Internship Report

ESTIMATION OF THE GROUND SPEED VECTOR OF AN AIRCRAFT USING AN EMBEDDED CAMERA

Marcos MEDRANO

Tutor AIRBUS:	Alain GUILLET	<alain.guillet@airbus.com>
	Pierre SCACCHI	<pierre.p.scacchi@airbus.com>
Tutor ENSEEIHT:	Simone GASPARINI	<simone.gasparini@enseeiht.fr>

2014

Abstract

Ground speed measurement of an aircraft rolling on ground may be significantly inaccurate at very low speeds due to the performances of the existing sensors (GPS, IRS, Tachymeters). This has motivated the search for a new sensor to provide a complementary estimation of the ground speed and improve the overall accuracy. In this work a vision-based approach is studied to measure speed of an aircraft from the sequence of images of an on-board camera. Different methods are evaluated together with the image processing techniques involved. The selected algorithms are implemented and tested under controlled scenarios. Finally, the sensitivity of the estimation is evaluated by varying different parameters of the camera.

Résumé

Les mesures de vitesse sol d'un avion qui roule au sol peuvent être relativement imprécises à très basse vitesse en raison des performances des capteurs existants (GPS, IRS, Tachymètres). Ceci a motivé la recherche d'un nouveau capteur permettant de fournir une estimation complémentaire de la vitesse sol et d'améliorer la précision globale. Dans ce travail, une approche basée vision est étudiée pour mesurer la vitesse d'un avion à partir d'une séquence d'images d'une caméra embarquée. Différentes méthodes sont évaluées ainsi que les techniques de traitement d'image associées. Les algorithmes sélectionnés sont mis en œuvre et testés dans des scénarios contrôlés. Enfin, la sensibilité de l'estimation est évaluée en faisant varier différents paramètres de la caméra.

Acknowledgements

I want to thank all members of the Research in Navigation (EYAN) and Stability and Control (EYCDR) departments for their hospitality and kindness.

I especially want to thank and express my gratitude to Alain Guillet and Pierre Scacchi, tutors of this work, for the guidance and support they have given me during my 6 month internship. I also want to thank Simone Gasparini for the comments and ideas he has suggested me at key stages of this work.

Finally, I want to thank all those who in one way or another, contributed to the success of my internship and who have not been mentioned here.

Index

LIST OF FIGURES **5**

I. INTRODUCTION **6**

1. Airbus 6
2. Team and methodology 6
3. Planning and specifications 7

II. A VISION-BASED APPROACH FOR GROUND SPEED ESTIMATION **8**

1. Context 8
2. Hypothesis and constraints 8
3. Image processing and analysis 9

III. EGO-MOTION ESTIMATION: STATE OF THE ART **12**

1. Discrete-time approach 12
2. Continuous-time approach 13
3. Other approaches 14
4. Planar scenes 14
5. Summary and selected methods 15

IV. DEVELOPMENT **16**

1. Proposed solution 16
2. Method validation 16
3. Initial prototype 17
4. Synthetic images in perspective 18
5. Scenario definition 19

V. ESTIMATION ISSUES **20**

1. Detected issues 20
2. Proposed improvements 21
3. Evaluation of the improvements 24
4. Summary 28

VI. RESULTS **29**

1. Ego-motion estimator 29
2. Performance measures 29
3. Results for the defined scenarios 29
4. Summary 35

VII. SENSITIVITY EVALUATION **36**

1. Parameters 36
2. Varying focal length 37
3. Varying camera position 38
4. Summary 39

CONCLUSIONS AND FUTURE WORK **41**

BIBLIOGRAPHY **42**

APPENDICES **43**

A.1 Mathematical description 43
A.2 Egomotion: method description. 47
A.x Results of sensitivity evaluation 55

List of Figures

Figure 1. Airbus sites in Europe and worldwide. 6

Figure 2. Organization chart of the departments involved in this work. 7

Figure 3. Examples of existing cameras on commercial aircrafts. 8

Figure 4. Example of feature matching between two images. 9

Figure 5. Optical flow emerging for translational (above) and rotational motion (below) over X, Y and Z axis. 10

Figure 6. An example image sequence for a camera perform a translational motion over a static scene. 10

Figure 7. Only relative information can be obtained without knowledge of the scene. 11

Figure 8. Image of an aircraft rolling on a runway (left), its motion can be associated with the motion of a vehicle. Attitude angles and axis of an aircraft (right). 11

Figure 9. Diagram of a vision-based approach for ground speed estimation. 16

Figure 10. Examples of Feature Detectors included in OpenCV: FAST (2007 features) and ORB (500). 18

Figure 11. Scenes from Visuel flight simulator overt the Toulouse airport. 18

Figure 12. Example of issues with patterns in the scene that may cause incorrect matching under certain motions. 20

Figure 13. Issues with feature detection and matching: features too close to each other and incorrect matching (above), correct matching but inconvenient position of features. 22

Figure 14. Effect of camera rotation. A camera rotated 20° downwards (above) allow to detect a significant feature displacement, while a camera looking forward (below) results in a small useful region to detect features. 23

Figure 15. Features could not be detected for a camera rotated downwards 40° (left), 70° (middle), and 90° (right). 23

Figure 16. Ground speed error obtained with the baseline (left), the filtering of matches (middle), and filtering of matches and homographies (right) for the TAXI scenario. 25

Figure 17. Ground speed error for the TAXI scenario using DLT-based (left) and RANSAC-based homography (right). 26

Figure 18. Ground speed error for the APPROACH scenario using DLT-based and RANSAC-based homography. 26

Figure 19. Ground speed error using a Washout Filter for the APPROACH scenario. Different values of T were used: no filter(top-left); T=1 (top-right); T=1.5 (bottom-left); T=3 (bottom-right). 27

Figure 20. Ground speed error (left) and velocity direction error (right) for the TAXI scenario. 30

Figure 21. Matches of features that were used to perform the estimation at the 19th second of the TAXI scenario. 30

Figure 22. Ground speed error (left) and velocity direction error (right) for the TAXI-R scenario. Regions marked in the graphs correspond respectively to forward motion, left turning, right turning and forward motion. 31

Figure 23. Ground speed error (left) and velocity direction error (right) for the TAKEOFF scenario. 32

Figure 24. Detected features for different instants of the TAKEOFF scenario: 16 features were detected over a region with fixed-distance marks (above); only 2 features over a region with only the runway centerline. 33

Figure 25. Ground speed error (left) and velocity direction error (right) for the APPROACH scenario. 34

Figure 26. Example of features detected over the clean area before the runway. 35

Figure 27. Visuel frames for different simulated focal lengths: 18mm (left), 45mm (middle) and 100mm (right). 36

Figure 28. Image captured by a real camera mounted on the vertical stabilizer of an aircraft (left) and the image generated in Visuel superposing the aircraft body (right). 37

Figure 29. The pinhole camera model. Points in the space are projected onto the image plane. 43

Figure 30. Epipolar geometry, two views of the same scene. 44

Figure 31. The camera-centered coordinate system. Image plane is centered on the z-axis and parallel to the x-z plane. 46

Figure 32. The ground plane equation is modeled in function of the rotational parameters of the aircraft. 52

Figure 33. Ground speed estimation for different focal lengths. Rows represent the scenarios: TAXI, TAXI-R, TAKEOFF and APPROACH while columns represent the focal lengths: 18mm (left), 30mm (middle) and 45mm (right). 56

Figure 34. Ground speed estimation for a 30mm (left) and 45mm (right) focal lengths of a camera mounted on the vertical stabilizer over TAXI, TAXI-R, TAKEOFF and APPROACH (from top to bottom). 57

I. Introduction

1. Airbus

Airbus is positioned as the first aircraft manufacturer in the world with more than 63,000 employees in multiple locations in Europe and worldwide. In 2013, Airbus delivered 626 aircrafts and won 1,503 net orders, the largest commercial performance in the history of aviation. As of July 2014 Airbus has received a total of 14.000 aircraft orders.

The company was born on 1969 as a consortium of aerospace manufacturers. In 1972, the A300 was released, the first twin-engine wide-body aircraft in the world. In 1984, the A320 was launched, the first civilian aircraft to have fly-by-wire controls, an initially controversial feature. By the end of 2000, Airbus launched the development of the A380, currently the world's largest passenger aircraft. In 2006, Airbus launched the A350 XWB (extra wide body), its first aircraft with both fuselage and wing structures made primarily of carbon-fiber-reinforced polymer.

Recently, on January 2014, EADS Group became Airbus Group after a restructuring of the group.

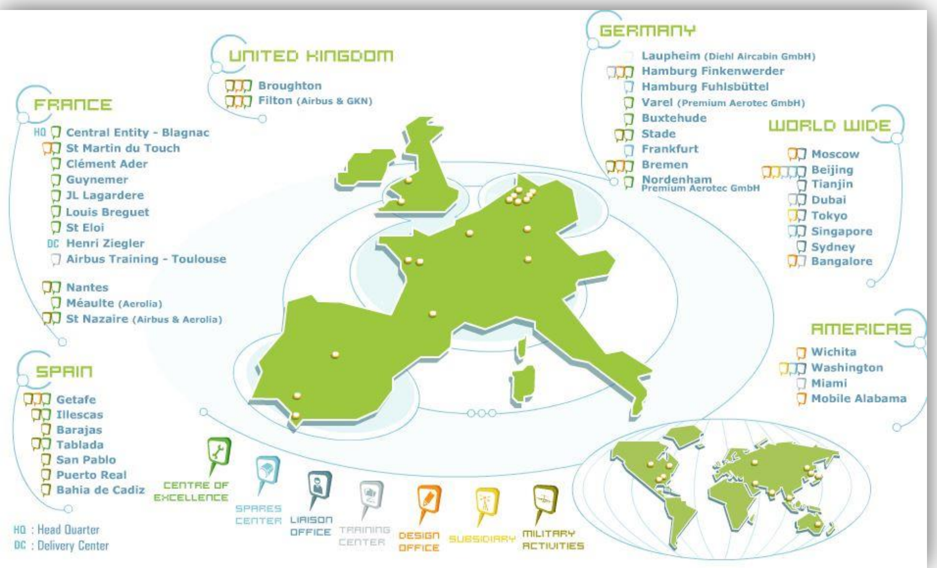


Figure 1. Airbus sites in Europe and worldwide.

2. Team and methodology

The idea that motivated this work was given by the EYAN and EYCDR departments. EYAN is in charge of Navigation Systems, including inertial reference systems and air data systems. Navigation systems contribute highly to the safety and smooth functioning avionics operations and the settings provided by these systems must meet strict requirements for accuracy, availability and integrity. The EYCD department is in charge of the development on Stability and Control of the aircraft. The research team on this subject works on the EYCDR branch of the EYCD department.

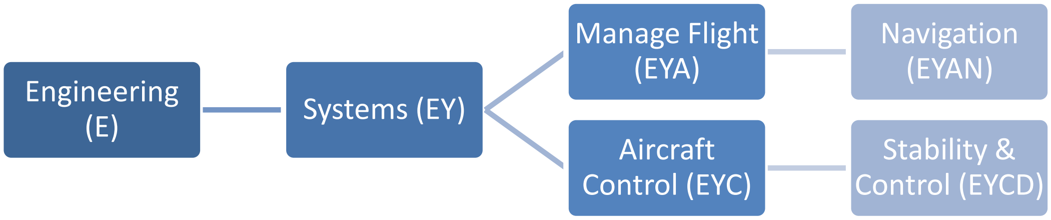


Figure 2. Organization chart of the departments involved in this work.

The internship took place within the EYCDR team at the Airbus site of Saint Martin du Touch. Additional collaborators of this work were Josep Boada-Bauxell (EYCDR), research engineer, and Victor Gibert (EYCDR) currently Ph.D student on a subject of vision-based flight control.

Periodical meetings were held together with the tutors of Airbus to discuss and follow the progress of the internship. The feedback obtained from these meetings allowed to validate decisions, to early identify issues and to define following steps.

Within the EYCDR service, a weekly meeting was held between the chief of the service, employees and interns. The chief of the service communicates relevant information from the department and then the progress on the work of each person was briefly discussed. Constraints and issues are openly discussed and tracked with a visual management methodology.

3. Planning and tools

This 6 month internship took place between March 3, 2014 and August 22, 2014. The following roadmap was initially proposed by the tutors of Airbus:

1. **Bibliography**: Problem understanding and review of associated techniques. State of the art: ego-motion, optical flow, projective geometry, 3D modelling, OpenGL.

2. **Development, basis of the estimation**: Measure the displacement (in magnitude and direction) between two successive images. Implement a simple example in Qt/C++ to validate calculations.

3. **Development, synthetic images in perspective**: Implement the above calculations on a more advanced prototype (generating synthetic images). Improve robustness by using previous states.

4. **Development, real images**: Test the calculations on real images. Integrate a video stream.

5. **Evaluation of robustness and sensibility**: complete the synthetic image generator to evaluate the robustness of the algorithms to disturbances (noise, lighting conditions and visibility).

6. **Integration with simulation tools**: Integrate the prototype in Simbox.

7. **Report writing**: finalize the report and prepare the defense of the internship.

The development was based on the following tools:

- Qt/C++ environment for prototype development.
- Matlab and Simulink environment for algorithm validation.
- OpenCV computer vision library.
- Visuel flight simulator.

II. A vision-based approach for ground speed estimation

1. Context

The ground speed of an aircraft is estimated today by 3 systems: the *Inertial Reference System* (IRS), the *Global Positioning System* (GPS) and the *Tachymeters* installed on the wheels. They all give reliable measurements for most of the aircraft operation. Nonetheless, when the aircraft is rolling on the ground at low speeds (below 10 knots) the measurements can be significantly inaccurate. Furthermore, the IRS suffers from integration drift over time resulting in an error of approximately 5knots after 8 hours of flight. The GPS-based estimation may be lost if the GPS satellites are lost and the tachymeters are unreliable at very low speeds.

This situation motivates the search for a new sensor to measure the ground speed and improve the overall measure accuracy. A vision-based approach aims to measure the ground speed of the aircraft by analyzing a sequence of images taken by a camera fixed to the aircraft. This image measurement of the ground speed can be used as an additional source of data to improve the measurement of existing sensors (a technique called hybridization or fusion).

The idea of using a vision-based approach to measure the ground speed is interesting in many ways:
- Its underlying physical principle is dissimilar and complementary from existing sensors (IRS, GPS).
- It is a nearly autonomous system (although it may require some aircraft parameters as input).
- Aircrafts are usually already equipped with cameras mounted on various locations (fuselage, vertical stabilizer, landing gear and cockpit).
- Cameras can also be used for other means like obstacles or runway detection and visual landing aid.

The main goal of this work is to research image processing algorithms for motion estimation and develop a software prototype to enable the estimation of the ground speed vector of an aircraft based on an images sequence, either synthetic or real.

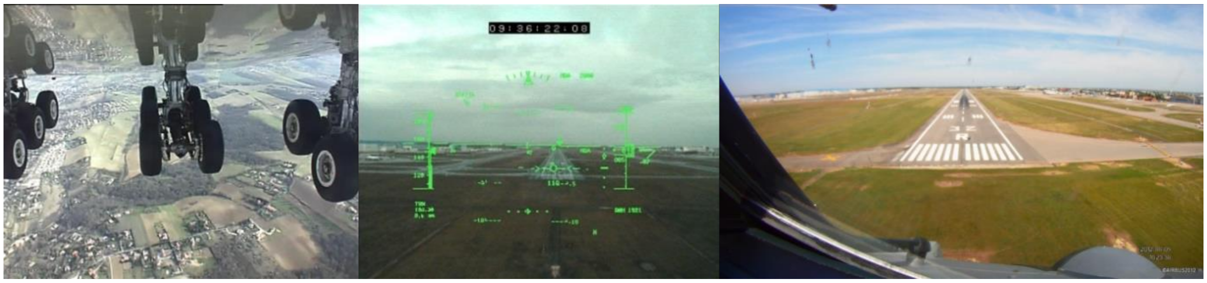


Figure 3. Examples of existing cameras on commercial aircrafts.

2. Hypothesis and constraints

As a first approach to the problem, a number of hypothesis and constraints were considered to simplify the analysis and calculation.

- A single calibrated camera will be considered.
- The camera will be fixed to the aircraft so there is no independent motion between them (c*amera motion* will be used as synonym to *aircraft motion*).
- There is no predefined place to locate the camera on the aircraft. Will be part of this work to find the best place to locate the camera.

- We will consider attitude angles (roll, pitch, yaw) and altitude of the aircraft to be known and available be used at any stage of the process.
- The altitude can be obtained from the air data system, which is the altitude above a given isobar line (1013 hPa or any other pressure selected by the pilot) or the altitude computed by the radio-altimeter system, which is the altitude above the ground just vertically below the aircraft. However the altitude from the air data system is not referenced to the ground, so it may significantly vary with the meteorological conditions, and the radio altimeter measures the directly vertical altitude above the ground surface, so it can fluctuate significantly depending on the ground surface.

3. Image processing and analysis

On the other hand, image processing and analysis algorithms are limited by the characteristics of the images and by the available information of the scene. In order to extract information from the images, a mathematical model is used to represent the physical camera.

Many image processing and analysis techniques have been developed to aid the interpretation of remote sensing images and to extract as much information as possible from the images. The choice of specific techniques or algorithms to use depends on the goals of each individual project.

Feature detection and matching

Feature detection and matching is the process of detect and match a set of *features* over a sequence of images. A feature is usually defined as an *interesting part* of an image leaving what is actually considered interesting to the application behind this process. Examples of features can be *edges*, *corners* (or *interests points*) and *regions*.

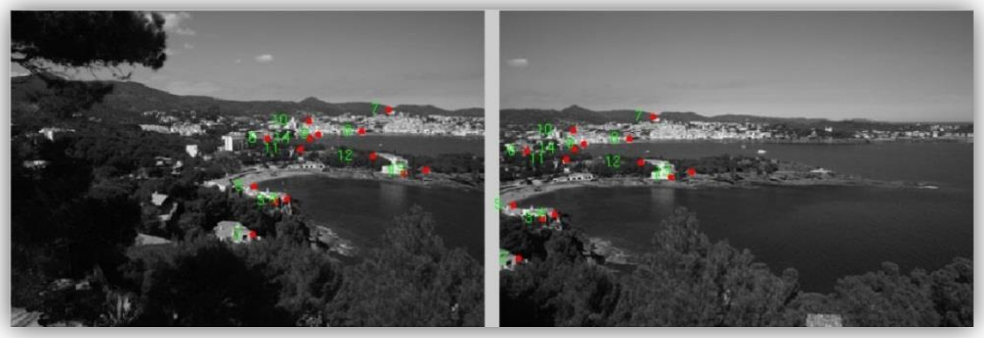


Figure 4. Example of feature matching between two images.

The process of feature detection and matching involves three main phases that are rather independent, different methods exist for each phase.

- <u>*Detection*</u>: low-level image analysis to detect most interesting features.
- <u>*Description*</u>: features must be described in a way that they can be compared with other features. This is usually achieved by grouping certain parameters of the feature (such as image coordinates, neighborhood information and scale or rotation parameters) in a vector called the *feature descriptor*.
- <u>*Matching*</u>: descriptor vectors from two images are matched using some similarity measure. A rather simple approach such as a brute-force all-to-all comparison performs well in practice.

Feature detection and matching are still subjects of research in the computer vision community. Improvements over existing methods, and even novel feature detectors, are proposed regularly.

Optical Flow

The optical flow is the apparent motion of pixels due to the relative motion between the camera and the scene. Since the optical flow represents the visual perception of changes in the scene, it is naturally affected by many factors such as lighting conditions and optical ambiguities.

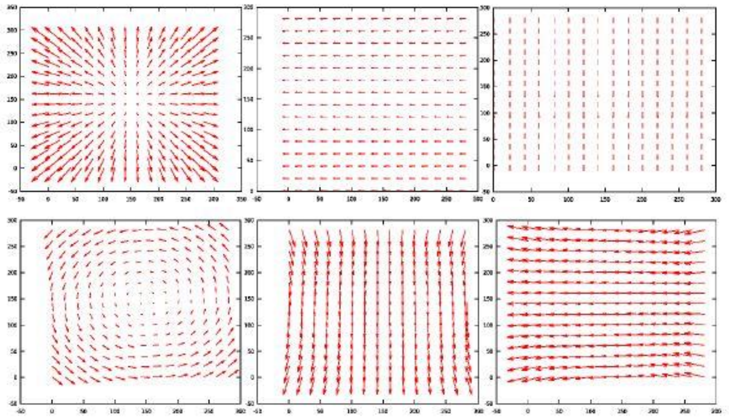


Figure 5. Optical flow emerging for translational (above) and rotational motion (below) over X, Y and Z axis.

Computation of the optical flow usually relies on the *brightness constancy* assumption which states that a feature in the scene does not change its brightness intensity between the two images. Additional constraints such as *local coherence* and *global smoothness* over the optical flow field are assumed depending on the selected method. The former assumes that the flow vector is constant over a spatial neighborhood of a pixel (usually a small square window) and the latter assumes that the resulting flow field should be globally smooth.

Ego-motion estimation

As a camera moves within an environment, a motion is observed in the image sequence. The goal is to recover the 3D motion of the camera (*ego-motion*) relative to the environment, from the 2D motion observed in the image sequence. The relative motion of the camera between two images is defined by its *motion parameters*, the rotational and translational components of the camera relative motion.

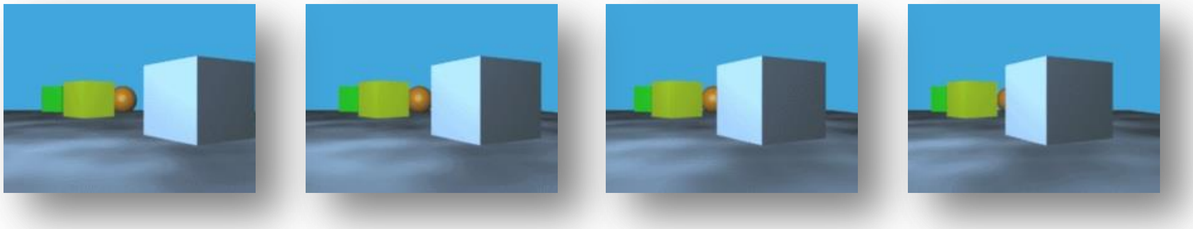


Figure 6. An example image sequence for a camera perform a translational motion over a static scene.

Solutions to the ego-motion problem come essentially from the analysis of the relationship between the 3D motion of a point in the world and the corresponding 2D motion observed on the image plane.

Without prior knowledge of the scene, the translational component can be recovered up to a scalar factor. Since there is no depth information on a single image, it is not possible, for example, to distinguish a small nearby object moving slowly from a big distant object moving fast.

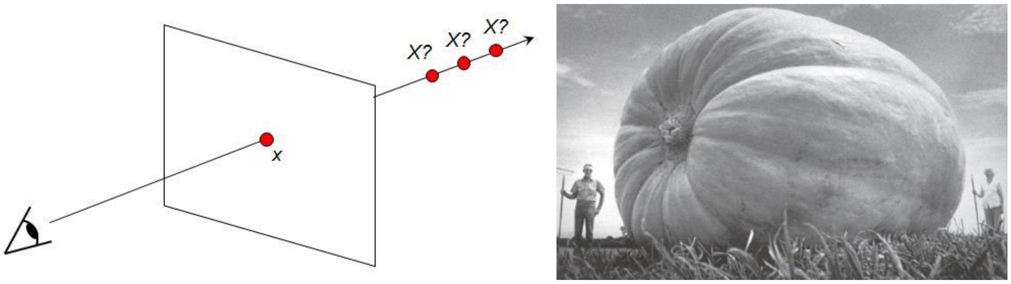


Figure 7. Only relative information can be obtained without knowledge of the scene.

As many other computer vision problems, ego-motion estimation is not considered to be successfully resolved for a generic case. Real applications require non trivial assumptions and simplifications in order to achieve reliable results.

Metric reconstruction

After the motion parameters between two images are estimated, information about the scene is required to compute the actual displacement of the camera relative to the environment. Two scenarios arise for the application being developed.

Aircraft rolling on the ground: While the aircraft is rolling on the ground, its motion can be associated with the motion of a vehicle. Vertical velocity can be assumed to be zero and the height of the camera above the ground can be assumed to be constant. Analytical expressions allow relating the coordinates of points on the ground with the projection of those points in the image plane.

Aircraft in flight: Velocity estimation relies on the measure of the altitude and the rotation angles of the aircraft. When the aircraft is relatively near the ground, situations such as moving objects should be taken into account in order to avoid incorrect measures.

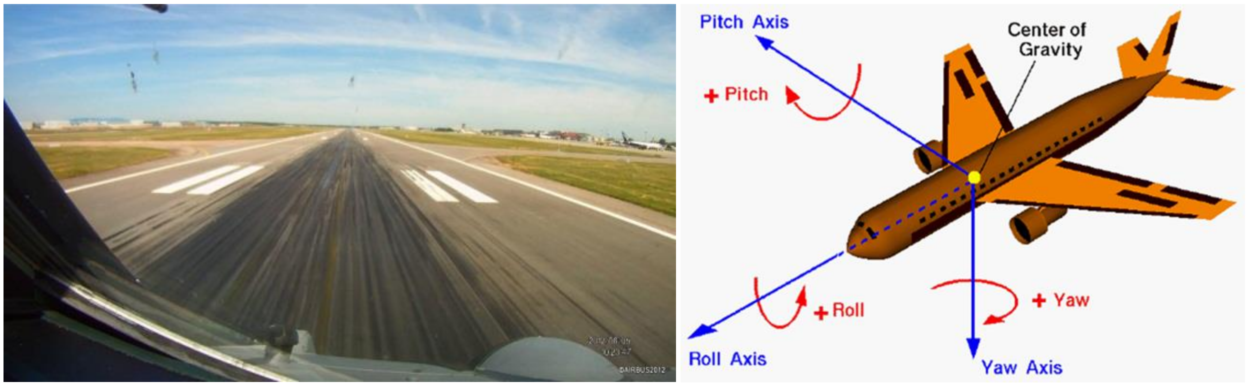


Figure 8. Image of an aircraft rolling on a runway (left), its motion can be associated with the motion of a vehicle. Attitude angles and axis of an aircraft (right).

III. Ego-motion estimation: state of the art

As a camera moves within an environment a motion is observed in the image sequence. The problem is then to estimate the relative 3D motion of the camera (the *ego-motion*) from the motion observed in the image sequence. The estimation process should output the *motion parameters* (R, t) between two or more images taken by the camera.

The motion observed in an image sequence can be evidently affected by other objects moving in the scene. As a first approach to the problem, a *static scene* can be assumed so the motion observed in the image sequence is only an effect of the camera motion. This constraint allows simplifying the equations and it can be removed later by applying techniques such as *segmentation* for different moving regions.

Complete ego-motion estimation is a complex task due mostly to the visual ambiguities in the interpretations of camera rotations and translations which are proven to be inherent to the problem and thus algorithm-independent (Adiv, Inherent ambiguities in recovering 3D Motion and structure from a noisy flow field, 1989) (Longuet-Higgins H. C., 1986).

As described before, the motion detected at each point in the image depends not only on the translation and rotation components of the ego-motion but also on the actual *depth* of the point. However, depth affects only the translational component of the motion since a rotational motion have the same effect for all the points regardless their distance to the camera.

Existing techniques to compute the ego-motion are usually categorized as *discrete-time* or *continuous-time* methods depending on whether the input is a set of point correspondences or image velocities.

Discrete-time approaches are based on tracking image features over a sequence of images while continuous-time approaches are based on the optical flow measured at some or all image points in a single image. There have also been proposed direct methods to recover the 3D motion from the spatiotemporal gradients of image brightness without the need of track features at all.

Regardless of the approach, there usually exist some scene configurations that prevent to uniquely define the geometry of the scene. This is the case of the so called *planar degeneracy*, essentially a configuration where all the points being tracked lie on a plane. In practice, this problem arises either when a scene has dominant planes or when the observed scene is far from the camera. Both rather typical cases for the application being considered of a camera attached to an aircraft.

1. Discrete-time approach

Discrete-time (or point-based) methods start by computing the epipolar geometry of the scene from the position of corresponding image features in successive frames. Once the geometry of the scene is defined, (essentially a set of parameters that characterize the scene) the relative orientation of the camera is deduced and the motion parameters are recovered[1].

The 3D displacement of the camera can be recovered using an intrinsic geometric constraint, called the *epipolar constraint,* between two images of the same scene (Longuet-Higgins H. , 1981). This was first described in the case of a calibrated scenario (where the internal parameters of the camera are known)

[1] For more detail, see the appendix *Epipolar Geometry*.

and it was generalized later removing this assumption. The algorithm is widely known as the *8-points algorithm* because it requires at least 8 points generally distributed in the scene.

The idea is to compute the so called *essential matrix*, which represents the geometric epipolar constraint of the scene, from sufficient point's correspondences. Once the essential matrix is computed, rotation and translation can be found from the essential matrix. In the case of a non-calibrated scenario a unique solution cannot be found since the matrix represents the 3D displacement up to a projective factor (in this case the matrix is called *fundamental matrix*).

The advantage of discrete-time methods is that they can recover exact 3D motion parameters, at least in a calibrated scenario, without making any assumption over the knowledge of the scene and that they are computationally simpler because the estimation involves linear algebraic techniques.

In principle, the use of the epipolar constraint gives a correct estimation of the camera displacement only when the displacement between the two images is relatively large. This is not generally the case of two consecutive frames from a video sequence where the high sampling rate results into small camera displacement and the estimation results become less reliable. A general conclusion found in the literacy is that discrete-time methods do not fit real-time applications, meaning that they actually require a large baseline to provide a reliable estimation.

2. Continuous-time approach

A differential approach is to recover the 3D velocity of the camera from the optical flow field (u, v) given at some or all image positions(x, y). The problem can be formulated in terms of nonlinear equations, with the motion parameters as unknowns, and then solved numerically. The main concern of these methods is their sensitivity to noise input data and its computational complexity.

Most of the reviewed methods started from a constraint that relates the optical flow (u, v) at some point, with the linear and angular velocities(V, ω). Since the motion parameters interact non-linearly in the motion equation, it is not possible to find six linear equations to estimate the six motion parameters. Therefore, rotation and translation components are generally estimated separately using one component to compute the other. This is usually referred as either a *translational-first* or a *rotational-first* approach.

From the state-of-the-art reviewed methods, the only rotational-first method found was proposed by Prazdny (Prazdny, 1980). It requires solving a system of nonlinear equations numerically and it resulted in a computational expensive method that requires a good initial guess to obtain reliable results.

It was noticed later that is easier to compute the translation component first. This is based on the observation that if two 3D points are at different depths and projects to nearby image locations, then the flow vector difference between them points toward the so-called focus of expansion (FOE) which represents the direction of heading. However, this approximation is only valid when the depth difference is large which correspond generally to occlusion boundaries. These are regions where it is particularly difficult to measure the optical flow.

Bruss and Horn proposed a global approach that chooses the 3D motion that minimizes a least-squares criterion (Bruss & Horn, 1983). They derive closed form for both rotation-only and translation-only motions and proposed a set of linear and non-linear equations for the general case that were solved using numerical optimization. Adiv proposed a similar approach but solving the equation for subdivided optical-flow field patches (Adiv, Determining three-dimensional motion and structure from optical flow generated by several moving objects, 1985). Instead of using iterative optimization, the translational component is searched over the space of translational motions for each patch. He also proposed to group patches that share the same translation thus performing a segmentation of different moving regions in the image.

Jepson and Heeger (Jepson & Heeger, 1992) proposed a linear subspace method using a set of constraint vectors to estimate the translation. This is based on the observation that the difference between any two flow vectors gives a constraint on translation independent of rotation. They also avoided the use of an iterative optimization by sampling the solution space of candidate translations.

The methods that use the bilinear constraint are all statistically biased. Kanatani (Kanatani, 1993) suggested both a continuous-time method (based on a differential version of the epipolar constraint) and a renormalization procedure to automatically remove the bias by compensating the unknown noise.

There has also been proposed a differential version for the essential matrix algorithm (Ma, Soatto, Kosecka, & Sastry, 2003) that adapts the solution to the continuous case.

3. Other approaches

Direct methods have also been proposed to estimate the ego-motion directly from image intensities (Hanna, 1991) (Negahdaripour & Horn, Direct Method for Locating the Focus of Expansion, 1989). In these methods, the ego-motion is computed from the spatiotemporal gradients of image brightness rather than from features positions or from optical flow vectors.

Even though these approaches do not rely on the need of a robust feature detector or a complete optical flow computation, they also have limitations and constraints. In principle, they do not seem well adapted for general motions (only for constrained motion) since a rotational motion is usually required.

4. Planar scenes

Planar scenes represent a special case for ego-motion estimation methods. When all the points being tracked lie on a plane in the scene, the standard ego-motion techniques become unreliable or provide not estimation at all. In discrete-time approaches, for example, the problem arises from the fact that an infinite number of matrices satisfy the epipolar geometry if all the points lie on a plane.

This problem is usually addressed by including in the algorithm the knowledge of the planarity of the scene, for example (Szeliski & Torr, 1998), or by using additional images, multi-view techniques (Pollefeys, Verbiest, & Van Gool, 2002).

The *homography decomposition* is a widely used technique to estimate the ego-motion between images of planar scenes (Faugeras & Lustman, 1988). The homography matrix constraints the position of the image points between the views in a similar way as the essential matrix approach. The decomposition of the estimated homography allows retrieving two solutions for the motion parameters between the views and additional hypothesis are usually required to disambiguate the solutions. Numerical and analytical methods have been proposed to decompose the homography matrix into the motion parameters (Malis & Vargas, 2007).

A relatively simple method was found in the literature that models the ground equation in terms of the rotational angles and height of an unmanned aerial vehicle (Oreifej, Lobo, & Shah, 2011). This makes it a method suitable to the application being considered where the camera is attached to an aircraft.

Direct methods that address the planar case have also been proposed (Negahdaripour & Horn, Direct Passive Navigation- Analytical Solution for Planes, 1985).

The planar case is particularly important for the current application because during most of the aircraft operation the camera will capture planar scenes. Although this depends on the orientation of the camera within the aircraft, both ground and flight operations may include planar scenes.

5. Summary and selected methods

Continuous approach seems to be better adapted to process a video stream where the motion of the camera is small relative to the sampling frame rate. In the other hand, discrete approach is relatively simpler and allows finding closed form solutions for the motion parameters. In general, both approaches have the constraints and limitations of the techniques which they rely on.

Continuous-time method are less computationally expensive as they are based on the optical flow computation, while discrete-time approaches, based on feature detection and matching are rather computationally expensive mainly for the need of computing feature descriptors.

	Discrete approach	Continuous approach
Underlying principle	Epipolar constraint	Epipolar constraint or motion parallax
Pre-requisites	Feature Matching	Optical Flow
Motion parameters	Motion parameters are usually estimated at the same time	One component is estimated first and then it is used to estimate the other.
Constraints	Requires significant camera displacement between images. Linear equations system. Least-squares approach.	Requires a small (differential) displacement between images. Non-linear equations system Numerical optimization is usually required

Even if discrete approach requires a significant displacement, which may imply latency in the order of a second, the estimation may still be used to improve the performance of existing sensors at a hybridization stage.

The continuous-time methods studied in this work seemed to differ mostly in the mathematical manipulations on the motion equations. They use different techniques to formulate the nonlinear equations required to solve the motion parameters (Tian, Tomasi, & Heeger, 1996). The choice was then based on the apparent simplicity and applicability to the present application.

Regarding the planar case, it may be an issue both in flight and ground maneuvers. If dominant planes are detected in the scene, the planar case must be considered in the algorithm to obtain reliable results.

Given that discrete approach seems relatively easy to implement, it was decided to retain one of each approaches to compare their results. The eight-point algorithm and its version for planar scenes were selected as representative methods for the discrete approach. The planar-ego method was selected as a continuous approach since it includes flight parameters in the motion equations and it addresses the planar case.

IV. Development

1. Proposed solution

The solution initially proposed for the visual-based ground speed estimation was based on the egomotion estimation problem.

Two paths were initially considered: a discrete approach based on feature matching, and a continuous approach based on the estimation of the optical flow.

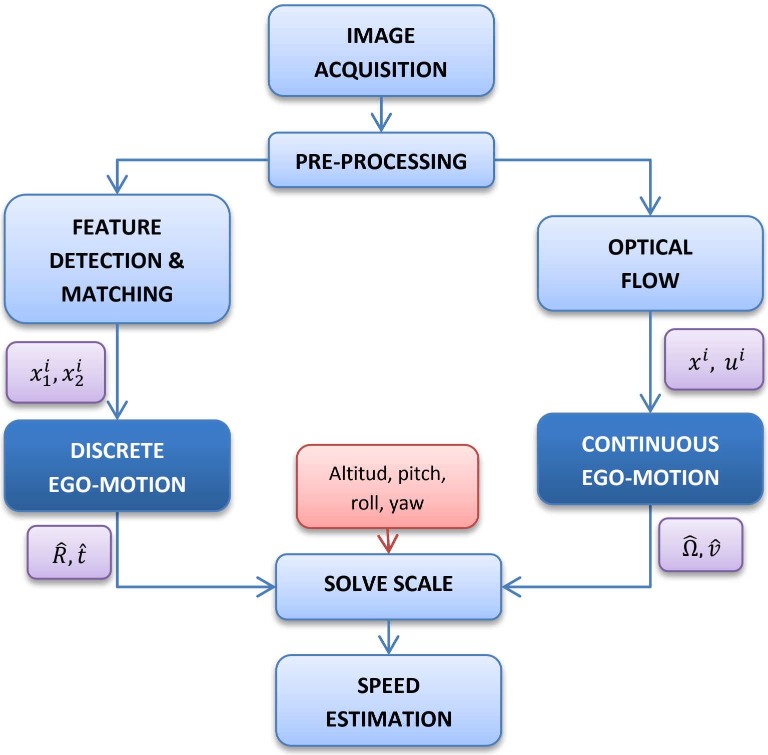


Figure 9. Diagram of a vision-based approach for ground speed estimation.

Discrete ego-motion estimates the discrete displacement between frame i and frame $i + k$ of an image sequence. In contrast, continous ego-motion analyses the images frame by frame and outputs an estimation of the instantaneous speed.

2. Method validation

The methods described in the previous section were validated in order to isolate the ego-motion estimation process from the other stages of the solution (avoiding the need for a feature matching and/or optical flow solution). This allows also to perform a quick validation of the results under a controlled scenario (known parameters) and to understand the implementation requirements.

The isolation of the ego-motion estimation was achieved by generating random 3D points of a scene and projecting them into two different camera frames with known relative rotation and translation.

The eight-points algorithm and the four-points algorithm for planar scenes were implemented in a *Matlab* environment. A few scripts were also implemented to allow the simulation of a scene and two images. The scripts allow quickly developing and testing new methods.

For the simulations performed with a completely planar scene, the eight-point algorithm did not work since the matrix of points correspondences does not have the rank required to estimate the essential matrix. Similarly, no estimation could be performed with the eight-points algorithm for a pure rotational motion, hence a translational motion between images is required.

In the other hand, the four-point method gave always a result whenever at least four-points are detected in the scene. The results, however, are not reliable if the points do not belong to a common plane because the homography matrix relates different views of the same plane.

Regarding the planar-ego method, the authors made available the Matlab implementation of the algorithms that was used to perform the validation of the method.

If their constraints are respected, all the methods recover reliably the translational and rotational component of motion for exact point's correspondences (noiseless case).

3. Initial prototype

The prototype was developed in a Qt/C++ environment using the OpenCV computer vision library. OpenCV library was built from sources in order to use the same C++ compiler included in the Qt environment. A guide was written during the internship to explain the building process in detail. See appendix *Building OpenCV for Qt on Windows*.

Selecting a Feature Tracking solution

Discrete ego-motion estimation relies on feature detection and matching. Desired characteristics of a feature detector and matcher can be its *quality* to detect good features, its *speed* and its *invariance* to image changes. For example brightness and scale changes should not affect feature detection.

The following feature detectors, extractors and matchers are currently available in OpenCV 2.4.8.x:

TYPE	INCLUDED IN OPENCV
Feature Detectors	Harris, Shi-Tomasi, FAST, SIFT, SURF, ORB
Descriptor Extractors	SIFT, SURF, BRIEF, BRISK, ORB, FREAK
Descriptor Matchers	Brute-force (norm L1 or L2), Brute-force Hamming, Flann-based matcher

Harris and Shi-tomasi detects corners by computing intensity gradient in all directions using a window of analysis. They are not scale-invariant thus they are not well adapted to our needs.

SIFT and SURF are robust, scale-invariant, alternatives. However, these methods are patented. FAST algorithm is a free alternative to SIFT/SURF, although its main disadvantage is that it detects a large ratio of outliers. Finally, ORB is another efficient alternative to SIFT/SURF proposed by OpenCV Labs. It is essentially a FAST detector with a BRIEF descriptor that achieves rotation-invariant and scale-invariant detection. This makes ORB the most promising detector for the current application.

Since an exhaustive comparison between all the possible combinations is not the main objective of this work, the selection was based on public benchmarks and on the results obtained for simple test cases.

A simple program was developed to perform feature detection and matching for images from a video sequence of a landing at the Toulouse airport.

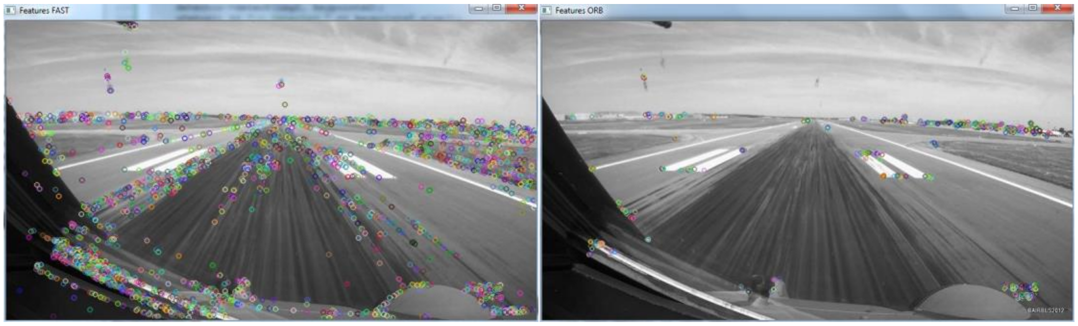


Figure 10. Examples of Feature Detectors included in OpenCV: FAST (left) and ORB (right).

As it can be seen from the images, undesirable features may be detected in the sky, in the airplane boundaries and even in the imperfections of the window itself (for an internal camera). This happened even though the images are relatively clean and noiseless. Robustness of feature detection should be analyzed in order to improve the overall accuracy of the estimation.

Implementation and testing

The initial prototype consisted in a program that processes frames from a video and computes discrete egomotion. It included abstraction classes for the feature tracking and egomotion estimation, the input video, region of interests, camera matrix and number of frames to skip between frames (frame step).

A known set of features was used as input for both the C++ and the Matlab implementations using the same camera parameters to detect and correct differences between algorithms.

The results obtained with a video of a real landing were not reliable as expected since the calibration information of the camera was not available and the sequence represented a hard challenge for the initial ego-motion implementation.

4. Synthetic images in perspective

The algorithms were tested under a more realistic scenario over synthetic images in perspective. A flight simulator was provided during the internship to test the robustness and sensitivity of the algorithms.

Visuel flight simulator

Visuel is a flight simulator that simulates a realistic environment using satellite images. It was developed in C++ using Qt and OpenGL. The simulator allows to control the aircraft and to adjust scene parameters.

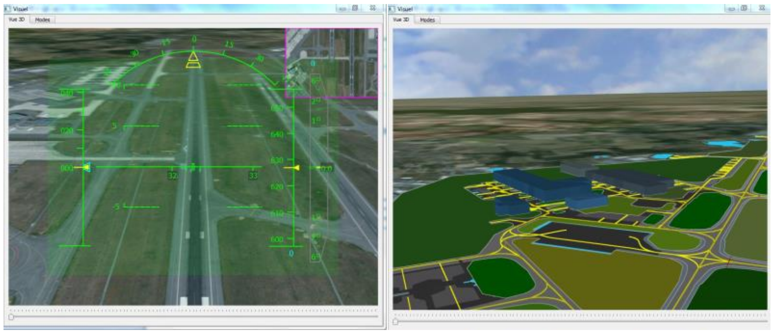


Figure 11. Scenes from Visuel flight simulator overt the Toulouse airport.

The following considerations were taken into account while working with Visuel.

- **Speed**: Wind is not simulated in Visuel which simplifies the aircraft speed computation. The vertical and ground speed are computed as follows $vert.speed = speed * \sin(FPA)$ and $GS = speed * \cos(FPA)$ where FPA (*flight path angle)* is the angle between the horizontal and the velocity vector, which describes whether the aircraft is climbing or descending.
- **Scene**: Every feature detected in the scene corresponds to a feature on the ground. The altitude used within the algorithm was the altitude above the ground level.
- **Views and camera position**: Visuel offers four different views to display the scene: Cockpit, Ground, Orbit and Pilot Fish. Only Cockpit and Pilot Fish views were used in this work.
- **Aircraft**: No aircraft model was included with Visuel. The simulations consisted essentially in a camera attached to an imaginary aircraft, following with the specified trajectories.

5. Scenario definition

The scenarios defined in Visuel attempted to cover the main maneuvers of the aircraft while it is rolling on the ground at low speeds. The interests was focused in measuring the performance of the estimator while varying specific parameters of the simulation, and not in simulating exact real conditions. Varying roll or pitch of the aircraft, for example, was not considered as important as performing a turn over the taxiway or an acceleration over the runway.

SCENARIO	DESCRIPTION
TAXI	Aircraft rolling on the taxiway at a low constant ground speed (20 knots). The aircraft performs a pure forward motion with no turns. The simulation time is 35 seconds.
TAXI-R	Aircraft rolling on the taxiway at a low constant ground speed (20 knots). The aircraft performs forward motion and two turns. The simulation time is 45 seconds.
TAKEOFF	Aircraft performing acceleration (speed from 0 to 130 knots) over the runway from the beginning of the runway. The simulation time is 20 seconds.
APPROACH	Aircraft approaching to the runway at a constant ground speed (129.8 knots) and constant FPA (-3°). The distance was set to 3km and the altitude was calculated so the aircraft lands over the runway. The simulation time is 50 seconds.

Other scenarios such as LANDING and CRUISE were also considered but, for the purpose of estimating the ground speed of the aircraft, a LANDING scenario will result almost symmetric to the TAKEOFF scenario and, in a similar way, a CRUISE scenario with constant altitude and constant ground speed will result almost equivalent to the TAXI or APPROACH scenarios.

The interest of making a distinction between the APPROACH and TAKEOFF (or LANDING) scenarios is that in a takeoff maneuver the aircraft goes fast over the runway (there is a fast scrolling of the features detected on the ground) and vary its ground speed, while in the approach maneuver the interest was to evaluate the estimation with a constant ground speed while varying of the altitude of the aircraft.

The initial conditions for each scenario were specified in Visuel as a set of parameters and the duration of each scenario was controlled manually.

V. Estimation Issues

1. Detected issues

The initial results obtained with the discrete feature-based ground speed estimation were not reliable. Even under the constrained TAXI scenario the magnitude of the error was considerable large, the estimation showed instability and the magnitude of the errors were not acceptable. The goal was then to detect the causes that lead to bad estimations and to propose and develop improvements to obtain an acceptable ground speed estimator.

The causes leading to a bad estimation were more likely to be found in the output of the feature matching stage and the homography estimation. It is important to notice that the whole estimation is relying strongly on the quality of these two stages.

Issues related to the selected matches: Analyzing the results of the feature matching stage a number of issues were identified.

- Features too close to each other. If the detected features are too close to each other, small errors in the features positions have a large impact on the estimated homography.
- Incorrect matches (*outliers*). Even if they represent a small percentage of the total matches, outliers can significantly degrade the homography estimation since all the matches are considered for the computation.

Issues related to the scene: The issues identified above can be partially avoided by filtering the selected matches before they get to the homography estimation stage. However, other issues are related to the characteristics of the scene and in principle they cannot be avoided.

- Degenerated features configurations. The algorithm expects that the features lie in a general configuration on the plane. If the features are located in a degenerated configuration, such as all features lying on the same line, the result is similar to the features being too close to each other. Small errors in the features position will have great impact on the estimated homography.
- Very few detected features. Whenever the number of detected features is less than 8 or 6 features, the homography matrix is significantly degraded. Having a large number of features correspondences helps to reduce the impact of the error in the features position.
- Patterns. Correctly matching features within patterns in the images is a real challenge. A feature from the pattern may be incorrectly matched within the pattern as it is the case in the image below. This applies for the patterns that may occur in the runway or road markings.

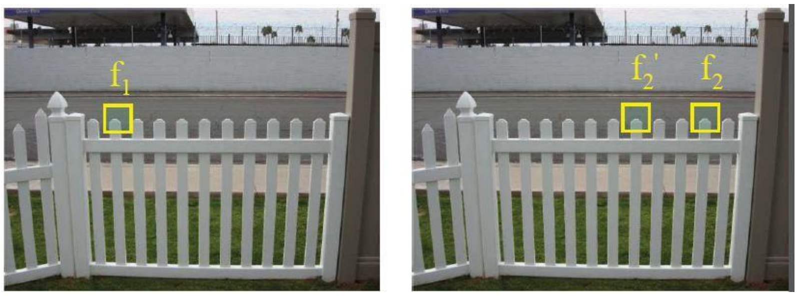


Figure 12. Example of issues with patterns in the scene that may cause incorrect matching under certain motions.

The situations described above allow concluding that the discrete feature-based motion estimation from the homography matrix has two main issues. The estimation is very sensitive to incorrect matches (outliers) from the features matching stage. In addition, small errors in the feature location seem to have great impact on the resulting homography (especially when using a small number of matches).

The estimation may also be degraded if the features are not located in a general configuration on the ground.

2. Proposed improvements

A few improvements were proposed and analyzed together with the tutors of Airbus and ENSEEIHT. The focus was mainly put on avoiding the use of outliers in the homography estimation stage and reducing the impact of errors in the features location, although other general improvements were also discussed.

Detect and filter bad matches

Before estimating the homography matrix, the matches can be filtered using different criteria:

- Feature displacement. If a feature does not have a significant displacement between frames (it has not moved), the match can be discarded since it will degrade the quality of the homography.
- Distance between features of other matches. For every match to be considered in the estimation, discard any other match whose features are closer than a certain threshold.
- Quality. Since the matches can be sorted by their quality[2], only the best matches can be considered for the homography estimation. This can be done either by selecting a fixed number of matches (the N best matches) or by using a threshold to discard matches.

These relatively simple changes can reduce the chances of getting a degraded homography matrix. However, as discussed before, they cannot guarantee a robust estimation since there can still occur cases where incorrect matches are not filtered.

Identify degraded homographies

The idea was to find indicators that help to predict if a homography will give rise to incorrect motion estimation. The **condition number** of the homography and the **re-projection error** were computed and tracked together with the ground speed estimation. The number of features that were used to compute the homography was also considered.

The following situations were detected whenever the estimated ground speed differed significantly from the actual ground speed:

- The condition number of the homography increased significantly.
- The computed re-projection error increased significantly.
- The number of features correspondences used was below 10.

These situations may be used to predict the quality of the homography matrix and avoid computing the motion from it. However, it should be noticed that the absence of these situations does not imply reliable motion estimation. The *goodness* of the homography matrix will indicate a correct modeling of

[2] Feature matching consists essentially in measuring the *distance* between feature descriptors. This distance (provided by the feature matcher) can be used as an indicator of the quality of the match.

the motion of the detected features, and because the detected matches may not model the scene correctly, this does not necessarily imply correct ground speed estimation.

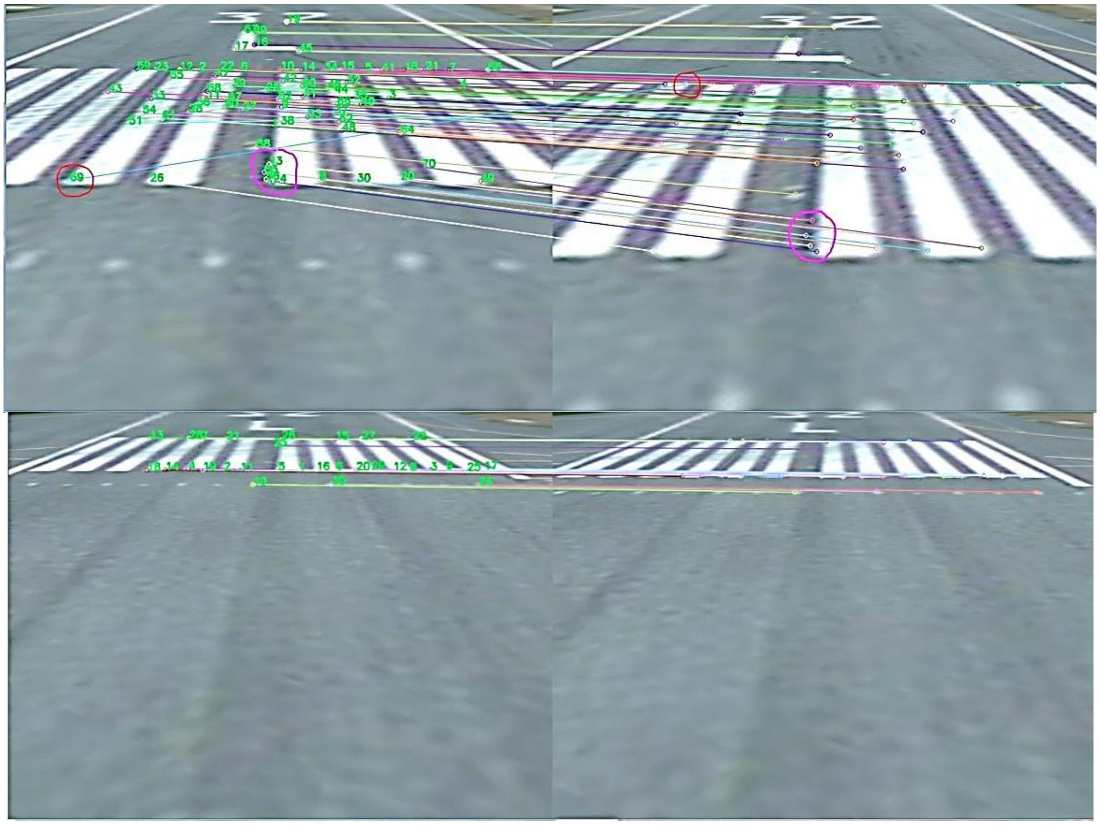


Figure 13. Issues with feature detection and matching: features too close to each other and incorrect matching (above), correct matching but inconvenient position of features.

Camera orientation

Rotating the camera between 15° and 30° downwards resulted in a more stable estimation. The reason again is related to the feature detection stage.

As the homography estimation is based on the measured feature displacement, rotating the camera downwards had the following advantages that tend to improve the overall estimation performance:

- There is a wider area in the images for features to be detected.
- There is a decrease in the feature position error.
- There is a more significant displacement between features.

When the camera is looking forward, only a small region of the image is useful to detect features and the computed displacement will be more sensitive to errors in the measured features positions. In a different way, if the camera is rotated downwards below 35° the resulting image does not have sufficient scene detail to detect features and the motion cannot be estimated.

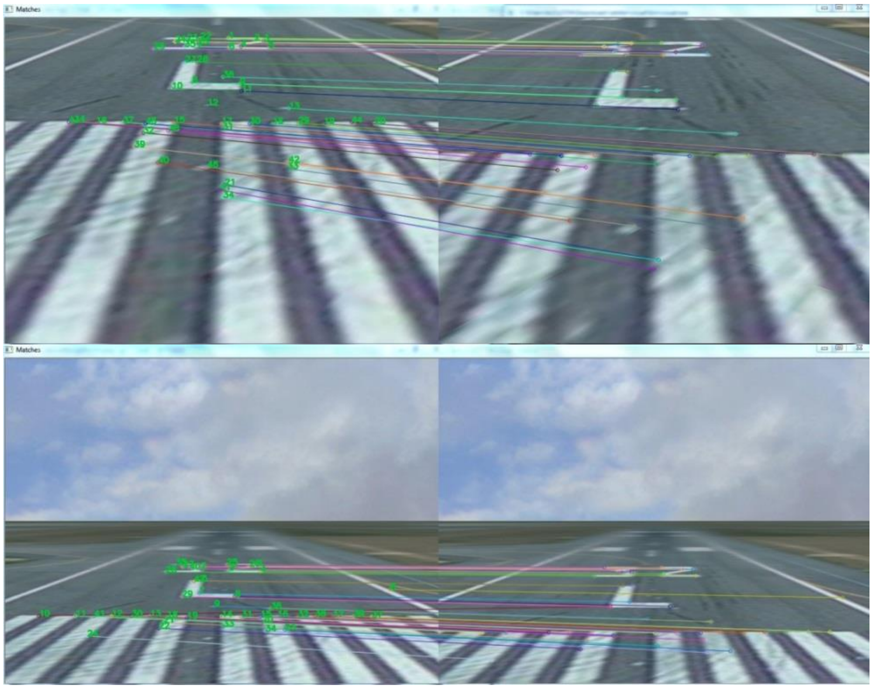


Figure 14. Effect of camera rotation. A camera rotated 20° downwards (above) allow to detect a significant feature displacement, while a camera looking forward (below) results in a small useful region to detect features.

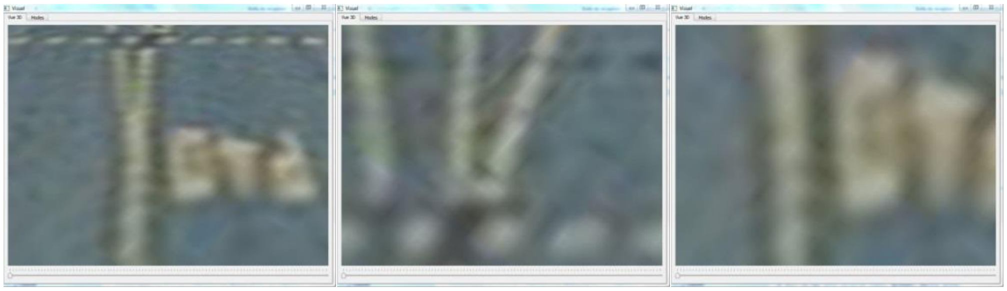


Figure 15. Features could not be detected for a camera rotated downwards 40° (left), 70° (middle), and 90° (right).

RANSAC

The *Random Sample Consensus* (RANSAC) is a non-deterministic method to estimate the parameters of a mathematical model from noisy data. It is widely used for the homography estimation since it addresses the problem of the robustness to outliers.

The basic idea is to iteratively compute a homography from a small random subset of the matches and measure the re-projection error of using the computed homography matrix. Specifying a threshold for the re-projection error, the method can count the number of correct matches (*inliers*) and the homography matrix that shows the largest consensus (largest set of inliers) is selected as solution.

Despite the effort that can be put to filter outliers before, they may nonetheless reach the homography estimation stage, so using the RANSAC method will result in a robust homography estimation by efficiently removing these outliers from the matches set. The improvement over the DLT estimator will be seen, evidently, whenever there are outliers in the set of computed matches.

Improve stability by using previous estimations

The stability of the final output of the ground speed estimation can also be improved by using the results of previous estimations. For this, the used of a *washout filter* was proposed by the tutors of Airbus.

A washout filter allows differentiating and smoothing the changes in the input signal improving the stability of the measurements. A discrete-time formula can be obtained using the bilinear transform: $S(k) = \frac{a-1}{a+1} S(k-1) + \frac{1}{T}\frac{a}{(a+1)}\left(E(k)–E(k-1)\right)$ where T is the time constant of the filter, e is the sampling period and $a = 2T/e$.

Within the ground speed estimator, the only parameter specified to the filter was the time constant T while e was taken as the time between the frames used to estimate the motion. This is computed as the frames step divided the number of frames per seconds of the simulation.

The input of the filter ($E(k)$) is the position of the aircraft; however the difference $E(k) - E(k-1)$ was used since it is the norm of the estimated translational vector.

$$currentSpeed = \frac{a-1}{a+1} previousSpeed + \frac{1}{T}\frac{a}{(a+1)} |velocity\ vector|$$

3. Evaluation of the improvements

The improvements proposed in the previous section were evaluated under the TAXI and APPROACH scenarios. The goal was to measure their impact on the overall estimation error over the initial estimation method without any improvement.

Performance measures

The following measures were considered to evaluate the performance of the estimation:

- The maximum error committed during the simulation (Max.)
- The mean of the error (ME)
- The mean of the absolute error (MAE)
- The Root Mean Squared Error (RMSE)
- The standard deviation of the error (SD)

The ratio between number of estimations performed over number of estimations attempted (estimation ratio) was also measured.

Filtering of matches and homographies

It can be seen from the graphs that the filtering of homographies allows to avoid the peaks in the ground speed. As expected, it is only avoiding the use of degraded homographies for the motion estimation.

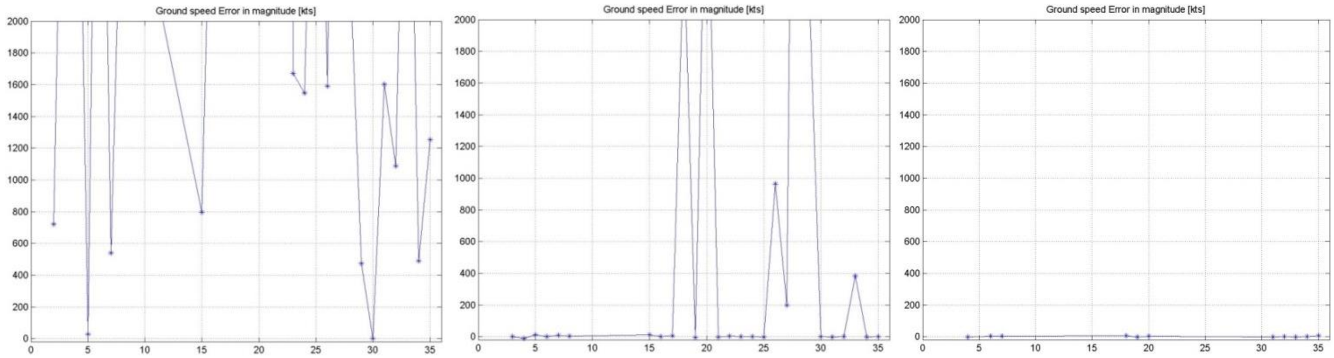
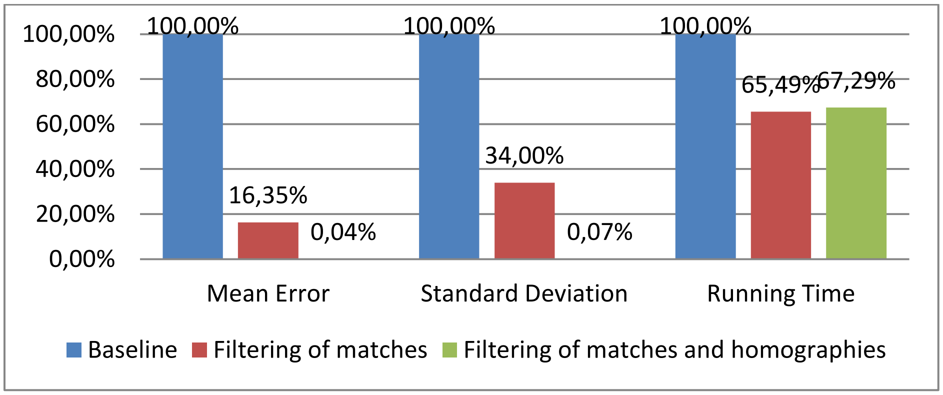


Figure 16. Ground speed error obtained with the baseline (left), the filtering of matches (middle), and filtering of matches and homographies (right) for the TAXI scenario.

Although the overall error is improved, it should be noticed that both filters increase the region where the estimation cannot be performed. For example, from the 35 attempts to perform an estimation in the TAXI scenario (1 estimation was attempted per second), 8 estimation were lost using the filtering of matches, and from 15 to 20 estimations using the filtering of matches and homographies. This corresponded to regions of the taxiway with no ground marks, not enough features can be detected.

These results show that the filtering of matches and homographies is essential to obtain an acceptable estimator. In order to complete the evaluation, the running time required to perform the estimation was also measured for each variant of the estimators over the TAXI scenario.



Even though the filtering of matches adds the need of comparing all the features between frames, it reduces the number of matches that are used to compute the homography. This resulted in a significant improvement in the running time required to perform estimation.

As for the filtering of homographies, it only adds comparison of values already computed of the homography matrix with some predefined thresholds. This had almost not impact in the running time.

Clearly the filtering of matches and homographies resulted in a gain-gain situation. Not only it achieves a decent error in mean and deviation, but also allows reducing the running time of the overall estimation.

Estimating the homography using RANSAC method

As described before, the RANSAC method allows efficiently removing incorrect matches at the homography estimation stage. This was compared to the DLT method that computes the homography considering all the matches.

The following graphs show the results obtained by estimating the homography using DLT and RANSAC for the TAXI and APPROACH scenarios. Since reasonable error magnitudes are now obtained, the scale of the graph was reduced to show the performance of the estimators in detail.

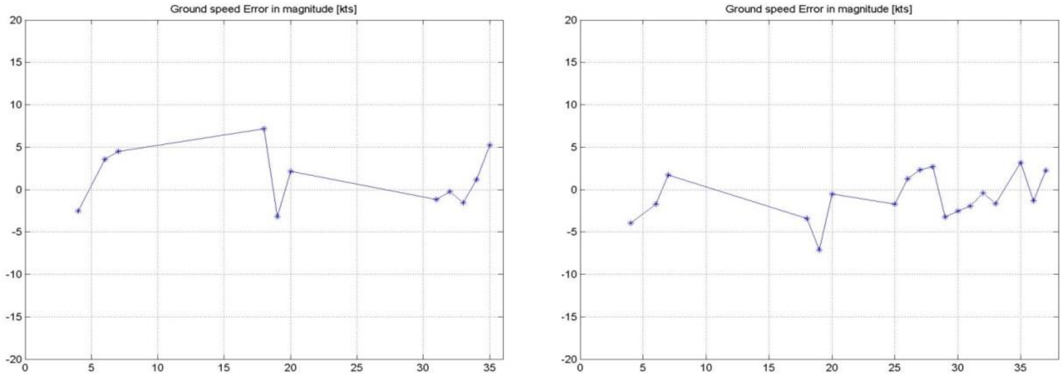


Figure 17. Ground speed error for the TAXI scenario using DLT-based (left) and RANSAC-based homography (right).

At a first glance, the difference between DLT and RANSAC method to estimate the homography cannot be fully appreciated. Since both methods are based on the same principle, the gain of using the RANSAC method is mostly appreciated whenever incorrect matches reach the homography estimation stage.

A new simulation was made under the APPROACH scenario to compare the results of both estimators.

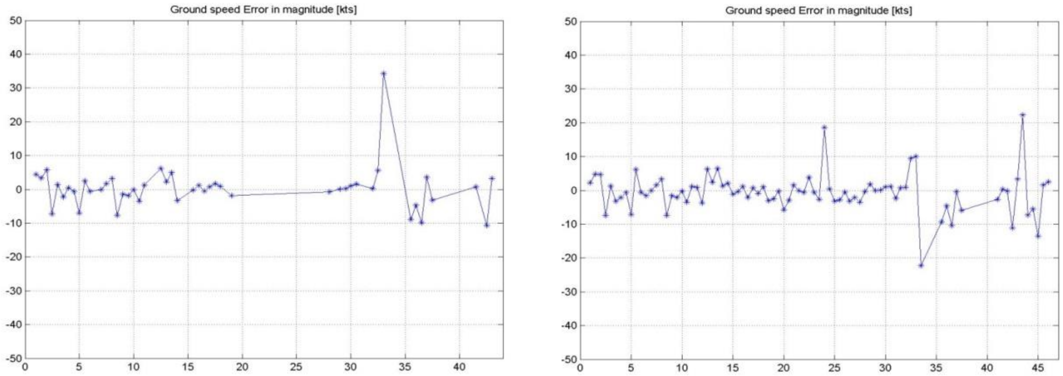


Figure 18. Ground speed error for the APPROACH scenario using DLT-based and RANSAC-based homography.

The fact that the RANSAC-based approach outputs more estimations than the DLT-based approach (18 against 11 estimations for the TAXI scenario; 82 against 47 for the APPROACH scenario) is related to the filters applied to the computed homographies. Since RANSAC avoids the use of incorrect matches the estimated homography results better conditioned than the DLT approach and hence are more likely to pass the filter of degraded homographies.

Although a great improvement cannot be appreciated in terms of the estimation error, the RANSAC method allows estimating less degraded homographies and outputs more values per unit of time.

Improve stability by using a Washout filter

A washout filter was used in order to smooth the output of the ground speed estimation. The main parameter of the filter is its time-constant, which controls the response of the filter to changes in the input signal. The results below show the output of the estimator for different values of T over the APPROACH scenario.

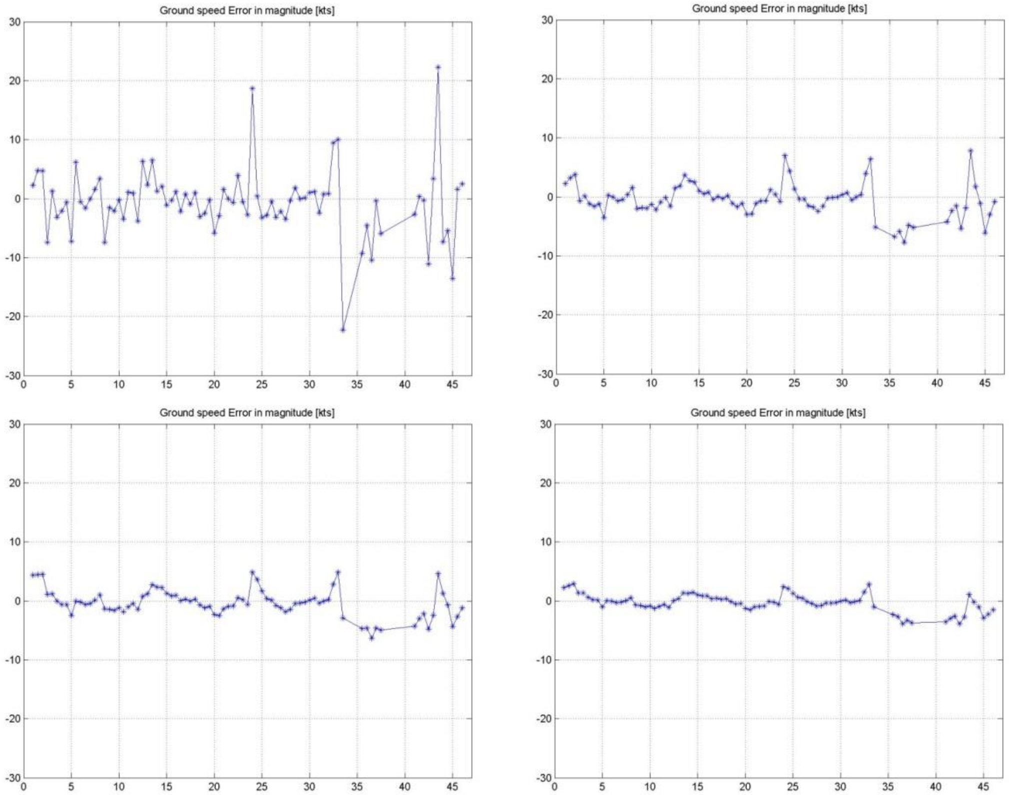


Figure 19. Ground speed error using a Washout Filter for the APPROACH scenario.
Different values of T were used: no filter(top-left); T=1 (top-right); T=1.5 (bottom-left); T=3 (bottom-right).

As expected, the washout filter allowed to smooth the input signal and to reduce the overall error under a constant speed scenario. Although the error is reduced as the time constant increases, under acceleration, the estimated ground speed will respond more slowly to changes measured in the images.

4. Summary

The following table shows the improvement of the ground speed error by applying the improvements to the discrete feature-based estimator for the TAXI scenario.

	Max.	ME	MAE	RMSE	SD	Est. ratio
DLT Homography (baseline)	24436	3927	3927	6204	4918	100 %
DLT Homography **+ filtering of matches and homographies**	7.16	1.38	2.96	3.55	3.43	32 %
RANSAC Homography **+ filtering of matches and homographies**	7.10	-0.90	2.39	2.80	2.74	52 %
RANSAC Homography **+ filtering of matches and homographies** **+ Washout filter (T=1.0)**	5.45	-1.29	2.05	2.43	2.14	43 %

The original DLT-based estimator without any of the proposed improvements was included here only as a reference since the obtained results are evidently unacceptable.

The improvements allowed to reduced considerably the magnitude of the error. For the TAXI scenario both mean and deviation of the error represent about a 20% of the real ground speed of the aircraft.

As explained before, DLT and RANSAC methods, outputted similar results for the TAXI scenario. However, the RANSAC method resulted in a greater estimation ratio, giving more estimations than the DLT method for the same scenario. The RANSAC estimation tend to output better conditioned homographies and thus they tend to pass more often the filtering of homographies.

Due to the non-deterministic nature of the RANSAC method, the simulation does not output always the same results for a specific scenario[3]. This was noticed mainly on the peaks of the errors and in the estimation ratio, but it did not impact significantly in the mean and deviations of the error measures.

Regarding the systematic error, all variants showed a small positive or negative bias. The number of estimations however, is not enough to reach a strong conclusion about the bias of the estimator. Longer scenarios should be defined in order to precisely evaluate this aspect. As for the available results, the measured bias represents about a 1 % of the real ground speed of the aircraft.

Finally, as described in the previous sections, all the improvements allowed to reduce not only the error measures but also the running time of the estimation. The filtering of matches reduces the time needed to estimate the homography and the RANSAC method avoids the pre-calculation and the need of the SVD decomposition that DLT requires.

Therefore, the selected estimator was the variant that performs filtering, computes the homography using RANSAC method and applies a washout filter on the output of the estimation.

[3] In fact, OpenCV's ORB feature detector and matcher has also a non-deterministic nature, giving sometimes a different set of matches for the same image pair.

VI. Results

1. Ego-motion estimator

The performance of the ego-motion estimator was measured using the discrete approach as it yielded reasonable results for the ground speed estimation.

The continuous approach, based on the optical flow, was implemented in the same environment and adapted to the same project. However, the development could not be completed and the preliminary results were not reliable. Additional work is required in order to detect the causes of the errors.

The results included in this section cover only the discrete, feature-based, estimation based on the homography decomposition.

2. Performance measures

The estimated ground speed and velocity vector were compared with the ground truth values obtained from Visuel flight simulator for all the defined scenarios. The vertical speed was also estimated for the APPROACH scenario and compared with the vertical speed of the aircraft.

The results were recorded in a CSV file and analyzed with a Matlab script in order to compute different error measures:

- The maximum error committed during the simulation (Max.) $max(error)$.
- The mean (average) of the errors (ME) $mean(error)$.
- The mean of the absolute errors (MAE) $mean(abs(error))$.
- The Root Mean Square Error (RMSE) $sqrt(mean(error^2))$.
- The Standard Deviation of the errors (SD) $std(error)$.

For the velocity direction error a single angle between the real direction and the estimated direction was considered. Thus the error lie between 0 and 180 degrees and the MAE equals the ME.

In addition to these error measures, the *estimation ratio* between succeeded and attempted estimations was computed for each simulation. A low ratio indicates that the estimation could not be performed during most of the simulation. As described before this happens either when the estimation does not seem reliable, or when not enough features could be detected.

3. Results for the defined scenarios

In this section the results for the ground speed estimation based on the discrete egomotion estimation are presented for each of the defined scenarios: TAXI, TAXI-R, TAKEOFF, APPROACH.

Scenario: TAXI

The table below shows the results obtained for the estimated ground speed and velocity vector for the TAXI scenario.

	Max.	ME	MAE	RMSE	SD
Ground speed [knots]	5.45	-1.24	2.01	2.36	2.07
Velocity vector [degrees]	10.40	2.82	2.82	3.56	2.24

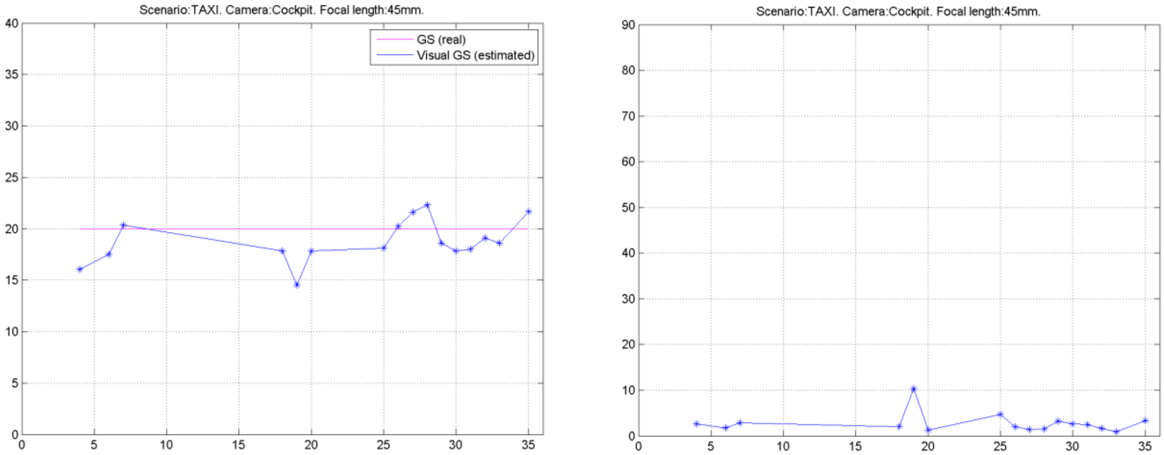


Figure 20. Ground speed error (left) and velocity direction error (right) for the TAXI scenario.

The ground speed of the aircraft was 20 knots and remained constant during the simulation. Hence, the obtained mean absolute error represents about a 10% of the aircraft ground speed and the RMSE about a 12%. Results show also a small negative bias for the ground speed estimation under this scenario although there are not so many estimations to reach a conclusion about the bias of the estimator.

The error in the direction of the estimated velocity vector, remained mostly below 5 degrees. A peak of 10.4 degrees can be observed at 19 seconds. This affected the ground speed estimation, resulting in a peak of 5.45 knots, about a 27% of the ground speed of the aircraft. As it can be seen from the image, this is probably due to the small number of features and to their relatively close locations.

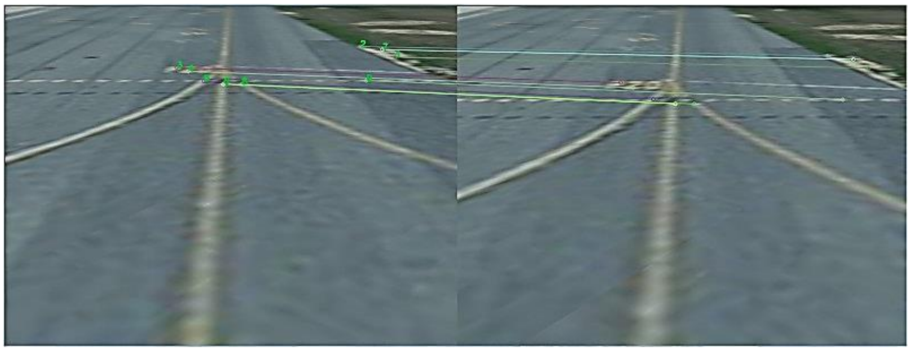


Figure 21. Matches of features that were used to perform the estimation at the 19th second of the TAXI scenario.

Under this scenario 35 estimations were attempted and 17 estimations were obtained (about a 48% of attempted estimations). This low ratio is also due to the lack of details in most parts of the taxiway.

Scenario: TAXI-R

The table below shows the results obtained for the estimated ground speed and velocity vector for the TAXI-R scenario.

	Max.	ME	MAE	RMSE	SD
Ground speed [knots]	5.11	-0.56	1.70	2.19	2.15
Velocity vector [degrees]	22.16	8.64	8.64	10.82	6.62

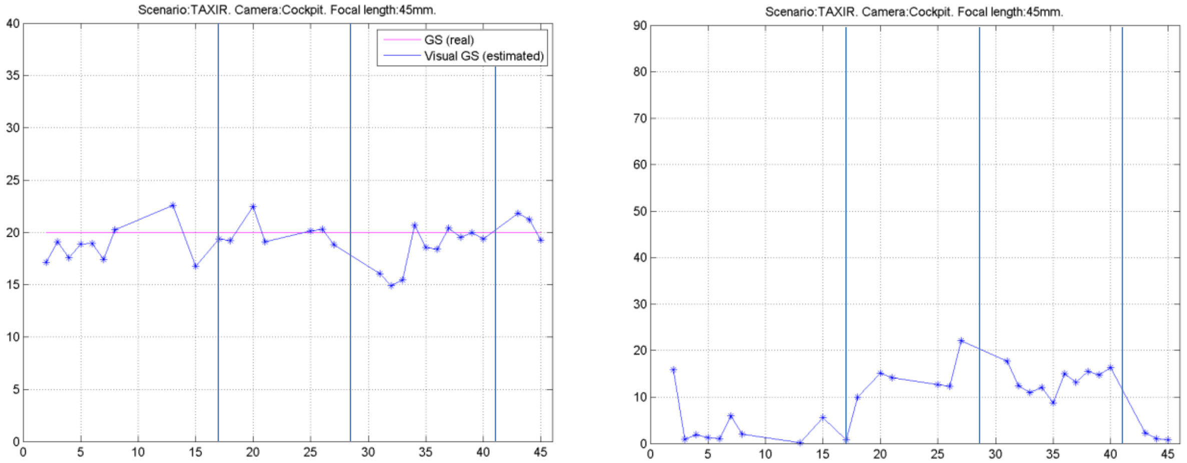


Figure 22. Ground speed error (left) and velocity direction error (right) for the TAXI-R scenario. Regions marked in the graphs correspond respectively to forward motion, left turning, right turning and forward motion.

Similarly to the TAXI scenario, the ground speed of the aircraft was 20 knots and remained constant during the simulation. However, in this scenario, the aircraft passes through a different taxiway and the trajectory includes two turns of about 90 degrees at 17 seconds and at 28.5 seconds. At 41 seconds the aircraft returns to the forward motion.

Results for the ground speed estimation are quite similar to the TAXI scenario. The MAE is about a 8.5% of the aircraft ground speed and the RMSE about a 11%. The results show a smaller bias in comparison to the TAXI scenario.

Rotation seems not to have an impact in the ground speed estimation, but it affects the estimation of the velocity vector. Regions where the aircraft is turning show an average error in the direction of about 14 degrees, while regions where the aircraft performs forward motion show an average error of 3 degrees. As in the previous case, a peak is sometimes observed under forward motion. In this case was observed at the start of the simulation with a magnitude of 15.9 degrees.

Under this scenario, 45 estimations were attempted and 32 estimations were obtained (about a 71% of attempted estimations). The improvement over the TAXI scenario is due to the greater number of taxiway details in the TAXI-R scenario which in turn results in a greater number of features detected.

Scenario: TAKEOFF

The table below shows the results obtained for the estimated ground speed and velocity vector for the TAKEOFF scenario.

	Max.	ME	MAE	RMSE	SD
Ground speed [knots]	29.56	-7.34	7.34	8.87	5.03
Velocity vector [degrees]	74.47	4.10	4.10	10.79	10.06

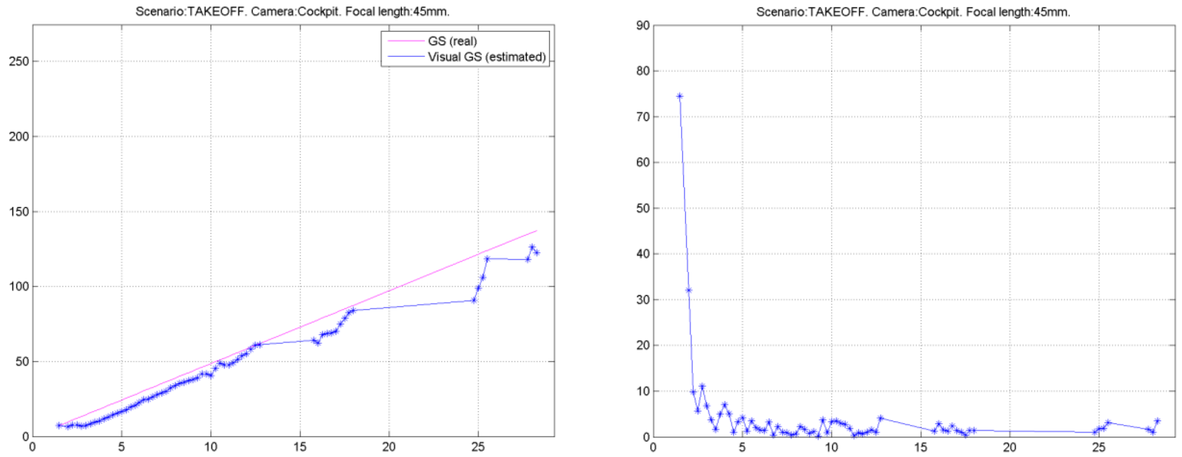


Figure 23. Ground speed error (left) and velocity direction error (right) for the TAKEOFF scenario.

In this scenario, the ground speed increases from 0 to 130 knots at a constant rate. The aircraft performs forward motion over the runway.

The ground speed estimation shows a delay compared to the aircraft ground speed. The estimation error caused by this delay remained mostly constant during the simulation with values between 6 and 8 knots. This effect was expected for two main reasons: the use of a washout filter and the estimation process itself.

The estimated speed passes through a washout filter in order to obtain a smooth output, which adds a delay to the estimation. This is inherent to the filter and depends on the time constant used in the filter.

Regarding the estimation process, essentially the displacement of the aircraft between to instants of time is being measured, which implies a delay inherent to the estimation process. The ground speed estimated at time t_k will be always related to the ground speed of the aircraft between t_{k-1} and t_k.

These reasons explain partially the error magnitudes obtained for this scenario. An approach to compensate the ground speed with an estimation of the acceleration of the aircraft was proposed by the tutors of Airbus[4].

During the main part of the acceleration (the period from 0s to 8s) the ground speed error remained below 30 knots, with a mean of 21.5 knots below the real ground speed. In contrast, over the region of constant speed, the error remained below 5.5 knots, with a mean of 2.1 knots (4.2% and 1.6% of the real ground speed).

[4] Details can be found in the Future Work section.

Regarding the estimation of the velocity vector, the results show a significant peak at the start of the simulation. This was probably due to the small displacement at the start of the simulation since the aircraft start from 0 knots and a fixed frame step was used[5]. For the rest of the simulation, the error in the velocity vector remained below 5 degrees.

Under this scenario, 80 estimations were attempted and 28 estimations were obtained (about a 35% of attempted estimations). The graph shows that estimation is performed in some regions, while gaps without estimations are observed for the periods 3s-4s, 5s-7s, 8s-13s, 14s-16s and 17s-19s. This corresponds to regions of the runway where only the centerline can be detected. This does not provide sufficient features to perform estimation. However, when fixed distance markers are detected, estimations are outputted again.

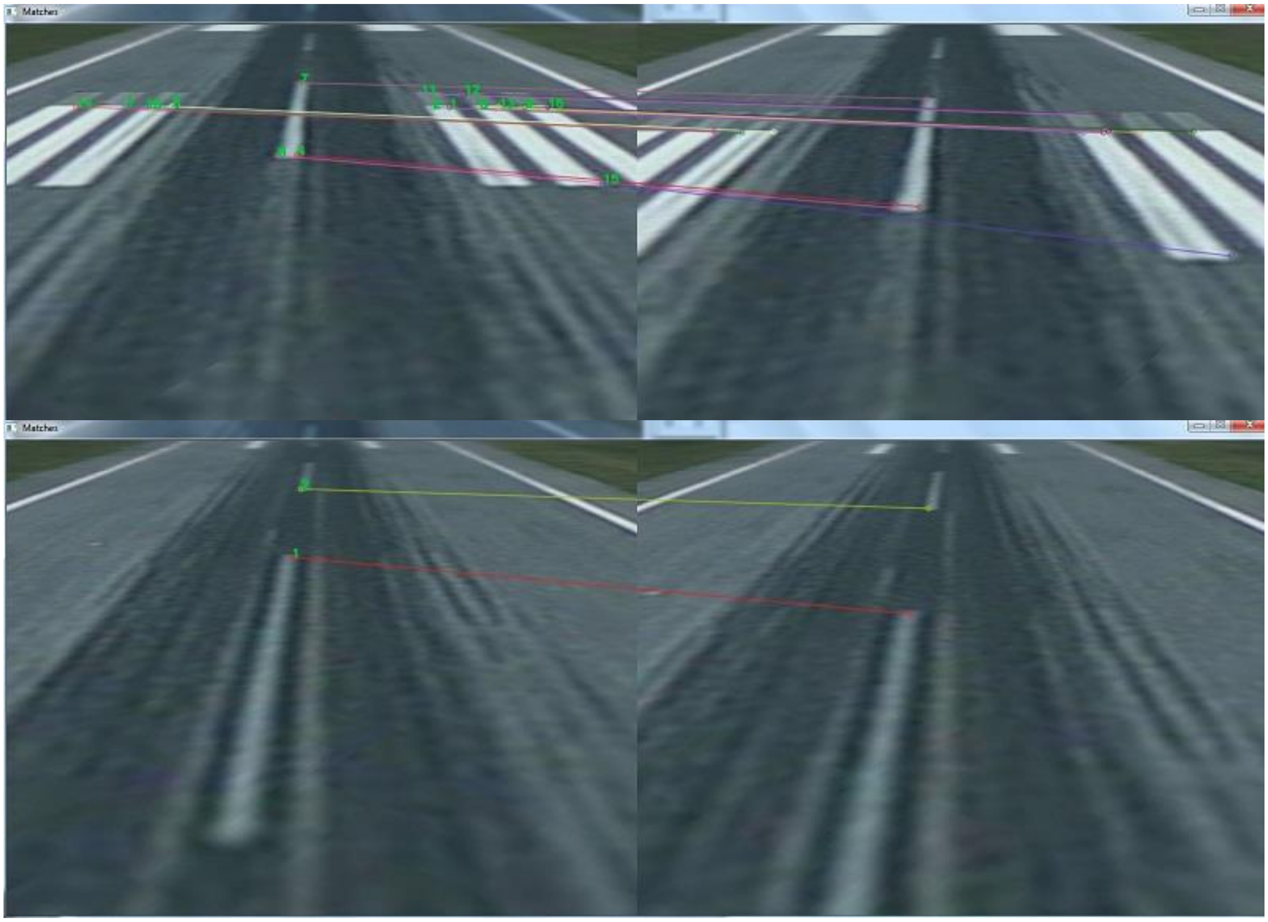


Figure 24. Detected features for different instants of the TAKEOFF scenario: 16 features were detected over a region with fixed-distance marks (above); only 2 features over a region with only the runway centerline.

[5] Adjusting the frame step dynamically was considered during this work. Details can be found in the appendix.

Scenario: APPROACH

The table below shows the results obtained for the estimated ground speed and velocity vector for the APPROACH scenario. In this case, results for the vertical speed are also included.

	Max.	ME	MAE	RMSE	SD
Ground speed [knots]	7.78	-0.59	2.08	2.84	2.79
Velocity vector [degrees]	7.26	1.13	1.13	1.71	1.29

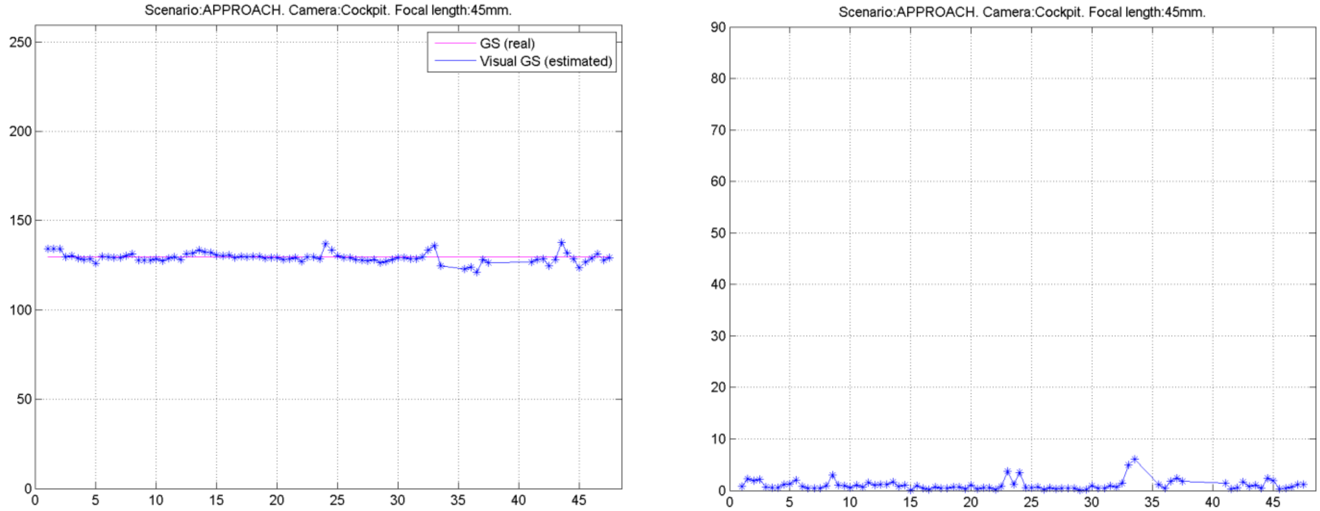


Figure 25. Ground speed error (left) and velocity direction error (right) for the APPROACH scenario.

The ground speed of the aircraft was 129.8218 knots and remained constant during the simulation. The aircraft performed a simplified approach maneuver, moving in a straight line to the runway with a fixed pitch of -3 degrees.

Results for the ground speed estimation were much better than expected. The MAE and the RMSE represents respectively about a 1.6% and a 2.2% of the aircraft ground speed. It should be recalled, however, that the simulation conditions do not represent reality accurately. In fact, both the aircraft maneuver and the scene simulated in Visuel are oversimplified, with a perfectly flat environment with no buildings nor moving objects.

The error in the velocity vector remained below 3 degrees during most of the simulation, a total of 94% of the estimations, while the remaining 6% were peaks between 3 and 7 degrees of error.

Under this scenario, 100 estimations were attempted and 85 estimations were obtained during most of the simulation. The regions where gaps are observed, near the end of the simulation, correspond to the clean area of the airport before the runway where not enough features could be detected.

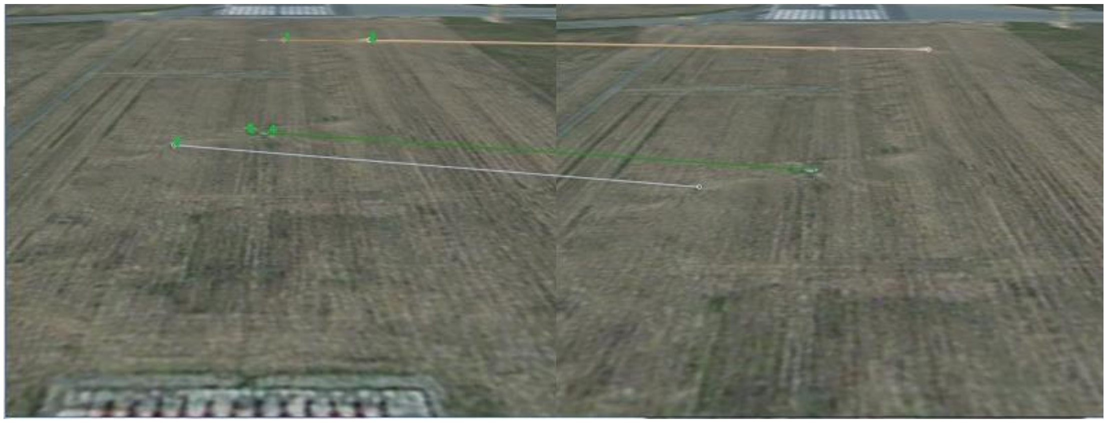


Figure 26. Example of features detected over the clean area before the runway.

4. Summary

This section summarizes the results obtained for the ground speed and the velocity vector over all the defined scenarios.

The following table shows the results for ground speed estimation. Values are expressed in knots.

GS module	Max.	ME	MAE	RMSE	SD
TAXI	5.45	-1.24	2.01	2.36	2.07
TAXI-R	5.11	-0.56	1.70	2.19	2.15
TAKEOFF	29.56	-7.34	7.34	8.87	5.03
APPROACH	7.78	-0.59	2.08	2.84	2.79

The following table shows the results for the velocity vector estimation. Values are expressed in degrees.

GS direction	Max.	ME	MAE	RMSE	SD
TAXI	10.40	2.82	2.82	3.56	2.24
TAXI-R	22.16	8.64	8.64	10.82	6.62
TAKEOFF	74.47	4.10	4.10	10.79	10.06
APPROACH	7.26	1.13	1.13	1.71	1.29

VII. Sensitivity evaluation

The performance of the estimation was evaluated while varying key parameters of the simulation. Only the estimation of the ground speed in module was evaluated.

It should be noticed that the TAKEOFF scenario considered in this section included a non-linear acceleration that do not represent an actual takeoff scenario. This was corrected later only for the results showed in the previous section.

1. Parameters

Focal length: Focal length is simulated in Visuel since is a physical measure of a lens property. It can be modified by adjusting the region of the space that is projected and showed in the screen. The following focal lengths configurations were used:

- 18 mm (wide-angle lens).
- 30mm (wide-angle lens).
- 45 mm (normal-angle lens), default in Visuel.
- 100 mm (telephoto lens).

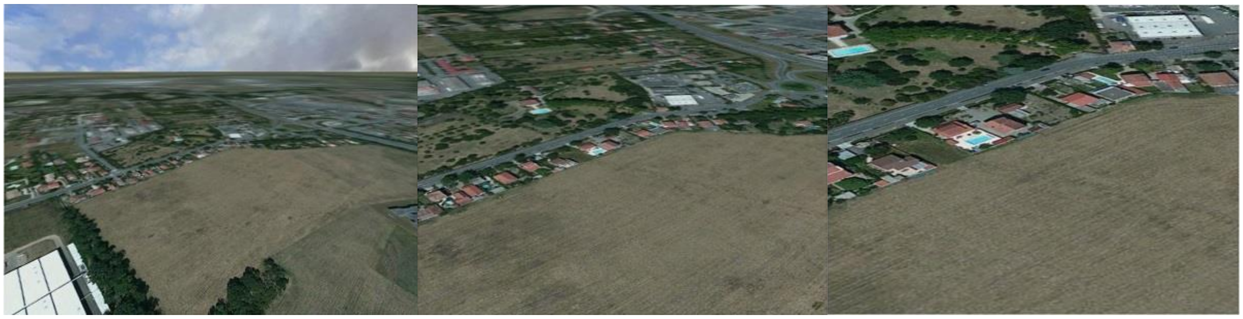


Figure 27. Visuel frames for different simulated focal lengths: 18mm (left), 45mm (middle) and 100mm (right).

Camera position: Within Visuel, the camera can be placed in principle anywhere in the scene. Two main positions were considered for the evaluation:

- COCKPIT: the camera is placed in the cockpit of the aircraft, looking forward and rotated 20 degrees downwards. A clear view of the scene was considered and no cockpit objects nor window details were simulated. The camera was placed 14m above the ground.
- VERTICAL STABILIZER: the camera is mounted on the edge of the vertical stabilizer, looking forward and rotated 20 degrees downwards. The camera was placed 17.1 m above the ground [6].

Cameras from the External and Taxi Aid Camera Systems (ETACS) were considered to define the possible camera positions. ETACS includes a camera installed under the fuselage, oriented towards the front of the aircraft; and a camera mounted on the leading edge of the vertical stabilizer, oriented toward the front of the aircraft.

[6] Visuel's "Pilot Fish" mode was used. It places a camera 50m behind the center of the aircraft with 20 degree of elevation looking at the center of the aircraft.

Since no aircraft model was available, for the camera mounted on the vertical stabilizer, the body of the aircraft was extracted from a real image and used to generate a mask. This mask allowed to cover the region of the image that is normally covered by the body of the aircraft and to simulate approximately the image of the real camera.

However, this only worked accurately under simulations with focal length similar to the real camera. When simulating a 18mm focal length, the camera could not be placed on the vertical stabilizer since the resulting image resulted distorted. In addition, using a 100mm focal length was not useful in this case since due to the zoom effect, the resulting image will contain only the fuselage of the aircraft.

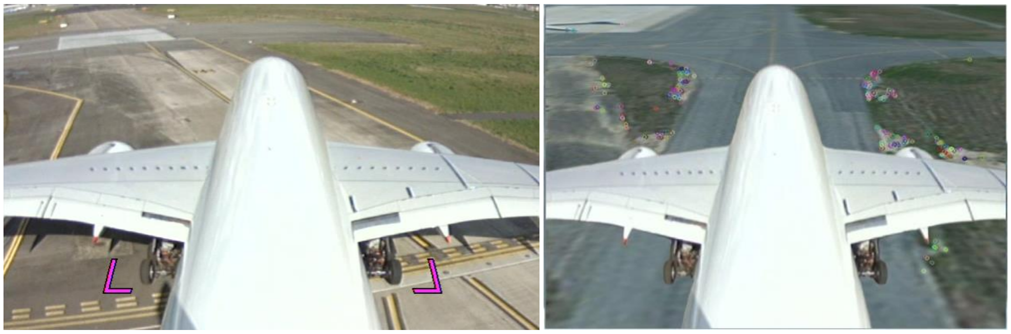
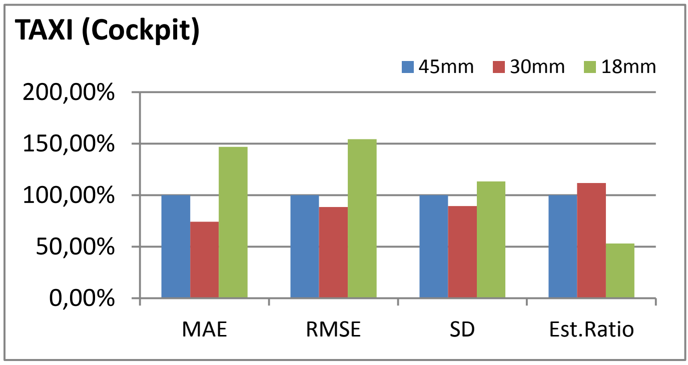


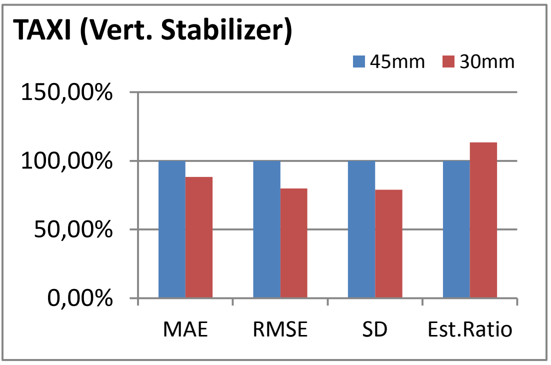
Figure 28. Image captured by a real camera mounted on the vertical stabilizer of an aircraft (left) and the image generated in Visuel superposing the aircraft body (right).

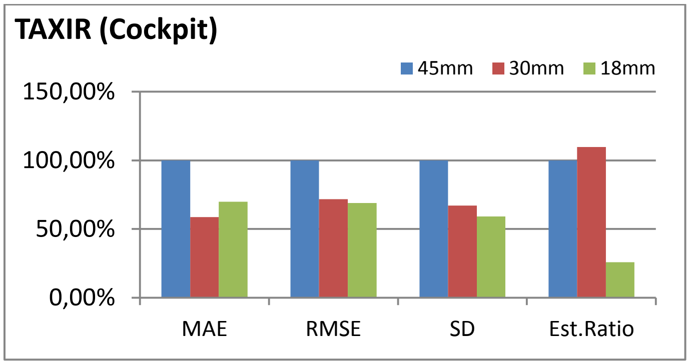
Cameras mounted on the landing gear, both oriented forward and oriented downwards, were also considered, but the results were not satisfactory for ground scenarios. For the camera oriented downwards the lack of detail does not allows to detect feature in the images, while for the camera oriented forward the region of the images that can be used to detect features is very restricted. Notice that unlike the real image, the scene simulated in Visuel does not have optic distortion in the borders of the image.

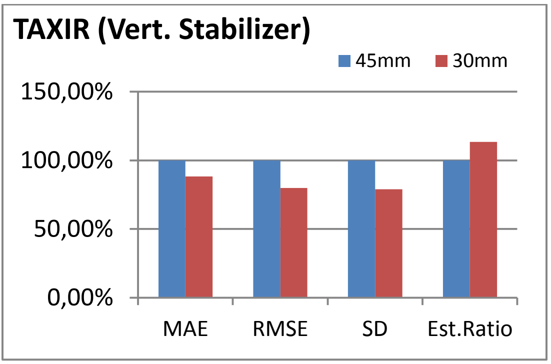
2. Varying focal length

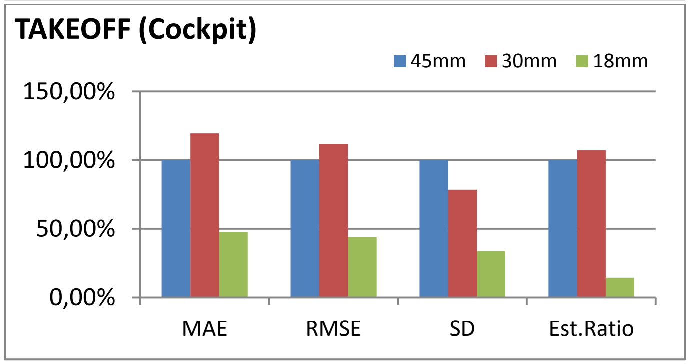
The following graphs show the relative improvements in the performance measures by scenario, considering the initial simulated 45mm focal length as the baseline.

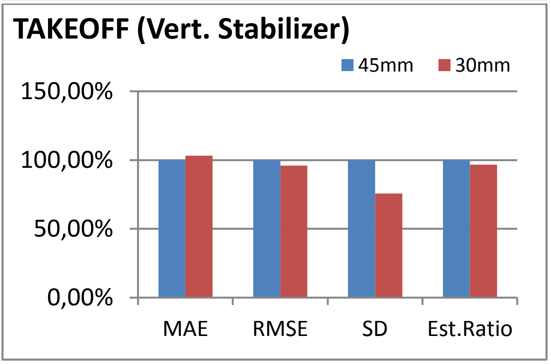


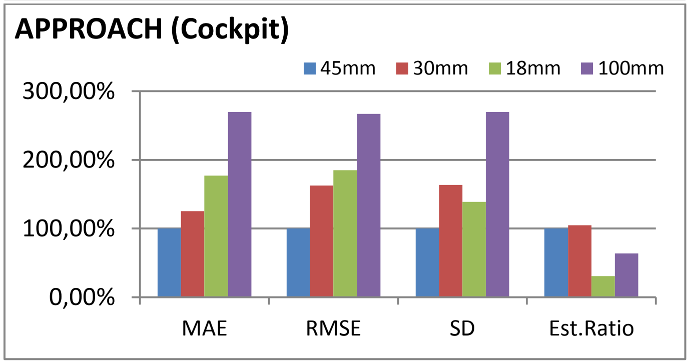


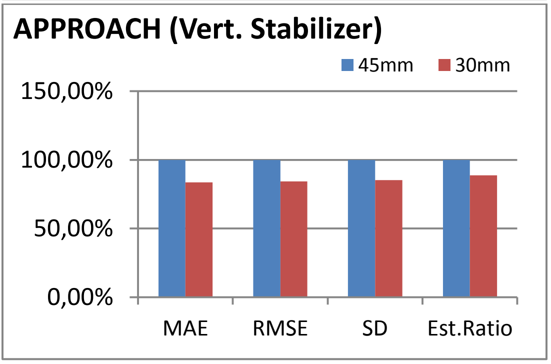






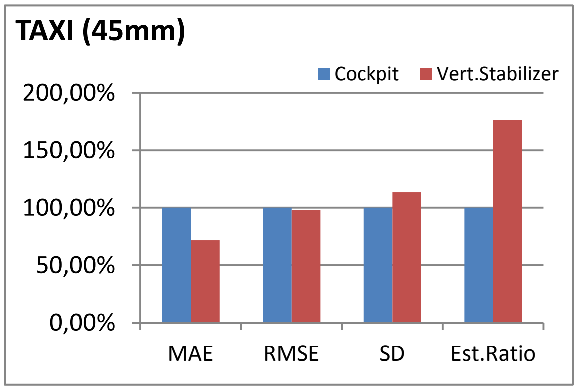


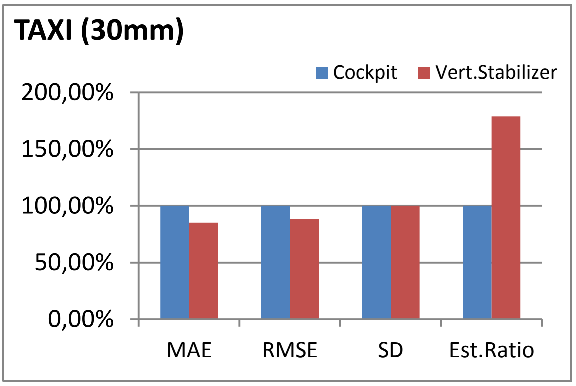


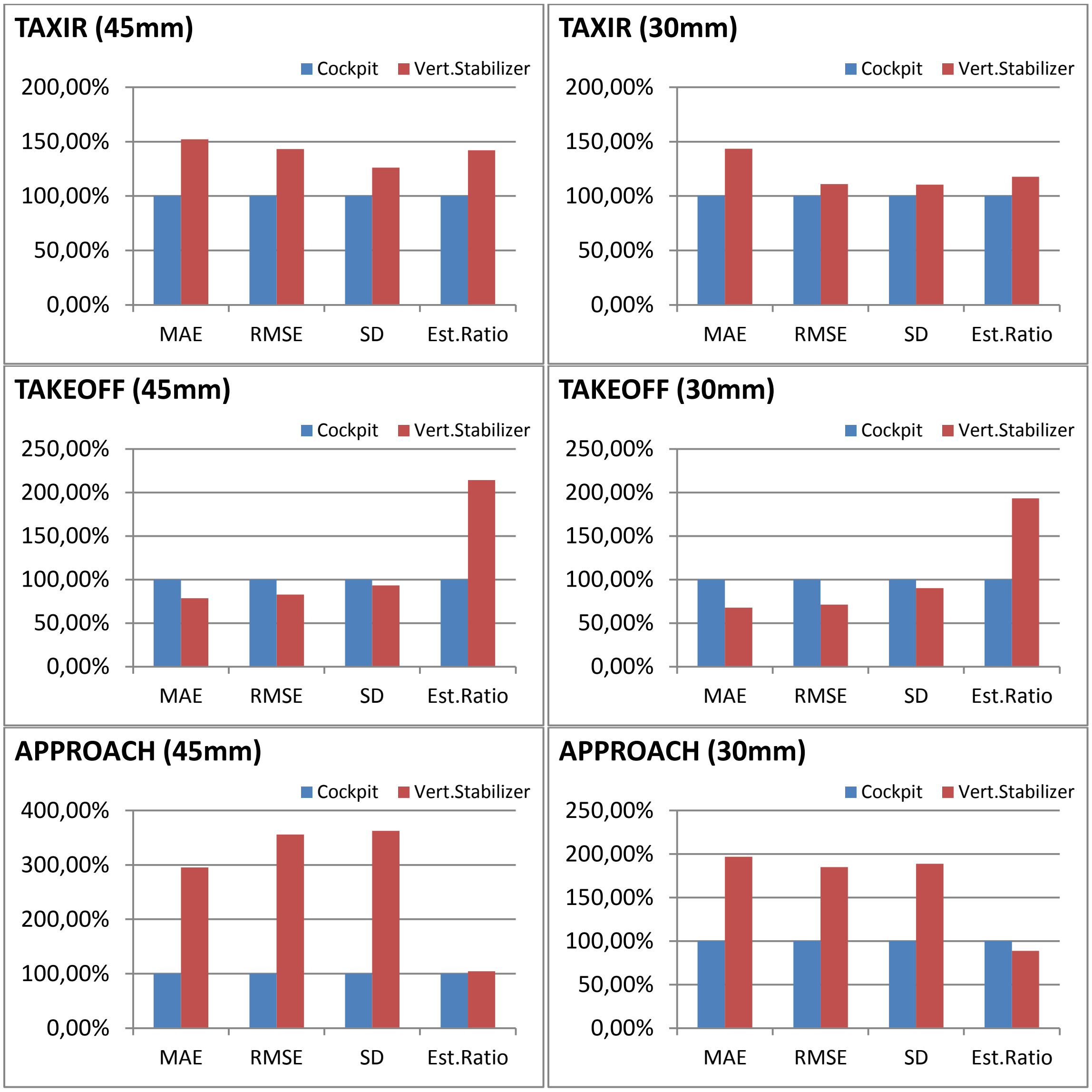


3. Varying camera position

The following graphs show the relative improvements in the performance measures by scenario, considering the camera fixed in the Cockpit as the baseline.







4. Summary

As it can be seen in the graphs, the focal length had a considerable impact on the overall performance of the estimator, both in the error measures and in the estimation ratio.

18 mm focal length: Considering the initial simulated 45mm focal length as the baseline, the simulation with an 18mm focal length resulted in a decrease of the estimation ratio, giving less than 50% of the estimation with the baseline. The worst performance was obtained over the TAKEOFF scenario, were only 4 estimations could be performed during the 20s of the simulation. This is related to the fewer number of features detected.

This also explain the variability of the error indicators for the 18mm focal length: over the TAXI and APPROACH scenarios, the error in mean and deviation increased notably, while over the TAXI-R and TAKEOFF scenario the error measures decreased. Since the estimator outputted very few values during simulations, a small number of correct or incorrect estimations have a non-negligible impact on the performance indicators. However, since the TAXI and APPROACH scenarios had the greater estimation ratios, it is reasonable to think that the 18mm configuration tends to increase the overall error in mean and deviation of the estimator.

30mm focal length: The results obtained with a simulated 30mm focal length seemed to improve the overall estimator performance.

For the camera fixed on the vertical stabilizer, the error measures showed a decrease of between 5% and 20% both in mean and deviation. For the camera fixed in the cockpit, this only happened over the TAXI and TAXI-R scenarios, while the TAKEOFF and APPROACH scenarios showed an increase in the overall error of the estimator[7]. More simulations should be performed in order to identify the tendency of the error measures. Different and longer scenarios should help for this purpose.

As for the estimation ratio, it resulted slightly increased in almost all simulations. An improvement between 5% and 10% was obtained in general. For the camera fixed on the vertical stabilizer, the estimation ratio shows in contrast a slightly decrease for the APPROACH and TAKEOFF. This value, however, does not represent a significant change in the performance of the estimator.

100mm focal length: The simulation performed with the 100mm focal length did not yield satisfactory results compared to the baseline. Both the error measures and the estimation ratio resulted degraded for the APPROACH scenario.

Vertical stabilizer: Considering the camera in the cockpit as the baseline, the relative improvement of the camera mounted on the vertical stabilizer shows a significant impact on the estimation ratio. For a fixed scenario, the estimation ratio resulted increased generally between 20% and 100% over the ground scenarios. For the APPROACH scenario the estimation ratio did not changed significant regarding the simulations with the camera placed in the cockpit.

These results are reasonable considering that the height difference of the camera location, about 17m within the simulations, is more significant when the aircraft is on the ground.

This however did not have a direct relation with the error measures. The TAXI-R and APPROACH scenarios showed a significant increase in the error mean and deviation, while the TAXI and the TAKEOFF scenario showed a slightly decrease. However, it should be noticed from the tables of the previous section, that the error mean and standard deviation remain within reasonable absolute values: between 5% and 10% of the real ground speed for the APPROACH scenario and about 10% for the taxi scenarios.

[7] There is an implicit randomness in the estimator results due to Feature Detector and the RANSAC method.

Conclusion

This work represented an exploratory path over the idea of a vision-based sensor to improve the ground speed estimation. The theory and techniques involved in this process were studied and presented in this work and the complete estimation process was developed and tested over realistic scenarios. I consider the objectives of the internship were met.

The discrete-time ego-motion estimation, although based on a simple feature-based homography decomposition, yielded satisfactory results. This suggests that the idea that motivated this study is feasible. Results showed error magnitudes between 5% and 10% of the aircraft ground speed for the ground scenarios which may help to improve current ground speed measurements at low speeds.

It should be noticed, however, that the simulations were made over ideal scenarios and many of the hypothesis would not be valid under real cases. The performance of the estimator could not be evaluated over real videos due to the considerable effort put in the improvement of the selected method and to the decision of evaluate the two approaches.

From a personal perspective, this internship represented my first experience in a research project and allowed me to work with an experienced team of engineers and researchers within a company that is the global leader in the aviation industry. I have developed a good knowledge in the field of computer vision, particularly with respect to motion estimation.

Future Work

I have categorized the tasks which I consider should be a reasonable continuation of this work towards a real application.

- Next steps: tasks which remained pending and I consider should be a continuation of this work. Much of the work has already been done in this internship and their results.
    - Review and test the planar-ego method to correct issues.
    - Measure the performance of the ground speed estimation under real videos.
- Further study: topics that emerged during this internship which may seem subtle but are certainly required to be analyzed.
    - Use a variable frame step to improve estimation performance.
    - Compensate estimated speed under scenarios with acceleration.
- Further work: tasks that may require more effort and are not related to research but to implementation on a real case.
    - Improve algorithm design and measure running time of the estimation stages.
    - Remove OpenCV dependency and migrate to pure C++.

Bibliography

Adiv, G. A. (1985). *Determining three-dimensional motion and structure from optical flow generated by several moving objects.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 7, no.4, pp.384-401.

Adiv, G. A. (1989). *Inherent ambiguities in recovering 3D Motion and structure from a noisy flow field.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, pp.477-489.

Bruss, A. R., & Horn, B. K. (1983). *Passive navigation.* Computer Graphics and Image Processing, 21:3–20.

Burger, W., & Bhanu, B. (1990). *Estimating 3D Egomotion from Perspective Image Sequence.* Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.12, no.11, pp.1040,1058.

Faugeras, O., & Lustman, F. (1988). *Motion and structure from motion in a piecewise planar environment.* INRIA-00075698, version 1, N° RR-0856.

Hanna, K. J. (1991). *Direct multi-resolution estimation of ego-motion and structure from motion.*

Hartley, R., & Zisserman, A. (2004). *Multiple View Geometry in Computer Vision.* Cambridge University Press, Second Edition, March 2004.

Heeger, D. J., & Jepson, A. D. (1992). *Subspace methods for recovering rigid motion.* International Journal of Computer Vision,7(2):95-117.

Horn, B. K. (1990). *Recovering Baseline and Orientation from Essential Matrix.*

Huang, T. S., & Fang, J. Q. (1984). *Solving three-dimensional small-rotation motion equations.* Computer Vision, Graphics, and Image Processing, Volume 26, Issue 2, Pages 183-206.

Irani, M., Rousso, B., & Peleg, S. (1997). *Recovery of Ego-Motion Using Region Alignment.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 3.

Jepson, A. D., & Heeger, D. J. (1992). *Linear Subspace Methods For Recovering Translational Direction.* Spatial Vision in Humans and Robots, pages 39-62. Cambridge University Press.

Kanatani, K. (1993). *3D Interpretation of Optical Flow by Renormalization.* International Journal of Computer Vision, 11(3):267–282.

Longuet-Higgins, H. (1981). *A computer algorithm for reconstructing a scene from two projections.*

Longuet-Higgins, H. C. (1986). *Visual Motion Ambiguity.* Vision Res. Vol. 26, No. 1, pp. 181-183.

Ma, Y., Soatto, S., Kosecka, J., & Sastry, S. (2003). *An Invitation to 3D vision.*

Malis, E., & Vargas, M. (2007). *Deeper understanding of the homography decomposition for vision-based control.*

Negahdaripour, S., & Horn, B. K. (1985). *Direct Passive Navigation- Analytical Solution for Planes.* In Proceedings of IEEE Conference on Robotics and Automation, p. 1157-1163.

Negahdaripour, S., & Horn, B. K. (1989). *Direct Method for Locating the Focus of Expansion.*

Oreifej, O., Lobo, N., & Shah, M. (2011). *Horizon Constraint for Unambiguous UAV Navigation in Planar Scenes.*

Pollefeys, M., Verbiest, F., & Van Gool, L. J. (2002). *Surviving dominant planes in uncalibrated structure and motion recovery.* ECCV '02 Proceedings of the 7th European Conference on Computer Vision-Part II. p837-851.

Prazdny, K. (1980). *Egomotion and relative depth map from optical flow.* Biological Cybernetics, 36:87-102.

Szeliski, R., & Torr, P. (1998). *Geometrically constrained structure from motion: Points on planes.* European Workshop on 3D Structure from Multiple Images of Large-Scale Environments (SMILE), p171-186.

Tian, T. Y., Tomasi, C., & Heeger, D. J. (1996). *Comparison of Approaches to Egomotion Computation.*

Tomasi, C., & Shi, J. (1993). *Direction of heading from image deformations.* Proceedings of IEEE Computer Vision and Pattern Recognition, pages 422-427.

Appendices

A.1 Mathematical description

The camera model

The *pinhole camera model* is the most commonly used mathematical model to describe the relationship between points in the scene and their projection onto the image plane. It is the mathematical representation of an ideal pinhole camera.

In a pinhole camera, light rays from the scene pass through the pinhole and generate an inverted image on the back wall of the camera. No lens is used to focus the light rays, the pinhole (or aperture) acts as a barrier to block most of the light rays and allows to generate a less blurred image.

The distance between the pinhole and the back wall of the camera is the effective focal length. Perspective projection is obtained, the height of the image objects depends on their distance to the pinhole and parallel lines that lie in a plane will converge to points on the horizon line.

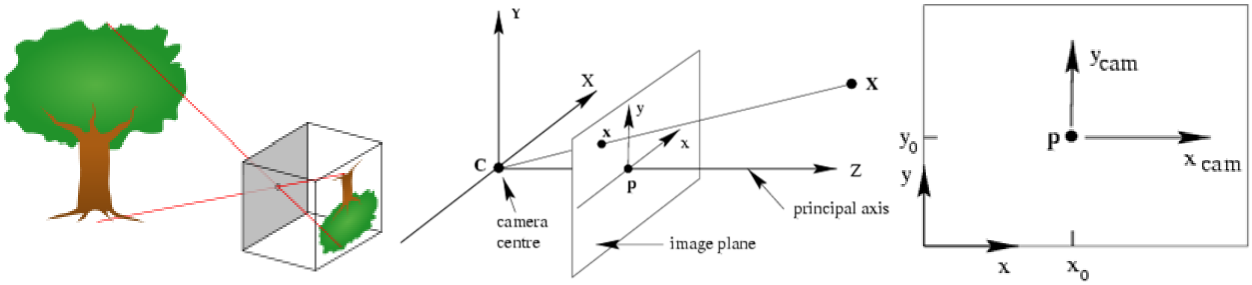


Figure 29. The pinhole camera model. Points in the space are projected onto the image plane.

A camera-centered coordinate system is considered with the z axis aligned with the *principal axis* of the camera (viewing direction). The *image plane* is located at a distance f of the camera center and is perpendicular to the principal axis[8]. Their intersection is called the *principal point*.

A physical point in the space $\boldsymbol{X} = [X, Y, Z]^T$ is projected onto the image plane at position $\boldsymbol{x} = [x, y]^T$ using[9] $w\tilde{\boldsymbol{x}} = C\tilde{\boldsymbol{X}}$.

Where w is a scale factor and C is is called the *camera matrix* containing the perspective projection and transformations that allow to pass from world coordinates to image coordinates.

$$C = K\,[R \mid \boldsymbol{t}]$$

K is called the *calibration matrix* and contains the *intrinsic parameters* of the camera, which depend only on the characteristics of the camera being modeled, and $[R \mid \boldsymbol{t}]$ is the matrix of *extrinsic parameters* that relates the camera coordinate system to an external world coordinate system[10].

[8] Notice that, within the mathematical model, the image plane can be located either in front or behind the camera center without altering equations. The former is usually preferred since it avoids the need of rotating the image.

[9] Homogeneous coordinates are used: $\tilde{\boldsymbol{x}} = [x, y, 1]^T$ and $\tilde{\boldsymbol{X}} = [X, Y, Z, 1]^T$.

[10] A *normalized camera matrix* is centered at the origin and it is represented by $C = K\,[\,I \mid \boldsymbol{0}\,]$.

Camera calibration

Camera calibration is the process of finding the parameters of the mathematical model of the camera. Although this can refers to both the intrinsic and extrinsic parameters, it generally refers only to the intrinsic parameters and the calibration matrix:

$$K = \begin{pmatrix} f\,k_u & s_{uv} & x_0 \\ 0 & f\,k_v & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Where f is the focal length, x_0, y_0 are the principal point coordinates in the image, k_u, k_v are the scale factors relating pixels to distance, and the s_{uv} is the skew coefficient which takes into account the potential non-orthogonality between rows and columns of the image sensor[11].

Depending on the image analysis being performed, the calibration process is not always required. In the literature, it is generally specified if an algorithm works with *calibrated* or *non-calibrated* cameras.

Epipolar geometry

The epipolar geometry[12] is essentially the projective geometry between two different views of the same scene. Given the projection of a scene point into one image, the corresponding projection in the other image is constrained to a line.

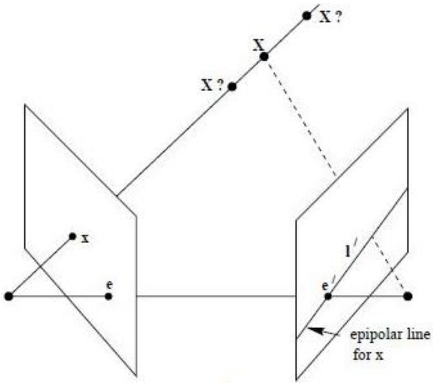


Figure 30. Epipolar geometry, two views of the same scene.

A scene point X is projected onto the images by the camera matrices C and C'. Since only the relative orientation between the cameras is considered, it can be assumed that $C = K\,[I|\mathbf{0}]$ and $C' = K\,[R|t]$, where K is the calibration matrix containing the same intrinsic parameters for both cameras.

The relationship between the projections x and x' is the so called *epipolar constraint*:

$$x' F\, x = 0$$

The matrix F is called the *fundamental matrix* and represents the epipolar geometry of the scene. Since the epipolar constraint is valid for any pair of matching points (x, x'), the fundamental matrix can be computed from a sufficient number of matching points without any information about the camera matrices. However, the relative orientation of the cameras cannot be uniquely retrieved. The

[11] The skew coefficient is specified by the manufacturer and in usually considered negligible for standard cameras.

[12] Proofs can be found on (Hartley & Zisserman, 2004).

fundamental matrix does not depend on the choice of the world frame hence motions that differ by a projective transformation will have the same fundamental matrix.

If the calibration matrix K is known, the image points $x = K\ [R|t]\ X$ can be normalized using its inverse as $\hat{x} = K^{-1}x = [R|t]\ X$. Then the epipolar constraint can be expressed in function of the normalized image coordinates:

$$\hat{x}'E\ \hat{x} = 0$$

Where E is called the *essential matrix* and it represents a specialization of the fundamental matrix for a calibrated scenario.

In the same way as the fundamental matrix, the essential matrix can be computed from enough point correspondences. However, the camera matrices (and hence their relative orientation) can be extracted from the essential matrix.

From an SVD decomposition of $E = UDV^T$, two solutions can be retrieved for t and two solutions for R giving rise to four possible solutions. From these solutions, only one corresponds to a point in front of both cameras, so a *positive depth* constraint is usually applied to discard unfeasible solutions.

Planar case: the homography matrix

When all the points lie on a plane, they share an additional constraint besides the epipolar constraint.

If $N = [n_1, n_2, n_3]^T$ is the normal vector of the plane P and $d > 0$ its distance to the optical center of the camera, then for every point X on the plane $N^T\boldsymbol{X} = n_1X + n_2Y + n_3Z = d \quad \Leftrightarrow \quad \frac{1}{d}N^T\boldsymbol{X} = 1$. Points correspondences are then related by:

$$x_2 = H\ x_1$$

Where H is the *homography matrix* that relates points in the ground from one image, to the corresponding point in the other image.

If the points do not lie all the same plane, the homography matrix is not unique and should be estimated minimizing an error measure. For this, the *re-projection error* is generally used, which attempts to measure the error committed by projecting the points using the homography matrix and its inverse.

$$reprojection\ error = \sum_{i=1}^{n} d\left(x_2^i, Hx_1^i\right)^2 + d\left(x_1^i, H^{-1}x_2^i\right)^2$$

The homography matrix can be decomposed into the motion and scene structure parameters:

$$H = \left(R + \frac{1}{d}t\ N^T\right)$$

where (R, t) are the motion parameters, N is the normal of the plane in the camera reference frame and d is the distance from the camera center to the plane. Although the decomposition is not unique, the correct solution can be recovered using some knowledge of the camera pose.

Motion equations

Consider a coordinate system attached to the camera, and an image plane located at $z = f$ (the focal length of the camera).

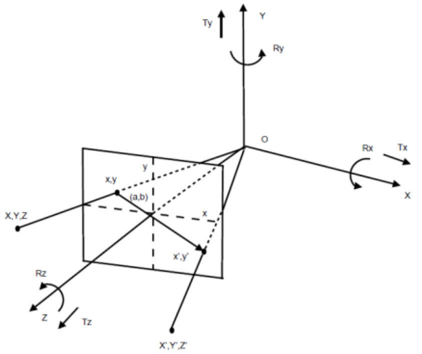


Figure 31. The camera-centered coordinate system.

If the camera performs a 3D motion (R, t) the scene point $\boldsymbol{X} = (X, Y, Z)$ appears to be moving relative to the camera with a motion $(-R, -t)$ and is observed at new world coordinates $\boldsymbol{X}' = (X', Y', Z')$. Under the differential motion assumption (Huang & Fang, 1984), the new coordinates of the point can be expressed in the so called *bilinear constraint*:

$$\boldsymbol{X}' = \Omega \times \boldsymbol{X}^T + V$$

Notice that for the previous equation to be valid for every point, the whole scene must behave like a rigid body.

Replacing the rotational velocity for its skew-symmetric matrix, the cross product becomes a matrix multiplication:

$$\boldsymbol{X}' = M_\Omega X + V = \begin{bmatrix} 1 & -\Omega_z & \Omega_y \\ \Omega_z & 1 & -\Omega_x \\ -\Omega_y & \Omega_x & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} X - Y\Omega_z + Z\Omega_y + V_x \\ X\Omega_z + Y - Z\Omega_x + V_y \\ -X\Omega_y + Y\Omega_x + Z + V_z \end{bmatrix}$$

The corresponding coordinates on the image plane (x', y') can be obtained using perspective projection:

$$x' = \frac{X'}{Z'} = \frac{X - Y\Omega_z + Z\Omega_y + V_x}{-X\Omega_y + Y\Omega_x + Z + V_z} \;\; ; \;\; y' = \frac{Y'}{Z'} = \frac{X\Omega_z + Y - Z\Omega_x + V_y}{-X\Omega_y + Y\Omega_\mathrm{x} + Z + V_z}$$

Since $x = X/Z$ and $y = Y/Z$, replacing $X = xZ$ and $Y = yZ$ in the previous equations and dividing by Z:

$$x' = \frac{x - \Omega_z y + \Omega_y + \frac{V_x}{Z}}{-\Omega_y x + \Omega_x y + 1 + \frac{V_z}{Z}} \;\; ; \;\; y' = \frac{\Omega_z x + y - \Omega_x + \frac{V_y}{Z}}{-\Omega_y x + \Omega_x y + 1 + \frac{V_z}{Z}}$$

The displacement of the image point (u, v) caused by the differential camera is:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \end{bmatrix} \left(\Omega \times \boldsymbol{X}^T + \frac{V}{Z} \right)$$

where u, v and Z are the components of the image velocity and the depth at each image position (x, y), and V and Ω are the translational and rotational velocities.

A.2 Egomotion: method description.

Eight-points algorithm

This section explains the *eight-points algorithm* as described in (Ma, Soatto, Kosecka, & Sastry, 2003).

As discussed before, the image points of two views of a rigid scene are related by the epipolar constraint. Each pair of corresponding image points $(\hat{x}, \hat{x}')$ give rise to an equation: $\hat{x}\ E\ \hat{x}' = 0$.

The algorithm receives as inputs the 2D point's matches in normalized image coordinates[13], estimates the essential matrix E and outputs the motion parameters(R, t). At least eight points located in general position are required in order to uniquely recover the motion parameters.

Estimation of E

The algorithm starts by computing a first approximation of E which satisfies the epipolar constraint but which may not have the structure of an essential matrix.

Expanding the epipolar constraint between two point's correspondences:

$$x_2^T\ E\ x_1 = [x_2, y_2, z_2] \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = x_2 x_1 e_{11} + \cdots + z_2 z_1 e_{33} = 0$$

This equation can be expressed as:

$$a^T E_s = 0$$

Where a^T is a vector containing the columns of the Kronecker product between x_2 and x_1 and E_s is the vector containing the columns of E.

Given the set of points matches (x_1^i, x_2^i) all the equations can be expressed as the system $\chi\ E_s = 0$ where $\chi = [a^1, a^2, \dots, a^n]^T \in \mathbb{R}^{n\times 9}$.

In order to solve the above equation the rank of χ must be eight, thus eight points are required to compute E_s. However, when more points correspondences are used, the E_s that minimizes a least-squares error function can be obtained:

$$\|\chi\ E^s\|^2$$

E_s can be chosen to be the eigenvector of $\chi^T \chi$ corresponding to its smallest eigenvalue.

Projection of E onto the essential space

The essential matrix estimated before may not satisfy the internal structure of the essential matrices. To enforce this, E need to be projected onto the space of essential matrices.

If $E' = U\ diag\{\lambda_1, \lambda_2, \lambda_3\}\ V^T$ with $\lambda_1 \geq \lambda_2 \geq \lambda_3$ is the SVD decomposition of the first approximation of E, the matrix belonging to the essential space that minimizes the distance to E' is given by:

$$E = U\ diag\left\{\frac{(\lambda_1 + \lambda_2)}{2}, \frac{(\lambda_1 + \lambda_2)}{2}, 0\right\}\ V^T$$

[13] Calibration matrix must be known in order to normalized the detected image points.

Decomposition of E: Numerical method.

Four solutions can be extracted from an SVD decomposition of $E = U\ D\ V^T$.

$$R_1 = U\ W\ V^T\ ;\ R_2 = U\ W^T\ V^T\ ;\ \ R_3 = R_1\ \ ;\ \ R_4 = R_2$$

$$t_1 = V\ W diag\{1{,}1{,}0\}V^T\ \ ;\ t_2 = V\ W^T diag\{1{,}1{,}0\}\ V^T\ \ ;\ t_3 = t_2\ \ ;\ \ t_4 = t_1$$

Where

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

Decomposition of E: Analytical method

Horn proposed an analytical method to recover the motion parameters directly from the elements of E (Horn, 1990). The two possible solutions for the translation and the relative orientation can be computed from:

$$t\ t^T = \frac{1}{2}\ \mathrm{trace}(EE^T)\ I - EE^T \quad ; \quad (t \cdot t)\ R = \mathrm{cofactors}(E)^T - t\ E$$

Where *trace* is the sum of the elements of the main diagonal and *cofactors* is the cofactor matrix obtaining from replacing each element e_{ij} with its cofactor (or minor).

Remarks

In order to solve the above equation uniquely for E_s (up to a scalar factor[14]) the rank of $\chi \in \mathbb{R}^{n\times 9}$ must be eight. If the points lie in certain *degenerate* configurations, such as all the points lying on a plane, the rank of χ will be less than eight and the equation cannot be solved uniquely.

In addition, it was assumed that $E \neq 0$ in the epipolar constraints. Since this is not valid only when $T = 0$, the described method requires a significant displacement even if a spurious translation will always be measured in a real scenario.

Four-points algorithm for planar cases (DLT)

This section explains the *four-points algorithm* as described in (Ma, Soatto, Kosecka, & Sastry, 2003). The method uses a *Direct Linear Transformation* algorithm so it is also referred as DLT.

Similarly to the eight-points algorithm, the first steps is to estimate the homography matrix H and normalize it. Then, the motion parameters and structure parameters can be recovered from it.

Estimate H

The planar epipolar constraint between two corresponding points can be constructed using $\widehat{\boldsymbol{x}_2}$.

$$\widehat{\boldsymbol{x}_2}\ H\ \boldsymbol{x}_1 = \begin{bmatrix} 0 & -z_2 & y_2 \\ z_2 & 0 & -x_2 \\ -y_2 & x_2 & 0 \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{21} & h_{23} \\ h_{31} & h_{31} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = 0$$

As before, we can rewrite the planar epipolar constraint as $a^T H_s = 0$ where $a \in \mathbb{R}^{9\times 3}$ is the Kronecker product of $\widehat{\boldsymbol{x}_2}$ and $\boldsymbol{x}_1$ and H_s is a vector consisting in the stacked columns of H.

[14] The epipolar constraint is homogeneous in E, thus multiplying by any non-zero constant gives another solution.

A matrix can be constructed by joining the constraints from all the point's correspondences (x_1^i, x_2^i) $i = 1, \dots, n$ $(n \geq 4)$[15] :

$$\chi = [a^1, a^2, \dots, a^n]^T \in \mathbb{R}^{3n \times 9}$$

The planar epipolar constraint for each points correspondence can then be expressed as $\chi\, H_s = 0$. In order to solve the equation we use a similar least-squares approach to find the H_s that minimizes the error function:

$$\|\chi\, H_s\|^2$$

Normalize the homography matrix H

The homography obtained from the previous step was estimated up to a scalar factor:

$$H_L = \lambda H = \lambda \left(R + \frac{1}{d} t\, N^T \right)$$

Since $|\lambda|$ is the second largest eigenvalue of H_L, the homography can be normalized and two possible solutions are obtained:

$$H = \frac{\pm\, H_L}{\sigma_2(H_L)} = \pm \left(R + \frac{1}{d} t\, N^T \right)$$

An additional positive depth constraint can be applied to get the correct sign of H. Since $\lambda_2\, \boldsymbol{x}_2 = H\, \lambda_1\, \boldsymbol{x}_1$ and $\lambda_1, \lambda_2 > 0$, then for every point correspondence:

$$x_2^T\, H\, x_1 > 0$$

The homography matrix H can then be uniquely recovered from enough point's correspondences.

Recover the motion parameters: Numerical method.

The motion parameters can be computed from the eigenvalues and eigenvectors of the homography decomposition $H^T H = V \Sigma V^T$. Where $\Sigma = \mathrm{diag}\{\sigma_1^2, \sigma_2^2, \sigma_3^2\}$; $\sigma_1^2 \geq \sigma_2^2 \geq \sigma_3^2 \geq 0$.

Since $\sigma_2 = 1$, the length of v_2 is preserved under the map of H: $H^T H v_2 = v_2$. Two other unit-length vectors can be constructed with the same property[16]:

$$u_1 = \frac{\sqrt{1 - \sigma_3^2}\, v_1 + \sqrt{\sigma_1^2 - 1}\, v_3}{\sqrt{\sigma_1^2 - \sigma_3^2}} \quad ; \quad u_2 = \frac{\sqrt{1 - \sigma_3^2}\, v_1 - \sqrt{\sigma_1^2 - 1}\, v_3}{\sqrt{\sigma_1^2 - \sigma_3^2}} \; ;$$

In addition, since v_2 is orthogonal to both u_1 and u_2, two sets of orthonormal bases of $\mathbb{R}^3$can be constructed: $\{v_2, u_1, \widehat{v_2} u_1\}$ and $\{v_2, u_2, \widehat{v_2} u_2\}$. The utility of these bases lies in the fact that their length is preserved by the application of H so the effects of applying it are due only to the rotational component:

$$H v_2 = R v_2 \quad ; \quad H u_i = R u_i \quad ; \quad \widehat{H v_2} H u_i = R \widehat{v_2} u_i \quad ; \quad i = 1{,}2$$

[15] Since the rank of $\widehat{\boldsymbol{x}_2}$ is 2, each point correspondence adds two constraints, thus at least 4 points are needed.

[16] H preserves the length of any vector from the subspaces $S_1 = span\{v_2, u_1\}$ and $S_2 = span\{v_2, u_2\}$.

This application can be grouped in a matrix form as follows:

$$U_i = [v_2, u_i, \widehat{v_2}u_i] \quad ; \quad W_i = \left[Hv_2, Hu_i, \widehat{Hv_2}Hu_i\right] \quad ; \quad i = 1{,}2$$

$$RU_i = W_i \quad ; \quad i = 1{,}2$$

Then the two possible solutions for the rotational component are $R_i = W_i U_i^T \; ; \; i = 1{,}2$.

The possible solutions for the plane normal and translational component are obtained directly from the above results:

$$N_i = \widehat{v_2}u_i \quad ; \quad \frac{1}{d}t_i = (H - R_i)N_i \quad ; \quad i = 1{,}2$$

Due to the sign ambiguity of the term $\frac{1}{d}tN^T$, the four possible decompositions are obtained:

$$R_i = W_i U_i^T \quad ; \quad N_i = \widehat{v_2}u_i \quad ; \quad \tfrac{1}{d}\, t_i = (H - R_i)N_i \quad ; \quad i = 1{,}2$$

$$R_i \quad ; \quad -N_i \quad ; \quad -\tfrac{1}{d}\, t_i \quad ; \quad i = 1{,}2$$

Recover the motion parameters: Analytical method

However, a method was proposed to compute the motion and structure parameters directly from H (Malis & Vargas, 2007). It introduces a symmetric matrix S:

$$S = H^T H - I = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{12} & s_{22} & s_{23} \\ s_{13} & s_{23} & s_{33} \end{bmatrix}$$

The plane normal can be obtained from any of the three elements of the diagonal (s_{ii}), however, using the largest s_{ii} will be the most well-conditioned option [17].

$$N'(s_{11}) = \begin{bmatrix} s_{11} \\ s_{12} \pm \sqrt{M_{S_{33}}} \\ s_{13} \pm \epsilon_{23}\sqrt{M_{S_{22}}} \end{bmatrix} \quad ; \quad N'(s_{22}) = \begin{bmatrix} s_{12} \pm \sqrt{M_{S_{33}}} \\ s_{22} \\ s_{23} \mp \epsilon_{13}\sqrt{M_{S_{11}}} \end{bmatrix} \quad ; \quad N'(s_{33}) = \begin{bmatrix} s_{13} \pm \epsilon_{12}\sqrt{M_{S_{22}}} \\ s_{23} \pm \sqrt{M_{S_{11}}} \\ s_{33} \end{bmatrix}$$

Where $M_{S_{ij}}$is the expressions of the minor corresponding to element s_{ij} , and $\epsilon_{ij} = sign\ (M_{S_{ij}})$ [18].

From the above equations, two different normal vectors are obtained and we must include the opposite of each to the set of possible solutions. The translational component for the reference frame[19] can be obtained from the expressions of the normal vectors:

$$t_1^*(s_{ii}) = \frac{\|t_e^*\|}{2}\left[\epsilon_{s_{ii}}\, \rho\, N_2(s_{ii}) - \|t_e^*\| N_1(s_{ii})\right]$$

[17] Notice that two solutions are obtained for any of the s_{ii} and that $N'(s_{11}) = N'(s_{22}) = N'(s_{33})$. Also, this normal vectors are not normalized so: $N(s_{ii}) = N'(s_{ii})/\|N'(s_{ii})\|$.

[18] The following *sign* function should be used: $\mathrm{sign}(a) = \{1\ if\ a \geq 0\ \ ;\ -1\ \mathrm{otherwise}\}$.

[19] $T_e^* = R_e^T T_e$

$$t_2^*(s_{ii}) = \frac{\|t_e^*\|}{2}\left[\epsilon_{s_{ii}}\,\rho\,N_2(s_{ii}) - \|t_e^*\| N_1(s_{ii})\right]$$

Where:

$$\epsilon_{s_{ii}} = \text{sign}(s_{ii})$$

$$\rho^2 = 2 + \text{trace}(S)\; +\; \nu$$

$$\|t_e^*\|^2 = 2 + \text{trace}(S) - \nu$$

$$\nu = 2\sqrt{1 + \text{trace}(S) - M_{S_{11}} - M_{S_{22}} - M_{S_{33}}}$$

From this, the rotation matrix and the relative translational vector can be obtained:

$$R_i = H\left(I - \frac{2}{\nu} t_i^* N_i^T\right)\;\;;\;\;\; t_i = R_i\, t_i^*$$

The number of solutions can be reduced from four to two by imposing a positive depth constraint. Since points must be in front of both cameras frames, verifying that $N^T e_3 = n_3 > 0$ will eliminate two of the computed solutions. The remaining two solutions are physically possible and without additional information they cannot be discarded[20].

Planar-ego

This section explains the Planar-ego method as described in (Oreifej, Lobo, & Shah, 2011).

The method addresses the planar case by modeling the ground plane in terms of the rotational parameters of an unmanned aerial vehicle. The ground plane equation is included in the motion equations to constraint the motion parameters.

The authors propose also a method to estimate the rotational parameters by using the horizon line although this is not used in this work since the rotational parameters are available and can be used at any time.

The ground plane equation

The equation of the ground plane in the camera coordinate system can be expressed as

$$\hat{n}\,\boldsymbol{X} = h$$

where $\hat{n}$ is the unit normal of the plane, $\boldsymbol{X} = (X, Y, Z)$ is a point on the ground plane and h is the height of the aircraft.

If the aircraft is level with the ground (roll and pitch are zero), the normal of the plane is simply $\hat{n} = \boldsymbol{n_0} = (0,1,0)^T$. However, when the rotational parameters are non-zero, the direction of the unit normal $\hat{n}$, must be rotated accordingly $\hat{n} = R_a\, R_b\, \boldsymbol{n_0}$.

[20] In fact, in the particular case where $T \sim N$, u_1 results equal to u_2 and the solutions are equivalent. Hence, in this case, imposing the positive depth constraint will lead to a unique solution.

Where R_a and R_b are rotational matrices that depends on the roll (α)and pitch (β) angles as follows.

$$R_a = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \; ; \; R_b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix}$$

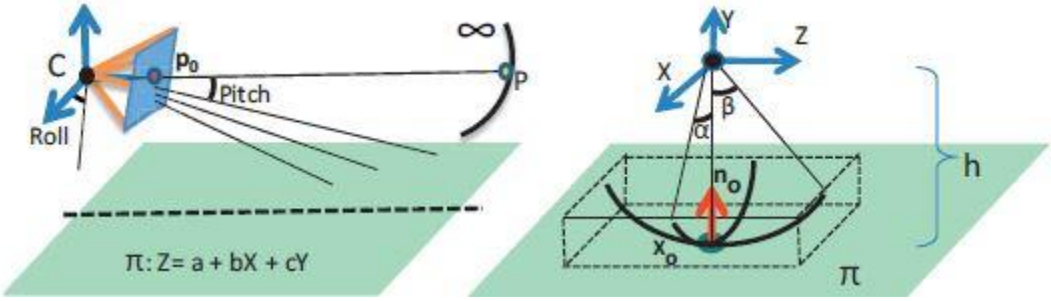


Figure 32. The ground plane equation is modeled in function of the rotational parameters of the aircraft.

The equation of the ground plane can then be expressed as follows: $Z = b\,X + c\,Y \; + a$ where $a = -h/\,(\sin\beta\cos\alpha)\,;\; b = \tan\alpha\,/\sin\beta \;;\; c = -1/\tan\beta$.

This allows obtaining a unique linear solution for the ego-motion. Recalling the motion equations:

$$u = f\left(V_x/Z + \Omega_y\right) - (x\,V_z/Z) - y\,\Omega_z - xy\,\Omega_x/f + x^2\,\Omega_y/f$$

$$v = f\left(V_y/Z + \Omega_x\right) - (y\,V_z/Z) + x\,\Omega_z - xy\,\Omega_y/f - y^2\,\Omega_x/f$$

Since all the points of the scene belong to the plane, the equation of the ground plane can be included in the previous equations to obtain:

$$u \;=\; q_1 + q_2\,x + q_3\,y + q_4\,x^2 + q_5\,x\,y \quad ; \quad v \;=\; q_6 + q_7\,x + q_8\,y + q_4\,x\,y + q_5\,y^2$$

Where

$$q_1 = f\;(V_x/a) + f\Omega_y \;;\; q_2 = -(V_x\,b/a) - V_z/a \;;\; q_3 = -(V_x\,c/a) - \Omega_z \;;\; q_4 = (V_z\,c/af) + \Omega_y/f$$

$$q_5 = (V_z c/af) - \Omega_x/f \;;\; q_6 = f\left(V_y/a\right) - f\Omega_x \;;\; q_7 = -\left(V_y\,b/a\right) + \Omega_z \;;\; q_8 = -(V_x\,c/a) + V_z/a$$

Estimating q coefficients and recovering motion parameters

Since the optical flow at each image point give rise to two equations, at least four points are needed to estimate the q coefficients. However, using more points will improve the estimation. As in the discrete-time approach, a least-squares approach is used.

The optical flow computed at each image point (u_i, v_i) is stacked together to form a vector as well as the coordinates multiplying the q coefficients.

$$U = (u_1\,..\;u_n\,v_1\,..\,v_n)^T \in \mathbb{R}^{2n\times 1} \quad ; \quad M = \begin{bmatrix} 1 & x & y & x^2 & xy & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & xy & y^2 & 1 & x & y \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \in \mathbb{R}^{2n\times 8}$$

The linear system $M\ Q = U$ can then be solved to obtain the q coefficients $Q\ =\ M^{+}\ U$ where M^{+} is the pseudoinverse of M.

Without any information about the scene or the rotational parameters, the system includes 8 non-linear equations in terms of the translational and rotational velocities, and the equation of the ground plane. The authors solved this problem by estimating the rotational parameters from the horizon line. However, when a, b and c can be computed directly from available measures, the equations become linear and the motion parameters can be directly recovered from them.

Other reviewed methods

There has been considerable work in the last thirty years to solve the problem of motion estimation from an image sequence. Different approaches and techniques have been proposed each of which behave well in some situations and have problems in others.

This is a brief description of the ego-motion estimation methods reviewed during this work. Most of them are continuous-time methods and they are cited quite often in the literacy. It is not the intention to perform an exhaustive review, nor to describe each method in detail, but to present the main characteristics of the methods that were found relevant for this work.

(Prazdny, 1980)**:** The method consists in solving a system of nonlinear equations relating the optical flow with the rotational component of motion (and thus independent of translation and depth) using a triple of image points. These constraints can be expressed as $n_3(n_1 \times n_2) = 0$ and $n_i = (\boldsymbol{\Omega} \times \boldsymbol{s}_i + \boldsymbol{v}_i) \times \boldsymbol{s}_i$ where $\boldsymbol{s}_i$ and $\boldsymbol{v}_i$ are the image coordinates and velocity. Prazdny combined the constraints locally and solved numerically the three third-order polynomial equations of three unknowns. The rotational component is obtained numerically and then is used to compute the translational component. The computational costs increase substantially with the number of points being tracked so the approach resulted computationally expensive.

(Bruss & Horn, 1983)**:** The methods started with the motion constraint: $V = (-\omega \times r) - T,$ where $T = (T_x, T_y, T_z)$, $\omega = (\omega_x, \omega_y, \omega_z)$ and $r = (X, Y, Z)$. From this constraint, they proposed a least-squares approach writing the equations in term of the optical flow, and separating the rotation and translations effects in the optical flow:

$$(u, v) = (u_R + u_T, v_R + v_T)$$

$$\varepsilon(T, R, Z) = \sum_{i=1}^{n} (u_i - u_{Ri} - u_{Ti})^2\ + (v_i - v_{Ri} - v_{Ti})^2$$

Normalizing the translation using $U = (U_x, U_y, U_z) = \frac{(T_x, T_y, T_z)}{|T_x, T_y, T_z|}$ and $\widetilde{Z}_\iota = \frac{|T_x, T_y, T_z|}{Z_i}$ they derive the objective function with respect to Z to find the optimal value and then rewrite the objective function $\varepsilon(U, R)$.

The resulting equations system contains three linear equations for the rotation and four non-linear equations on translation. An estimation of the rotation can be obtained as a function of translation ($R(T)$), substituting this rotation into the bilinear constraint a nonlinear constraint on T is obtained.

Finally the estimated translation is used to compute R. They proposed to solve the equations using an iterative numerical method where translation can be obtained by searching the translational vector that minimize the error over all image velocities in a unitary sphere (|T|=1).

(Burger & Bhanu, 1990): A continuous approach based on the computation of the *fuzzy FOE,* a region where the FOE can be found, that estimate the speed of a vehicle is computed from detected on the ground.

The first step is to transform the camera-centered coordinate system (XYZ) to a new rotated coordinate system $(X'Y'Z')$ to line up the z-axis with the direction of translation (FOE). If the FOE is located at (x_f, y_f), the required *pan* and *tilt* rotations are computed (θ_f, ϕ_f). A given image point (x, y), is a projection of an unknown 3D points that lie on the straight line. For points on the ground (x_g, y_g), the scale factor κ can be estimated given that $Y = -h$. The distance of the 3D points (their Z_g coordinate) can then be obtained:

$$Z_g = -h \frac{x_s \sin\theta_f - y_s \cos\theta_f \sin\phi_f + f \cos\theta_f \cos\phi_f}{y_s \cos\phi_f + f \sin\phi_f}$$

If the depth of a point is measured at two different time instances t, t' the displacement and velocity can be computed as $\Delta Z_g(t, t') = Z_g - Z_g'$ and $V_{g(t,t')} = \frac{\Delta Z_g(t,t')}{t\prime - t} = \frac{Z_g - Z_g'}{t' - t}$.

(Heeger & Jepson, 1992) and (Jepson & Heeger, 1992): Set of methods based on motion parallax, called subspace methods. Translation is estimated minimizing this constraint over all image velocities in a unitary sphere ($|T| = 1$).

(Tomasi & Shi, 1993)**:** Translational-first approach that estimates translation from *image deformations*. It uses motion parallax information estimating T from image deformations (defined as the change of angular distance between pair of image points as the camera moves). Image deformations are independent of camera rotation so a bilinear constraint on T is derived. This can be minimized and solved for T on a subset of point pairs using the *variable projection method* on the unit sphere |T|=1. This involves solving 3 translation parameters and N depth parameters, where N is the number of points. If N is large, the algorithm becomes very expensive.

(Kanatani, 1993)**:** Uses a continuous version of the epipolar constraint in terms of essential parameters and twisted flow (a rotated version of the velocity vector). Kanatani noted that the least-squares estimates of T are systematically biased. He analyzed the statistical bias and proposed a renormalization method that removes it by automatically compensating for the unknown noise.

(Irani, Rousso, & Peleg, 1997): The authors proposed a method to estimate the ego-motion in a static scene directly from image intensities avoiding the need for a reliable feature detector and complete optical flow estimation. The general idea of the method is:

1. Detect a region corresponding to a *planar* surface and compute its 2D motion parameters.
2. Register images frames (image warping) to cancel the 3D camera rotation in the 2D motion. The residual flow field points to the *focus of expansion* (FOE).
3. Compute translation by estimating the position of the FOE using the registered images.
4. Compute rotation using translation.

The FOE from two registered image frames, indicates the direction of the translational motion of the camera, and thus provides a first estimation of the camera motion.

Ego-motion is reliably recovered except in the case of a pure translation in the *x-y* plane (an example of the planar degeneracy). Lastly, the method relies on the detection of the 2D planar surface in the scene.

A.3 Results of sensitivity evaluation

This section contains the results obtained for the sensitivity evaluation of the ground speed estimation.

Varying focal length

The tables below show the ground speed error measures obtained for a camera fixed in the cockpit and in the vertical stabilizer.

Camera fixed in the Cockpit

TAXI	Max.	ME	MAE	RMSE	SD	Estimation ratio
18mm	8.01	-2.79	2.79	3.52	2.28	25.7 %
30mm	4.97	-1.01	1.41	2.02	1.80	54.3 %
45mm	5.45	-1.19	1.90	2.28	2.01	48.6 %

TAXI-R	Max.	ME	MAE	RMSE	SD	Estimation ratio
18mm	3.04	-1.17	1.37	1.77	1.42	17.8 %
30mm	7.24	-0.92	1.15	1.84	1.61	75.6 %
45mm	5.53	-1.02	1.96	2.57	2.40	68.9 %

TAKEOFF	Max.	ME	MAE	RMSE	SD	Estimation ratio
18mm	11.98	-7.49	7.49	8.27	4.06	5.0 %
30mm	33.60	-18.87	18.87	21.03	9.43	37.5 %
45mm	29.48	-14.68	15.79	18.84	12.02	35.0 %

APPROACH	Max.	ME	MAE	RMSE	SD	Estimation ratio
18mm	17.21	-3.63	3.63	5.22	3.83	26.0 %
30mm	29.49	-0.93	2.57	4.58	4.51	89.0 %
45mm	7.93	-0.67	2.05	2.82	2.76	85.0 %
100mm	23.67	-1.48	5.53	7.53	7.45	54.00%

Camera fixed on the Vertical Stabilizer

TAXI	Max.	ME	MAE	RMSE	SD	Estimation ratio
30mm	6.16	-0.17	1.20	1.79	1.80	97.14%
45mm	8.70	0.09	1.36	2.24	2.28	85.71%

TAXI-R	Max.	ME	MAE	RMSE	SD	Estimation ratio
30mm	4.93	1.04	1.65	2.04	1.78	88.89%
45mm	10.47	2.12	2.98	3.68	3.03	97.78%

TAKEOFF	Max.	ME	MAE	RMSE	SD	Estimation ratio
30mm	27.24	-12.35	12.76	14.95	8.50	72.50%
45mm	32.57	-19.93	12.38	15.60	11.22	75.00%

APPROACH	Max.	ME	MAE	RMSE	SD	Estimation ratio
30mm	33.80	0.45	5.06	8.47	8.52	79.00%
45mm	43.66	-1.30	6.06	10.04	10.00	89.00%

The following graphs show the ground speed error for the camera mounted in the Cockpit.

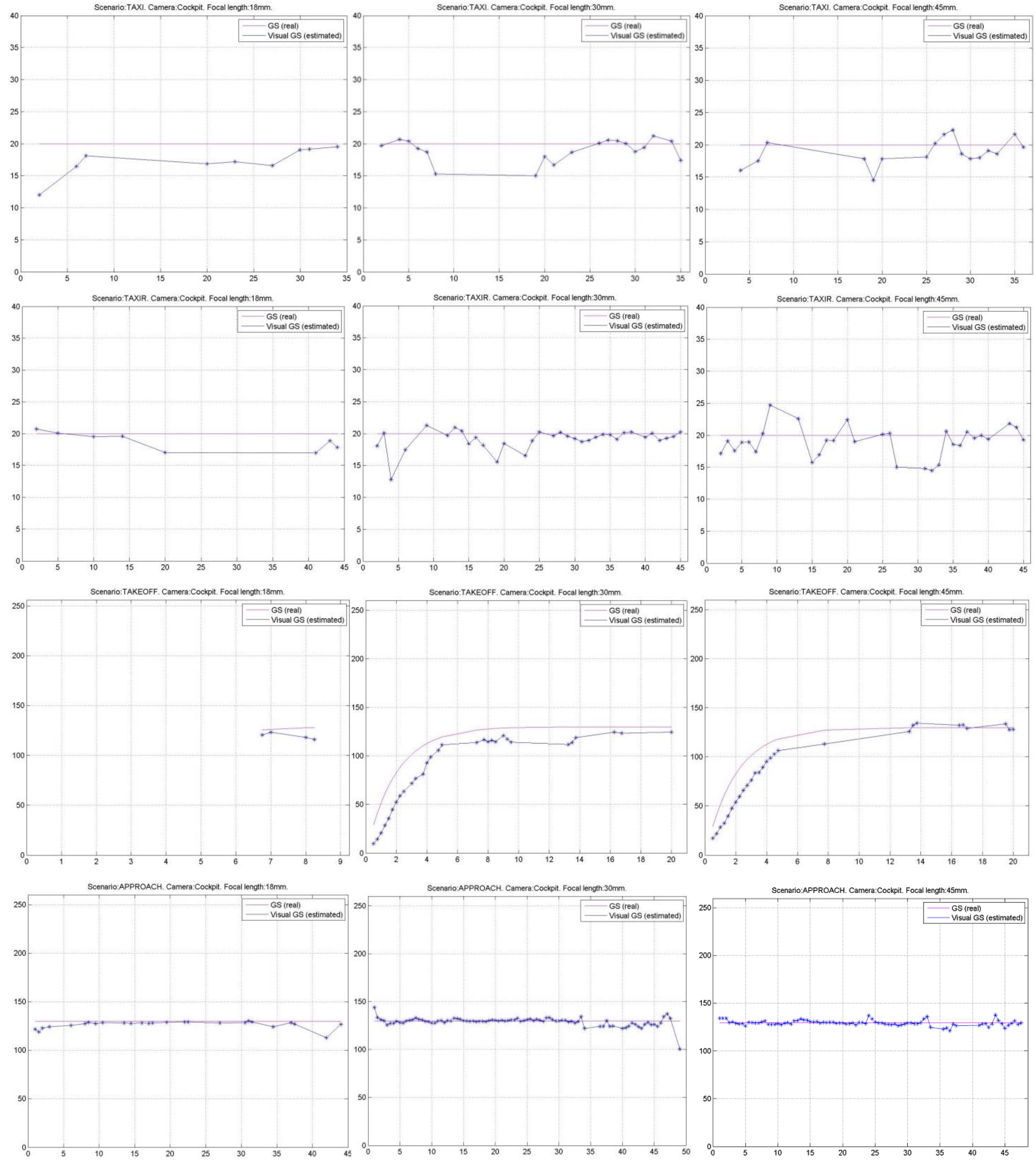


Figure 33. Ground speed estimation for different focal lengths. Rows represent the scenarios: TAXI, TAXI-R, TAKEOFF and APPROACH while columns represent the focal lengths: 18mm (left), 30mm (middle) and 45mm (right).

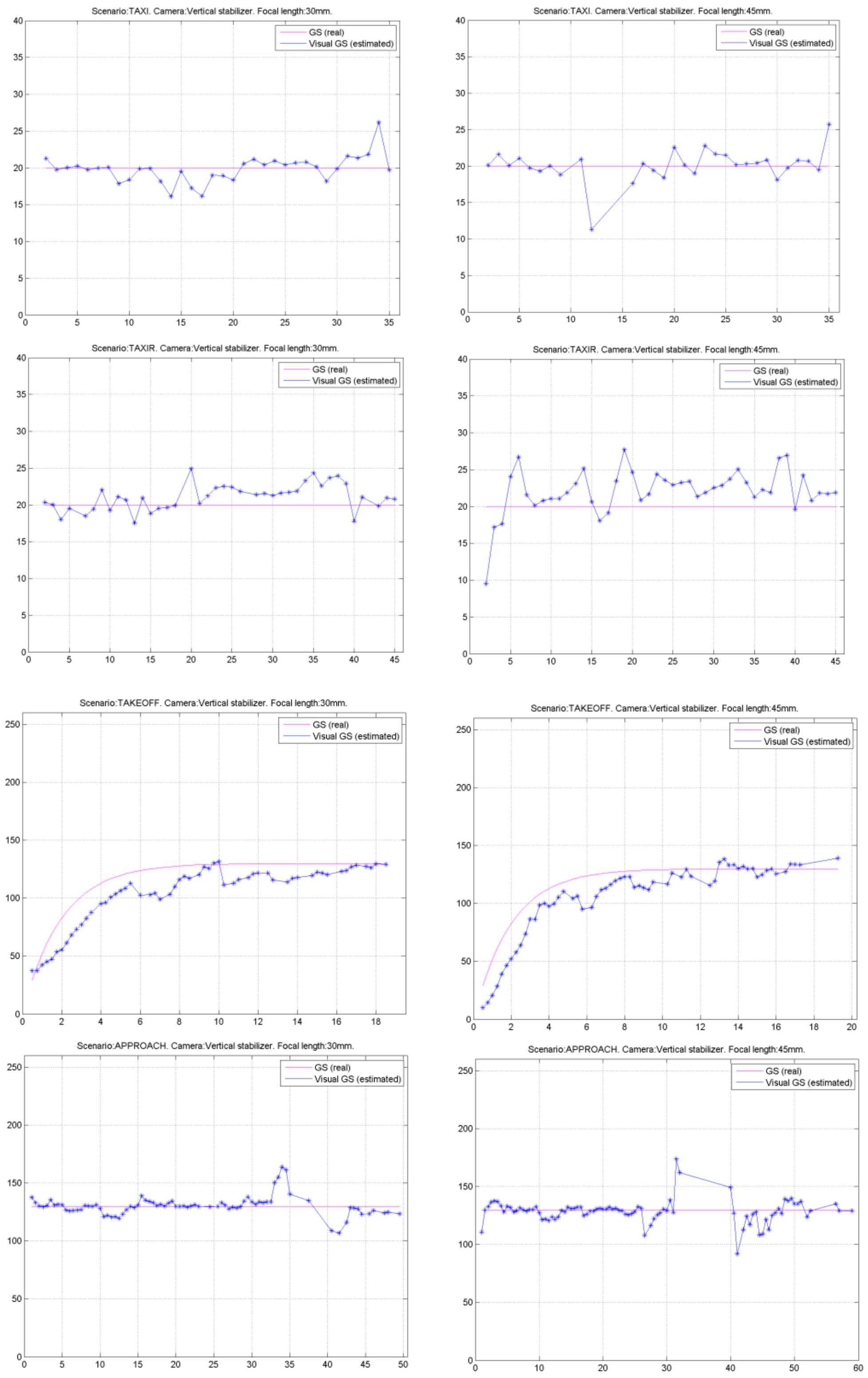


Figure 34. Ground speed estimation for a 30mm (left) and 45mm (right) focal lengths of a camera mounted on the vertical stabilizer over TAXI, TAXI-R, TAKEOFF and APPROACH (from top to bottom).

Varying camera position

The results showed in the previous were compared by varying the position of the camera and fixing the focal length of the camera. The results obtained for a 30mm focal length are shown below.

45mm fixed Focal length

TAXI	Max.	ME	MAE	RMSE	SD	Estimation ratio
Cockpit	5.45	-1.19	1.90	2.28	2.01	48.6 %
V. Stabilizer	8.70	0.09	1.36	2.24	2.28	85.71%

TAXI-R	Max.	ME	MAE	RMSE	SD	Estimation ratio
Cockpit	5.53	-1.02	1.96	2.57	2.40	68.9 %
V. Stabilizer	10.47	2.12	2.98	3.68	3.03	97.78%

TAKEOFF	Max.	ME	MAE	RMSE	SD	Estimation ratio
Cockpit	29.48	-14.68	15.79	18.84	12.02	35.00 %
V. Stabilizer	32.57	-19.93	12.38	15.60	11.22	75.00%

APPROACH	Max.	ME	MAE	RMSE	SD	Estimation ratio
Cockpit	7.93	-0.67	2.05	2.82	2.76	85.00 %
V. Stabilizer	43.66	-1.30	6.06	10.04	10.00	89.00%

30mm fixed Focal length

TAXI	Max.	ME	MAE	RMSE	SD	Estimation ratio
Cockpit	4.97	-1.01	1.41	2.02	1.80	54.3 %
V. Stabilizer	6.16	-0.17	1.20	1.79	1.80	97.14%

TAXI-R	Max.	ME	MAE	RMSE	SD	Estimation ratio
Cockpit	7.24	-0.92	1.15	1.84	1.61	75.6 %
V. Stabilizer	4.93	1.04	1.65	2.04	1.78	88.89%

TAKEOFF	Max.	ME	MAE	RMSE	SD	Estimation ratio
Cockpit	33.60	-18.87	18.87	21.03	9.43	37.5 %
V. Stabilizer	27.24	-12.35	12.76	14.95	8.50	72.50%

APPROACH	Max.	ME	MAE	RMSE	SD	Estimation ratio
Cockpit	29.49	-0.93	2.57	4.58	4.51	89.0 %
V. Stabilizer	33.80	0.45	5.06	8.47	8.52	79.00%