, the true angular speed and the linear acceleration are computed at each time step of $0.01s$ (respectively, at the time step i, we denote them with $\boldsymbol{\Omega_i^{true}}$ and $\boldsymbol{A_{v\ i}^{true}}$). Starting from them, the IMU sensors are simulated by randomly generating the angular speed and the linear acceleration at each step according to the following: $\boldsymbol{\Omega_i} = N\left(\boldsymbol{\Omega_i^{true}} - \boldsymbol{\Omega_{bias}}, P_{\Omega_i}\right)$ and $\boldsymbol{A_i} = N\left(\boldsymbol{A_{v\ i}^{true}} - \boldsymbol{A_{gi}} - \boldsymbol{A_{bias}}, P_{A_i}\right)$ where:

- N indicates the Normal distribution whose first entry is the mean value and the second its covariance matrix;
- P_{Ω_i} and P_{A_i} are the covariance matrices characterizing the accuracy of the IMU;
- $\boldsymbol{A_{gi}}$ is the gravitational acceleration in the local frame and $\boldsymbol{A_{bias}}$ is the bias affecting the data from the accelerometer;

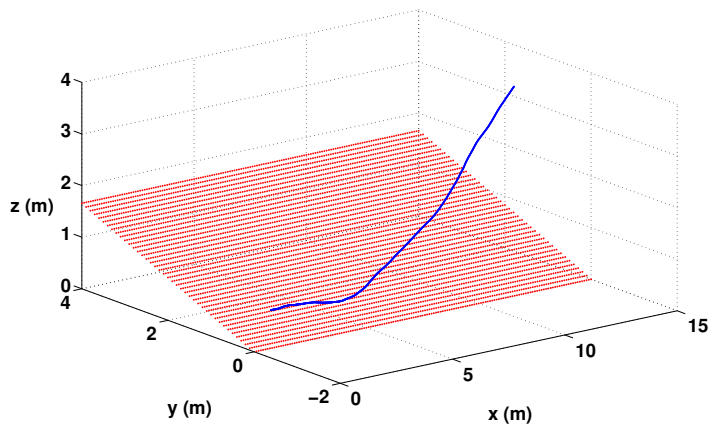


Figure 4.17: A typical vehicle trajectory in our simulations.

- $\boldsymbol{\Omega_{bias}}$ is the bias affecting the data from the gyroscope.

In all the simulations we set both the matrices P_{Ω_i} and P_{A_i} diagonal and in particular: $P_{\Omega_i} = \sigma^2_{gyro} I_3$ and $P_{A_i} = \sigma^2_{acc} I_3$, where I_3 is the identity 3×3 matrix. We considered several values for σ_{gyro} and σ_{acc}, in particular: $\sigma_{gyro} = 1\ deg\ s^{-1}$ and $\sigma_{acc} = 0.01\ ms^{-2}$.

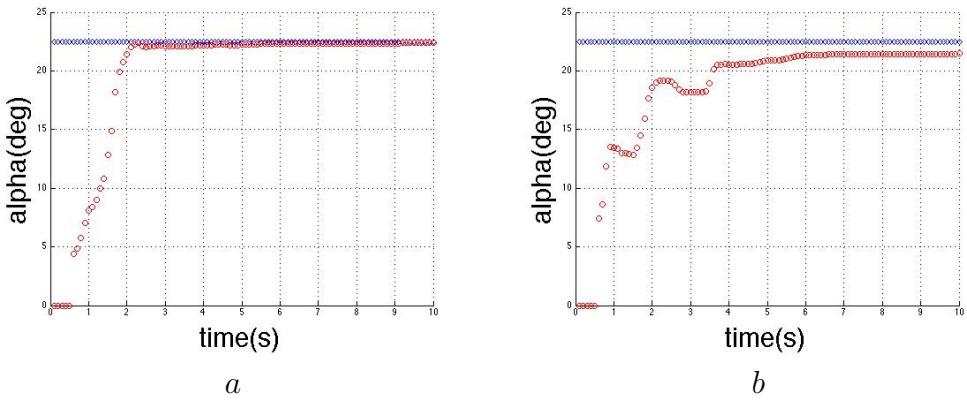
Regarding the camera, the provided readings are generated in the following way. By knowing the true trajectory, and the position and the orientation of the planar surface, the true bearing angles of the laser spot in the camera frame are computed[1]. They are computed each $0.1s$. The parameter L is set equal to $0.3m$. Then, the camera readings are generated by adding to the true values zero-mean Gaussian errors whose variance is equal to $(1\ deg)^2$ for all the readings.

Simulation Results Figure 4.17 displays a typical $3D$ trajectory obtained in our simulations. The figure also displays the planar surface, consisting of a plane, which makes an angle of $\alpha = \frac{\pi}{8}\ rad = 22.5\ deg$ with the gravity.

Figures 4.18 a and b display the estimated α respectively in the case without and with bias. The values of the biases adopted in our simulations are: $\boldsymbol{\Omega_{bias}} = [0.03\ 0.03\ 0.03]^T (deg\ s^{-1})$ and $\boldsymbol{A_{bias}} = [0.03\ 0.03\ 0.03]^T (ms^{-2})$. As expected, the estimation in presence of bias becomes worse. However, the error on the estimated α in presence of bias is smaller than $1\ deg$.

Figures 4.19, 4.20, 4.21 and 4.22 a and b display respectively the estimated P, R, v_o and d. In each figure, both the cases of unbiased and biased inertial

[1]This is obtained also by knowing that the laser pointer has the same orientation as the camera and that it is located at the position $[L,\ 0,\ 0]$



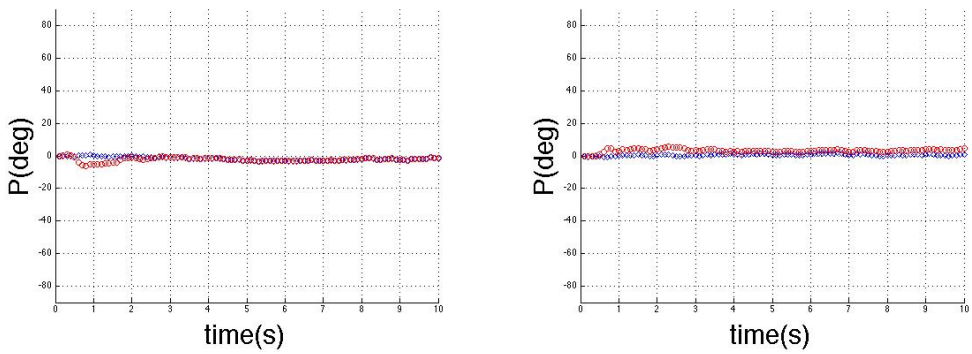
a b

Figure 4.18: Estimated α in absence (a) and in presence of bias (b) on the inertial data. Blue dots indicate ground true values while red discs indicate the estimated values.

measurements are displayed. We initialized the filter by using a value of the initial observable state which differs from the ground truth by a relative error in the range $[10,\ 20]\%$.
We also evaluated the robustness of the filter with respect to systematic errors on the imu-camera calibration and laser-camera calibration. Specifically, we performed simulations by introducing errors of one cm and one deg on the calibration parameters. These systematic errors affect the estimated α (the difference with respect to the ground truth is in the range $[4,\ 6]deg$) while for all the other observable modes the effect is negligible (less than one deg for R and P and less than $1cms^{-1}$ and $1cm$ respectively for v_o and d).

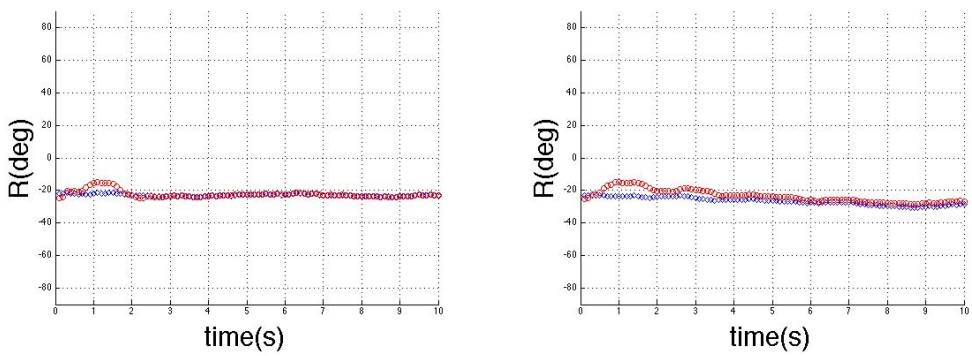
4.3.5.2 Preliminary experiments

In this section we provide preliminary results obtained by using a data set provided by the autonomous system laboratory at ETHZ in Zurich. The data are provided together with a reliable ground-truth, which has been obtained by performing the experiments at the ETH Zurich Flying Machine Arena [60], which is equipped with a Vicon motion capture system. As previously said, the observations of the laser spot are simulated. This was possible thanks to the fact that a reliable ground truth was provided together with the inertial data. In particular, given the true trajectory, we simulated the same planar surface described in the previous section. By having the true vehicle configuration it was possible to create the observations performed by the camera on the laser spot produced by a laser pointer as in the simulations (see the last paragraph in 4.3.5.1).



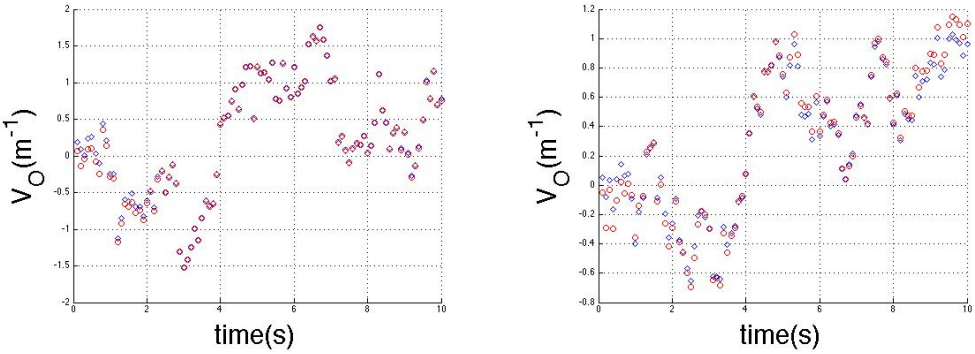
a *b*

Figure 4.19: Estimated P in absence (a) and in presence of bias (b) on the inertial data. Blue dots indicate ground true values while red discs indicate the estimated values.



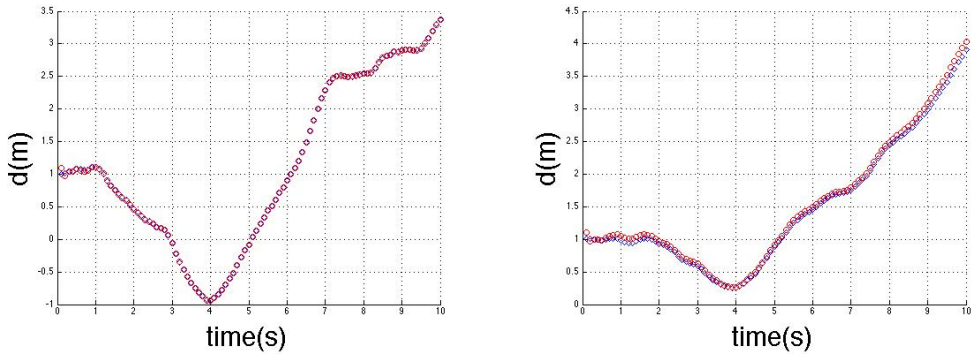
a *b*

Figure 4.20: Estimated R in absence (a) and in presence of bias (b) on the inertial data. Blue dots indicate ground true values while red discs indicate the estimated values.



a *b*

Figure 4.21: Estimated v_0 in absence (a) and in presence of bias (b) on the inertial data. Blue dots indicate ground true values while red discs indicate the estimated values.



a *b*

Figure 4.22: Estimated d in absence (a) and in presence of bias (b) on the inertial data. Blue dots indicate ground true values while red discs indicate the estimated values.

Figure 4.23 displays the estimated α. Figures 4.24 a and b display the estimated P and R and figures 4.25 a and b display the estimated v_o and d. All the observable modes are estimated with very good accuracy. Additionally, we remark that the convergence of the filter occurs in less than half second for all the observable modes.

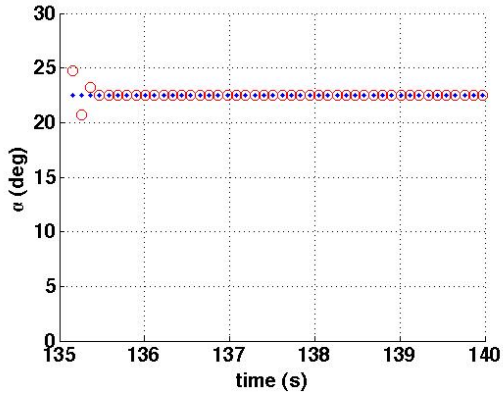
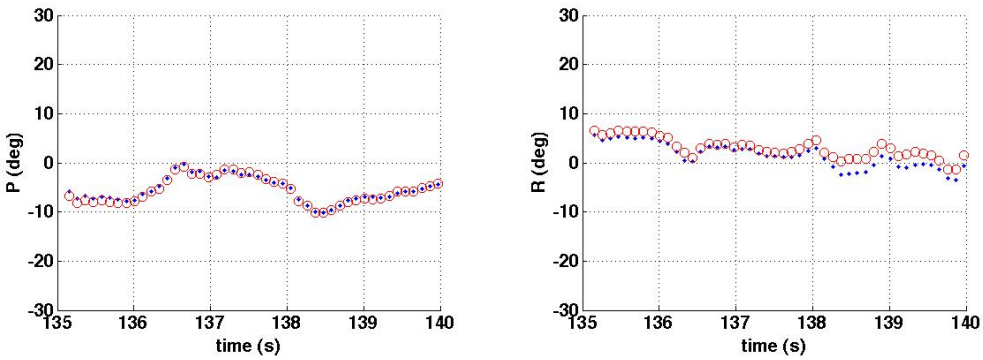
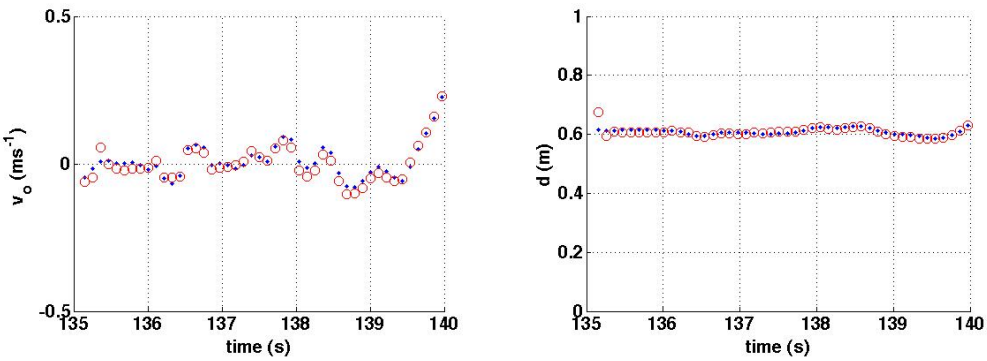


Figure 4.23: Estimated α in the experiment. Blue dots indicate ground true values while red discs indicate the estimated values.



a b

Figure 4.24: Estimated P (a) and R (b) in the experiment. Blue dots indicate ground true values while red discs indicate the estimated values.



a *b*

Figure 4.25: Estimated v_o (a) and d (b) in the experiment. Blue dots indicate ground true values while red discs indicate the estimated values.

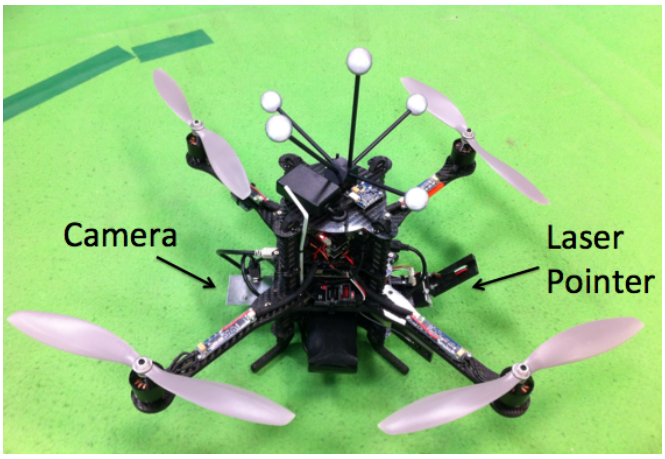


Figure 4.26: The Pelican quadcopter equipped with a monocular camera and a laser module and passive markers.

4.3.5.3 Camera-laser module calibration

In order to test the Camera-laser module calibration techinique, we equipped our robot platform (*Pelican* quadrotor from *Ascending Technologies*) with a monocular camera (*Matrix Vision mvBlueFOX*, *FOV*: 130 *deg*) and a Laser Module (*SparkFun TTL, wavelength: 650nm, poweroutput: 0.45-0.8mW*).

The Laser module and the monocular camera are mounted on a fixed baseline, and the latter is calibrated using the Camera Calibration Toolbox for Matlab [17]. The calibration between IMU and camera has been performed using the Inertial

Measurement Unit and Camera Calibration Toolbox by Lobo [57].

The results of the Camera-laser module calibration described in Section4.3.2 are the following: $\theta = 47.1\ deg$, $\phi = -3.1\ deg$ and $L_x = -0.146\ m$, $L_y = -0.005\ m$. The resulting L is $0.100\ m$. Figure 4.15 represents some steps of the calibration.

4.3.6 Conclusion

In this section we considered an aerial vehicle equipped with a monocular camera and inertial sensors. Additionally, a laser pointer is mounted on the vehicle and it produces a laser spot. The laser spot is observed by the monocular camera and it is the unique point feature used in the proposed approach. We focused our attention to the case when the vehicle moves in proximity of a planar surface and in particular when the laser spot belongs to this surface. The work provided two main contributions. The former is the analytical derivation of all the observable modes, i.e. all the physical quantities that can be determined by only using the inertial data and the camera observation of the laser spot during a short time-interval. Specifically, it is shown that the observable modes are: the distance of the vehicle from the planar surface; the component of the vehicle speed, which is orthogonal to the planar surface; the relative orientation of the vehicle with respect to the planar surface; the orientation of the planar surface with respect to the gravity. Once the observed modes have been derived, a local decomposition of the original system has also been provided. This decomposition separates the observable modes from the rest of the system and allowed us to introduce a simple recursive method to perform the estimation of all the observable modes (second contribution). The use of a virtual laser spot feature overcomes the limits of feature tracking algorithms and makes our approach suitable to work even in dark or featureless environment. The method is validated by using synthetic data. The validation with real data is in progress. We presented a low-cost low-weight sensor suite and a low computational complexity approach in the framework of aerial navigation. It can be integrated in the framework of autonomous takeoff and landing, safe touch-down and low altitude manoeuvres. However, we want to emphasize that both the contributions are very general and can be applied in other frameworks. In particular, in all the environment where GPS is denied and where the most of objects have planar surfaces (e.g. in an indoor or city-like environment). For instance, these contributions could be used in the framework of humanoid robotics (where visual and inertial sensing are often adopted and the navigation usually occurs in an indoor environment).

Chapter 5

Conclusions

This dissertation tackles the problems of data association and pose estimation of a camera-IMU system, with a focus on MAVs navigation. Due to the limited computation resources of MAVs, a particular attention is devoted to the study of low computational complexity techniques.

Two methods to perform outlier detection on computationally-constrained micro aerial vehicles are presented. The algorithms rely on on-board IMU measurements to calculate the relative rotation between two consecutive camera frames and the reprojection error to detect the inliers. The first method assumes that vehicle's motion is locally planar, while the second method includes the general case of a 6DoF motion. Even if the 5-point RANSAC is the "golden standard method" for 6DoF motion estimation, experimental results shows that the proposed Me-RE and 2-point RANSAC algorithms can be used as a first choice before committing to the 5-point RANSAC due to their very low computational complexity. Considering that the Me-RE algorithm relies on the local planar motion assumption, we remark that it can replace the 5-point algorithm if the motion of the vehicle is smooth and the camera framerate is high. We show that the Me-RE algorithm outperforms standard RANSAC-based methods by up to two orders of magnitude in speed, while being able to identify the majority of the inliers. The motion can then be refined applying standard methods [89], [38] to the remaining inliers. We show that in the case of a monocular camera mounted on a quadrotor vehicle, motion priors from IMU can be used to discard wrong estimations in the framework of a 2-point-RANSAC based approach.

For what concerns the pose estimation problem, two low computational complexity algorithms are provided.

The first is a new approach to perform MAV localization by only using the data provided by an Inertial Measurement Unit and a monocular camera. The approach exploits the so-called "planar ground assumption" and the geometric constraints encoded in a virtual pattern made by three natural point features

belonging to the ground plane. It is based on a closed solution which provides the vehicle pose from a single camera image, once the roll and the pitch angles are obtained by the inertial measurements. This makes the approach very simple in terms of computational complexity and robust since the closed form solution makes unnecessary any initialization. The very low computational cost of the proposed approach makes it suitable for pose control in tasks like hovering, autonomous take off and landing.

In order to overcome the limits of feature tracking algorithms and to provide a solution for featureless or dark environments, we introduce the concept of virtual features. We consider an aerial vehicle equipped with a monocular camera, inertial sensors and a laser pointer. We suppose that the vehicle is moving in the surrounding of a planar surface whose orientation is unknown (assumption that holds in indoor or city-like environments and on landing platforms) and that the laser spot produced by the laser pointer belongs to this surface. The laser spot is then observed by the monocular camera and it represents the unique point feature used in the proposed approach. The difference with respect to classical vision and IMU data fusion problems is that in this case the feature is moving jointly with the vehicle. We analytically derive all the physical quantities (called "observable modes") that can be determined by only using the information contained in the inertial data and the camera observation of the laser spot during a short time-interval. Specifically, it is shown that the physical quantities we can estimate are: the distance of the vehicle from the planar surface; the component of the vehicle speed which is orthogonal to the planar surface; the relative orientation of the vehicle with respect to the planar surface; the orientation of the planar surface with respect to the gravity. Once having derived the observable modes, a local decomposition of the original system is provided. This decomposition separates the observable modes from the rest of the system and allowed us to introduce a simple recursive method to perform the estimation of all the observable modes. Additionally this decomposition provides the link between them and the sensor data. This corresponds to write the equations characterizing the dynamics and the observations only in terms of the observable modes and it is performed by using an extension of the Kalman Canonical decomposition for nonlinear systems [64]. Once obtained the local decomposition, we estimate the observable subspace by using an Extended Kalman Filter.

All the algorithms described in this dissertation have been developed in the framework of MAVs navigation. However, they can be used wherever is required low computational complexity and low payload budget.

5.1 Research Outlook

Since the performed experiments provided encouraging results, future work will include supplementary performance evaluations.

Moreover, the contributions presented in this dissertation can be extended in the following directions:

1. The first one aims to the data association issue.

   1.1 *1-point algorithm* - Real-time on-board implementation, considering the standard deviation of the distribution of the α computed from all the feature correspondences (3.10) as an index of reliability of the Me-RE algorithm. The feature matching set will be preprocessed at each timestamp with the Me-RE algorithm. If the variance of the aforementioned distribution will be higher than a predefined threshold, the resultant motion hypothesis will be discarded and a 5-point algorithm will be run on the whole matching set. On the contrary, the 5-point algorithm will be run only on the resultant inlier set.

   1.2 *2-point algorithm* - Real-time on-board implementation, considering smart policies for the choice of the pairs of features to use (based for example on the feature positions in the image plane and not only on their relative position) in order to reduce the computational complexity of the approach.

2. The second one addresses the pose estimation problem.

   2.1 *Virtual patterns* - Implementation with natural features with smart policies for the choice of the triplet of features to use. Additionally it would be interesting to analyse the computational complexity versus the robustness of the algorithm while considering virtual patterns made by more than three features.

   2.2 *Virtual features* - Real-time on-board implementation to perform autonomous landing and safe touchdown tasks in dark or featureless environments. It is therefore important to consider laser patterns (instead of a single spot) in order to improve the precision. Additionally, we remarked that the performance depends on the laser-module configuration in the camera frame. Hence, the performance can be improved by using a laser module with a servomotor which changes the configuration according to the vehicle trajectory.

Bibliography

[1] Ascending Technologies GmbH. http://www.asctec.de. xii, xv, 3, 72, 74

[2] Delta Drone Innovative Aeronautics. http://www.deltadrone.fr. xii, 3

[3] KMelRobotics. http://kmelrobotics.com. xii, 3, 41, 52

[4] Mikrokopter. http://www.mikrokopter.de. xii, 3

[5] Skybotics UAV navigation solutions. http://www.skybotix.com. 2

[6] Vicon Motion Capture System. http://www.vicon.com. 7

[7] N. Abdelkrim, N. Aouf, A. Tsourdos, and Brian White. Robust nonlinear filtering for ins/gps uav localization. In *16th Mediterranean Conference on Control and Automation*, pages 695–702, 2008. 6

[8] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy. Stereo vision and laser odometry for autonomous helicopters in gps-denied indoor environments. In *Proceedings of SPIE*, 2010. 7

[9] M.W. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart. Onboard imu and monocular vision based control for mavs in unknown in- and outdoor environments. In *Proc. of International Conference on Robotics and Automation*, Shanghai, May 2011. 74

[10] L. Armesto, J. Tornero, and M. Vincze. Fast ego-motion estimation with multi-rate fusion of inertial and vision. *Int. J. Rob. Res.*, 26(6):577–589, June 2007. 60

[11] A. Bachrach, R. He, and N. Roy. Autonomous flight in unknown indoor environments. *International Journal of Micro Air Vehicles*, 1(4):217–228, December 2009. 7

[12] A. Bachrach, R. He, and N. Roy. Autonomous flight in unstructured and unknown indoor environments. In *in Proceedings of European Conference on Micro Aerial Vechicles (EMAV)*, 2009. 7

[13] Y. Bar-Shalom and T.E. Fortmann. *Tracking and data association*, volume 179 of *Mathematics in Science and Engineering*. Academic Press Professional, Inc., San Diego, CA, USA, 1987. 88

[14] M. Bloesch, S. Weiss, D. Scaramuzza, and R. Siegwart. Vision based mav navigation in unknown and unstructured environments. In *Proc. of The IEEE International Conference on Robotics and Automation (ICRA)*, May 2010. 60

[15] S. Bouabdallah. *Design and control of quadrotors with application to autonomous flying*. PhD thesis, Lausanne, 2007. xii, 3, 4, 6

[16] S. Bouabdallah, A. Noth, and R. Siegwart. Pid vs lq control techniques applied to an indoor micro quadrotor. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, Sept 2004. xii, 3, 4

[17] Jean-Yves Bouguet. Camera calibration toolbox for matlab. 2004. 24, 41, 52, 72, 81, 95

[18] D. Brescianini, M. Hehn, and R. D'Andrea. Quadrocopter pole acrobatics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013. 7

[19] William L. Brogan. *Modern Control Theory (3rd Ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1991. 86

[20] P. Castillo, R. Lozano, and A.E. Dzul. *Modelling and Control of Mini-Flying Machines*. Springer, 2005. 71

[21] T. Cheviron, T. Hamel, R. Mahony, and G. Baldwin. Robust nonlinear fusion of inertial and visual data for position, velocity and attitude estimation of uav. In *Proc. of International Conference on Robotics and Automation*, Rome, Italy, April 2007. 61

[22] P Corke, J Lobo, and J Dias. An introduction to inertial and visual sensing. *The International Journal of Robotics*, 26:519–535, 2007. 14

[23] P. Corke, J. Lobo, and J. Dias. An introduction to inertial and visual sensing. *Int. J. Rob. Res.*, 26(6), June 2007. 60

[24] Peter I. Corke. *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer, 2011. 34, 38, 42, 49, 51

[25] Raffaello D'Andrea. Flying machine enabled construction. http://www.idsc.ethz.ch/Research_DAndrea/Archives/Flying_Machine_Enabled_Construction, December 2011. FRAC Centre, Orlans, France. xii, 9

[26] Andrew J Davison. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403–1410. IEEE, 2003. 30

[27] J. Dias, M. Vinzce, P. Corke, and J. Lobo. Editorial : Special issue : 2nd workshop on integration of vision and inertial sensors. *The International Journal of Robotics Research*, 26(6):515–518, June 2007. 14

[28] D. Eberli, D. Scaramuzza, S. Weiss, and R. Siegwart. Vision based position control for mavs using one single artificial landmark. In *Proc. of International Conference & Exhibition on Unmanned Aerial Vehicles*, Dubai, June 2010. 61

[29] JonathanT. Erichsen and J.Margaret Woodhouse. *Human and Animal Vision*. Springer London, 2012. 14

[30] Jay Farrell. *Aided navigation: GPS with high rate sensors*. McGraw-Hill New York, 2008. 68

[31] O.D Faugeras and S. Maybank. Motion from point matches: multiplicity of solutions. *International Journal of Computer Vision*, 4(3):225–246, 1990. 32

[32] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 31

[33] Tanskanen P. Fraundorfer F. and M. Pollefeys. A minimal case solution to the calibrated relative pose problem for the case of two unknown orientation angles. In *European Conf. Computer Vision*, pages 269–282, 2010. 32

[34] P. Gemeiner, P. Einramhof, and M. Vincze. Simultaneous motion and structure estimation by fusion of inertial and vision data. *Int. J. Rob. Res.*, 26(6), June 2007. 60

[35] K. Gillingham and F.H. Previc. Spatial orientation in flight. *Technical Report, Brooks Air Force Base, Texas*, 1993. 16

[36] V. Grabe, M. Riedel, HH Bulthoff, P.R. Giordano, and A. Franchi. The telekyb framework for a modular and extendible ros-based quadrotor control. In *submitted to ECMR*. IEEE, 2013. 42, 53

[37] N. Guenard, T. Hamel, and V. Moreau. Dynamic modeling and intuitive control strategy for an "x4-flyer". In *Control and Automation, 2005. ICCA '05. International Conference on*, June 2005. xii, 3, 4

[38] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2000. 33, 34, 45, 48, 51, 55, 97

[39] S. Hrabar, T. Merz, and D. Frousheger. Development of an autonomous helicopter for aerial powerline inspections. In *Applied Robotics for the Power Industry (CARPI), 2010 1st International Conference on*, Oct 2010. 4

[40] M. Hwangbo and T. Kanade. Visual-inertial uav attitude estimation using urban scene regularities. In *Proc. of International Conference on Robotics and Automation*, Shanghai, May 2011. 61

[41] Alberto Isidori. *Nonlinear Control Systems*. Springer-Verlag New York, Inc., 3rd edition, 1995. 86

[42] F. John. *Partial Differential Equations*. Springer, 1982. 84

[43] E. Jones, A. Vedaldi, and S. Soatto. Inertial structure from motion with autocalibration. In *Proceedings of the Internetional Conference on Computer Vision - Workshop on Dynamical Vision*, 2007. 60, 67

[44] J. Kelly and G.S. Sukhatme. Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration. In *CIRA'09*, pages 360–368, 2009. 60, 67

[45] J. Kelly and G.S. Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *International Journal of Robotics Research*, pages 56–79, 2011. 24, 60, 67

[46] J. Kim and S. Sukkarieh. Real-time implementation of airborne inertial-slam. *Robot. Auton. Syst.*, 55(1):62–71. 60

[47] B.M. Kitt, J. Rehder, A.D. Chambers, M. Schonbein, H. Lategahn, and S. Singh. Monocular visual odometry using a planar road model to solve scale ambiguity. In *Proc. of the European Conference on Mobile Robots*, Orebro, Sweden, 2011. 62

[48] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nara, Japan, November 2007. 61

[49] L. Kneip, M. Chli, and R. Siegwart. Robust real-time visual odometry with a single camera and an imu. In *Proc. of The British Machine Vision Conference (BMVC)*, Dundee, Scotland, August 2011. 32

[50] E. Kruppa. Zur ermittlung eines objektes aus zwei perspektiven mit inner orientierung. In *Sitz. –Ber. Akad. Wiss, Wien, Math. Naturw. Kl., Abt. IIa.*, volume 122, pages 1939–1948, 1913. 32

[51] Jack B. Kuipers. *Quaternions and rotation sequences : a primer with applications to orbits, aerospace, and virtual reality.* Princeton Univ. Press, 1999. 78

[52] V. Kumar and N. Michael. Opportunities and challenges with autonomous micro aerial vehicles. *Int. J. Rob. Res.*, 31(11):1279–1291, 2012. 2, 4

[53] Vijay Kumar. Robots that fly ... and cooperate. http://www.ted.com/talks/vijay_kumar_robots_that_fly_and_cooperate.html, March 2012. xii, 9

[54] Anthony Lawrence. *Modern Inertial Technology: Navigation, Guidance, and Control.* Springer, 2nd edition edition, 1998. 18

[55] J. G. Leishman. The Breguet-Richet Quad-Rotor Helicopter of 1907. http://www.enae.umd.edu/AGRC/Aero/Breguet.pdf, May 2002. 3

[56] Q. Lindsey, D. Mellinger, and V. Kumar. Construction with quadrotor teams. *Autonomous Robots*, 33:323–336, 2012. 7

[57] J. Lobo and J. Dias. Relative pose calibration between visual and inertial sensors. *The International Journal of Robotics Research*, 26(6):561–575, 2007. xiii, 25, 34, 41, 45, 49, 52, 72, 96

[58] Jorge Lobo. Inertial Sensor Data Integration in Computer Vision Systems. Master's thesis, University of Coimbra, 2002. 19

[59] HC Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, MA Fischler and O. Firschein, eds*, pages 61–62, 1987. 37, 51

[60] S. Lupashin, A. Schllig, M. Sherback, and R. D'Andrea. A simple learning strategy for high-speed quadrocopter multi-flips. In *ICRA*, pages 1642–1648. IEEE, 2010. 91

[61] D. Mackenzie. A flapping of wings. *Science*, 335(6075):1430–1433, 2012. 2

[62] M. Maimone, Y. Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186, 2007. 30

[63] A. Martinelli. Closed-form solution for attitude and speed determination by fusing monocular vision and inertial sensor measurements. In *ICRA*. IEEE, 2011. 60, 67

[64] A. Martinelli. State estimation based on the concept of continuous symmetry and observability analysis: The case of calibration. *IEEE Transactions on Robotics*, 27(2):239–255, 2011. 83, 84, 98

[65] A. Martinelli. Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale and bias determination. *Transaction on Robotics*, 28:44–60, 2012. 60, 61, 62, 66, 67

[66] A. Martinelli. Observability properties and deterministic algorithms in visual-inertial structure from motion. *Foundations and Trends in Robotics*, 3(3):139–209, 2013. 60, 67

[67] A. Martinelli. Closed-form solution of visual-inertial structure from motion. *International Journal of Computer Vision*, 106:138–152, 2014. 71

[68] A. Martinelli, C. Troiani, and A. Renzaglia. Vision-aided inertial navigation: Closed-form determination of absolute scale, speed and attitude. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2011. 67

[69] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, and S. Tadokoro. Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 29(5):832–841, 2012. 4

[70] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International journal of computer vision*, 65(1-2):43–72, 2005. 29

[71] Faraz M. Mirzaei and Stergios I. Roumeliotis. A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation. *IEEE Transactions on Robotics*, 24(5):1143–1156, 2008. 24

[72] A.I. Mourikis and S.I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3565–3572, Rome, Italy, April 10-14 2007. 60, 67

[73] O. Naroditsky, X. Zhou, J. Gallier, Stergios I Roumeliotis, and K. Daniilidis. Two efficient solutions for visual odometry using directional correspondence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):818–824, 2012. 32

[74] A. Natraj, C. Demonceaux, P. Vasseur, and P.F. Sturm. Vision based attitude and altitude estimation for uavs in dark environments. In *IROS*. IEEE, 2011. 61

[75] David Nistér. An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):756–770, 2004. 32, 37, 51

[76] D Ortin and JMM Montiel. Indoor robot motion based on monocular images. *Robotica*, 19(3):331–342, 2001. 32

[77] Johan Philip. A non-iterative algorithm for determining all essential matrices corresponding to five point pairs. *The Photogrammetric Record*, 15(88):589–599, 1996. 32

[78] O. Pizarro, R. Eustice, and H. Singh. Relative pose estimation for instrumented, calibrated imaging platforms. In *DICTA*, pages 601–612. Citeseer, 2003. 32

[79] P. Pounds. Design, Construction and Control of a Large Quadrotor micro air vehicle. `http://www.eng.yale.edu/pep5/P_Pounds_Thesis_2008.pdf`, 2007. PhD thesis. xii, 3, 4

[80] R. Ritz, M. Mueller, and R. D'Andrea. Cooperative quadrocopter ball throwing and catching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4972–4978, 2012. 7

[81] I. Sa and P. Corke. Vertical infrastructure inspection using a quadcopter and shared autonomy control. 92:219–232, 2014. 4

[82] D. Scaramuzza. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International journal of computer vision*, 95(1):74–85, 2011. 32

[83] D. Scaramuzza. Performance evaluation of 1-point-ransac visual odometry. *Journal of Field Robotics*, 28(5):792–811, 2011. 32

[84] D. Scaramuzza, M.C. Achtelik, L. Doitsidis, F. Fraundorfer, E.B. Kosmatopoulos, A. Martinelli, M.W. Achtelik, M. Chli, S.A. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G.H. Lee, S. Lynen, L. Meier, M. Pollefeys, A. Renzaglia, R. Siegwart, J.C. Stumpf, P. Tanskanen, C. Troiani, and S. Weiss. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments. *ACCEPTED with minor revision for the IEEE Robotics and Automation Magazine*, 2013. 4

[85] D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. *Robotics & Automation Magazine, IEEE*, 18(4):80–92, 2011. 32

[86] D. Scaramuzza, F. Fraundorfer, and R. Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4293–4299. IEEE, 2009. 32

[87] A. Schoellig, H. Siegel, F. Augugliaro, and R. DAndrea. *So You Think You Can Dance? Rhythmic Flight Performances with Quadrocopters*. Springer International Publishing, 2014. 7

[88] R. Siegwart, I.R. Nourbakhsh, and D. Scaramuzza. *Introduction to Autonomous Mobile Robots*. Mit Press, 2011. xiii, 29, 30

[89] H. Stewénius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, 2006. 32, 45, 48, 51, 55, 97

[90] Bill Triggs. Routines for relative pose of two calibrated cameras from 5 points. *MOVI - IMAG-INRIA Rhône-Alpes / GRAVIR, Technical Report*, 2000. 32

[91] C. Troiani, S. Al Zanati, and A. Martinelli. 1-point-based monocularmotion estimation for computationally-limited micro aerial vehicles. In *European Conference on Mobile Robotics (ECMR)*, 2013. 61

[92] C. Troiani and A. Martinelli. Vision-aided inertial navigation using virtual features. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct 2012. 77

[93] C. Troiani, A. Martinelli, C. Laugier, and D. Scaramuzza. 1-point-based monocularmotion estimation for computationally-limited micro aerial vehicles. In *European Conference on Mobile Robotics (ECMR)*, 2013. 34

[94] C. Troiani, A. Martinelli, C. Laugier, and D. Scaramuzza. 2-point-based outlier rejection for camera-imu systems with applications to micro aerial vehicles. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. 45

[95] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344, August 1987. 24

[96] M. Turpin, N. Michael, and V. Kumar. Trajectory design and control for aggressive formation flight with quadrotors. *Autonomous Robots*, Feb. 2012. 7

[97] M. Veth and J. Raquet. Fusion of low-cost imaging and inertial sensors for navigation. In *Journal of the Institute of Navigation*, volume 54, 2007. 60

[98] C. Vincenzo Angelino, V.R. Baraniello, and L. Cicala. High altitude uav navigation using imu, gps and camera. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 647–654, July 2013. 7

[99] S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012. 61

[100] S. Weiss, D. Scaramuzza, and R. Siegwart. Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments. *J. Field Robot.*, 28(6):854–874, November 2011. 61

[101] Stephan M. Weiss. Vision based navigation for micro helicopters, 2012. 24, 61

[102] J. Willmann, F. Augugliaro, T. Cadalbert, R. D'Andrea, F. Gramazio, and M. Kohler. Aerial Robotic Construction: Towards a New Field of Architectural Research. *International journal of architectural computing*, 10:439–460, 2012. 7

[103] N. Yazdi, F. Ayazi, and K. Najafi. Micromachined inertial sensors. *Proceedings of the IEEE*, 86(8):1640–1659, 1998. 19

[104] B. Yun, K. Peng, and B. Chen. Enhancement of gps signals for automatic control of a uav helicopter system. In *Proc. of International Conference on Robotics and Automation*, Rome, Italy, April 2007. 6

[105] T. Zhang, Y. Kang, M. Achtelik, K. Kiihnlenz, and M. Buss. Autonomous hovering of a vision/imu guided quadrotor. In *Proc. of International Conference on Mechatronics and Automation*, Changchun, China, August 2009. 61

[106] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, 1998. 24

List of publications

Journals

- Troiani, C., Martinelli, A., Laugier, C., D. Scaramuzza, "Low computational complexity algorithms for MAVs Vision-Aided Inertial Navigation", *Robotics and Autonomous Systems* (RAS), submitted, 2014.

- Scaramuzza D., Achtelik M.C., Doitsidis L., Fraundorfer F., Kosmatopoulos E.B., Martinelli A., Achtelik M.W., Chli M., Chatzichristofis S.A., Kneip L., Gurdan D., Heng L., Lee G.H., Lynen S., Meier L., Pollefeys M., Renzaglia A., Siegwart R., Stumpf J.C., Tanskanen P., Troiani C., Weiss S., "Vision-Controlled Micro Flying Robots: from System Design to Autonomous Navigation and Mapping in GPS-denied Environments", ACCEPTED for the *IEEE Robotics and Automation Magazine*, 2013.

Conferences

- Troiani, C., Martinelli, A., Laugier, C., D. Scaramuzza, "2-Point-based Outlier Rejection for Camera-Imu Systems with applications to Micro Aerial Vehicles", *IEEE International Conference on Robotics and Automation* (ICRA), 2014.

- Troiani, C., Martinelli, A., Laugier, C., D. Scaramuzza, "1-Point-based MonocularMotion Estimation for Computationally-Limited Micro Aerial Vehicles", *European Conference on Mobile Robotics* (ECMR), 2013.

- Troiani, C., Al Zanati, S., Martinelli, A., "A 3 points vision based approach for MAV localization in GPS denied environments", *European Conference on Mobile Robotics* (ECMR), 2013.

- Troiani, C., Martinelli, A., "Vision-aided inertial navigation using virtual features", *IEEE International Conference on Intelligent Robots and Systems* (IROS), 2012.
- Martinelli, A., Troiani, C., Renzaglia, A., "Vision-aided inertial navigation: Closed-form determination of absolute scale, speed and attitude", *IEEE International Conference on Intelligent Robots and Systems* (IROS), 2011.