Linköping studies in science and technology. Dissertations.
No. 1533

Navigation and Mapping for Aerial Vehicles Based on Inertial and Imaging Sensors

Zoran Sjanic

Department of Electrical Engineering
Linköping University, SE–581 83 Linköping, Sweden

Linköping 2013

Cover illustration: Background constitutes of a composite image of Washington D.C. where an optical image from Google Maps is used in the upper-right and a SAR image from Sandia National Laboratories is used in the lower-left corner. For the foreground, a front view image of the rotor Unmanned Aerial Vehicle Saab V-200 Skeldar is used.

Linköping studies in science and technology. Dissertations.
No. 1533

Navigation and Mapping for Aerial Vehicles Based on Inertial and Imaging Sensors

Zoran Sjanic

`zoran@isy.liu.se`
`www.control.isy.liu.se`
Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE–581 83 Linköping
Sweden

ISBN 978-91-7519-553-7 ISSN 0345-7524

Printed by LiU-Tryck, Linköping, Sweden 2013

Za Almu

Abstract

Small and medium sized Unmanned Aerial Vehicles (UAV) are today used in military missions, and will in the future find many new application areas such as surveillance for exploration and security. To enable all these foreseen applications, the UAV's have to be cheap and of low weight, which restrict the sensors that can be used for navigation and surveillance. This thesis investigates several aspects of how fusion of navigation and imaging sensors can improve both tasks at a level that would require much more expensive sensors with the traditional approach of separating the navigation system from the applications. The core idea is that vision sensors can support the navigation system by providing odometric information of the motion, while the navigation system can support the vision algorithms, used to map the surrounding environment, to be more efficient. The unified framework for this kind of approach is called Simultaneous Localisation and Mapping (SLAM) and it will be applied here to inertial sensors, radar and optical camera.

Synthetic Aperture Radar (SAR) uses a radar and the motion of the UAV to provide an image of the microwave reflectivity of the ground. SAR images are a good complement to optical images, giving an all-weather surveillance capability, but they require an accurate navigation system to be focused which is not the case with typical UAV sensors. However, by using the inertial sensors, measuring UAV's motion, and information from the SAR images, measuring how image quality depends on the UAV's motion, both higher navigation accuracy and, consequently, more focused images can be obtained. The fusion of these sensors can be performed in both batch and sequential form. For the first approach, we propose an optimisation formulation of the navigation and focusing problem while the second one results in a filtering approach. For the optimisation method the measurement of the focus in processed SAR images is performed with the image entropy and with an image matching approach, where SAR images are matched to the map of the area. In the proposed filtering method the motion information is estimated from the raw radar data and it corresponds to the time derivative of the range between UAV and the imaged scene, which can be related to the motion of the UAV.

Another imaging sensor that has been exploited in this framework is an ordinary optical camera. Similar to the SAR case, camera images and inertial sensors can also be used to support the navigation estimate and simultaneously build a three-dimensional map of the observed environment, so called inertial/visual SLAM. Also here, the problem is posed in optimisation framework leading to batch Maximum Likelihood (ML) estimate of the navigation parameters and the map. The ML problem is solved in both the straight-forward way, resulting in nonlinear least squares where both map and navigation parameters are considered as parameters, and with the Expectation-Maximisation (EM) approach. In the EM approach, all unknown variables are split into two sets, hidden variables and actual parameters, and in this case the map is considered as parameters and the navigation states are seen as hidden variables. This split enables the total problem to be solved computationally cheaper then the original ML formulation. Both optimisation problems mentioned above are nonlinear and non-convex requiring good initial solution in order to obtain good parameter estimate. For this purpose a method for initialisation of inertial/visual SLAM is devised where the conditional linear structure of the problem is used to obtain the initial estimate of the parameters. The benefits and performance improvements of the methods are illustrated on both simulated and real data.

Populärvetenskaplig sammanfattning

Obemannade flygande farkoster, eller Unmanned Aerial Vehicles (UAV) på engelska, är en typ av farkoster där operatören inte sitter ombord utan styr den från marken. Dessa används idag, i huvudsak, av militären för olika typer av uppdrag, men det finns en stor potential för framtida civil användning, speciellt inom övervaknings- och kartläggningsuppdrag. Dessutom ger det faktum att ingen människa sitter i UAV:en en viktig fördel över bemannade farkoster, nämligen att även uppdrag som skulle kunna vara farliga nu kan utföras utan att liv riskeras. För att göra UAV-operatörer ännu effektivare i utförandet av uppdragen, måste farkosterna bli mer och mer autonoma, dvs. ha förmågan att själva utföra vissa uppgifter, som till exempel flyga en fördefinierad rutt. En kritisk komponent i detta är navigeringssystemet, som har som huvuduppgift att beräkna och förse olika andra system med farkostens position, höjd, fart och orientering. För att göra detta, används olika mätsensorer som mäter UAV:ens rörelse eller dess läge i förhållande till andra objekt med kända positioner. Kraven på navigeringssystemet för att kunna klara av att ge stöd till övriga system är att det måste ha en hög noggrannhet och tillförlitlighet.

Idag finns det navigeringshjälpmedel som ger möjlighet till en väldigt noggrann navigering, som det globala satellit navigeringssystemet, eller Global Positioning System (GPS) på engelska. Det systemet använder satelliter som kretsar runt jorden för att bestämma sin egen position. Ett problem som dessa har är att tillgången till dem inte alltid kan förutsättas beroende på olika omständigheter. Satelliter kanske inte syns från nuvarande position eller är det någon som avsiktligt stör ut dessa signaler. I sådana situationer är ett navigeringssystem som är oberoende av externa hjälpmedel högst önskvärt.

Denna avhandling presenterar olika metoder som, genom att använda information från olika sensorer som en typisk UAV bär med sig, skapar ett navigeringssystem som kan klara de krav som ställs på det. Sensorerna som används är främst bildalstrande sensorer, såsom optisk kamera och radar, men även tröghetssensor som mäter UAV:ens rörelse. Metoderna bygger på att de bildalstrande sensorerna observerar omgivningen UAV:en flyger i, och på så sätt räknar ut hur den har rört sig. Detta löses med ett matematisk ramverk som kallas optimering, där det gäller att hitta den bästa lösningen till det ställda problemet.

En annan fördel sensorer såsom kamera eller radar har är att med hjälp av dessa kan kartor över omgivningen skapas samtidigt som man löser navigeringsproblemet. Denna procedur kallas Simultaneous Localisation and Mapping, eller förkortat SLAM, på engelska och är huvudmetoden som utnyttjas i denna avhandling. De resulterande kartorna kan användas för olika syften, t.ex. att skapa en karta över ett landskap där förändringar av omgivningen kan ha skett som efter en naturkatastrof. De resultat som har erhållits visar att de använda metoderna har en betydande potential att användas i praktiken för att skapa ett robust navigeringssystem.

Acknowledgments

First of all, I would like to thank all the people that make this little thing that I have done possible, Prof. Fredrik Gustafsson, my supervisor, and Dr. Thomas Schön, my co-supervisor, for all the guidance and help during this journey. Also, my thanks go to Prof. Lennart Ljung and people at Saab Aeronautics in Linköping for giving me this opportunity in the first place. All my former and present bosses; Dr. Predrag Pucar, Niklas Ferm, Jonas Palm, Björn Kullberg, Tobias Jansson and research responsible Dr. Gunnar Holmberg. Your support during this time was priceless, and don't worry, soon I'll be back to actually do something useful. Thanks go to the Control Theory group's boss Prof. Svante Gunnarsson for taking such a good care of the group and to all the administrators Ulla Salaneck, Åsa Karmelind and Ninna Stensgård for taking care of all the practicalities.

During this long journey, you can't survive alone, and that is why some other people deserve my thanks. Naturally, people at the Control Theory group with which I, in some way, worked with, Dr. Martin Enqvist, Lic. Roger Larsson, Dr. Christian Lundquist, Dr. David Törnqvist, Dr. Fredrik Gunnarsson, Dr. Carsten Fritsche, Dr. Umut Orguner and especially "my partner in crime" Lic. Martin "Morgan" Skoglund. I would also like to thank people from different parts of Saab, Dr. Patrik Dammert, Dr. Hans Hellsten, Dr. Per-Johan Nordlund, Dr. Ola Härkegård and Dr. Sören Mollander for the help and good discussions. It was a pleasure to cooperate with all of you.

And since "all work and no play makes Jack a dull boy", it is important to have fun besides all the work. Therefore it is nice that a whole bunch of people in the Control Theory group know how to have fun. So thanks to (in order of appearance) Dr. Jonas Callmer, Dr. Karl Granström, Dr. Gustaf Hendeby, Dr. Henrik Ohlsson, Dr. Christian Lundquist, Dr. Christian Lyzell, Dr. Emre Özkan, Dr. Henrik Tidefelt, Dr. Ragnar Wallin, Dr. Daniel Ankelhed, Lic. Patrik Axelsson, Lic. André Carvalho Bittencourt, Niclas Evestedt, Lic. Rikard Felkeborn, Ylva Jung, Lic. Sina "my name is to long to fit on the Amex card" Koshfetrat Pakazad, Lic. Roger Larsson, Jonas Linder, George Mathai, Isak Nielsen, Hanna Nyqvist, Michael Roth, Marek Syldatk, Lic. Niklas Wahlström. And Jonas, Kalle and Christian Lu., thanks for sharing so many nights out both in Sweden and elsewhere.

Special thanks go to people that shared the working room with me during all these years (and actually kept their mental sanity(?!)) Lic. Daniel Petersson, Lic. Fredrik "quiz" Lindsten and Manon Kok. Thanks for making our room a fun and nice place to work in.

All this work would be impossible without an infinite support from my family and friends, thanks for all the support. And Alma, I know that this took toooooo much time but it is soon finished ⌣̈, so thanks for having patience and trust in me during all this time ♡.

Last, but not least, thanks to the LINK-SIC Industry Excellence Center and Saab for financing my studies.

Linköping, September 2013
Zoran Sjanic

Contents

II Publications

Notation

ABBREVIATIONS

Abbreviation	Meaning
3GPP LTE	3rd Generation Partnership Project Long Term Evaluation
ADC	Air Data Computer
BFGS	Broyden-Fletcher-Goldfarb-Shanno (quasi-Newton algorithm)
CARABAS	Coherent All RAdio BAnd Sensing
CML	Constrained Maximum Likelihood (estimate)
CMAP	Constrained *Maximum a Posteriori* (estimate)
DCT	Discrete Cosine Transform (focus measure)
DOF	Degrees Of Freedom
EKF	Extended Kalman Filter
EM	Expectation-Maximisation
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IDP	Inverse Depth Parametrisation
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KF	Kalman Filter
LS	Least-Squares
MAP	*Maximum a Posteriori* (estimate)
MC	Monte Carlo (simulation)
MEMS	Micro Electro-Mechanical Sensors
ML	Maximum Likelihood (estimate)

ABBREVIATIONS

Abbreviation	Meaning
MV	Minimum Variance (estimate)
NLS	Nonlinear Least-Squares
PG	Phase Gradient
RAR	Real Aperture Radar
RMSE	Root Mean Square Error
SAR	Synthetic Aperture Radar
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localisation and Mapping
SML	Sum-Modified-Laplacian (focus measure)
SURF	Speeded Up Robust Features
TG	Tenengrad (focus measure)
UAV	Unmanned Aerial Vehicle
VGA	Video Graphics Array

SYMBOLS AND OPERATORS

Notation	Meaning
x_t	States vector
y_t	Measurements vector
u_t	Known inputs vector
$f_t(x_t, u_t)$	System dynamics
$h_t(x_t, u_t)$	Measurement function
w_t	Process noise
e_t	Measurement noise
Q_t	Process noise variance
R_t	Measurement noise variance or range
θ	Parameter vector
$\hat{x}$	Estimate of x
$\mathcal{N}(\mu, P)$	Gaussian distribution with mean μ and variance P
$x_{0:N}, \mathcal{X}$	Short notation for $\{x_0,\ x_1,\ \ldots,\ x_N\}$
p^X, p^Y, p^Z	Position in Cartesian coordinates
X, Y, Z	Position in Cartesian coordinates (alternative notation)
v^X, v^Y, v^Z	Velocity in $X-$, $Y-$ and $Z-$direction, respectively
a^X, a^Y, a^Z	Acceleration in $X-$, $Y-$ and $Z-$direction, respectively
q_0, q_1, q_2, q_3	Quaternions
R	Rotation matrix
$\omega^X, \omega^Y, \omega^Z$	Angular velocity around $X-$, $Y-$ and $Z-$direction, respectively
I	Optical or (complex-valued) SAR image
$\mathbb{R}$	Set of real numbers
$\mathcal{O}$	Ordo, in order of
$\dot{x}_t$	Time derivative of x_t
$\sim$	Is distributed according to
$\\|x\\|_{P^{-1}}$	$P^{-1}-$weighted norm of x, $\sqrt{x^T P^{-1} x}$
x^*	Complex conjugate of x
x^T	Transpose of a vector or a matrix
$\arg\min_x$	Minimising argument with respect to x
$\arg\max_x$	Maximising argument with respect to x
s. t.	subject to

SYMBOLS AND OPERATORS

Notation	Meaning
$\in$	Is member of
$\notin$	Is not member of
$\mathrm{diag}\{a, b, c\}$	3×3 matrix with elements $a,\ b$ and c on the diagonal and zeros otherwise
$\lvert x \rvert$	Absolute value of x or magnitude of a complex number x
$\mathbf{dim}(x)$	Dimension of x
$\arg\{x\}$	Argument of a complex number x
$\Re\{x\}$	Real part of a complex number x
$\Im\{x\}$	Imaginary part of a complex number x
$A * B$	Convolution of matrices A and B
$\mathrm{Tr}(A)$	Trace of a matrix A
$\det(A)$	Determinant of a matrix A
$\frac{\partial}{\partial x} f(x, y)$	Partial derivative of $f(x, y)$ with respect to x

Part I

Background

1

Introduction

In this introductory chapter a background to the work performed in this thesis and a motivation for it will be presented. Also, a brief introduction to Unmanned Aerial Vehicles is provided. Furthermore, all the contributions the author has been involved in are listed and briefly described. This list is divided between publications that are included in the Part II of the thesis and the ones that are not. The chapter is finalised with the outline of the thesis.

1.1 Background and Motivation

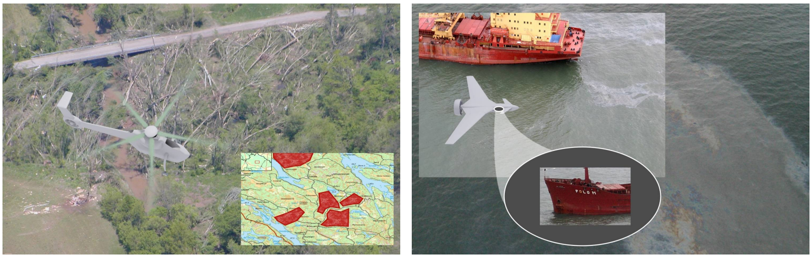
Throughout the history, the exploratory nature of the people has driven them to travel and explore their world. This has led to a need for the art of navigation, i.e., knowing where we are in relation to some known environment. Navigation is used in many different applications, like sailing with boats or flying with aircraft. Many of these applications should not be possible without navigation, like intercontinental flights for example. Navigation also allows for the creation of the systems that can assist human operators in performing different tasks (or even completely remove them), where operating the actual vehicle, like e.g., flying, is the most basic task that the operator can be assisted with. An example of aerial vehicles where a human operator's involvement is partially or completely removed is the Unmanned Aerial Vehicles (UAV). The most obvious thing in the case of UAVs is that the operator is not actually sitting in the UAV, but rather is controlling it remotely from the ground. In this way, the UAVs can carry out missions that might be dangerous and where a risk for a loss of an aircraft is high. Another case might be long reconnaissance missions lasting for 24 hours, where it is impossible to have one pilot or crew to operate for so long, like for Northrop-Grumman Global Hawk, Figure 1.1.

Today, most of the UAVs are used for military purposes, such as reconnaissance of the

Figure 1.1: *Northrop-Grumman Global Hawk fixed wing UAV. Image: Northrop-Grumman website.*

battlefield or enemy's forces. When it comes to the civilian applications, the UAVs are not present in a bigger scale today. However, there is a huge potential for using the UAVs in the civilian sector, such as police or fire departments or environment surveying missions, like assessing the damage after catastrophes like floods or earthquakes. In Figure 1.2 some of these concepts are illustrated. Actually, saying that UAVs are not used in the civilian sector is, from the technical point of view, not entirely correct. Auto-pilots are used in all commercial airplanes for about 95% of the flight time implying that it is actually the "manned" UAVs that are used every day. The potential is reinforced by the fact that many of the tasks that the UAVs might perform are usually long, boring or, as mentioned, dangerous. Also, an advantage of using UAVs over an manned aircraft is that an operator should be able to concentrate on the mission instead of concentrating on the basic tasks, like flying an aircraft for example. It is therefore crucial to build UAVs with as much autonomy and support in the basic functions as possible. This implies that onboard systems must be able to handle different situations, that may arise during the mission, without the constant monitoring of the operator.

One of the most important parts of the total system to enable this is the navigation system of the UAV. The navigation system provides all the important quantities needed for the UAV to operate autonomously. In order to be able to do this with high reliability, the navigation system must have high accuracy and robustness. This could be realised with the navigation aids that give high quality navigation performance, like Global Navigation Satellite Systems (GNSS) of which Nav-Star Global Positioning System (GPS) (GPS, 2008) is the most famous one. However, the availability of these external navigation aids cannot be assumed in many applications. This can be caused by different things, like occlusions rendering the reception of the satellite signals impossible or even mali-



***(a)** Disaster damage assessing.* ***(b)** Oil spill identification.*

***(c)** Power lines surveying.* ***(d)** Riots monitoring.*

***Figure 1.2:** Some of the examples of UAV usage areas in the civilian sector. Images: © Saab AB.*

cious jamming of the signals by opposing military force. In all these cases, some kind of navigation system that is independent of the external systems, like satellites, is highly desirable. Such problem can, in general, be solved with expensive and usually big and heavy components, e.g., high-grade Inertial Navigation Systems (INS), like it is done in commercial or fighter aircraft. Since another reason for using UAVs is that they should be cheaper and smaller than manned aircraft, this is not a very good option. An example of this is Saab V-200 Skeldar rotor UAV, Figure 1.3, which weighs 200 kg and has a length of approximately 4 m. In this case the less the components weigh the more fuel can be carried, which in turn gives longer mission time. The conclusion is that tactical performance, like endurance, puts the requirements on the choice of the components and the design of the UAV. The end result is that the used equipment, including the navigation system, must be cheap (in this context cheap means a couple of thousand USD class), small and lightweight. These constraints are often in conflict with the previously stated requirements on accuracy and robustness.

In the cases where UAVs are used for surveillance and reconnaissance missions, it is important to create an overview of the environment where the operation is carried out. For example, during a flood the configuration of the landscape changes and it is important to obtain the latest information about the status of the environment. In the more military oriented applications, an UAV can be used to create an up-to-date map of the interesting area for tactical evaluation of the mission or if the prior map was not available. Furthermore, the maps can also be used for navigation purposes, if the UAV carries sensors that can

Figure 1.3: *Saab V-200 Skeldar rotor UAV. Image: © Saab AB. Photographer: Stefan Kalm.*

observe the environment and in that way relate these observations to the maps, serving as an alternative to the high-precision external aids, like GPS.

The work in this thesis is motivated by the above-mentioned aspects of the navigation systems in UAVs. Due to restrictions on weight and size of the equipment in small UAVs, high-performing inertial navigation systems, which are both large in size and heavy, are not a feasible option. Less accurate inertial navigation systems are prone to drift in navigation parameters making them dependent on external aids. Fortunately, UAVs are often equipped with imaging sensors like optical camera or imaging radar, usually due to the mission requirements, allowing us to use these and support the navigation system. Since these sensors primarily observe the surrounding environment, it is also possible to actually create the map of this environment at the same time as localising the UAV. This procedure is known as Simultaneous Localisation and Mapping, or SLAM (Durrant-Whyte and Bailey, 2006; Bailey and Durrant-Whyte, 2006), and will be used as an unified framework in order to perform stable navigation and mapping of the environment. These maps can be used for different purposes, ranging from tactical evaluation of the terrain to creating high-fidelity maps or even three-dimensional models of the previously uncharted areas. They can in turn be later used to facilitate the navigation for other vehicles.

1.2 Contributions

1.2.1 Included Publications

Articles included in this thesis are listed below in chronological order together with a short description and contribution of each paper.

> Z. Sjanic and F. Gustafsson. Navigation and SAR Auto-focusing Based on the Phase Gradient Approach. In *Proceedings of 14th International Confer-*

> *ence on Information Fusion*, Chicago, USA, July 2011.

This article, **Paper C**, describes a method for iterative trajectory refinement and UHF SAR auto-focusing based on the filtering framework. Inertial measurements (accelerations) are used together with an estimate of the range time derivative, or phase gradient, in an Extended Kalman Filter (EKF) to obtain a stable estimate of the position. The main contribution is formulation and incorporation of the phase (or range) derivative into the filtering framework enabling sequential solution of the navigation and auto-focusing problem. Furthermore, the importance of using the exact expression for the range and its derivative in UHF SAR case is also shown.

> Z. Sjanic, M. A. Skoglund, T. B. Schön, and F. Gustafsson. A Nonlinear Least-Squares Approach to the SLAM Problem. In *Proceedings of 18th IFAC World Congress*, Milano, Italy, August/Septemeber 2011.

In this paper, which is **Paper D**, a *Maximum a posteriori* (MAP) batch formulation of the inertial/visual SLAM is proposed resulting in a nonlinear least squares optimisation problem. The inherent structure and sparsity of the resulting NLS problem is utilised to efficiently obtain the metrically correct solution to the 6-Degrees-of-Freedom (6-DoF) estimate of the platform's states, i.e., position, velocity and orientation, and a three dimensional map of the environment. Also, for the parametrisation of the landmark location an Inverse Depth Parametrisation (IDP) is used which also gives the special structure to the NLS problem. The benefit of IDP, which uses six parameters, over the regular three-parameter parametrisation is faster convergence of the landmark depth if an EKF is used to estimate the states and the map. Here the EKF is applied to the initialisation procedure of the estimated parameters and to obtain the data association.

> Z. Sjanic and F. Gustafsson. Simultaneous Navigation and Synthetic Aperture Radar Focusing. *Provisionally accepted to IEEE Transactions on Aerospace and Electronic Systems*, August 2013a.

An optimisation based method for simultaneous navigation and SAR auto-focusing is proposed in this contribution, **Paper A**, resulting in basically a Maximum Likelihood (ML) formulation. Here, the accelerations sensors are used together with SAR images, utilising image focus measures, image entropy in particular, to obtain the likelihood function. This likelihood is quite a complicated function of the optimisation variables, which are here the platform's states, making the direct gradient calculation difficult. However, as an important part of the article's contribution, an efficient way of calculating this gradient is devised, which is based on the chain rule for differentiation. The solution is made even more efficient by using quasi-Newton method where the Hessian matrix is estimated from gradient information during the iterations.

> Z. Sjanic and F. Gustafsson. Navigation and SAR focusing with Map Aiding. *Submitted to IEEE Transactions on Aerospace and Electronic Systems*, June 2013b.

This paper, **Paper B**, devises a method for solving the navigation and SAR focusing problem with an aid from prior information in form of optical maps. The basic idea is based on the image matching approach, called Chamfer matching, which is, in this contribution, reformulated in the nonlinear least squares optimisation framework. With this approach it is possible to solve for both the global position and direction of flight, since the map has known global coordinates, and for the flight trajectory, since the assumption is that the most focused image will give the matching cost with the smallest value.

> M. A. Skoglund, Z. Sjanic, and F. Gustafsson. Initialisation and Estimation Methods for Batch Optimisation of Inertial/Visual SLAM. *Submitted to IEEE Transactions on Robotics*, September 2013.

Inertial/visual SLAM problem formulation in optimisation form, as most optimisation problems, requires an initial value of the optimisation parameters. The nonlinear and non-convex nature of the posed problem may cause suboptimal solutions due to bad initialisation and consequently convergence to a local minimum. In Paper D EKF is used to initialise the parameters and data association, but that solution is not feasible for large data sets due to the cubic complexity for calculating the matrix inverse involved in the EKF iteration. This article, **Paper E**, contributes with a multi-stage procedure for the initialisation of the nonlinear inertial/visual SLAM optimisation problem. The approach is based on both pure visual methods, like image based rotation estimation and appearance based data association, and combined inertial/visual methods, where an almost linear formulation of the problem is utilised to efficiently obtain an initial solution and data association. It is also demonstrated that this estimate leads to a better initial values than straight forward initialisation with measurements only.

> Z. Sjanic, M. A. Skoglund, and F. Gustafsson. Expectation-Maximisation Maximum Likelihood Estimation for Inertial/Visual SLAM. *Submitted to IEEE Transactions on Robotics*, September 2013b.

Here, **Paper F**, an Expectation-Maximisation (EM) approach to the ML formulation of the SLAM problem is proposed. The EM framework is usually used when the ML problem is difficult to solve by introducing a new set of unknowns, so called hidden variables. The problem is then split into two subproblems, Expectation step (E-step) where hidden variables are estimated conditioned on the old parameters and Maximisation step (M-step) where parameters are estimated based on the hidden variables from the E-step by solving an optimisation problem. This framework is highly applicable to inertial/visual SLAM where the platform's states are seen as hidden variables and the map is represented as parameters. In this particular case, the E-step is approximately, but efficiently, solved with Extended Rauch-Tung-Striebel Smoother and the M-step is, also efficiently, solved with quasi-Newton method. Since the smoother step is computationally cheap and the optimisation problem in the M-step has fewer variables than the full ML formulation, the total computational time can be lower for the big problems.

> Z. Sjanic, F. Gunnarsson, C. Fritsche, and F. Gustafsson. Cellular Network Non-Line-of-Sight Reflector Localisation Based on Synthetic Aperture Radar Methods. *Submitted to IEEE Transactions on Antennas and Propagation*, September 2013a.

This short paper, **Paper G**, contributes with an application of a nonparametric multi-static SAR technique to cellular network data. The data consists of measured range to base stations which contains both line-of-sight and non-line-of-sight signals between the phone and the base stations, the latter ones caused by the signal reflections from the buildings in the urban environment. This data is used to estimate a map (or a multi-static SAR image) of the strongest reflectors. Furthermore, with this map, it is possible to find reflector-trajectory correspondence which can be used for outlier rejection in a cellular network localisation application.

1.2.2 Other Publications

Other articles which are not included in the thesis are listed below. Also here, the description and contribution are described.

> R. Larsson, Z. Sjanic, M. Enqvist, and L. Ljung. Direct Prediction-error Identification of Unstable Nonlinear Systems Applied to Flight Test Data. In *Proceedings of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 2009.

In this contribution a system identification method for unstable and nonlinear systems based on direct prediction-error approach is devised. The predictor is based on the nonlinear state space description of the system and calculated using EKF. The methods are demonstrated on estimation of aerodynamic parameters for the fighter aircraft.

> Z. Sjanic and F. Gustafsson. Simultaneous Navigation and SAR Auto-focusing. In *Proceedings of 13th International Conference on Information Fusion*, Edinburgh, UK, July 2010.

This article introduces an early concept of solving navigation and SAR focusing problem simultaneously and in the optimisation framework. The ideas presented here can be seen as a predecessor to the much more thorough work done in Paper A.

> Z. Sjanic and F. Gustafsson. Fusion of Information from SAR and Optical Map Images for Aided Navigation. In *Proceedings of 15th International Conference on Information Fusion*, Singapore, July 2012.

In this paper a method for matching of SAR images and optical map in order to aid the navigation is proposed. In this way a global position and direction of flight can be estimated. The method is based on the nonlinear least squares optimisation approach which also allows for the estimate of the parameter covariance. This contribution is the basis for Paper B.

1.3 Thesis outline

The thesis is divided into two parts, theoretical and subject background in the first part and edited versions of published papers in the second part. Chapter 1, this chapter, introduces the main problem and the motivation for solving it. It also briefly describes UAVs and lists all the included publications in Part II of the thesis and scientific contributions that are included in these articles. Chapter 2 briefly introduces the estimation theory that is used. UAV navigation systems and all the utilised sensors, i.e., inertial sensors, optical camera and SAR, and their properties are described in Chapter 3. Finally, the first part is concluded with Chapter 4 where some concluding remarks are discussed and suggestions about the future work are given.

2

Estimation and Sensor Fusion Framework

In this chapter an overview the sensor fusion framework with short estimation theory and Simultaneous Localisation and Mapping (SLAM) framework is described, where only the main ideas and concepts are mentioned.

2.1 Estimation Theory for Dynamical Systems

The primary task of the sensor fusion function is to utilise the information from all the available sensors and fuse it together in order to improve certain properties of the output entities of interest. It can be that sensors have noisy measurements, imperfections due to biases or low signal fidelity. In some cases, the sensors do not measure quantities that we want directly, and they must be calculated or estimated in some way. The general approach to this problem is called estimation, where sensor fusion is one particular instance. In a way, we can say that the navigation system is an estimator of the navigational quantities, which in turn describe the motion state of the UAV, given some measurements of this motion. Systems in motion are usually denoted dynamical systems, which means that an input to such a system at a particular time will affect future behavior of the system. The opposite case is the static system where the input at the particular time affects the output at that time only.

In this framework, a particularly useful description of the dynamical systems is the state space description, usually in the form

$$x_{t+1} = f_t(x_t, u_t, w_t) \tag{2.1a}$$

$$y_t = h_t(x_t, u_t, e_t) \tag{2.1b}$$

where x_t are the states of the system, usually the quantities that need to be estimated, u_t are the known inputs, w_t is the system noise, y_t are the measurements and e_t is the measurement noise. The function $f_t(\,\cdot\,)$ describes the dynamics of the system and $h_t(\,\cdot\,)$ is a function that relates the measurements and the states of the system. Note that these functions may vary in time. This is a rather general model of a system and one that is often used is

$$x_{t+1} = f_t(x_t, w_t) \tag{2.2a}$$

$$y_t = h_t(x_t) + e_t \tag{2.2b}$$

where noise term in the measurement equation appears in an additive fashion and the known input u_t is omitted. This poses no practical problems since, in most cases, the additive measurement noise is a plausible model and a known input can be modelled through the time dependency of f_t and h_t. In the rest of this thesis u_t will mostly be omitted for notational convenience, and be included only when it is necessary. The descriptions of the dynamics and measurements (2.1) and (2.2) are both in discrete time, which is suitable for implementation in computers and because most of the modern sensors deliver data in sampled form. Usually, the system dynamics is dependent on the moving platform, so it can be fixed for each application. In our case the dynamics of the system will be modelled as aircraft dynamics which will be described below.

Another, and a bit more general, way of describing the system (2.2) is in the form of conditional probability density functions, or pdf, for state time transition and measurements

$$x_{t+1} \sim p(x_{t+1}|x_t) \tag{2.3a}$$

$$y_t \sim p(y_t|x_t) \tag{2.3b}$$

Equation (2.3b) is also known as likelihood function. From the system perspective these two descriptions are equivalent. The model above are referred to as Markov process', i.e., the state at time t is only dependent of the state at time $t-1$. In a similar way the measurement at time t is conditionally independent of the states in all times except the state at time t. As an example of model (2.3), take equation (2.2b) and suppose that the measurement noise, e_t, has Gaussian distribution with zero mean and variance R_t, $e_t \sim \mathcal{N}(0, R_t)$. This will, for one time instance, yield

$$p(y_t|x_t) = p_{e_t}(y_t - h(x_t)) = \frac{1}{\det\{2\pi R_t\}^{1/2}} e^{-\frac{1}{2}(y_t - h_t(x_t))^T R_t^{-1}(y_t - h_t(x_t))} \tag{2.4}$$

The estimation problem can now be posed as determining the states $x_{0:N} = \{x_0, \ldots, x_N\}$ given all the measurements $y_{1:N} = \{y_1, \ldots, y_N\}$ and the system model (2.2) or (2.3).

To define the solution to this problem, we can start with the simple method called *Maximum Likelihood* (ML) (Fisher, 1912) which is a well known statistical method where a model parameters, usually called θ, are estimated from a set of measurements, assumed to be generated from this model as

$$y_t \sim \varphi(y_t|\theta) \tag{2.5}$$

where $\varphi(y_t|\theta)$ represents the likelihood of the measurements parametrised with θ. The

ML solution is obtained by solving the maximisation problem

$$\hat{\theta}^{\mathrm{ML}} = \arg\max_{\theta} \varphi(y_{1:N}|\theta) \tag{2.6}$$

Using (2.3b) and identifying $\varphi = p$ and $\theta = x_{0:N}$, ML formulation is obtained and the method can be applied to the state estimation problem as

$$\hat{x}_{0:N}^{\mathrm{ML}} = \arg\max_{x_{0:N}} p(y_{1:N}|x_{0:N}) \tag{2.7}$$

where the joint likelihood for all the measurements, $p(y_{1:N}|x_{0:N})$, is used. In many practical applications the measurement noise is assumed to be independent in time, which is quite reasonable assumption in these cases. This implies that the joint likelihood can be written as

$$p(y_{1:N}|x_{0:N}) = p(x_0)\prod_{t=1}^{N} p(y_t|x_t) \tag{2.8}$$

where $p(x_0)$ is so called prior of the states, which acts as a belief about the state values at time 0, i.e., before any measurements have been obtained.

The ML approach above does not utilise the state transition pdf, (2.3a), which can actually be useful information. In order to include state dynamics, we can use a conceptually slightly different approach than ML. In ML the true parameter θ^0 is considered as deterministic. If the parameter is seen as stochastic instead, i.e., its properties can be described with some pdf, $p(\theta)$, in the same way as for the measurements which are stochastic, we can "augment" the likelihood with this pdf simply by multiplying these together as $p(y|x)p(x)$. This is one way of incorporating the prior knowledge of the states and with help from the Bayes' rule (Bayes, 1763)

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{2.9}$$

a so called posterior pdf of the states given measurements is obtained. Now $x_{0:N}$ that maximises this pdf can be obtained leading to so called *Maximum a Posteriori* estimate (MAP)

$$\hat{x}_{0:N}^{\mathrm{MAP}} = \arg\max_{x_{0:N}} p(x_{0:N}|y_{1:N}) = \arg\max_{x_{0:N}} p(y_{1:N}|x_{0:N})p(x_{0:N}) \tag{2.10}$$

Here, $p(y_{1:N})$ is omitted since it does not depend on $x_{0:N}$ and does not influence the maximisation solution. This formulation together with Markov and measurement model independency assumptions leads to the following MAP optimisation

$$\hat{x}_{0:N}^{\mathrm{MAP}} = \arg\max_{x_{0:N}} p(x_0)\prod_{t=1}^{N} p(y_t|x_t)p(x_t|x_{t-1}) \tag{2.11}$$

where $p(x_0)$ is as before the prior of the states before any measurements have arrived. Notice that if state prior is chosen as so called flat, i.e., $p(x) =$ constant, then the solution to MAP and ML will coincide. However, conceptually in MAP the parameters (or states) are still seen as stochastic while in ML they are not.

One powerful feature of the optimisation formulations (2.7) and (2.10) is that any kind of constraints on the states can be added, for example if states represent some physical entities which must be in certain bounds. Another example is if the prior distribution is degenerated, i.e., some of the states are deterministic. In these cases both maximisation problems can be augmented with constraints leading to *Constrained Maximum Likelihood*

$$\hat{x}_{0:N}^{\text{CML}} = \arg\max_{x_{0:N}} \; p(y_{1:N}|x_{0:N}) \qquad \text{s. t. } x_{0:N} \in \mathbb{X} \tag{2.12}$$

or *Constrained Maximum a Posteriori* formulation

$$\hat{x}_{0:N}^{\text{CMAP}} = \arg\max_{x_{0:N}} \; p(x_{0:N}|y_{1:N}) \qquad \text{s. t. } x_{0:N} \in \mathbb{X} \tag{2.13}$$

where $\mathbb{X}$ is a constraint set for the states. This formulation gives a great deal of flexibility in the problem description, allowing us to pose the problem in a form that is easy to solve. In most cases it will be assumed that involved probability functions have Gaussian form, i.e., the state and measurement noises are Gaussian. In that case it is appropriate to optimise a negative logarithm of the criterion resulting in a minimisation problem. Since the negative logarithm is a monotonously decreasing function the solutions will be the same.

One important special case of the system and measurement models is the case where both the dynamic and the measurement models are linear, the prior has Gaussian distribution and the process and the measurement noise are Gaussian and white, which gives

$$f_t(x_t, w_t) = F_t x_t + G_t w_t \tag{2.14a}$$

$$h_t(x_t) = H_t x_t \tag{2.14b}$$

$$x_0 \sim \mathcal{N}(\bar{x}, P_0), \; w_t \sim \mathcal{N}(0, Q_t), \; e_t \sim \mathcal{N}(0, R_t) \tag{2.14c}$$

Here, P_0 is the initial state covariance matrix, i.e., the uncertainty of x_0 and $\bar{x}$ is its prior value. The formulation of (2.11) in this case becomes (with negative logarithm applied)

$$\hat{x}_{0:N}^{\text{MAP}} = \arg\min_{x_{0:N}} \|x_0 - \bar{x}\|^2_{P_0^{-1}} + \sum_{t=1}^{N} \|y_t - H_t x_t\|^2_{R_t^{-1}} + \|x_t - F_{t-1}x_{t-1}\|^2_{(G_t Q_t G_t^T)^{-1}} \tag{2.15}$$

which is a weighted least squares problem. This example also shows a possible degeneracy issue in the problem formulation. To pose this problem it is required that the matrix $G_t Q_t G_t^T$ is actually invertible, and that is not always the case. In such problems, the formulations (2.12) and (2.13) are really useful.

All the methods presented above assume that all the measurements are available at the estimation time. If that is the case, the solution $\hat{x}_{0:N}$ is called smoothed estimate. Smoothing in this context is defined as an estimate in certain time that depends on the past and future measurements. If that is not the case, i.e., if estimate depends only on past measurements, we obtain a filtered estimate. One important advantage of the filtering is that it can be implemented in a sequential manner. The general sequential solution to (2.11) is given by recursive Bayesian filtering. The posterior distribution for each time instant can be obtained as, see e.g., Gustafsson (2010) for complete derivation,

$$p(x_1|y_0) = p(x_0) \qquad \text{(Initialisation)} \tag{2.16a}$$

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \tag{2.16b}$$

$$p(y_t|y_{1:t-1}) = \int_{\mathbb{R}^n} p(y_t|x_t)p(x_t|y_{1:t-1})\,\mathrm{d}x_t \tag{2.16c}$$

$$p(x_t|y_{1:t-1}) = \int_{\mathbb{R}^n} p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})\,\mathrm{d}x_{t-1} \tag{2.16d}$$

The procedure (2.16) defines a very general nonlinear filter that can be applied to a large class of dynamic and measurements models. However, the closed form solution exists only in a few cases. One such case is (2.14) and it can be shown that procedure (2.16) becomes the Kalman Filter (KF) (Kalman, 1960) which is both the Minimum Variance (MV) and the Best Linear Unbiased Estimator (BLUE). If all the measurements are available at the estimation time, i.e., smoothing solution is possible, the solution to (2.15) can be obtained with the Kalman smoother instead of solving it directly. If the model is not linear, some approximate solutions to solve (2.16) must be applied, where Extended Kalman Filter (EKF) (Kailath et al., 2000) and Particle Filter (PF) (Gordon et al., 1993) are the most common approaches.

2.2 Simultaneous Localisation and Mapping

The main idea of Simultaneous Localisation and Mapping (SLAM) is to estimate a map of the surrounding environment from a moving platform, while simultaneously localising the platform (Durrant-Whyte and Bailey, 2006; Bailey and Durrant-Whyte, 2006). Localisation in this context includes estimation of navigation states, which includes position, velocity and attitude. However depending on the sensor setup, some of the states cannot be estimated. This will be handled from case to case. The main prerequisite for mapping is that some of the available sensors actually observe the environment. Typical sensors that can do that are laser ranging sensors, radars and different kinds of cameras, e.g., infra-red or visible light cameras. In this work, radar (and in particular imaging radar or SAR) and optical camera will be used as the environment observing sensors. The maps generated by different sensors will also differ in their appearance. The advantage is that different maps may be used for different purposes depending on situation at hand. For example a SAR image can be seen as a map of the imaged environment where certain features of the environment are accentuated. In a similar way, an optical camera will produce another kind of map which might be useful in a different situation. How these maps are used is a question of tactical behaviour and will not be discussed in more detail here.

2.2.1 SLAM System Models

To put the SLAM problem in the sensor fusion framework described in Section 2.1 we have to define the dynamics and the measurement models or equivalently the state transition pdf and the likelihood function. To do that we need to introduce two coordinate systems which are important for the navigation. The first one is the local navigation frame, also called world frame, and it has a fixed position in the world. The second one is the body frame, which is aligned with the moving platform, and moves and rotates together with it. The navigation states basically describe body frame's position, velocity, acceleration and orientation relative the navigation frame. In SLAM the navigation frame is usually free to be placed arbitrarily, since most of the measurements are relative. However, the estimation results from SLAM can be combined with the prior information, like global maps, in order to fix the origin of the navigation frame to some global coordinate system.

In most cases the dynamics of the platform can be described by simple model using basic equations of motion. For linear motion, where states normally are position, linear velocity and acceleration, a double integrator model is used as (Farrell and Barth, 1999)

$$\dot{p}_t = v_t \tag{2.17a}$$

$$\dot{v}_t = a_t \tag{2.17b}$$

$$\dot{a}_t = w_t^a \tag{2.17c}$$

where p_t, is the position of the platform expressed in the local navigation frame, v_t is its velocity and a_t is the acceleration. w_t^a is the unknown input to the system, here modelled as a jerk. The dimension of the states, i.e., p, v and a as well as the noise w^a, will depend on the application, but will usually be 2 or 3. This is continuous time description of the system, but for the implementation we need a discrete time description. The model above can be discretised exactly if a zero order hold assumption is made. In that case the discrete time model is

$$p_{t+1} = p_t + T_s v_t + \frac{T_s^2}{2} a_t + \frac{T_s^3}{6} w_t^a \tag{2.18a}$$

$$v_{t+1} = v_t + T_s a_t + \frac{T_s^2}{2} w_t^a \tag{2.18b}$$

$$a_{t+1} = a_t + T_s w_t^a \tag{2.18c}$$

where T_s is the sampling time. This model can be compactly written in a matrix form as

$$x_{t+1} = F x_t + G w_t^a \tag{2.19a}$$

$$x_t = \begin{bmatrix} p_t \\ v_t \\ a_t \end{bmatrix}, \quad F = \begin{bmatrix} I & T_s I & \frac{T_s^2}{2} I \\ 0 & I & T_s I \\ 0 & 0 & I \end{bmatrix}, \quad G = \begin{bmatrix} \frac{T_s^3}{6} I \\ \frac{T_s^2}{2} I \\ T_s I \end{bmatrix} \tag{2.19b}$$

where I and 0 are the identity and zeros matrices of appropriate dimensions.

The rotational dynamics in $\mathbb{R}^3$ can be parametrised with so called Euler angles, denoted roll, ϕ, pitch, θ, and yaw, ψ, see Figure 2.1. These angles describe the rotation of one three-dimensional coordinate system relative another. In this case the coordinate systems are the local navigation and platform's body frames. The dynamics of these angles is

$$\dot{\phi}_t = \omega_t^X + \sin(\phi_t)\tan(\theta_t)\omega_t^Y - \cos(\phi_t)\tan(\theta_t)\omega_t^Z \tag{2.20a}$$

$$\dot{\theta}_t = \cos(\phi_t)\omega_t^Y + \sin(\phi_t)\omega_t^Z \tag{2.20b}$$

$$\dot{\psi}_t = -\frac{\sin(\phi_t)}{\cos(\theta_t)}\omega_t^Y + \frac{\cos(\phi_t)}{\cos(\theta_t)}\omega_t^Z \tag{2.20c}$$

$$\dot{\omega}_t^X = w_t^{\omega X} \tag{2.20d}$$

$$\dot{\omega}_t^Y = w_t^{\omega Y} \tag{2.20e}$$

$$\dot{\omega}_t^Z = w_t^{\omega Z} \tag{2.20f}$$

and $\omega_t^{\{X,Y,Z\}}$ are the angular velocities and $w_t^{\{X,Y,Z\}}$ are the unknown input terms, here modelled as angular accelerations. In contrast to the linear motion dynamics (2.19), this one is nonlinear. Although some nonlinear dynamics is possible to discretise exactly, it is not the case here and some approximate method, like Euler sampling, must be used. It can immediately be seen that there exists a singularity in the dynamics for $\theta_t = \pm\pi/2$. In this case the roll and the yaw angles are undefined. Fortunately, there is another representation of the rotations in $\mathbb{R}^3$, with the help of the quaternions, that does not suffer from these limitations (Kuipers, 1999; Shuster, 1993).

Quaternions are defined in four dimensional space as $q = [q_0\ q_1\ q_2\ q_3]^T$, $q_i \in \mathbb{R}$. To represent the rotation in $\mathbb{R}^3$, q is constrained to the unit sphere, i.e., $q^T q = 1$. The transformation from Euler angles to quaternions is defined as

$$q_0 = \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \tag{2.21a}$$

$$q_1 = \sin(\phi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \tag{2.21b}$$

$$q_2 = \cos(\phi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \tag{2.21c}$$

$$q_3 = \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\phi/2)\sin(\theta/2)\cos(\psi/2) \tag{2.21d}$$

and from quaternions to Euler angles as

$$\phi = \text{atan2}(2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)) \tag{2.22a}$$

$$\theta = \text{arcsin}(2(q_0q_2 - q_1q_3)) \tag{2.22b}$$

$$\psi = \text{atan2}(2(q_0q_3 + q_1q_2), 1 - 2(q_2^2 + q_3^2)) \tag{2.22c}$$

where atan2(· , ·) is $\arctan$ function defined for all four quadrants. The dynamics of the quaternions can be expressed as

$$\dot{q}_t = \frac{1}{2}\underbrace{\begin{bmatrix} 0 & -\omega_t^X & -\omega_t^Y & -\omega_t^Z \\ \omega_t^X & 0 & \omega_t^Z & -\omega_t^Y \\ \omega_t^Y & -\omega_t^Z & 0 & \omega_t^X \\ \omega_t^Z & \omega_t^Y & -\omega_t^X & 0 \end{bmatrix}}_{S(\omega_t)} q_t = \frac{1}{2}\underbrace{\begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}}_{\tilde{S}(q_t)} \omega_t \tag{2.23}$$

We see that the dynamics is still non-linear, but the non-linearities are much simpler than for the Euler angles (it is actually bilinear if angular rates are considered as input). A nice property of the quaternion dynamics is that, if we assume zero order hold, i.e., constant

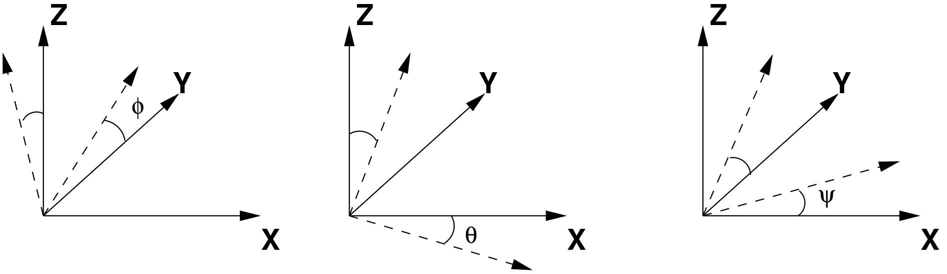


Figure 2.1: *Illustration of Euler angles, from left to right: roll angle (ϕ, rotation around x-axis), pitch angle (θ, rotation around y-axis) and yaw angle (ψ, rotation around z-axis). Solid line coordinate system represents local navigation frame and dashed coordinate system is the body frame.*

angular rate between sampling times, the discrete time model is

$$q_{t+1} = \exp\left(\frac{T_s}{2} S(\omega_t)\right) q_t \tag{2.24}$$

which also preserves the norm of the quaternion. Note that the second form of the dynamics, the one with $\tilde{S}(q)$, cannot be discretised in this way and some other strategy must be applied, e.g., Euler sampling.

Given the Euler angles or quaternions it is possible to calculate any vector coordinates in different systems. The simplest way to perform that is via a rotation matrix, R. As a function of Euler angles, the rotation matrix is defined as three consecutive rotations around each of the three axis, as shown in Figure 2.1. Since the order of rotations is important, different rotation matrices can be obtained for different orders given the same Euler angles. The standard order in aeronautics is $z-y-x$, i.e., rotation is first performed around z-axis, then around y-axis and finally around x-axis. The total rotation can then be composed of three consecutive rotations as (with shorthand notation $s\cdot = \sin(\,\cdot\,)$ and $c\cdot = \cos(\,\cdot\,)$)

$$\begin{aligned}\mathsf{R}(\phi,\theta,\psi) = &\begin{bmatrix}1 & 0 & 0\\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi\end{bmatrix}\begin{bmatrix}c_\theta & 0 & -s_\theta\\ 0 & 1 & 0\\ s_\theta & 0 & c_\theta\end{bmatrix}\begin{bmatrix}c_\psi & s_\psi & 0\\ -s_\psi & c_\psi & 0\\ 0 & 0 & 1\end{bmatrix} = \\ &\begin{bmatrix}c_\theta c_\psi & c_\theta s_\psi & -s_\theta\\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta\\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta\end{bmatrix}\end{aligned} \tag{2.25a}$$

and as a function of quaternions, the corresponding matrix is

$$\mathsf{R}(q) = \begin{bmatrix}q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2+q_0q_3) & 2(-q_0q_2+q_1q_3)\\ 2(q_1q_2-q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3+q_0q_1)\\ 2(q_1q_3+q_0q_2) & 2(-q_0q_1+q_2q_3) & q_0^2-q_1^2-q_2^2+q_3^2\end{bmatrix} \tag{2.26}$$

These rotation matrices express how to rotate a vector given in navigation frame to a vector given in body frame. To emphasise this relationship, if necessary, superscript will be used to denote which frame vector is expressed in. For example the gravity vector, g, always has coordinates $[0\ 0\ -9.81]$ in the local navigation frame, which we can write as g^n. If we want to find its representation in a body frame, g^b, we simply multiply the rotation matrix, R^{bn}, with g^n

$$g^b = \mathsf{R}^{bn}\, g^n \tag{2.27}$$

Superscript bn denotes the rotation from navigation to body frame. To obtain the inverse rotation, body to navigation frame, nb, an orthonormality property of the rotation matrix is used to obtain

$$\mathsf{R}^{nb} = (\mathsf{R}^{bn})^{-1} = (\mathsf{R}^{bn})^T \tag{2.28}$$

i.e., the transpose of the matrix denotes the inverse rotation. In this thesis only R will denote R^{bn} and R^T will denote R^{nb} unless the frame superscript is absolutely needed.

The next thing that is needed is a measurement or likelihood function. Its task is to describe how the information from the used sensors, i.e., the measurements, is related to the platform's states and the environment that we want to map. It usually has a form as in (2.1b), (2.2b) or (2.3b), but generally any function that relates sensor measurements, platform's states and the surrounding environment can be used in the ML and MAP optimisation criteria. Basically, any function of the form

$$F(x_{0:N}, \mathsf{M}, y_{1:N}) \tag{2.29}$$

can be used. Here, the map, M, denotes the representation of the environment that should be mapped. Also, this function will in most cases correspond to negative log-likelihood, implying that it should attain minimum value for the correct states and map parameters. This representation can be quite different depending on the used sensors, for example a SAR will produce an image which will be considered as a map of the environment. In the next chapter, Chapter 3, where the used sensors are described in detail, concrete examples of these functions will be given. Note also that some of the sensors will not measure the environment, and in that case, the function will simply not depend on the map.

3

Navigation System and Sensors

In this chapter an overview of the typical UAV navigation system and the sensors used in the estimation methods introduced in Chapter 2 is done. The utilised sensors and their properties are described as well as how they are incorporated into the sensor fusion framework.

3.1 UAV Navigation System

The definition of the navigation system in a UAV is the system which measures and delivers all interesting navigational quantities to the users of these data. Navigational quantities of interest are usually

- 3D position in some global coordinate system
- 3D velocity
- 3D acceleration
- attitude (pitch, roll and heading)
- 3D angular rate

which can either be directly measured or estimated in some way. In a basic configuration the UAV navigation system can look like the one in Figure 3.1. Basic sensors used here are Inertial Measurement Unit (IMU), Global Positioning System (GPS) and Air Data Computer (ADC). IMU senses specific forces and angular moments that platform exhibits and measures these giving accelerations and angular rates (or velocities) of the platform. In principle, the position and velocity could be calculated directly by integration of the

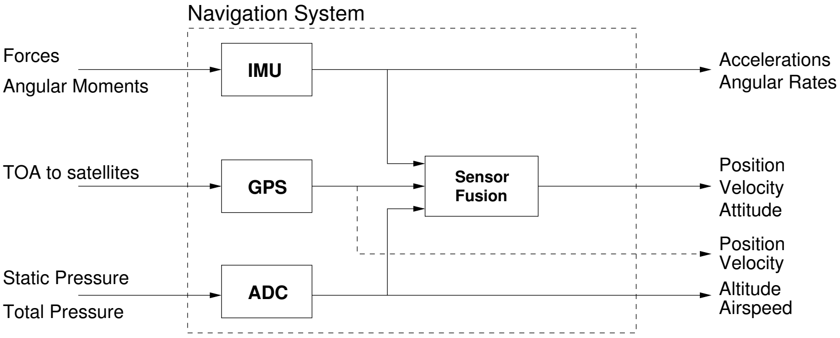


Figure 3.1: *Overview of the typical UAV navigation system. IMU is Inertial Measurement Unit, GPS is Global Positioning System and ADC is Air Data Computer. Dashed output from the GPS is used to indicate that raw GPS position could be used, but it is usually the fused one that is used in practice.*

acceleration based on the solution of (2.17)

$$p_t = p_0 + \int_0^t v_s \, \mathrm{d}s \tag{3.1a}$$

$$v_t = v_0 + \int_0^t a_s \, \mathrm{d}s \tag{3.1b}$$

where p_t is the position, v_t is the velocity and a_t is the acceleration of the platform expressed in some local navigation frame. Since the accelerations are measured relative the so called body frame which rotates together with the platform, a rotation with the attitude must be performed, $a_t = \mathsf{R}^{nb}\, a_t^m$, where superscript m denotes the measurements and R is the rotation matrix between body and navigation frame. The rotation is directly dependent on the attitude of the platform which can also be integrated from the angular velocities in a similar way. This approach can be applied when the sensor accuracy in terms of bias and noise is high, otherwise the velocity and position will drift over time rendering the system unusable. The solution is to use other sensors which will estimate the bias and give more stable estimate. More detailed description of IMU is presented in Section 3.2.

Global Positioning System (GPS) is a version of the Global Navigation Satellite System (GNSS). Another example is Russian GLONASS. Here the GPS receiver onboard the platform, receives Time-of-Arrival signals from the satellites orbiting the Earth. Since the satellite positions are known quite precisely, it is possible to, with trilateration, estimate the position (and velocity) of the platform with high accuracy.

Air Data Computer (ADC) measures static and total pressure of the air around the platform. Static pressure is directly proportional to the altitude of the platform, while total pressure is the sum of the static and dynamic pressures. Dynamic pressure is a consequence of the platform's movement through the air. In this way, the platform's airspeed, the speed in relation to the surrounding air, can be calculated. ADC is not utilised directly in this work since other sensors can provide same information, but can easily be incorporated into the presented framework.

The used sensors are able to calculate almost all navigational parameters by themselves, it is basically only the attitude that needs to be estimated in some way from the raw sensor data. However, in order to get good dynamical behavior, e.g., remove noise or the effects of the different sampling times, the sensor fusion methods described in Section 2.1 must be used.

As mentioned above, different sensors, when fused, give the desired performance of the navigation system. In the configuration above, the performance is heavily dependent on the GPS, since it gives the stable, drift-free position and velocity information. This is however not desirable, since GPS is easily jammed or even spoofed. Jamming is a situation when signal is deliberately disturbed leading to the loss of GPS positioning. Spoofing is, on the other hand, a case when a GPS signal is maliciously altered locally such that it gives false and misleading positioning. This is, in a way, a worse case since false position can lead to severe consequences, like UAV crash in urban area. To mitigate these risks, alternative means of navigation must be devised. One way of doing this is introducing other sensors which, in combination with sensor fusion, could give an estimate of the navigational parameters which is drift-free and has good accuracy. Here, two imaging sensors, Synthetic Aperture Radar (SAR) and regular optical camera, will be used. Figure 3.2 shows the addition of these sensors to the navigation system. The reason why these particular sensors are chosen as a complement is that most UAVs have a camera, either as a part of the standard equipment or as a mission specific sensor. SAR is a more mission specific sensor, but it has certain advantages compared to an optical sensor since the wavelengths a radar operates on make it weather-insensitive. This property is important in many situations where high availability and all-weather capability is expected. In Section 3.3 and Section 3.4 the sensor models and functional principles for SAR and optical cameras will be described, and how they are used in the sensor fusion framework.

Another quality that above-mentioned imaging sensors have, is that they give some kind of information about the environment they are observing. In the case of the sensors that will be used here, the information is an image of the surroundings. As already described, this information will allow us to build a map of the environment, at the same time as we estimate the navigation states.

3.2 Inertial Measurement Unit

In this section, inertial sensors, which is a collective name for accelerometers and gyroscopes, will be described. These are usually combined in a single Inertial Measurement Unit (IMU) and will measure three dimensional acceleration, $[a^X \; a^Y \; a^Z]^T$, and angular velocity, $[\omega^X \; \omega^Y \; \omega^Z]^T$. This setup is usually called 6 Degrees-Of-Freedom (6-DOF) inertial sensor. The accelerometers measure the specific force in an inertial frame. For example on the surface the earth, at rest the accelerometer will measure earth's gravity force and force exhibited by earth's rotation, so called coriolis effect. The basic functional principle of the accelerometer is a damped mass on a spring. When a force is applied to the

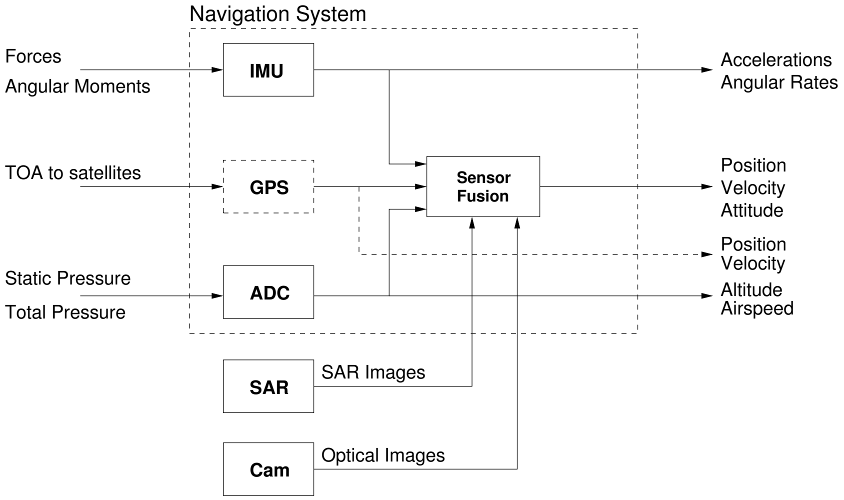


Figure 3.2: *UAV navigation system with addition of alternative sensors, SAR and optical camera. Both are imaging sensors, but images have different properties making them complementary. GPS block is dashed out here to emphasise that it can be left out.*

accelerometer, the force component parallel to the longitudinal axis will move the mass elongating the spring. The elongation is simply proportional to the applied force (or its longitudinal component) as

$$F = ma^X = kX \tag{3.2}$$

where m is the mass, k is the spring coefficient and X is the elongation. The technology for building accelerometers based on this spring mass principle has been developed during the recent years resulting in cheaper components. One such technology is Micro Electro-Mechanical Sensors (MEMS) where piezo-electrical effect is used. In this case, the silicon based rod is bent when affected by the force creating current which will be proportional to the applied force. The big advantage of this construction is that it can be made very small and light. This is a very attractive property of MEMS components, since both space and weight are limited in a typical UAV. However, the downside is that they are very sensitive for temperature variations giving large bias and noise terms.

The gyroscopes, or gyros, measure the rotational velocities around the fixed axis in the inertial frame. The basic technology how this is performed can be based on a principle of gyroscopic precession (Meriam and Kraige, 1998) used in mechanical gyros, e.g., the spinning or MEMS gyros, or light interference effect used in fiber optical and ring laser gyros. The precession is a principle where a rotating mass spinning around an axis will start to rotate around another axis when an external torque is applied around the third axis. This is illustrated in Figure 3.3 where the disc is spinning with angular velocity ω around x-axis and the torque M is applied around y-axis. This will cause the disc to precess

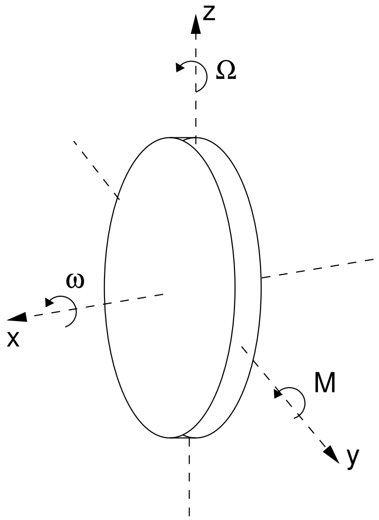


Figure 3.3: *Illustration of the gyroscopic precession principle. A cylindrically shaped mass spinning with angular velocity ω around x-axis will precess with angular velocity Ω around z-axis when a constant torque M is applied around y-axis.*

around z-axis with angular velocity Ω and the relationship between these is

$$M = I\Omega\omega \tag{3.3}$$

where I is the disc's moment of inertia. Spinning gyros use real spinning discs to measure the angular velocity while MEMS gyros use vibrating elements instead of rotating, but the underlying principle is the same. Fiber optical and ring laser gyros are using completely different principle based on the Sagnac effect. Here two beams of light are sent through a long optical fiber coil in opposite directions. When the coil rotates the phase shift in these two beams will cause interference pattern that can be observed and measured. The size of the interference is directly proportional to the angular velocity. In order to make errors small, optical gyros must be made large and heavy making them impractical for small UAV applications. On the other hand the accuracy and stability of the optical gyros is the best one with small bias and noise errors. MEMS gyros have the worst performance and purely mechanical ones are in the middle range.

For practical reasons, IMU is rigidly mounted, or strapped, in the platform's frame (body frame), i.e., it has a strap-down mounting. In this way the need for moving and potentially heavy parts is removed. This means that the acceleration is measured in the body frame. In order to use acceleration for navigation accelerations must be expressed in the local navigation frame, (this was used in (3.1)). The implication of this is that acceleration must be transformed to the local navigation frame in order to use it in the navigation application as already mentioned in Section 3.1. The measurement equation (2.2b) for the accelerometer is then

$$y_t^a = \mathsf{R}_t(a_t - g) + b_t^a + e_t^a \tag{3.4}$$

where R_t, as already described, is a rotation matrix that expresses rotation between navigation and body frame, and $g = [0\;0\;-9.81]^T$ is the gravitation acceleration expressed in the local navigation frame. The bias term b_t^a can be included in the model if the estimate of it is not compensated for or if the estimate of it is sought. e_t^a is the measurement noise term, usually assumed to be white and Gaussian with $e_t \sim \mathcal{N}(0, R_t^a)$.

The measurement equation for the gyros is even simpler, it is

$$y_t^\omega = \omega_t + b_t^\omega + e_t^\omega \tag{3.5}$$

where, as in acceleration case, b_t^ω and e_t^ω denote bias and noise terms respectively, with $e_t^\omega \sim \mathcal{N}(0, R_t^\omega)$.

Notice also that the output from the IMU can either be used as a measurement y_t or as an input u_t in the dynamics. Since both approaches have their advantages and disadvantages, which one is actually used will depend on the specific application.

3.3 Synthetic Aperture Radar

In this section the basic principles of the Synthetic Aperture Radar (SAR) are explained. It starts with the Real Aperture Radar (RAR), continues with the methods to create high resolution images and mentions some of the effects associated with SAR.

3.3.1 Real and Synthetic Aperture Radar

In the simplest setup a radar image can be created with the moving platform carrying the side-looking radar and flying above the scene to be imaged. By sending and receiving radar pulses along the trajectory, a range-azimuth image is created, see Figure 3.4. Since the energy in each radar pulse is spread out over the scene, every pulse will produce a one-dimensional image according to a simple principle; each echoed pulse is received and gated in range bins according to the time, t, it takes to receive the pulse, using the relation $R = tc/2$, where c denotes the speed of light. This means that each range bin will contain the total energy reflected from the scene on that specific range. When all of these simple one-dimensional images are stacked next to each other, a full RAR image is created. RAR images will have the imaged scene smeared across the azimuth, giving very poor effective resolution which can be seen in Figure 3.5, where a simulated example of the RAR image of two point targets is depicted. The resolution in the azimuth direction for the RAR images is governed by the basic laws of the electromagnetism which state that the radar lobe width is dependent of the antenna size and the wavelength of the carrier according to λ/d, where d is antenna size and λ is the wavelength. This will give the resolution $\lambda R/d$ where R is the range to the imaged scene. We see that by decreasing the wavelength or increasing the antenna we can make the lobe narrower and increase the resolution. The wavelength is usually fixed to a specific value and there is not much freedom of decreasing it, and besides there are other unwanted effects if low wavelengths are used like cloud occlusions. There are also limits regarding how large antenna a flying platform can carry which limits that parameter as well. However, by using the movement of the platform a long antenna can be synthesised and the resolution of the images is drastically increased. This principle leads to Synthetic Aperture Radar (Cutrona et al.,

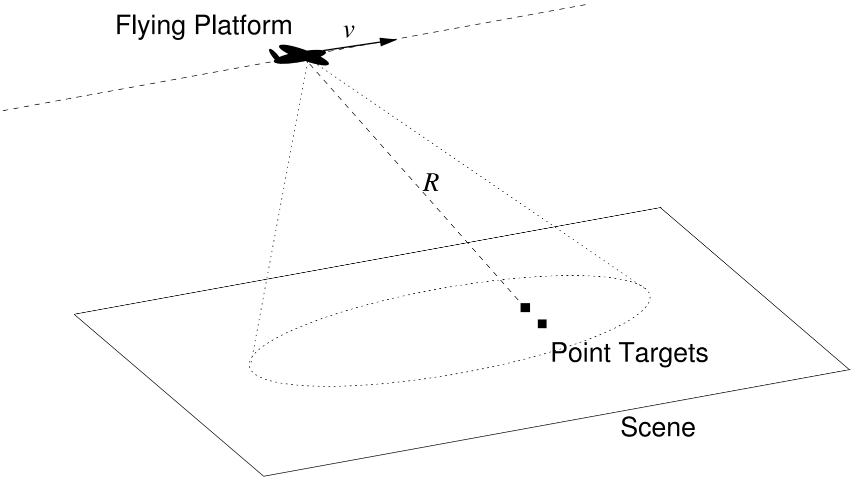


Figure 3.4: *Side-looking radar geometry. The flying platform is moving along the straight trajectory parallel to the scene with speed v while it transmits and receives radar pulses. R is the range from the platform to the target.*

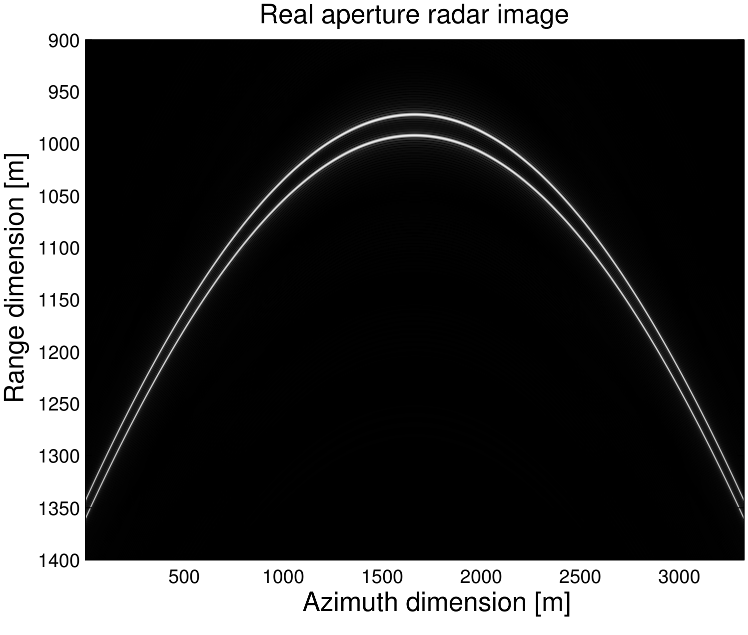


Figure 3.5: *Real aperture radar image of the two point targets in Figure 3.4.*

1961).

As mentioned above the movement of the platform can be utilised to improve the resolution in the azimuth direction. During the motion of the platform the scene, which can be assumed to consist of point scatterers, will travel through the radar lobe and the slant range to it will vary, see Figure 3.4. This slant range variation can be compensated for each radar echo and all of the echos can be integrated in order to produce an image which can be expressed as

$$I^{\mathrm{SAR}} = \int_{D_t} I_t^{\mathrm{RAR}}(R) g(R,t)\, \mathrm{d}t \tag{3.6}$$

where I^{SAR} is the SAR image, $I_t^{\mathrm{RAR}}(R)$ is the RAR image (or raw radar data), $g(R,t)$ includes possible demodulation to baseband and lobe weight terms, R is the range dimension, t is the azimuth dimension of the raw data (or time dimension), and D_t is the azimuth (or time) domain of the RAR image. Image creation can be performed in the image (or time) domain or in the frequency domain. Some of the most important frequency domain methods are the Range-Doppler (Walker, 1980), the Fourier-Hankel (Fawcett, 1985; Hellsten and Andersson, 1987; Andersson, 1988) and the ω-K migration methods (Rocca, 1987; Cafforio et al., 1991; Milman, 1993). The frequency domain methods are generally fast, but has a downside that they assume straight trajectories in order to work properly. However, in reality the trajectory will never be straight, especially if the flying platform is a small UAV. This will cause image distortions and auto-focusing is more complicated. This opens up for the use of time domain methods, which are slower, but can handle any trajectory shape. This of course is an important benefit, particularly if SAR images are to be used for trajectory estimation, and time domain methods will be considered from now on.

One of the most known time domain methods is back-projection (Natterer, 1986). In the back-projection procedure each saved radar echo, which is one dimensional, is back-projected onto a two dimensional area. In this way a poor quality image of the scene is obtained. Now we can sum up all these back-projected images in order to obtain the full SAR image. This is equivalent to the integration operation in (3.6), except that integration becomes summation due to the discrete data. This process can be described for each pixel (i,j) in SAR image as in

$$I_{ij}^{\mathrm{SAR}} = \sum_{t=1}^{N} z_t(R_t^{ij}) \tag{3.7a}$$

$$R_t^{ij} = \|p_t - s^{ij}\|_2 \tag{3.7b}$$

where p_t is the position of the platform and s^{ij} is the position in the scene which corresponds to the pixel (i,j). $z_t(R)$ is the compensated raw data in sampled form. Figure 3.6 describes this procedure in a schematic way and Figure 3.7 is the resulting image produced with the back-projection method if the same raw data used for RAR from Figure 3.5 is used.

The main downside of this kind of procedure is that the number of operations needed to synthesise an image is proportional to $\mathcal{O}(KMN)$ for a $K \times M$ image created from

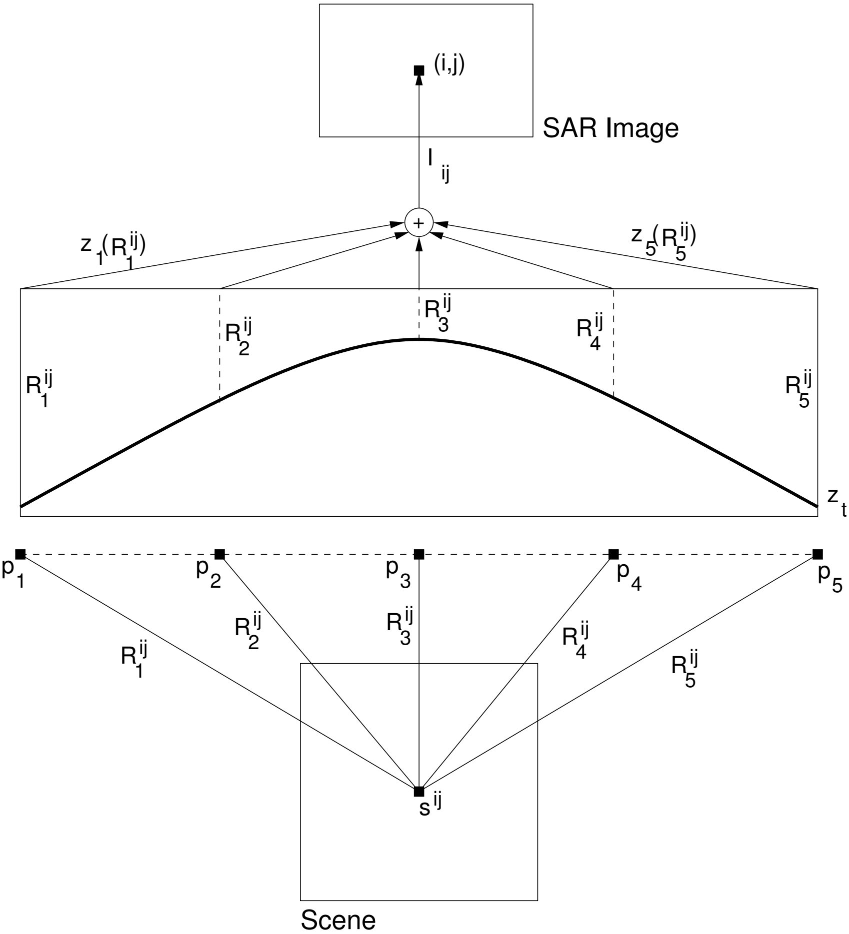


Figure 3.6: *Back-projection operation schematically described. The scene consists of only one point target in this illustration. The figure is not to scale.*

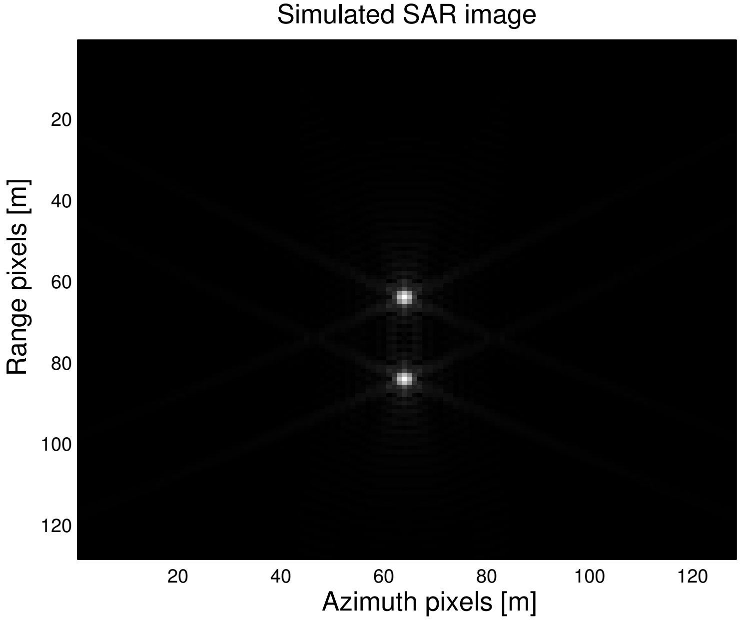
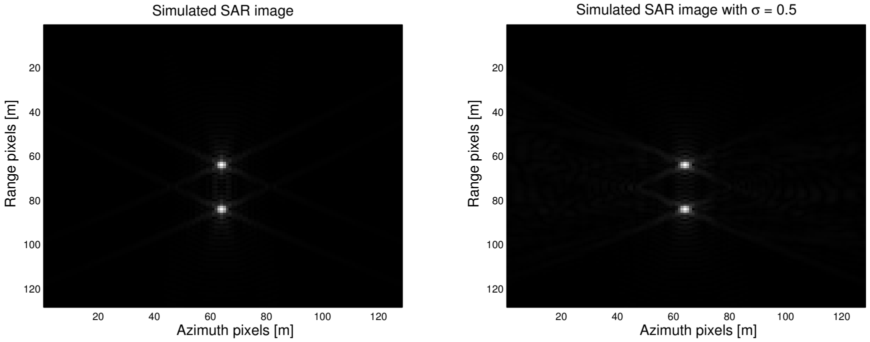


Figure 3.7: *Synthetic aperture radar image of the two point targets from Figure 3.4.*

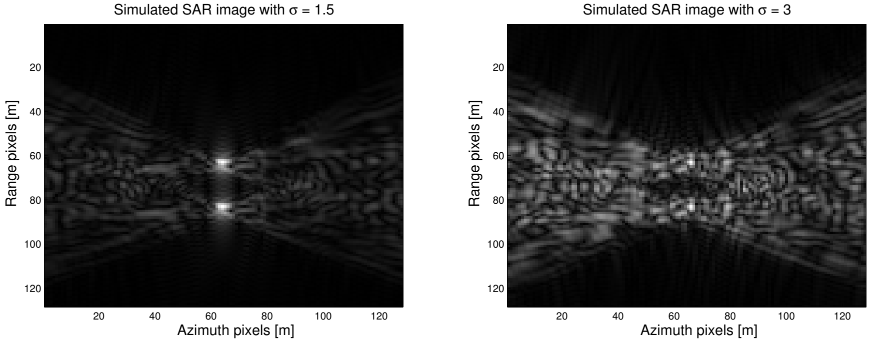
N radar echos. This can be a large number for large images and long aperture times. However during recent years a modification to the original back-projection, called fast factorised back-projection, has been developed. This method can actually create the SAR image in $\mathcal{O}(KM \log N)$ number of operations (Ulander et al., 2003). This implies considerable time saving and together with the development of the computers, this allows to consider real time SAR imaging.

It is now clear that in order to perform the back-projection (or factorised fast back-projection) operation the trajectory of the platform must be known or otherwise the resulting image will be distorted. The image distortion can manifest itself in many ways, from pure translation through geometric distortion to defocusing. Since translation and geometric distortions are hard to measure if the true scene is unknown, these will not be considered here. Defocusing, which is a very common distortion, is measurable, at least seen as an image property. The main source of defocusing of the SAR images is the error in the trajectory estimate used for creation of the images. In the frequency domain methods, a straight trajectory is used, and if the real trajectory deviates from this assumption, it will cause image defocusing. In the time domain methods, despite the fact that the general trajectory form is used, deviations from the real trajectory will cause the back-projected sub images to be shifted. The summation operation of the sub images will then cause defocusing. To illustrate this, the simple two point image from Figure 3.7 can be used. If Gaussian white noise with different variances is added to the cross-track position of the platform, images as in Figure 3.8 are obtained. In Figure 3.8a, the image is created with the same trajectory as data were acquired, which results in a perfectly fo-



***(a)** Focused SAR image of two point targets.*

***(b)** Unfocused SAR image of two point targets with* $\sigma = 0.5\ \mathrm{m}$.



***(c)** Unfocused SAR image of two point targets with* $\sigma = 1.5\ \mathrm{m}$.

***(d)** Unfocused SAR image of two point targets with* $\sigma = 3\ \mathrm{m}$.

***Figure 3.8:** Example SAR images with different perturbed trajectories.*

cused image. In the other three images the cross-track position noise was $\mathcal{N}(0, \sigma^2)$ where $\sigma = \{0.5, 1.5, 3\}$ $[\mathrm{m}]$ and the images are created under the assumption that the trajectory was linear. This results in defocused images, and the degree of defocusing depends on the noise variance. Much effort has been spent to correct for this, see for example Oliver and Quegan (2004); Wahl et al. (1994); Xi et al. (1999); Morrison and Munson (2002); Xing et al. (2009). Traditionally, these methods are open-loop type, meaning that the image is created with assumptions of linear flight trajectory and focusing is done afterwards in an open-loop way discarding possible flight trajectory information. This is a consequence of the off-line image generating process where the trajectory is no longer interesting. In the setup where SAR images are generated on-line, an idea, as already mentioned, is to use information from the image defocusing and navigation system together. In the next section, different focus measures, that can be used in this procedure are covered.

3.3.2 Measuring Focus

Sum-Modified-Laplacian

The Sum-Modified-Laplacian focus measure is defined as

$$\begin{aligned}\nabla^2_{\mathrm{ML}} I_{ij} =& |2I_{ij} - I_{(i-1)j} - I_{(i+1)j}| + \\ & |2I_{ij} - I_{i(j-1)} - I_{i(j+1)}| \end{aligned} \tag{3.8a}$$

$$\mathrm{SML}(I) = \sum_{i=2}^{K-1} \sum_{j=2}^{M-1} \nabla^2_{\mathrm{ML}} I_{ij} \cdot \mathbf{I}_{[\nabla^2_{\mathrm{ML}} I_{ij} \geq T]} (\nabla^2_{\mathrm{ML}} I_{ij}) \tag{3.8b}$$

where I_{ij} is the $K \times M$ image grey-scale intensity for pixel coordinate (i, j), $\mathbf{I}$ is the indicator function, and T is the threshold value. This measure has its maximum for the most focused images. Since (2.29) is a minimisation criterion, SML can be inverted or negated to fit into this criterion.

Tenengrad

The Tenengrad focus measure is defined as

$$\mathrm{TG}(I) = \sum_{i=2}^{K-1} \sum_{j=2}^{M-1} S_{ij}^2 \cdot \mathbf{I}_{[S_{ij} > T]} (S_{ij}) \tag{3.9}$$

where S_{ij} is the Sobel gradient in pixel coordinate (i, j),

$$S_{ij} = \sqrt{(S^r_{ij})^2 + (S^c_{ij})^2} \tag{3.10}$$

where S^r and S^c are row and column Sobel gradients respectively, T is threshold value and I is the $K \times M$ image. The Sobel gradients are obtained by convolving the image with the row and column Sobel kernels D^r and D^c,

$$S^r = D^r * I \tag{3.11a}$$

$$S^c = D^c * I \tag{3.11b}$$

$$D^r = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{3.11c}$$

$$D^c = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3.11d}$$

Just as SML, TG has it maximum for the most focused images. This problem is circumvented in the same way as for the SML.

Discrete Cosine Transform

Further, the Discrete Cosine Transform (DCT) can be used to measure image focus, see Kristan et al. (2006). The main idea is that focused images have higher frequency components than unfocused images. The focus measure based on DCT is defined as

$$\mathrm{DCT}(I) = 1 - \frac{\sum_{\omega=1}^{T}\sum_{\nu=1}^{T}|D(\omega,\nu)|^2}{(\sum_{\omega=1}^{T}\sum_{\nu=1}^{T}|D(\omega,\nu)|)^2} \tag{3.12}$$

where

$$D(\omega,\nu) = \alpha_\omega\alpha_\nu \sum_{i=0}^{K-1}\sum_{j=0}^{M-1} I_{ij}\cos\left(\frac{\pi(2i+1)\omega}{2K}\right)\cos\left(\frac{\pi(2j+1)\nu}{2M}\right) \tag{3.13a}$$

$$\alpha_\omega = \begin{cases} \frac{1}{\sqrt{K}}, & \omega = 0 \\ \sqrt{\frac{2}{K}}, & 1 \le \omega \le K-1 \end{cases} \tag{3.13b}$$

$$\alpha_\nu = \begin{cases} \frac{1}{\sqrt{M}}, & \nu = 0 \\ \sqrt{\frac{2}{M}}, & 1 \le \nu \le M-1 \end{cases} \tag{3.13c}$$

is the Discrete Cosine Transform of the image I and T is threshold. Even DCT behaves as TG and SML, i.e., it attains the maximum value for the sharpest images.

Entropy

Another measure of the image focus is image entropy calculated as

$$E_1(I) = -\sum_{k=1}^{256} p_k \log_2(p_k) \tag{3.14}$$

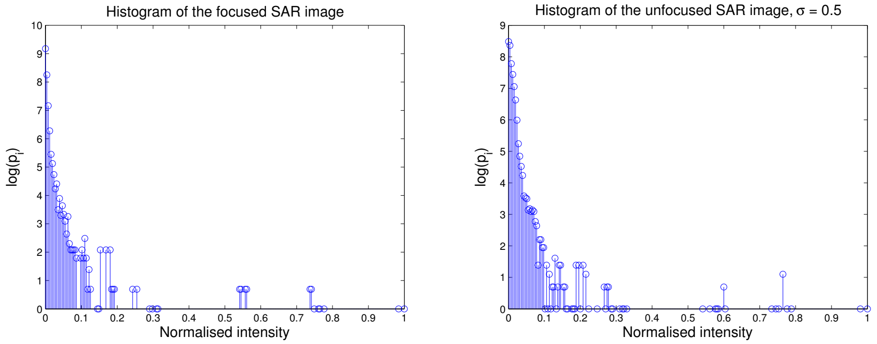
where p_i is an approximated grey level distribution of the $K \times M$ grey-scale image. A grey-scale SAR image is taken as the amplitude image $|I_{ij}|$, where I_{ij} is here considered as the complex-valued SAR image. p_i can be obtained from the image histogram, calculated as

$$p_k = \frac{\{\text{\# of pixel values } |I_{ij}|\} \in [k-1,k]}{KM} \tag{3.15a}$$

$$k \in [1,256] \tag{3.15b}$$

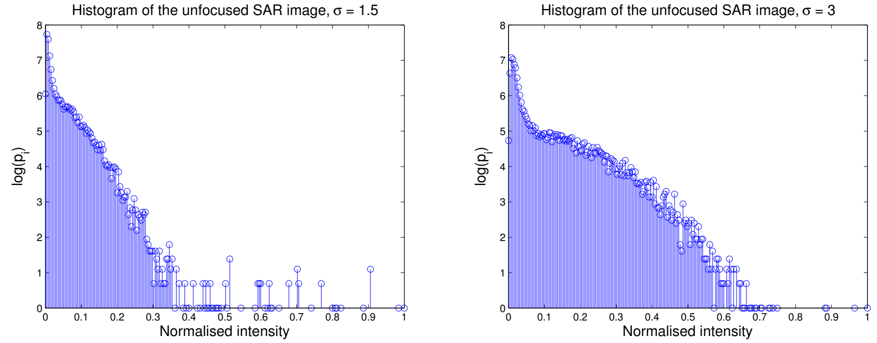
The more focused the image is, the higher the entropy is (Ferzli and Karam, 2005), exactly as for the previous mentioned measures. Note however that entropy measure is primarily used for optical images, whose defocus (or rather unsharpness) have different nature from the defocus of SAR images. In Section 3.3.4 it will be shown that the entropy for a SAR image is lower the more focused the image is. Example histograms and entropy values for the images in Figure 3.8 are given in Figure 3.9.

An alternative definition of entropy (and more frequently used in the SAR context) is



***(a)** Histogram of the focused image. $E_1(I) = 2.66$.*

***(b)** Histogram of the unfocused image with $\sigma = 0.5$ m. $E_1(I) = 3.32$.*



***(c)** Histogram of the unfocused image with $\sigma = 1.5$ m. $E_1(I) = 5.72$.*

***(d)** Histogram of the unfocused image with $\sigma = 3$ m. $E_1(I) = 7.47$.*

***Figure 3.9:** Histograms for the images in Figure 3.8 and corresponding entropy 1 values. Note the log-scale on y-axis.*

(Yegulalp, 1999; Xi et al., 1999; Morrison and Munson, 2002),

$$E_2(I) = -\sum_{i=1}^{K}\sum_{j=1}^{M} p_{ij}\ln(p_{ij}) \tag{3.16a}$$

$$p_{ij} = \frac{|I_{ij}|^2}{\sum_{i=1}^{K}\sum_{j=1}^{M}|I_{ij}|^2} \tag{3.16b}$$

This entropy, on the other hand, will have its minimum for both optical and SAR images, as will be shown in the evaluations in Sections 3.3.3 and 3.3.4.

3.3.3 Evaluation on Test Optical Image

For the purpose of evaluation of the different focus measures, the image in Figure 3.10 is used. The evaluation will be performed for different thresholds where it is applicable and for different degrees of defocus (or unsharpness).

Threshold Dependence

In order to evaluate the performance of the different measures, the ratio between focused and unfocused images, $F(I_{\text{focused}})/F(I_{\text{unfocused}})$, is plotted for the test image as a function of threshold T (F is one of TG, SML or DCT). Unfocusing is performed with the Gaussian low-pass filter kernel with size 5×5 pixels and with standard deviation $\sigma = 1.5$. The results are plotted in Figure 3.11. Since entropy does not depend on the threshold, only one value is obtained for this case, $E_1(I_{\text{foc}})/E_1(I_{\text{unfoc}}) = 6.99/6.80 = 1.03$ and $E_2(I_{\text{foc}})/E_2(I_{\text{unfoc}}) = 12.40/12.42 = 0.99$.

Blur Kernel Dependence

In Figure 3.12 the ratio $F(I_{\text{focused}})/F(I_{\text{unfocused}})$ is plotted for different standard deviations of the 5×5 Gaussian low-pass kernel, in the range $\sigma \in [0,\ 4]$. Higher variance will yield a less focused image. For the measures that depend on the threshold, the threshold values are chosen based on the results from Section 3.3.3, i.e., the threshold values that give high ratio between focused and unfocused images. For this particular evaluation they are $T_{\text{TG}} = 350$, $T_{\text{SML}} = 270$ and $T_{\text{DCT}} = 2$.

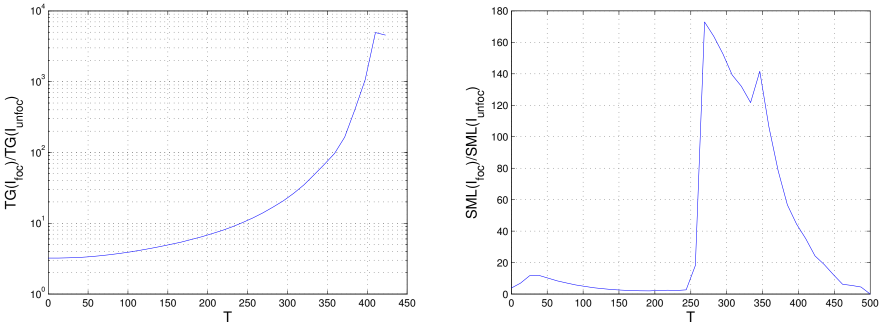
3.3.4 Evaluation on Test SAR Images

For SAR images, basically the same evaluation as above will be performed, but in this case defocusing will be caused by the different trajectory errors, which is the main reason for SAR image defocusing.

Threshold Dependence

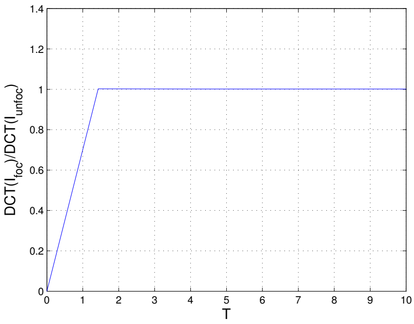
To get a feeling for how different focus measures perform on SAR images as a function of the threshold, the same evaluation is done as in Section 3.3.3. The example SAR image used in the evaluation is in Figure 3.13. Defocusing is here obtained with adding range direction noise with standard deviation $\sigma = 1.5$. Results are plotted in Figure 3.14. Exactly as above, since entropy does not depend on the threshold, there is only one value for each σ. In particular, $\sigma = 1.5$ gives $E_1(I_{\text{foc}})/E_1(I_{\text{unfoc}}) = 1.91/5.12 = 0.37$ and $E_2(\tilde{I}_{\text{foc}})/E_2(\tilde{I}_{\text{unfoc}}) = 4.28/8.10 = 0.53$.

Figure 3.10: *Test image used for evaluation of the focus measures. Image is courtesy of University of Southern California, Los Angeles, California, USA.*



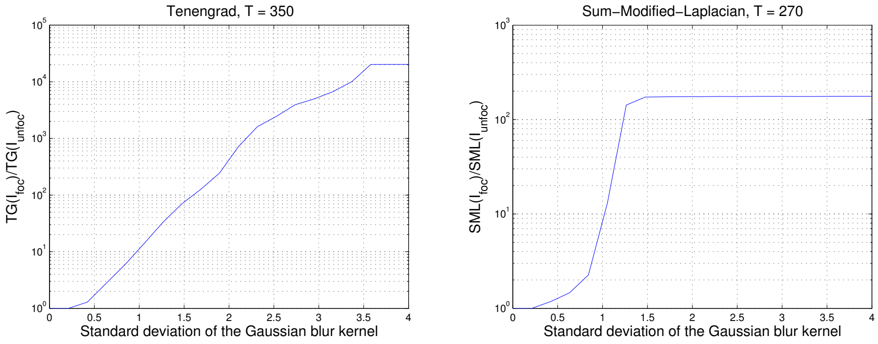
(a) *$TG(I_{\text{foc}})/TG(I_{\text{unfoc}})$, log-scale on the y-axis.*

(b) *$SML(I_{\text{foc}})/SML(I_{\text{unfoc}})$*



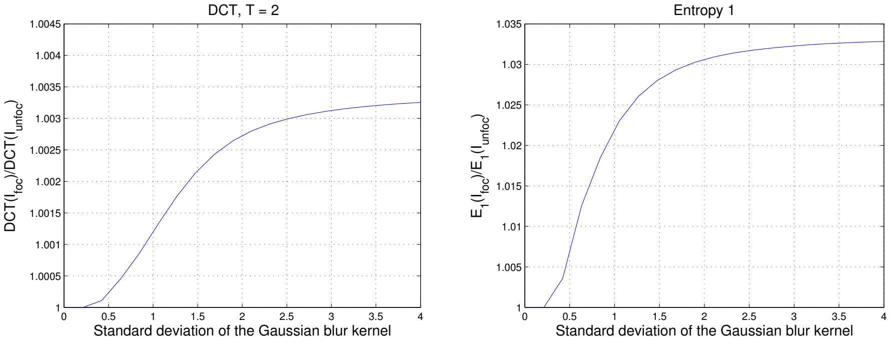
(c) *$DCT(I_{\text{foc}})/DCT(I_{\text{unfoc}})$*

Figure 3.11: *Ratio of the different focus measures for focused and unfocused optical images as a function of the threshold.*



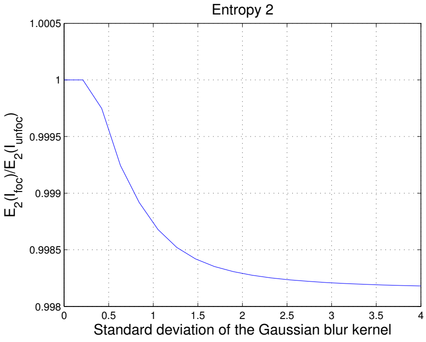
(a) *$TG(I_{\text{foc}})/TG(I_{\text{unfoc}})$, log-scale on the y-axis.*

(b) *$SML(I_{\text{foc}})/SML(I_{\text{unfoc}})$, log-scale on the y-axis.*



(c) *$DCT(I_{\text{foc}})/DCT(I_{\text{unfoc}})$.*

(d) $E_1(I_{\text{foc}})/E_1(I_{\text{unfoc}})$



(e) $E_2(I_{\text{foc}})/E_2(I_{\text{unfoc}})$

Figure 3.12: *Ratio of the different focus measures for focused and unfocused optical images as a function of the standard deviation of the Gaussian blur kernel.*

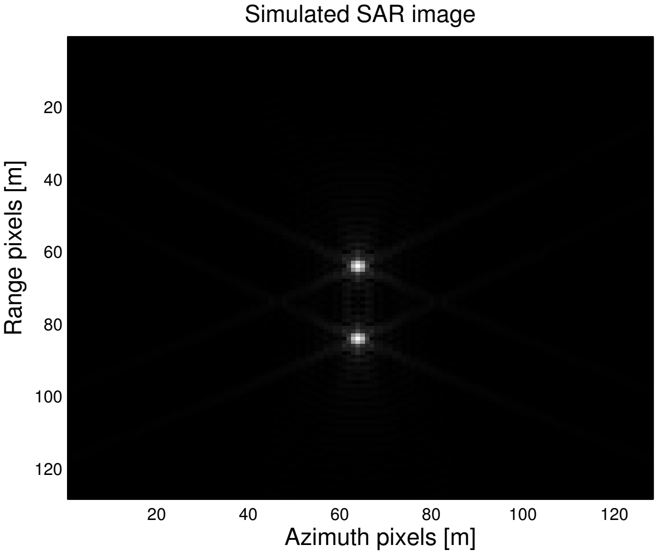


Figure 3.13: *Example SAR image used for evaluation of the focus measures.*

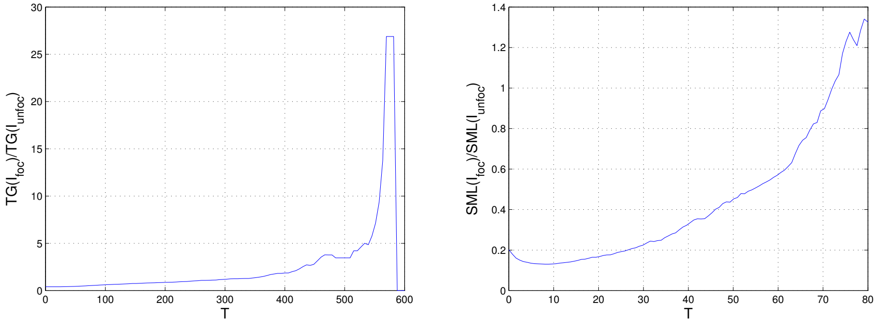
Position Blur Dependence

Another test of the focus measures that is performed is how they depend on the variance of the noise in the range direction for a given threshold, similar to the evaluation in Section 3.3.3. The simulation is performed with the same noise realisation, but different variances, i.e., position in range direction is

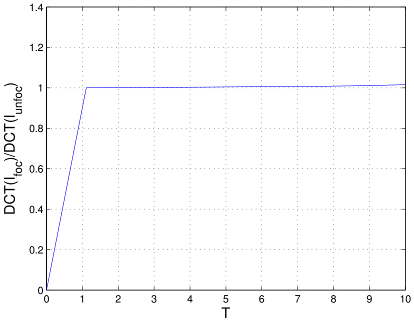
$$Y_t = \sqrt{Q} w_t \tag{3.17}$$

where w_t is $\mathcal{N}(0,1)$ and $Q \in [0,\ 9]\ [\mathrm{m}^2]$ (standard deviation is between 0 and 3 m). The result is depicted in Figure 3.15. From these plots it can be noticed that DCT and entropy 2 measures behave as expected, i.e., the ratio between focused and unfocused image is monotonically increasing as a function of the standard deviation for DCT and decreasing for entropy 2 measure. For TG, SML and entropy 1, however, there is a difference between optical and SAR images. For optical images, all three measures will have their maxima for focused images, while for SAR images a minimum value is obtained instead. This behaviour can be explained both by the look of the test image, two bright points and lot of dark area, and the nature of the SAR images, i.e., SAR images usually look like they are negative optical images and contain more dark areas than optical images.

In order to obtain an image which is more informative, a scene in Figure 3.16 is created. A smaller image is created to minimise the dark area. This image should be more representative for the SAR images. The simulation above is modified and the maximum value for the standard deviation of the noise is set to 1.5 instead of 3. This is done because the smaller image might cause rand effects which will negatively influence the focus measures. The results are depicted in Figure 3.17. It can be noticed in these plots that TG and SML behave as expected now, but SML has a highly non-convex form. This indicates

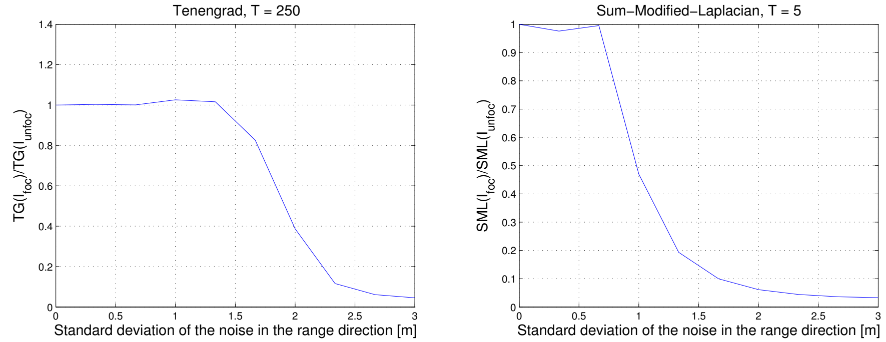


(a) $TG(I_{\text{foc}})/TG(I_{\text{unfoc}})$ **(b)** $SML(I_{\text{foc}})/SML(I_{\text{unfoc}})$

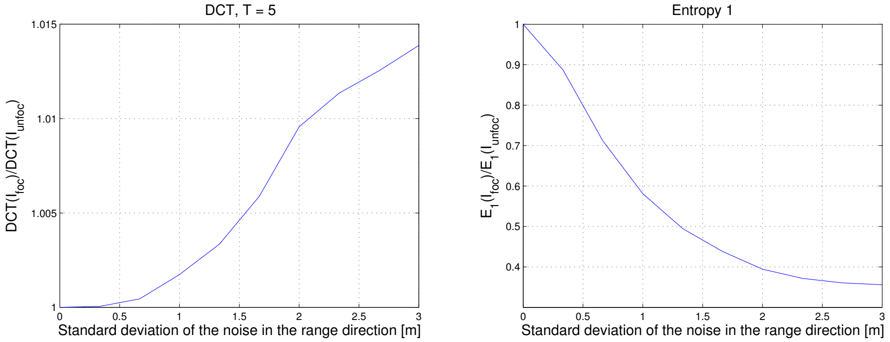


(c) $DCT(I_{\text{foc}})/DCT(I_{\text{unfoc}})$

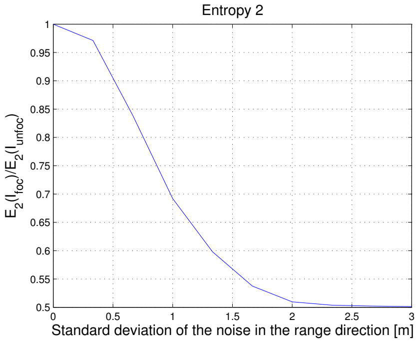
Figure 3.14: *Ratio of the different focus measures for focused and unfocused SAR images as a function of the threshold.*



(a) *$TG(I_{\text{foc}})/TG(I_{\text{unfoc}})$* ***(b)*** *$SML(I_{\text{foc}})/SML(I_{\text{unfoc}})$*



(c) *$DCT(I_{\text{foc}})/DCT(I_{\text{unfoc}})$* ***(d)*** $E_1(I_{\text{foc}})/E_1(I_{\text{unfoc}})$



(e) $E_2(I_{\text{foc}})/E_2(I_{\text{unfoc}})$

Figure 3.15: *Ratio of the different focus measures for focused and unfocused SAR images as a function of the standard deviation of the noise in the range direction.*

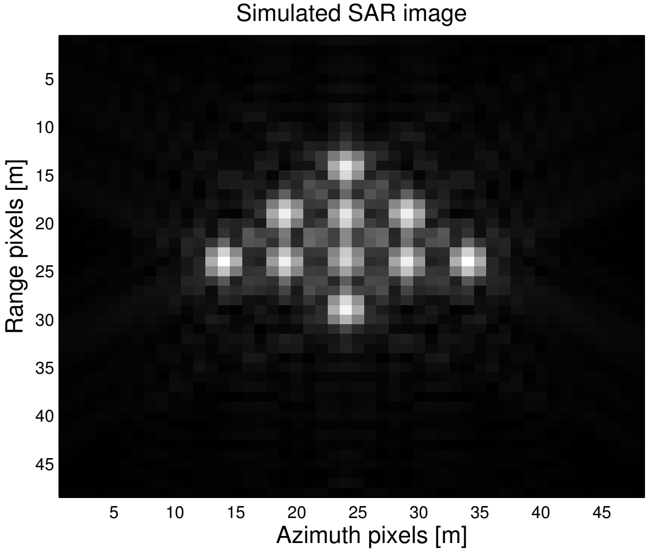


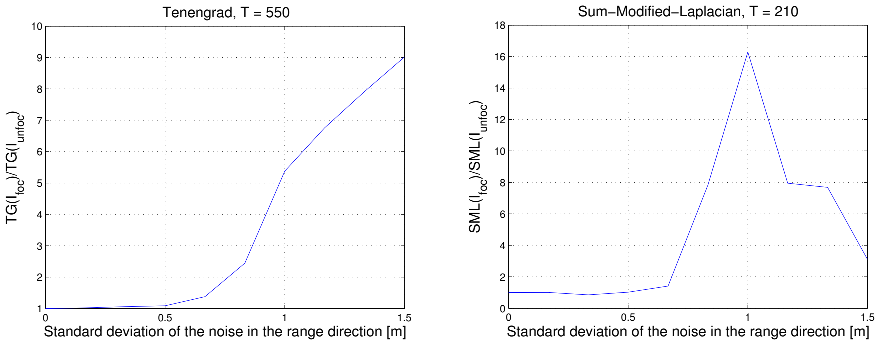
Figure 3.16: *Example SAR image with more informative scene.*

that SML might not be suitable measure for the SAR images. Entropy 2 and DCT still behave as expected, and entropy 1, just as in the previous case, attains minimum for the most focused SAR image.

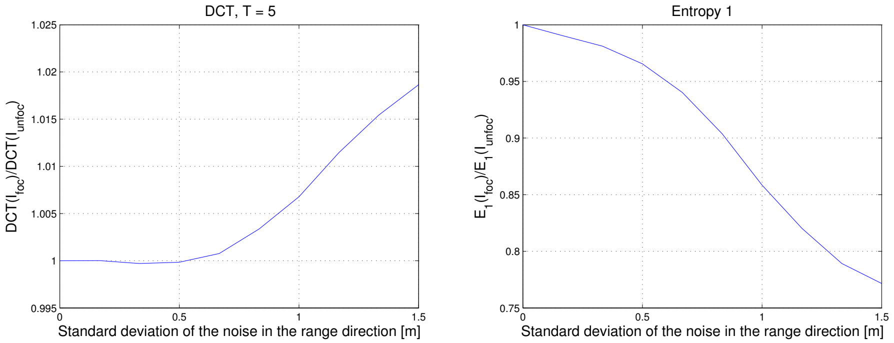
Evaluation on Perturbed Trajectory

In order to evaluate the focus measures behaviour for different trajectories on the simulated SAR images, simulations with different trajectory errors are performed. The trajectory is simulated with the model (2.19) and trajectory errors consist of different incorrect initial conditions on velocity in azimuth direction, v_0^X, and acceleration in range direction, a_0^Y. The nominal initial values are chosen as $v_0^X = 100$ m/s and $a_0^Y = 0$ m/s^2 and acceleration error is varied between -0.045 and 0.045 m/s^2 and velocity error is varied between 99.962 and 100.038 m/s. Those values are chosen as 3σ-values of the state errors from the assumed typical navigation system performance in an UAV, see Paper A. The noise, w_t^a, is set to zero in these simulations, i.e., the trajectory is completely deterministic. This has been done in order to be able to illustrate focus measures in a two-dimensional plot. On the other side, all focus measures, as functions of the state noise are convex and impose no problems in the minimisation step. In Figure 3.18 trajectory examples with some different acceleration and velocity errors according to Table 3.1 are shown.

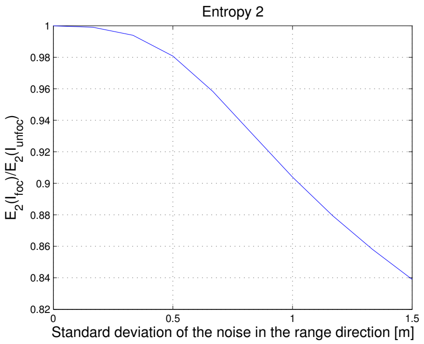
All five focus measures, TG with threshold value $T = 550$, SML with $T = 170$, DCT with $T = 5$ and entropy measures 1 and 2, are compared on the image in Figure 3.16. The thresholds are chosen according to the results from the simulations in Section 3.3.4. Results are depicted in Figure 3.19 where 1σ-, 2σ- and 3σ-standard deviations of the initial states v_0^X and a_0^Y are also drawn. The contours in the plots are the level curves of the focus measures as a function of the error in the initial states. The level curves for the



(a) $TG(I_{\text{foc}})/TG(I_{\text{unfoc}})$ **(b)** $SML(I_{\text{foc}})/SML(I_{\text{unfoc}})$



(c) $DCT(I_{\text{foc}})/DCT(I_{\text{unfoc}})$ **(d)** $E_1(I_{\text{foc}})/E_1(I_{\text{unfoc}})$



(e) $E_2(I_{\text{foc}})/E_2(I_{\text{unfoc}})$

Figure 3.17: *Ratio of the different focus measures for focused and unfocused SAR images with more informative scene as a function of the standard deviation of the noise in the range direction.*

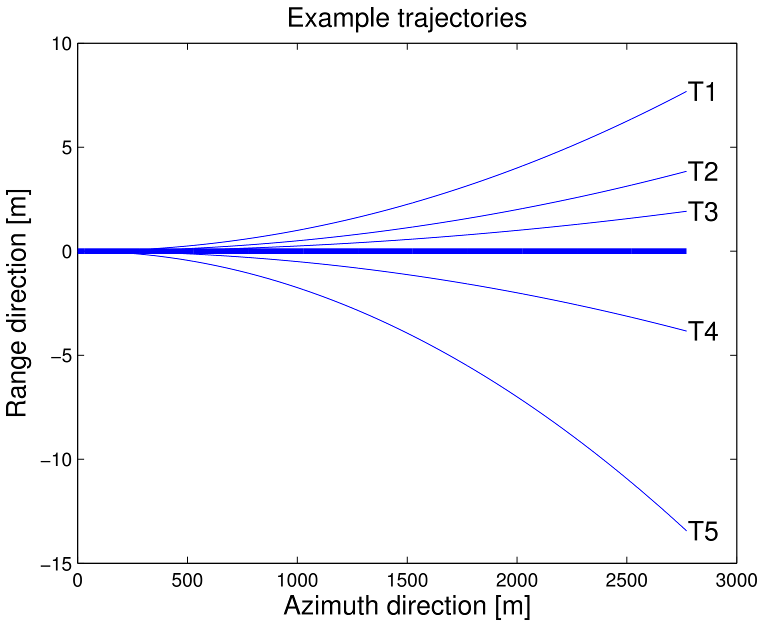


Figure 3.18: *Example trajectories from Table 3.1. Thick line: no errors in initial values.*

Trajectory	**Error in** a_0^Y $[\mathrm{m/s^2}]$	**Error in** v_0^X $[\mathrm{m/s}]$
T1	0.02	-0.005
T2	0.01	-0.01
T3	0.005	0.005
T4	-0.01	0.02
T5	-0.035	0.005

Table 3.1: *Errors in initial acceleration and velocity for example trajectories shown in Figure 3.18.*

measures with the maximum for the correct values of the initial states, like TG or SML, are inverted.

Further, plots where only one parameter, v_0^X or a_0^Y, is varied are presented in Figure 3.20 (all measures are normalised between 0 and 1). From all these figures it looks like that all measures except entropy 2 have several local minima and are highly non-convex. We also see that TG and SML do not have minimum value for the correct velocity value and that DCT has its minimum in the wrong value of the acceleration. The entropy measures perform fairly well, and entropy 1 has much sharper global minimum than entropy 2. Based on this, the entropy 1 and the entropy 2 measures look as the most attractive measures since they attain minimum value for the correct values of the states, they do not have any threshold to tune and entropy 2 is also smooth and convex in the interesting region of the state values.

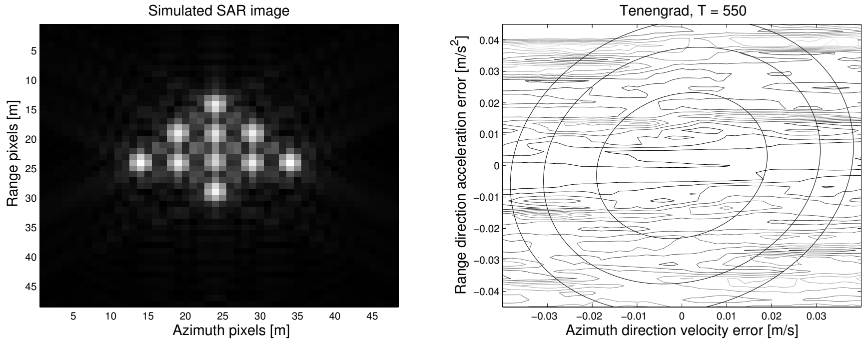
The focus measures' performance is also tested on a more unstructured scene illustrated in Figure 3.21a. This scene is created by randomly placing 150 point targets and assigning them a random reflectivity. The focus measures for this scene are shown in Figure 3.21. Here it can be seen that the measures look even worse and not even the entropy 1 measure has its global minimum for the correct values of the states. Entropy 2 is however still convex and smooth in the vicinity of the correct values of the states, and is still the most promising alternative to use as a focus measure in the minimisation criterion.

The focus measures above are evaluated on images that have no noise, i.e., the images are perfect. In reality, that is not the case, and images contain some noise caused by the noise in the radar measurements. Therefore, the same two scenes are used again, but white Gaussian noise with variance $\sigma^2 = 1.5$ is added to the radar echos. The images obtained with this setup are depicted in Figure 3.22a and Figure 3.23a. The focus measures for these two images are shown in Figure 3.22 and Figure 3.23. It can be seen in these plots that entropy 2 measure is still smooth and convex in the vicinity of the true values of the navigation states. The only thing that happens with entropy 2 is that the value of the function is different, but the principal form is the same as for the noise free case.

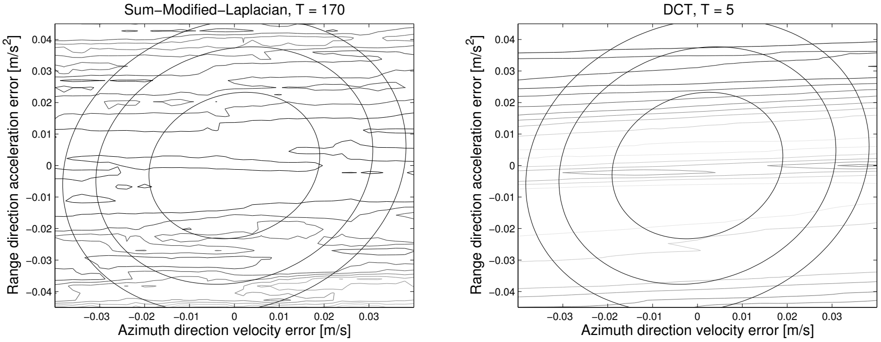
Another thing that can be noticed from these plots is that the entropy 1 measure does not have a pronounced global minimum in the case of structured scene, as it has for the noise free case. It looks similar to the case with unstructured scene. This is not surprising since the unstructured scene will behave like the image noise for the focus measure. It also looks like TG measure behaves better for the noisy case, at least around the true values of the navigation states. The explanation might be that TG measure has a threshold that can filter out the noise.

The conclusion from the evaluation above is that entropy 2 measure works quite fine for both cases, with and without noise in the radar measurements, and it is the smoothest measure of all tested ones. Entropy 1 could be used as well, at least for the scenes with structure, while TG seems to work for the case with noise in images. The drawback of the TG measure is the threshold that must be tuned to the different imaged scenes.

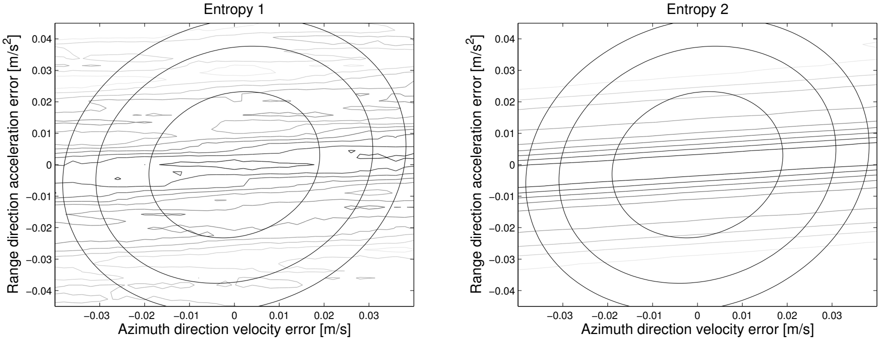
Another interesting thing that can be seen in this evaluation is that there is an ambiguity in the focus measure between initial X-direction velocity and Y-direction acceleration. It can be seen that almost all measures have a pronounced diagonal form. Very similar



(a) *SAR image of the structured scene.* ***(b)*** *TG focus measure with* $T = 550$.

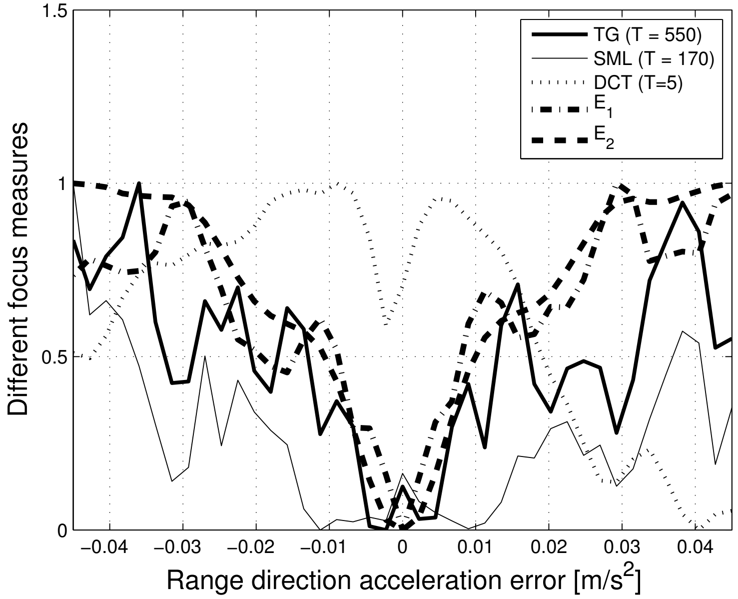


(c) *SML focus measure with* $T = 170$. ***(d)*** *DCT focus measure with* $T = 5$.

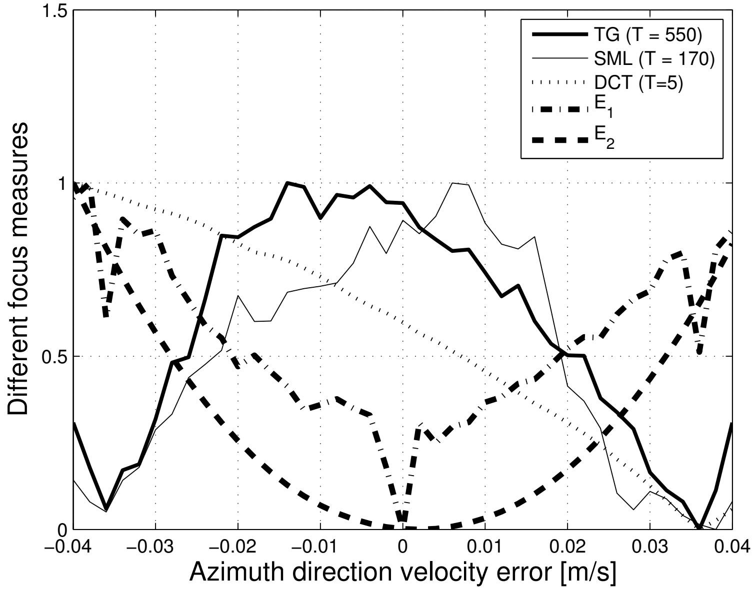


(e) *Entropy 1 focus measure.* ***(f)*** *Entropy 2 focus measure.*

Figure 3.19: *Focus measures for the image with more informative structured scene. Standard deviations of the states are also drawn.*

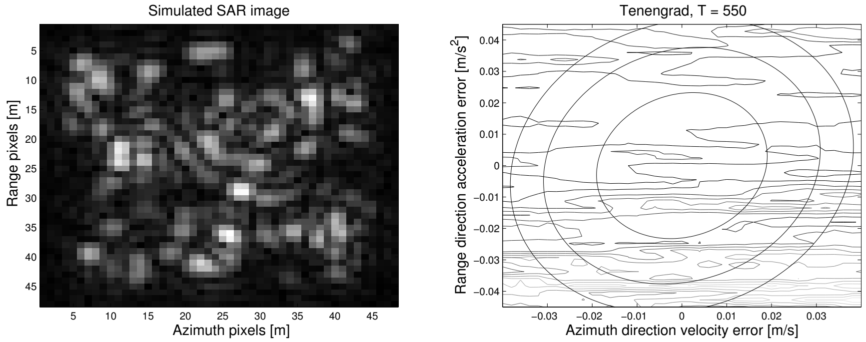


***(a)** Focus measures with no velocity error.*



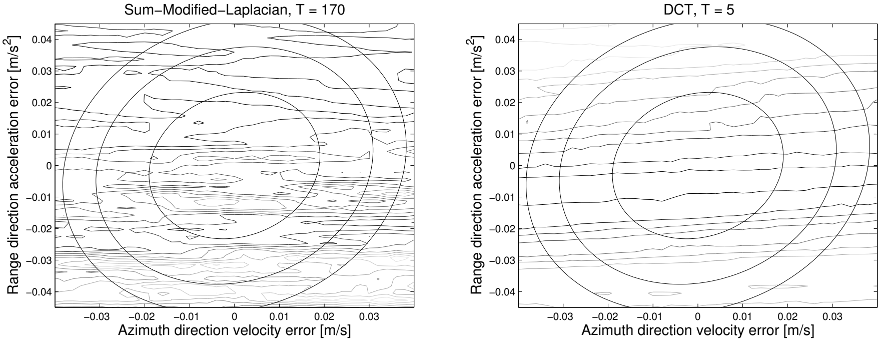
***(b)** Focus measures with no acceleration error.*

***Figure 3.20:** Focus measures for the structured scene with only error in a_0^Y and v_0^X.*



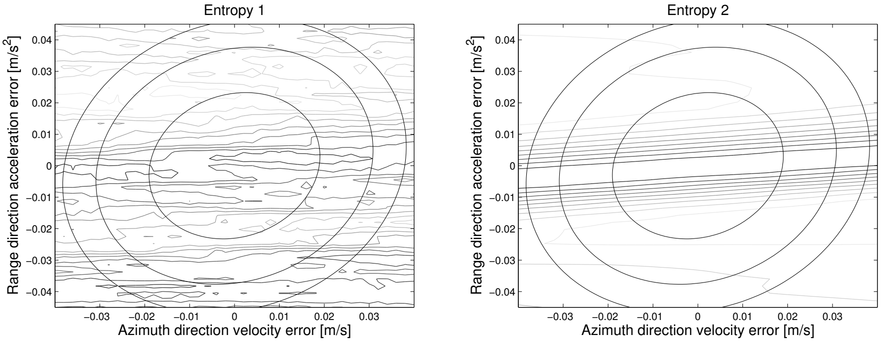
***(a)** SAR image of the unstructured scene.*

***(b)** TG focus measure with $T = 550$.*



***(c)** SML focus measure with $T = 170$.*

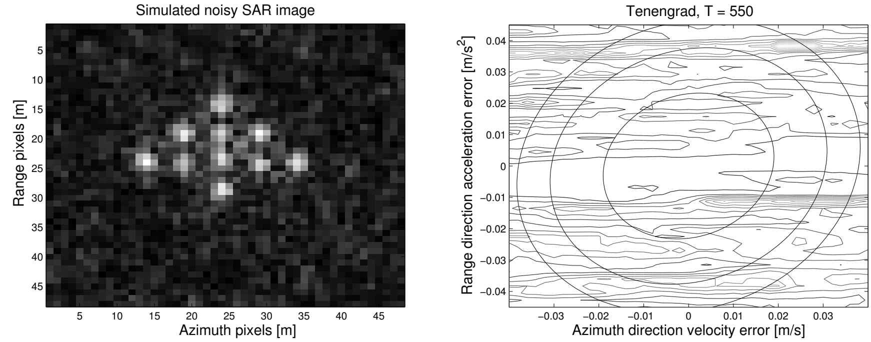
***(d)** DCT focus measure with $T = 5$.*



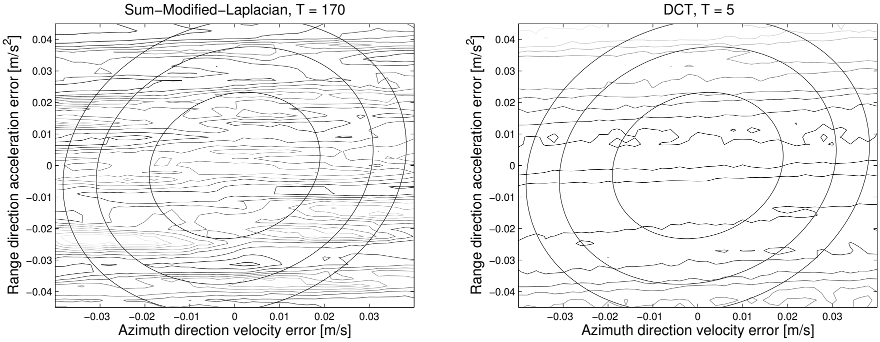
***(e)** Entropy 1 focus measure.*

***(f)** Entropy 2 focus measure.*

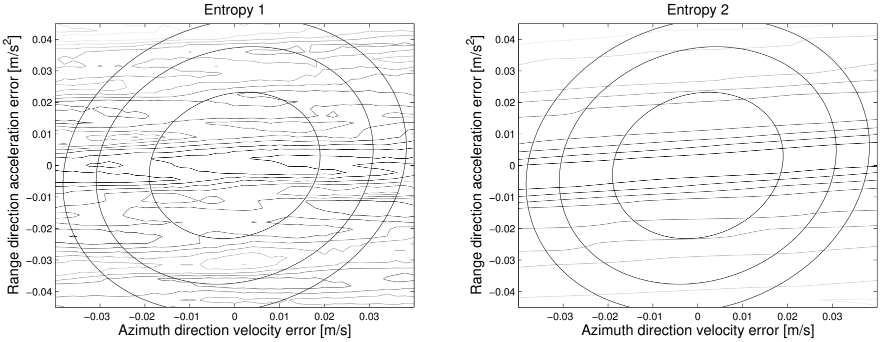
***Figure 3.21:** Focus measures for the image with more informative unstructured scene.*



***(a)** Noisy SAR image of the structured scene.*　***(b)** TG focus measure with $T = 550$.*

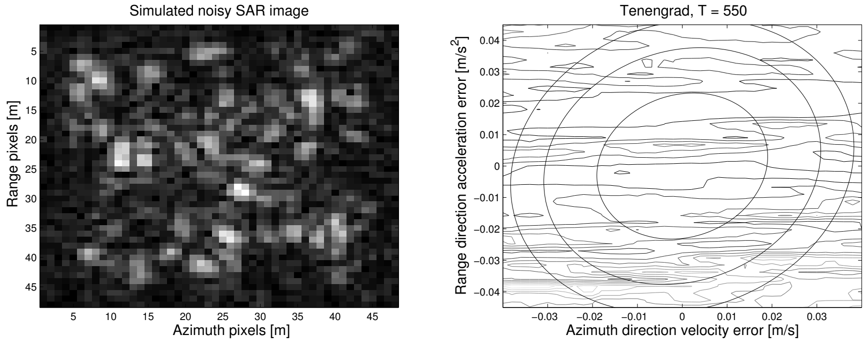


***(c)** SML focus measure with $T = 170$.*　***(d)** DCT focus measure with $T = 5$.*



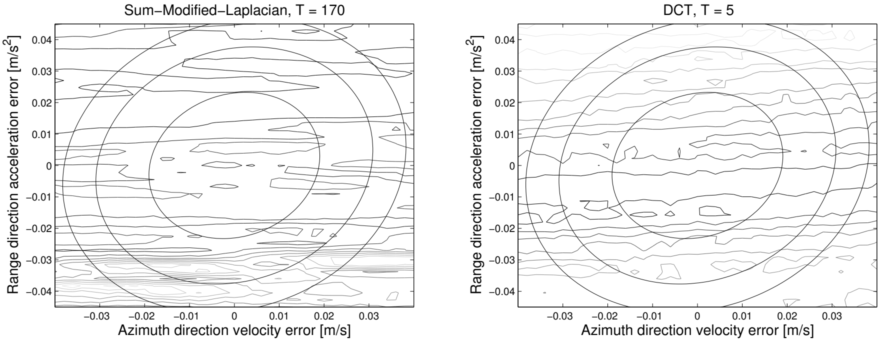
***(e)** Entropy 1 focus measure.*　***(f)** Entropy 2 focus measure.*

Figure 3.22: *Focus measures for the noisy image with more informative structured scene. White Gaussian noise with variance $\sigma^2 = 1.5$ is added on the radar echoes.*



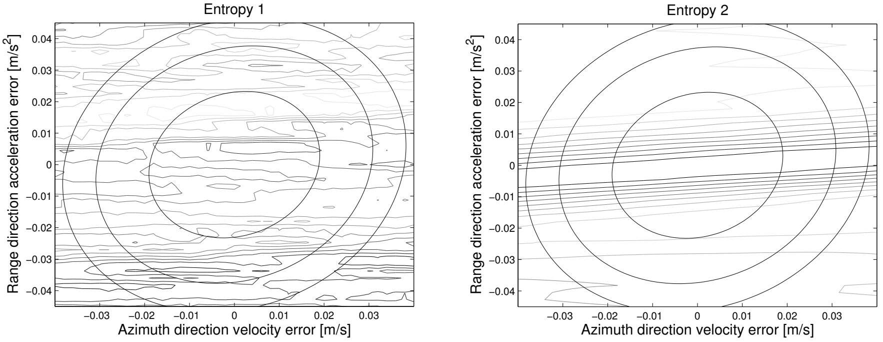
***(a)** Noisy SAR image of the unstructured scene.*

***(b)** TG focus measure with $T = 550$.*



***(c)** SML focus measure with $T = 170$.*

***(d)** DCT focus measure with $T = 5$.*



***(e)** Entropy 1 focus measure.*

***(f)** Entropy 2 focus measure.*

***Figure 3.23:** Focus measures for the noisy image with more informative unstructured scene. White Gaussian noise with variance $\sigma^2 = 1.5$ is added on the radar echoes.*

focus value is obtained if acceleration and velocity errors are both negative as well as if they are both positive. This is most clearly visible in the Entropy 2 measure. In practice this means that many different trajectories give equally focused image. This behavior of the focus measure puts limitations on estimation performance with focus measure only. In this case, the sensor fusion framework, where several different sensors are used, allows us to estimate the platform's states better. In this case inclusion of the IMU will improve the estimation performance.

3.4 Optical Camera

In the previous two sections IMU and SAR principles have been described and measurement functions for these sensors were defined. The last sensor that is used here is a regular optical camera. In the following section a function of an optical camera will be described. The basic projective geometry model of the pinhole camera will be explained, as well as how this model is used to formulate a measurement function used in the sensor fusion framework.

3.4.1 Projective Geometry Model

Generally speaking, an optical camera is a device that creates images of the surrounding environment, much like SAR does. The main difference is that the camera is a passive sensor, i.e., it does not transmit any signal or energy in order to create an image[1]. Instead it receives a visible light and detects the total light energy with an array of photo-sensitive cells. Each cell will be irradiated by the portion of the light from the environment, represented as rays of light, creating the actual image value at this cell. These values are then saved in memory and this is what we call an image. The number of these cells, or pixels, is deciding the image resolution, usually expressed as the amount of pixels in horisontal and vertical directions. For example, so called Video Graphics Array (VGA) format, although originally used to refer to display hardware invented by IBM, is today also used for specific resolution, namely 640×480 pixels, (IBM, 1992). All modern cameras use optics, consisting of one or several lenses to refract the light, i.e., change its propagation path, in order to place the image plane on the right place inside the camera, i.e., on the sensor array. A quite simple model of the lens, thin lens, can be used to explain this mechanism. As depicted in Figure 3.24, rays passing through the point P, and which are parallel with the optical axis, will deflect its path through the lens and pass through the focal point placed on the focal length, f_0, from the lens. In addition, rays passing through the center of the lens, so called Optical Center, OC, will pass unchanged. These two properties of the thin lens will define where image of P, p, will be placed, namely on the intersection of these rays. By changing the focal length we can make the image larger or smaller, which is known as zooming. Note also that the actual image is placed upside-down. With help from the principle of similar triangles we obtain the fundamental equation of the thin lens

$$\frac{1}{f_0} = \frac{1}{Z} + \frac{1}{z} \tag{3.18}$$

---

[1] This is true if we disregard from the flash that many cameras frequently use. However the flash is used to enhance the image quality and is not essential for the image creation

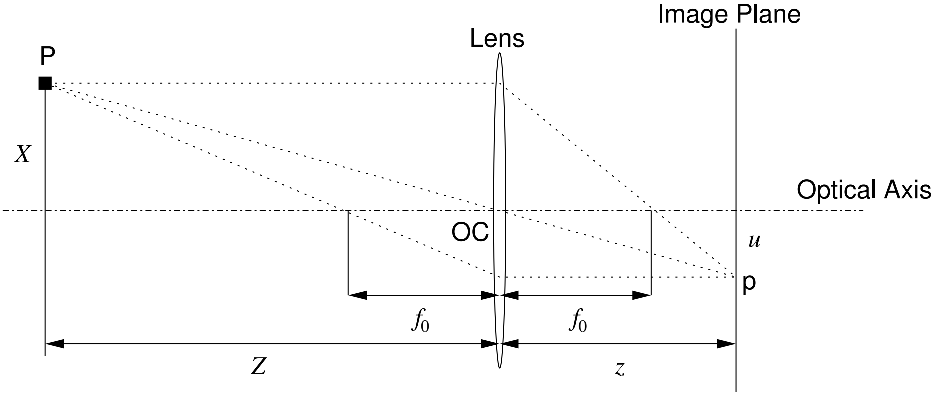


Figure 3.24: *Basic thin lens geometry. Point* P *on a distance Z from the lens is imaged as a point* p *on the image plane placed on a distance z. f_0 is defined as the focal length and* OC *is the Optical Center of the camera.*

With some algebraic transformations of this equation we can obtain the u-coordinate of p as

$$u = -\frac{Xf_0}{Z - f_0} = -\frac{(z - f_0)X}{f_0} \tag{3.19}$$

where the minus sign denotes the upside-down image position.

Although this equation describes a thin lens principle quite accurately, another, and simpler, model will be used to describe the projective nature of the camera. This model is called *Pinhole Camera Model* and is obtained by letting the size of the lens go to zero, (Ma et al., 2004). In that case, only the rays that are going through the optical center will contribute to the image creation. This also implies that f_0 will also go to zero and by rewriting (3.19) as

$$-\frac{Z - f_0}{X} = \frac{f_0}{u} \tag{3.20a}$$

$$-\frac{z - f_0}{u} = \frac{f_0}{X} \tag{3.20b}$$

and letting f_0 go to zero we obtain

$$-\frac{Z}{X} = -\frac{z}{u} = 0 \Rightarrow u = z\frac{X}{Z} \tag{3.21}$$

This is a perfect pinhole camera projective model. In Figure 3.25 this setup is depicted. Here the image plane is actually moved in front of the optical center in order to simplify expressions (minus sign is omitted in that way) and also, somewhat ambiguously, the distance between the optical center and the image plane is called focal length in the pinhole camera case and will be denoted f instead of z. In a two-dimensional image plane, the

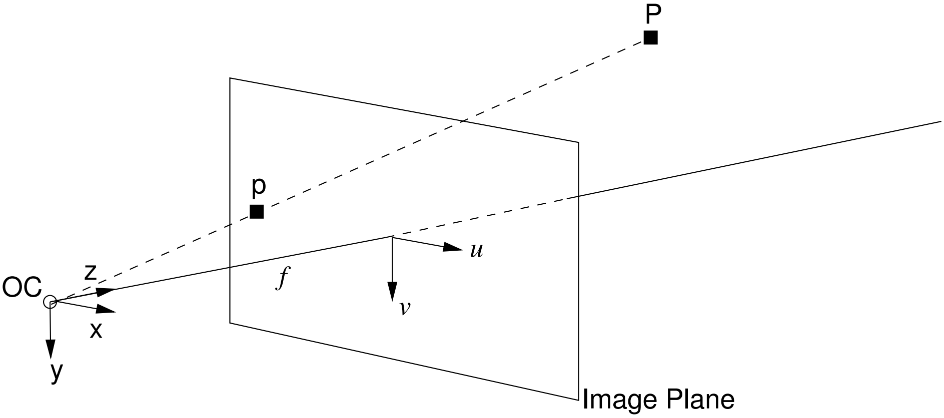


Figure 3.25: *Illustration of the projective geometry of the pinhole camera model.* OC *is the Optical Center of the camera,* p *is the image of the point in 3D,* P*, i.e., its projection in the image plane.* f *is the focal length of the camera, which is a distance between optical center and image plane.* u *and* v *are the normalised pixel coordinates.*

ideal pinhole camera projective model is then

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{3.22}$$

where $[X\ Y\ Z]^T$ are the 3D coordinates of the imaged point P expressed in the camera frame xyz, see Figure 3.25, and $[u\ v]^T$ are the coordinates of the image point p. Often, the focal length of the camera is known, which means that we can compensate for it and obtain the normalised pinhole camera model, where the focal length is equal to one. This model will be defined via an operator $P : \mathbb{R}^3 \to \mathbb{R}^2$ as

$$\begin{bmatrix} u \\ v \end{bmatrix} = P([X^c\, Y^c\, Z^c]^T) = \frac{1}{Z^c} \begin{bmatrix} X^c \\ Y^c \end{bmatrix} \tag{3.23}$$

where we have explicitly written imaged point's coordinates in camera frame, c, and here $[u\ v]^T$ is denoting the normalised coordinates of the image point. The main property of the projective model is its inherent depth ambiguity, i.e., without prior knowledge of the scene, it is impossible to resolve if the observation comes from the large scene that is far away, or from a small scene that is close to the camera. This is similar to the focus measure ambiguity mentioned for the SAR images. This behavior can easily be seen in the structure of (3.23). If X and Y are scaled by a constant, making the scene larger (smaller), then it can be compensated by scaling Z with the same constant making the distance to the scene bigger (smaller). This corresponds to moving along the line OC-P in Figure 3.25. If the scene is unknown and we have several camera observations from different viewpoints, this ambiguity can be resolved, and this is known as triangulation, see for example Hartley and Sturm (1997). However, if both camera positions and the scene are unknown, the

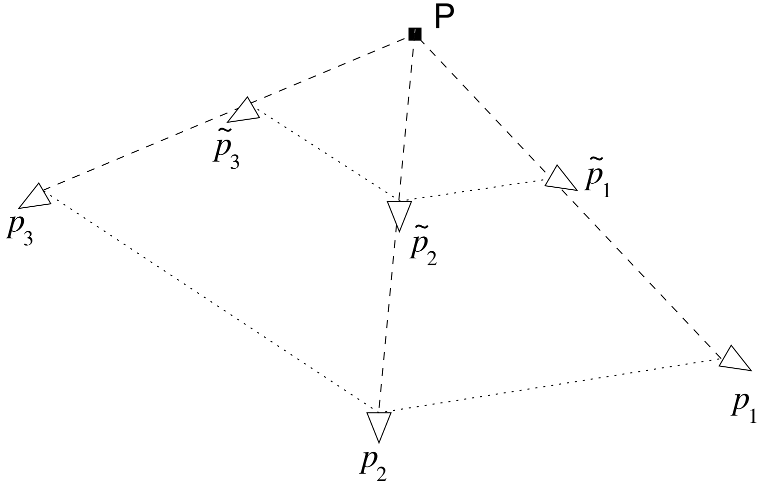


Figure 3.26: *Depth ambiguity of the moving camera. A camera moving fast in positions* $\{p_1,\ p_2,\ p_3\}$ *perceives the point* P *in exactly the same way as a slowly moving camera in positions* $\{\tilde{p}_1,\ \tilde{p}_2,\ \tilde{p}_3\}$.

problem is much more difficult. If camera is moving, the depth ambiguity manifests itself as a possible speed scaling, which in turn causes position scaling. This can be seen in Figure 3.26 where a camera located in positions $\{p_1,\ p_2,\ p_3\}$ gives the same images of the point P as a camera positioned in $\{\tilde{p}_1,\ \tilde{p}_2,\ \tilde{p}_3\}$ but these two cameras are moving with different speeds. In order to resolve this ambiguity, we need additional measurements of the motion. As mentioned in Section 3.3.4, exactly as for the SAR, an IMU can provide that kind of measurements and the sensor fusion framework gives the means necessary to solve the problem.

The normalised coordinates above assume that the actual image has infinite resolution, and they are expressed in metric units, e.g., meters. In practice, the actual digital image consists of discrete image elements, pixels, with coordinates $[p_x\ p_y]^T$ in the image plane. In order to use the projective model we need to transform the pixel coordinates to the normalised coordinates. This will be done by an affine model, involving scaling and shifting. Scaling is done to transform from pixel size to metric size and shifting is done because the normalised coordinates have their origin in the middle of the image plane, i.e., the point where optical axis and image plane intersect, while the pixel coordinates have their origin in the upper-left corner of the image. If these, so called intrinsic camera parameters, are given, the total transformation can be written as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{f}\left(\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}\begin{bmatrix} p_x \\ p_y \end{bmatrix} - \begin{bmatrix} s_x o_x \\ s_y o_y \end{bmatrix}\right) \tag{3.24}$$

where s_x and s_y are the scaling parameters (in m/pixel) and o_x and o_y are the coordinates of the middle of the image plane (in pixels). The intrinsic camera parameters can be estimated, for a particular camera, with readily available software, see for example

Bouguet (2010). This model assumes that there are no nonlinear distortions caused by the lens system for example, but even that kind of calibration is possible to perform giving a slightly more complicated model, see e.g., Ma et al. (2004). In the model (3.23) the 3D coordinates are referenced to the camera frame, but usually both platform's states and imaged environment points are referenced to the local navigation frame. For that reason we need to transform these coordinates to the camera coordinates. Just as in the IMU case, this transformation is performed as

$$m^c = \mathsf{R}^{cn}(m^n - p^n) \tag{3.25}$$

where $m^c = [X^c \; Y^c \; Z^c]^T$ are the environment point's, called landmark, coordinates in the camera frame, c, R^{cn} is the rotation matrix between navigation and camera frame, m^n is the coordinates of the landmark expressed in the navigation frame and, as before, p^n is the position of the platform in the navigation frame. The resulting measurement equation, for one landmark, can finally be written as

$$y_t^m = \begin{bmatrix} u_t \\ v_t \end{bmatrix} = P(\mathsf{R}_t(m - p_t)) + e_t^m = h^m(x_t, m) + e_t^m \tag{3.26}$$

where the frame notation has been omitted, and we added the time dependence and noise which is, as before, assumed to be white and Gaussian, $e_t^m \sim \mathcal{N}(0, R_t^m)$. The Equation (3.26) can be generalised for several landmarks by stacking all the measurements of the corresponding M landmarks as in

$$\underbrace{\begin{bmatrix} u_t^1 \\ v_t^1 \\ \vdots \\ u_t^{\mathsf{N}_{y_t}} \\ v_t^{\mathsf{N}_{y_t}} \end{bmatrix}}_{y_t^m} = \underbrace{\begin{bmatrix} P(\mathsf{R}_t(m^{c_t^1} - p_t)) \\ \vdots \\ P(\mathsf{R}_t(m^{c_t^{\mathsf{N}_{y_t}}} - p_t)) \end{bmatrix}}_{h_t(x_t, \mathsf{M}_t)} + e_t^m \tag{3.27}$$

where N_{y_t} denotes the number of measurements at time t and the correspondence variables, c_t^i, encodes the measurement-landmark assignment, $y_t^i \leftrightarrow m^j$, which gives a subset of the landmarks at time t, $\mathsf{M}_t = \{m^j\}$, $j \in \{1, \ldots, M | \, c_t^i = j\}$. This relation can be further generalised to collect the measurements for all the times, $1 : N$, by stacking (3.27) to obtain

$$\underbrace{\begin{bmatrix} y_1^m \\ \vdots \\ y_N^m \end{bmatrix}}_{y_{1:N}} = \underbrace{\begin{bmatrix} h_1(x_1, \mathsf{M}_1) \\ \vdots \\ h_N(x_N, \mathsf{M}_N) \end{bmatrix}}_{h_{1:N}(x_{0:N}, \mathsf{M})} + \underbrace{\begin{bmatrix} e_1^m \\ \vdots \\ e_N^m \end{bmatrix}}_{e_{1:N}} \tag{3.28}$$

In this way, a measurement function, or likelihood function, on the form (2.29) is obtained, where all the M landmarks $m^1, \ldots, m^M$ are collected into the parameter M. This parameter serves as a map of the environment that is observed by the camera. This representation is conceptually different from the map obtained with SAR in a sense that the camera based map is parametric, with point landmarks, while the SAR based map is simply an image obtained with nonparametric methods. In any case, since both kind of maps can be related to the platform's states, we can use similar estimation methods for both the states and the map.

3.4.2 Image Feature Extraction

In the previous section it was explained how to obtain the normalised image coordinates from the pixel coordinates and how to relate the normalised coordinates to the platform's states. In this section the techniques how to obtain interesting points in the images, represented by pixel coordinates and called image features, will be introduced.

As already mentioned, the observation model for the camera is parametric, meaning that the environment is encoded with discrete points in three dimensions, landmarks. The landmarks are, except in very special occasions, not given in advance, implying that they must be found and initialised during the estimation, i.e., we must use the images to extract the landmarks. In practice, this means that some of the points in images will be remembered in order to track them across several images and in that way establish the measurements of the environment needed for the estimation. The ability to track the landmarks in several images puts some requirements on which points in the images to choose. So the task for the feature extractor is to find image points that are relatively easy to find in as many images as possible. One way of accomplishing that is to find corners in the image. A reason the corner points are a good choice is that they have a distinct gradient change in all directions around a corner point. A simple edge would have change in gradient in only one direction (across edge) and a flat region has no change in gradient, as expected. By image gradient, we consider e.g., Sobel gradient calculated as in (3.11). A simple way to measure this response is to build an approximation to Hessian matrix in each pixel as in

$$H_{ij} = \begin{bmatrix} \sum_w (S^r)^2 & \sum_w S^r S^c \\ \sum_w S^r S^c & \sum_w (S^c)^2 \end{bmatrix} \tag{3.29}$$

where the sum is taken over a window, w, centered on pixel (i, j). The size of the window is a parameter that is free to choose and it determines the size of the surrounding where a feature is to be determined. Usually this window is combined with some low-pass kernel, e.g., a Gaussian kernel. By taking a simple transformation of the matrix above

$$C_{ij} = \det(H_{ij}) - \kappa \operatorname{Tr}(H_{ij})^2 \tag{3.30}$$

where $\det(\,\cdot\,)$ denotes the determinant and $\operatorname{Tr}(\,\cdot\,)$ the trace of the matrix, we obtain a function called Harris strength. The scalar κ is seen as a tuning parameter which governs the sensitivity between corner and edges. The Harris strength will have a high value for the pixels where the corner is located and we can detect corners simply by seeking the local maxima of C. This is exactly the principle behind a very popular Harris corner detector (Harris and Stephens, 1988) and many implementations are available. As an illustration of the Harris corner detector, consider the image in Figure 3.27. A resulting Harris strength and 30 corner features are shown in Figure 3.28.

Although simple to understand and implement, the Harris corner detector provides us only with the corner positions. In order to find the same corner in another image, we need some kind of corner identity as well. Since the Harris detector does not provide that, a common

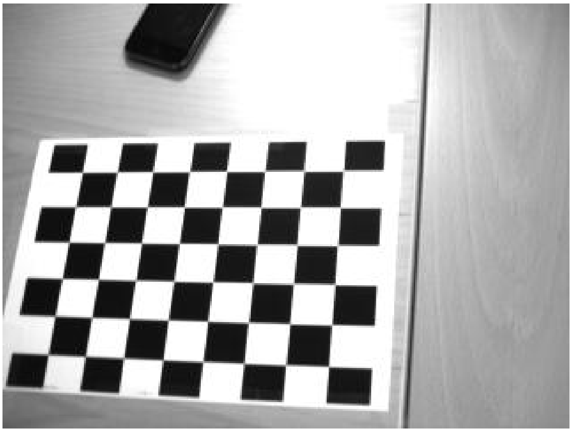
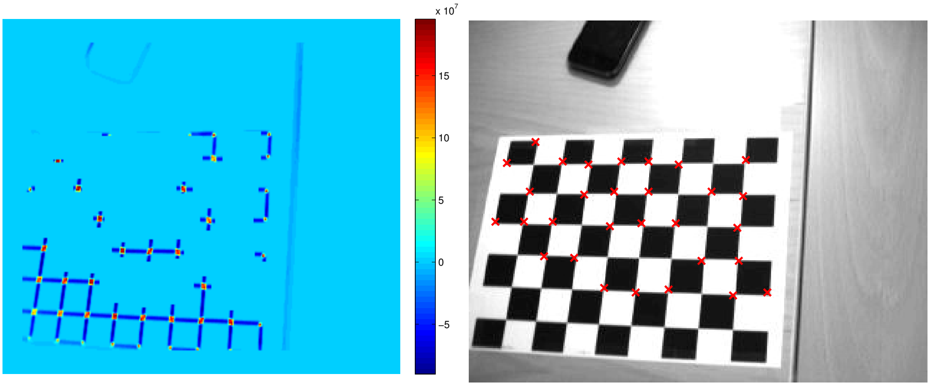


Figure 3.27: *Example image for the illustration of the Harris corner detector.*



(a) *Harris strength for the image in Figure 3.27.*

(b) *30 Harris corners for the image in Figure 3.27.*

Figure 3.28: *Harris strength and features for the example image.*

approach is to use a small image patch around the corner as a feature identity. This allows for feature matching by a correlation method, for example. This method works well for pure translational movement and without changing the scale of the image. To overcome that limitation, other feature detectors, that have scale and rotation invariance, have been proposed, and the most popular ones are Scale Invariant Feature Transform (SIFT) (Lowe, 1999) and Speeded Up Robust Features (SURF) (Bay et al., 2006). These feature detectors fall into the class of blob-detectors and use Difference of Gaussians (DoG) or determinant of Hessian (DoH) approach to detect key-points at different image scales. The key-points are chosen, just as in Harris detector case, as local maxima of the DoG or DoH operators. In addition to the invariance to rotation and scaling, these detectors also provide a feature descriptor, i.e., a feature identity that is needed. For SIFT it is a 128 dimensional vector which is composed of accumulated image gradients from small image regions in the direct surrounding of the key-point. In this way, a compact descriptor representation is obtained and it allows us to use it for feature matching across the images.

Given the feature detector of our choice, together with the measurement model (3.28), we can now use the camera as a sensor in order to estimate the navigation and environment parameters.

4

Concluding Remarks

This chapter concludes the whole thesis by summarising the main conclusions from the publications included in Part II, in a “bird-eye view” fashion, and giving some suggestions about future work. For more detailed treatment of the conclusions and the future work the reader is referred to each paper.

4.1 Conclusions

In this work, a sensor fusion and optimisation based estimation approach to navigation and mapping, also called SLAM, is presented. The approach is mainly aimed for applications to unmanned aerial vehicles and, as such, it is using sensors that are available on these platforms. Furthermore, the chosen sensors are independent on the external infrastructure, like satellites in case of GNSS. The benefit this sensor setup gives is the higher availability of the UAV in the cases where the external infrastructure cannot be assumed available. In particular, the sensors that are used here are the inertial and imaging sensors measuring the UAV’s ego-motion and the surrounding environment, respectively. Inertial sensors considered here are the accelerometers and gyros, while imaging sensors are the optical camera and SAR. In the SAR case, images can also be used to obtain the global UAV position by matching these to a map with known global coordinates.

The proposed methods are implemented and demonstrated on both simulated and real data. The results obtained for both considered imaging sensors are showing that they, in combination with IMU, can be used for navigational purposes as well as for mapping of the surroundings. Since the nature of the errors in the imaging sensors are different from the inertial errors, i.e., velocity and position ambiguity versus drift, they serve as a good complement to each other producing metrically correct solutions, i.e., resolving the ambiguities, and reducing the drift caused by integrating the IMU errors.

The structure of the SLAM problem can be exploited, mostly for the inertial/visual and partly for inertial/SAR case, and together with the optimisation formulation used to efficiently obtain the smoothed estimate of the interesting parameters. In the inertial/visual case the problem is formulated as nonlinear least squares for which efficient algorithms can be used. Furthermore, for the inertial/visual case, certain conditional structure can also be used to obtain a good initial solution of the parameters. This is highly desirable since bad initialisation of the nonlinear optimisation problem may lead to convergence to local minima.

For the inertial/SAR case, both sequential and batch formulations of the navigation and SAR image focusing are proposed. For the batch formulation an efficient calculation method of the optimisation-criterion gradient is devised and a quasi-Newton method can be applied to solve the optimisation problem in fewer iterations than only using pure gradient search. In the sequential solution an Extended Kalman Filter is used with a system model where range gradient is estimated from the SAR raw data and used as a measurement function. The optimisation formulation of the joint navigation and image focusing is further successfully extended by using maps to find global position and flight direction of the UAV.

The SAR technique has also been applied to cellular phone time-of-arrival data to estimate the non-line-of-sight signal propagation in urban environment. This can also be viewed as mapping of the dominant reflector environment. In particular, a multistatic extension of the basic SAR back-projection method has been applied to real cellular phone data and preliminary results show good potential.

4.2 Future Work

A quite broad area that involves navigation and mapping has been treated in this thesis. In most cases the optimisation framework has been used to estimate the parameters, which usually represent navigational states and some parametrisation of the environment map. Currently, a naïve implementation of these problems is done where standard MATLAB generic optimisation functions are used. By exploiting the structure of the problems, tailored solvers could be implemented which potentially can make execution more efficient from both the execution time and memory usage.

Another interesting option that can be explored is the Moving Horizon Estimation (MHE) where a smoothed estimate within a fixed-length window is obtained and the optimisation methods proposed here can easily be adapted for MHE. This would also give a decreased execution time, since the window size can be kept constant which means the the problem size is also approximately constant.

Bibliography

L. E. Andersson. On the Determination of a Function from Spherical Averages. *SIAM Journal on Mathematical Analysis*, 19(1):214–232, 1988. doi: 10.1137/0519016. URL `http://link.aip.org/link/?SJM/19/214/1`.

T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, Septemeber 2006.

H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. In *Proceedings of European Conference on Computer Vision ECCV2006*, pages 404–417, May 2006.

T. Bayes. An Essay towards solving a Problem in the Doctrine of Chances. *The Philosophical Transactions of the Royal Society of London*, 53:370–418, 1763.

J. Y. Bouguet. Camera Calibration Toolbox for Matlab. `www.vision.caltech.edu/bouguetj/calib_doc/`, 2010.

C. Cafforio, C. Prati, and F. Rocca. SAR data focusing using seismic migration techniques. *IEEE Transactions on Aerospace and Electronic Systems*, 27(2):194 –207, March 1991. ISSN 0018-9251. doi: 10.1109/7.78293.

L. J. Cutrona, W. E. Vivian, E. N. Leith, and G. O. Hall. A high-resolution radar combat-surveillance system. *IRE Transactions on Military Electronics*, MIL-5(2):127–131, April 1961. ISSN 0096-2511. doi: 10.1109/IRET-MIL.1961.5008330.

H. Durrant-Whyte and T. Bailey. Simultaneous Localization and Mapping: Part I. *IEEE Robotics & Automation Magazine*, 13(12):99–110, June 2006.

J. Farrell and M. Barth. *The global positioning system and inertial navigation.* McGraw-Hill Professional, 1999.

J. A. Fawcett. Inversion of N-Dimensional Spherical Averages. *SIAM Journal on Applied Mathematics*, 45(2):336–341, 1985. ISSN 00361399. URL `http://www.jstor.org/stable/2101820`.

R. Ferzli and L. J. Karam. No-reference objective wavelet based noise immune image sharpness metric. In *Proceedings of International Conference on Image Processing, ICIP 2005*, volume 1, pages I–405–8, Septemeber 2005.

R. A. Fisher. On an absolute criterion for fitting frequency curves. *Messenger of Mathematics*, 41:155–160, 1912.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings of Radar and Signal Processing*, 140(2):107 –113, April 1993. ISSN 0956-375X.

Global Positioning System Standard Positioning Service Performance Standard. GPS, US Government, 2008.

F. Gustafsson. *Statistical Sensor Fusion.* Studentliteratur, Lund, 2010.

C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of Fourth Alvey Vision Conference*, pages 147–151, 1988.

R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68 (2):146 – 157, 1997. ISSN 1077-3142.

H. Hellsten and L. E. Andersson. An inverse method for the processing of synthetic aperture radar data. *Inverse Problems*, 3(1):111, 1987. URL `http://stacks.iop.org/0266-5611/3/i=1/a=013`.

IBM VGA Technical Reference Manual. IBM, 1992.

T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation.* Prentice-Hall, Upper Saddle River, New Jersey, 2000.

R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

M. Kristan, J. Pers, Perse M., and S. Kovacic. A Bayes-spectral-entropy-based measure of camera focus using a discrete cosine transform. *Pattern Recognition Letters*, 27(13): 1431 – 1439, 2006. ISSN 0167-8655. doi: DOI:10.1016/j.patrec.2006.01.016.

J. B. Kuipers. *Quaternions and Rotation Sequences.* Princeton University Press, 1999.

R. Larsson, Z. Sjanic, M. Enqvist, and L. Ljung. Direct Prediction-error Identification of Unstable Nonlinear Systems Applied to Flight Test Data. In *Proceedings of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 2009.

D. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV99)*, pages 1150–1157, Corfu, Greece, September 1999.

Y. Ma, S. Soatto, J. Košecká, and S. S. Sastry. *An Invitation to 3-D Vision.* Springer Science+Business Media, LLC, 2004.

J. L. Meriam and L. G. Kraige. *Engineering Mechanics, Dynamics, 4th ed.* John Wiley & Sons, 1998. ISBN 0-471-24167-9.

A. S. Milman. SAR Imaging by Omega-K Migration. *International Journal of Remote Sensing*, 14(10):1965–1979, 1993.

R. L. Jr. Morrison and D. C. Jr. Munson. An experimental study of a new entropy-based SAR autofocus technique. In *Proceedings of International Conference on Image Processing, ICIP 2002*, volume 2, pages II–441–4, September 2002. doi: 10.1109/ICIP.2002.1039982.

F. Natterer. *The Mathematics of Computerised Tomography*. New York: Wiley, 1986.

C. Oliver and S. Quegan. *Understanding Synthetic Aperture Radar Images*. The SciTech Radar and Defense Series. SciTech, 2004. ISBN 1-891121-31-6.

F. Rocca. Synthetic Aperture Radar: a New Application for Wave Equation Techniques. *Stanford Exploration Project SEP-56*, pages 167–189, 1987. URL `http://sepwww.stanford.edu/oldreports/sep56/56_13.pdf`.

M. D. Shuster. A survey of attitude representations. *The Journal of Astronautical Sciences*, 41(4):439–517, October 1993.

Z. Sjanic and F. Gustafsson. Simultaneous Navigation and SAR Auto-focusing. In *Proceedings of 13th International Conference on Information Fusion*, Edinburgh, UK, July 2010.

Z. Sjanic and F. Gustafsson. Navigation and SAR Auto-focusing Based on the Phase Gradient Approach. In *Proceedings of 14th International Conference on Information Fusion*, Chicago, USA, July 2011.

Z. Sjanic and F. Gustafsson. Fusion of Information from SAR and Optical Map Images for Aided Navigation. In *Proceedings of 15th International Conference on Information Fusion*, Singapore, July 2012.

Z. Sjanic and F. Gustafsson. Simultaneous Navigation and Synthetic Aperture Radar Focusing. *Provisionally accepted to IEEE Transactions on Aerospace and Electronic Systems*, August 2013a.

Z. Sjanic and F. Gustafsson. Navigation and SAR focusing with Map Aiding. *Submitted to IEEE Transactions on Aerospace and Electronic Systems*, June 2013b.

Z. Sjanic, M. A. Skoglund, T. B. Schön, and F. Gustafsson. A Nonlinear Least-Squares Approach to the SLAM Problem. In *Proceedings of 18th IFAC World Congress*, Milano, Italy, August/Septemeber 2011.

Z. Sjanic, F. Gunnarsson, C. Fritsche, and F. Gustafsson. Cellular Network Non-Line-of-Sight Reflector Localisation Based on Synthetic Aperture Radar Methods. *Submitted to IEEE Transactions on Antennas and Propagation*, September 2013a.

Z. Sjanic, M. A. Skoglund, and F. Gustafsson. Expectation-Maximisation Maximum Likelihood Estimation for Inertial/Visual SLAM. *Submitted to IEEE Transactions on Robotics*, September 2013b.

M. A. Skoglund, Z. Sjanic, and F. Gustafsson. Initialisation and Estimation Methods for Batch Optimisation of Inertial/Visual SLAM. *Submitted to IEEE Transactions on Robotics*, September 2013.

L. M. H. Ulander, H. Hellsten, and G. Stenstrom. Synthetic-aperture radar processing using fast factorized back-projection. *IEEE Transactions on Aerospace and Electronic Systems*, 39(3):760–776, July 2003. ISSN 0018-9251. doi: 10.1109/TAES.2003.1238734.

D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and C. V. Jr. Jakowatz. Phase gradient autofocus - a robust tool for high resolution SAR phase correction. *IEEE Transactions on Aerospace and Electronic Systems*, 30(3):827–835, July 1994. ISSN 0018-9251. doi: 10.1109/7.303752.

J. L. Walker. Range-Doppler Imaging of Rotating Objects. *IEEE Transactions on Aerospace and Electronic Systems*, AES-16(1):23–52, 1980. ISSN 0018-9251. doi: 10.1109/TAES.1980.308875.

L. Xi, L. Guosui, and J. Ni. Autofocusing of ISAR images based on entropy minimization. *IEEE Transactions on Aerospace and Electronic Systems*, 35(4):1240–1252, October 1999. ISSN 0018-9251. doi: 10.1109/7.805442.

M. Xing, R. Jiang, X.and Wu, F. Zhou, and Z. Bao. Motion Compensation for UAV SAR Based on Raw Radar Data. *IEEE Transactions on Geoscience and Remote Sensing*, 47 (8):2870–2883, August 2009. ISSN 0196-2892. doi: 10.1109/TGRS.2009.2015657.

A. F. Yegulalp. Minimum entropy SAR autofocus. In *7th Adaptive Sensor Array Processing Workshop*, March 1999.

Part II

Publications

Paper A

Simultaneous Navigation and Synthetic Aperture Radar Focusing

Authors: Zoran Sjanic and Fredrik Gustafsson

Edited version of the paper:

Z. Sjanic and F. Gustafsson. Simultaneous Navigation and Synthetic Aperture Radar Focusing. *Provisionally accepted to IEEE Transactions on Aerospace and Electronic Systems*, August 2013a.

Preliminary version:

Technical Report LiTH-ISY-R-3063, Dept. of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden.

Simultaneous Navigation and Synthetic Aperture Radar Focusing

Zoran Sjanic and Fredrik Gustafsson

Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
`{zoran,fredrik}@isy.liu.se`

Abstract

Synthetic Aperture Radar (SAR) equipment is a radar imaging system that can be used to create high resolution images of a scene by utilising the movement of a flying platform. Knowledge of the platform's trajectory is essential to get good and focused images. An emerging application field is real-time SAR imaging using small and cheap platforms with poorer navigation systems implying unfocused images. This contribution investigates a joint estimation of the trajectory and SAR image.

1 Introduction

A general method for creating high resolution radar images from low resolution radar data, or real aperture images, is to use relative motion between radar antenna and the imaged scene and integrate all the partial real aperture images taken along the flown trajectory (Cutrona et al., 1961). Traditionally, this operation is performed in the frequency domain using FFT like methods, e.g., the Fourier-Hankel method (Fawcett, 1985; Hellsten and Andersson, 1987; Andersson, 1988) or the ω-K migration methods (Cafforio et al., 1991; Rocca, 1987; Milman, 1993). The common denominator of these methods is that they assume that the aircraft's (or antenna's) flown path is linear and that is generally not the case in practice. If the trajectory is not linear the integration will result in an unfocused image. It is possible to partly correct for the deviation from the nonlinear trajectory but then the methods become computationally inefficient. Another approach is to perform integration in time domain by means of solving the back-projection integral (Natterer, 1986).

This process is illustrated in Figure 1 where two points are imaged. Each column of the low resolution real aperture radar image on the bottom is back-projected to the sub images, which means that column is mapped to the two-dimensional sub image. These sub images are in turn summed to the final synthetic aperture radar image on the top. The simulation plots of this setup is also depicted in Figure 2. Even in this process it is assumed that the radar antenna's flown path is linear with constant altitude and heading, but the method can be extended to non-linear tracks as well. However exact inversion is

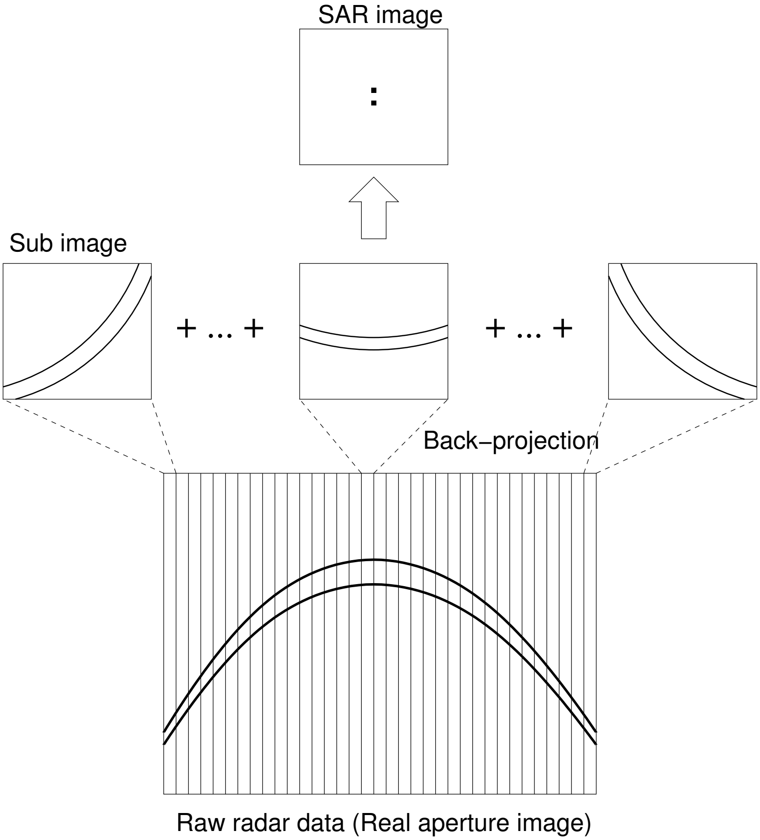
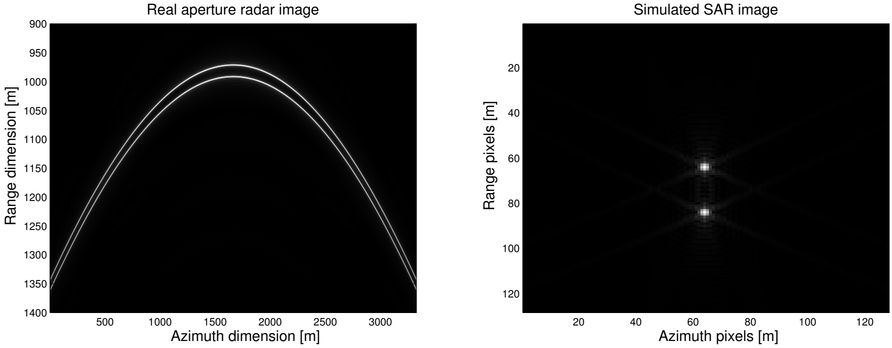


Figure 1: Illustration of the back-projection method for creating of the SAR images. Figure is not to scale.

not guaranteed. The main disadvantage of this method is the large amount of operations required to create an image, where the complexity is proportional to $\mathcal{O}(NKM)$ for $K \times M$ pixels image using an aperture with N positions. However, by means of coordinate transformation, an approximation to exact back-projection can be performed, which is called Fast Factorised Back-projection, see Ulander et al. (2003). The complexity of this algorithm is proportional to $KM \log N$ operations, which for large N implies an important saving. With this faster algorithm it should be possible to create images in real time, possibly in dedicated hardware. Since back-projection algorithms are dependent on knowledge of the antenna's position in order to get focused images, the image focus can be measured and used for estimation of the trajectory.

An example of this is depicted in Figure 3, where two point targets are imaged. In Figure 3a, a linear path is simulated, which results in a perfectly focused image. In the other three images the variation in range position was added as $\mathcal{N}(0, \sigma^2)$ where $\sigma = \{0.5, 1.5, 3\}$ $[\mathrm{m}]$ and the images are created with an assumption that the path was linear. This gives unfocused images as depicted. Traditional methods for auto-focusing are



*(**a**) Real aperture radar image of the two points.*

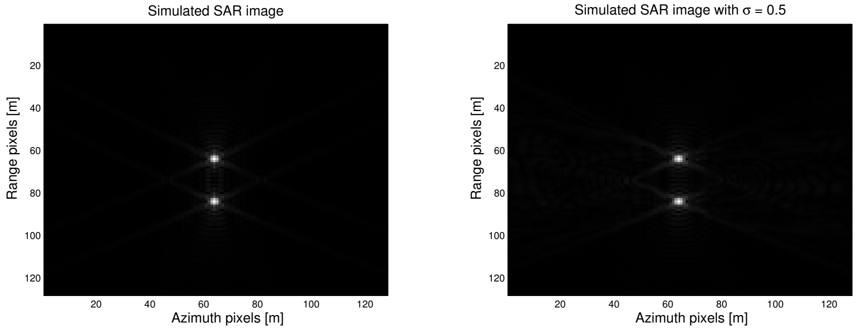
*(**b**) Synthetic aperture radar image of the two points.*

Figure 2: *Real and synthetic aperture radar images of the two point scene.*

mostly open-loop type methods where either SAR images or raw radar data are used, for example Phase Gradient Auto-focus (PGA), (Oliver and Quegan, 2004; Wahl et al., 1994; Xing et al., 2009; Fienup, 1989; Morrison and Munson, 2002). The significant common denominator for all these methods is that the image is created with assumptions on linear flight trajectory and focusing is done afterwards in an open-loop way discarding eventual flight path information. This is a consequence of the off-line image generating process where the trajectory is no longer interesting.

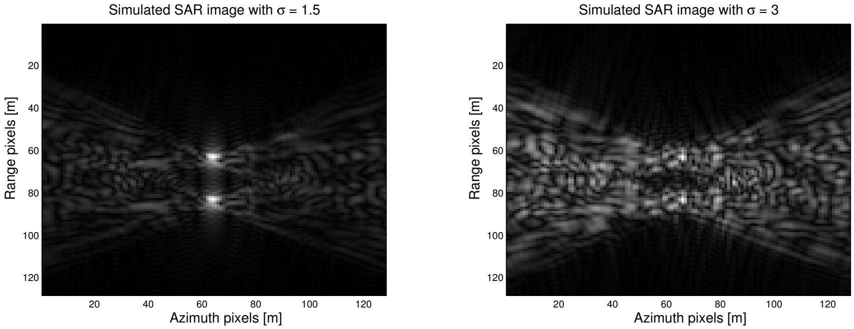
In the setup where SAR images are generated on-line an idea is to use information from the image focus and navigation system, like measured accelerations, together and in a sensor fusion framework try to obtain the best solution to both image focusing and navigation simultaneously. In the view of this approach for SAR images, the problem is closely related to the Simultaneous Localisation and Mapping (SLAM), (Durrant-Whyte and Bailey, 2006; Bailey and Durrant-Whyte, 2006), where a map of the unknown environment is estimated jointly with the platform's position. The SLAM problem has been well studied during recent years and many different solution methods have been proposed. One method that has been quite successful is to solve the SLAM problem in the sensor fusion framework. In the SAR application, the map of the environment from SLAM, is the unknown scene that is imaged and can be seen as the two dimensional map of point reflectors. The problem of positioning the platform is the same in SLAM. However, the main difference is that we consider a non-parametric SAR image rather than a parametric map of point reflectors, that would be a too restrictive assumption in SAR imaging. That is, though there are many conceptual similarities of joint navigation and mapping, the state of the art algorithms cannot be applied here.

This contribution applies a sensor fusion framework, where the SAR image together with a focus measure is interpreted as a "sensor" that contains information about the position of the platform. The image creating and auto-focusing methods described above can be illustrated in Figure 4. The method based on the sensor fusion can be implemented in



***(a)** Focused SAR image of two point targets.*

***(b)** Unfocused SAR image of two point targets with $\sigma = 0.5$.*



***(c)** Unfocused SAR image of two point targets with $\sigma = 1.5$.*

***(d)** Unfocused SAR image of two point targets with $\sigma = 3$.*

***Figure 3:** Example SAR images with different perturbed trajectories.*

a centralised or decentralised manner. In this work we are focused on the decentralised manner.

The outline is as follows. Section 2 summarises notation and makes a high-level mathematical formulation of the approach. Section 3 introduces the navigation framework and system and measurement models used. Section 4 describes the image focus measures that will be used in the auto-focus procedure. In Section 5 an optimisation framework and methods are introduced and their usage is explained. Numerical examples for the simulated images are covered in Section 6 and for the real SAR data in Section 7. Finally, conclusions and future work are discussed in Section 8.

2 Notation and Problem Formulation

Let the complex raw radar data be denoted $z_t(R)$, where t is the time index, and where $z_t(R)$ denotes the returned radio energy corresponding to distance R to the scene from

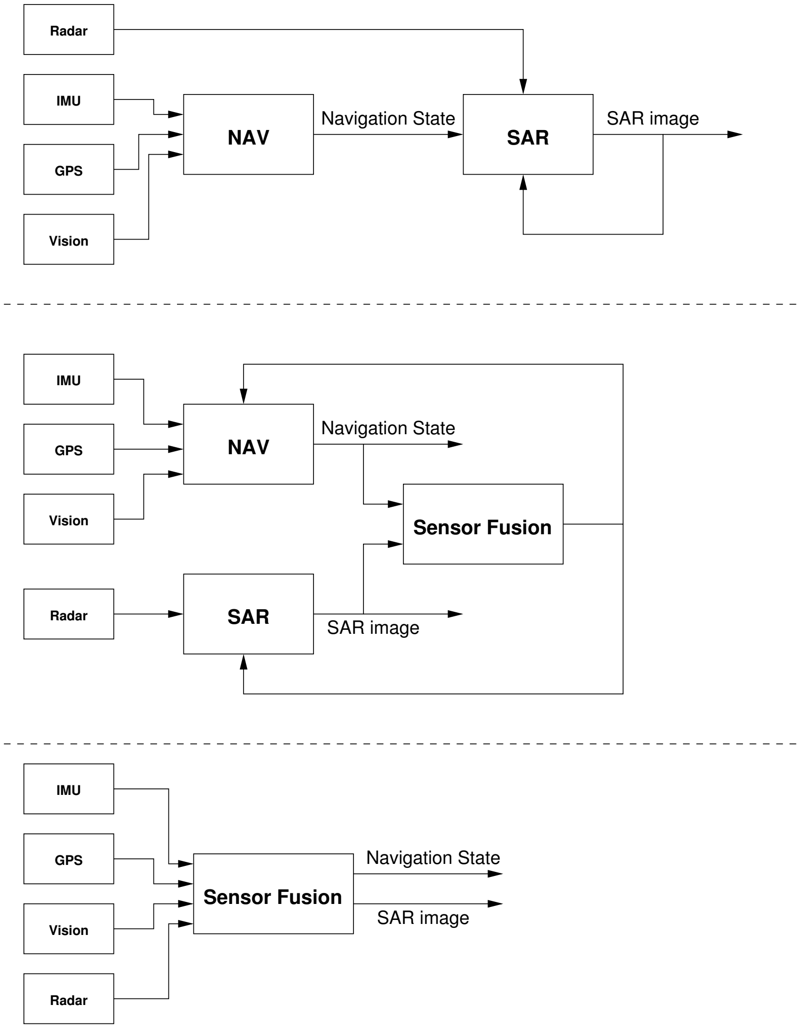


Figure 4: *Top: SAR architecture where navigation data is used in an open-loop manner. Middle: SAR architecture where navigation and SAR data is used together in a decentralised sensor fusion framework. Bottom: SAR architecture where navigation and SAR data is used together in a centralised sensor fusion framework.*

the radar. Further, let x_t denote the state vector of the platform, which includes position and orientation (radar pose). The back-projection method of producing the images, see Figure 1, can alternatively be expressed as integration per image pixel. For each pixel (i, j) in a complex valued image I, the total energy from each radar pulse is integrated by summing all the values from the raw data given the range from the platform to the point in the scene corresponding to the pixel (i, j). This can be expressed as

$$I_{ij} = \sum_{t=1}^{N} z_t(\|p_t - s_{ij}\|_2) \tag{1}$$

where p_t is the 3D position of the radar and s_{ij} is the coordinate of the scene point that is mapped to the pixel (i, j) in the image. Now, for a SAR system on the UAV platform, the pose cannot be assumed to be known. Instead, we have access to an estimated state $\hat{x}_t$, and the estimated SAR image becomes

$$\widehat{I}_{ij} = \sum_{t=1}^{N} z_t(\|\hat{p}_t - s_{ij}\|_2). \tag{2}$$

This estimated SAR image will be out of focus, since all the contributions from raw data will now be scattered due to the error in position estimate, see Figure 3 for an example of this.

The key idea in this contribution is to perform a parametric focusing. To enable this, we will make use of a focus measure $F(\widehat{I})$, with the property that

$$F(\widehat{I}) > F(I^o), \quad \widehat{I} \neq I^0, \tag{3a}$$

$$x^0_{1:N} = \arg\min_{x_{1:N}} F(\widehat{I}) \tag{3b}$$

where I^0 denotes the true SAR image and $x^0_{1:N}$ the true state sequence.

We will, however, not optimise the focus w.r.t. the pose blindly. We will optimise focus jointly with the filtering problem, in that the states obey the state dynamics and observations as well as possible.

As already noted, building up the image of size KM pixels with an aperture of N time points requires a huge computational effort ($\mathcal{O}(NKM)$). It may seem that an outer loop that performs focusing will increase the computational burden at least an order of magnitude more. However, we will show that the gradient of the focus measure can be computed efficiently, using the chain rule

$$\frac{\partial F(x)}{\partial x} = \frac{\partial F}{\partial I}\frac{\partial I}{\partial R}\frac{\partial R}{\partial x}. \tag{4}$$

The first and third factor can be derived analytically, while the second one can be computed from the already computed sub-images with only a little overhead.

In order to evaluate the performance of the estimation methods, some performance measures are needed. A popular measure for the parameter estimate is Root Mean Square Error (RMSE) defined as

$$\text{RMSE}(\hat{\theta}) = \sqrt{\frac{\sum_{k=1}^{N}(\hat{\theta}_k - \theta^0)^2}{N}} \tag{5}$$

where $\hat{\theta}_1, \ldots, \hat{\theta}_N$ are the unbiased estimates of the true scalar parameter θ^0. To assess the quality of the obtained SAR images, the power of the error image can be used. This can be defined as

$$\widetilde{P} = \frac{\sum_{i=1}^{K}\sum_{j=1}^{M}|\widehat{I}_{ij} - I_{ij}^0|^2}{KM} \tag{6}$$

where $\widehat{I}$ is the $K \times M$ complex SAR image obtained with the estimation procedure and I^0 is the perfect focused SAR image, i.e., created with the true trajectory.

3 Navigation Framework

An Inertial Navigation System (INS) in an aircraft integrates accelerometer and gyroscope data and corrects the state with aiding sensors such as barometer and GPS using a general dynamics and measurement equations

$$\begin{aligned} x_{t+1} &= f(x_t, w_t) && \text{(7a)} \\ y_t &= h(x_t) + e_t && \text{(7b)} \end{aligned}$$

where x_t are states of the system, w_t denotes the process noise with variance Q_t, e_t is measurement noise with variance R_t and y_t are the measurements. Usually an Extended Kalman filter is applied to estimate the states, see e.g., Farrell and Barth (1999). In this work, a simplified, yet useful, model of the dynamics will be assumed which will give simpler expressions in the algorithms.

Parameter	Accuracy (1-σ)	Stat. acc. (1-σ)
Position	$3\ \mathrm{m}$	$0.093\ \mathrm{m}$
Velocity	$0.4\ \mathrm{m/s}$	$0.012\ \mathrm{m/s}$
Acceleration	$0.06\ \mathrm{m/s^2}$	$0.015\ \mathrm{m/s^2}$

Table 1: *Accuracy and stationary accuracy for the navigation parameters*

3.1 Aircraft Model

In this setup, the following 2-DOF linear INS time discrete dynamics is used, (Farrell and Barth, 1999),

$$x_{t+1} = Fx_t + Gw_t \tag{8a}$$

$$F = \begin{bmatrix} I_2 & T_sI_2 & \frac{T_s^2}{2}I_2 \\ 0_{2\times 2} & I_2 & T_sI_2 \\ 0_{2\times 2} & 0_{2\times 2} & I_2 \end{bmatrix} \tag{8b}$$

$$G = \begin{bmatrix} \frac{T_s^3}{6}I_2 \\ \frac{T_s^2}{2}I_2 \\ T_sI_2 \end{bmatrix} \tag{8c}$$

$$x_t = [X_t\ Y_t\ v_t^X\ v_t^Y\ a_t^X\ a_t^Y]^T \tag{8d}$$

$$w_t = [w_t^X\ w_t^Y]^T \tag{8e}$$

where T_s is the sampling time, X is the position in azimuth direction and Y is the position in range direction, v^X and v^Y are the velocities in the X- and Y-directions respectively and a^X and a^Y are the accelerations in X- and Y-directions respectively. This model is used for the whole trajectory. Since this model is linear and time invariant the stationary Kalman filter can be used to estimate x_t giving $\hat{x}_t$ and its corresponding covariance P_t.

3.2 Navigation Performance

Due to the fact that the system is time invariant and linear, the covariance of the estimate will converge to the stationary covariance $\bar{P}$. This covariance can be calculated as

$$\bar{P} = F\bar{P}F^T - F\bar{P}H^T(H\bar{P}H^T + R)^{-1}H\bar{P}F^T + GQG^T \tag{9}$$

where F and G are defined above, and H is the linearised measurement equation $h(x_t)$. In our case, it is chosen as $H = I_6$, since we can assume that all states are measured by the navigation system. For a typical navigation system used in an UAV, the performance for these parameters (assumed to be measurement noise) can be summarised according to Table 1. System noise covariance, Q, which represents disturbance on states, like wind turbulence, can be taken as $\mathrm{diag}\{0.25,\ 0.25\}$. With these values, the stationary covariance is as given in the third column in Table 1.

4 Focus Measures

We here review and compare two common focus measures.

4.1 Two Entropy Measures

One common focus measure in SAR or image processing literature is image entropy calculated as

$$E_1(I) = -\sum_{k=1}^{256} q_k \log_2(q_k) \tag{10}$$

where q_k is an approximated grey level distribution of the $K \times M$ grey-scale image $|I|$, where I is the complex-valued SAR image. It can be obtained from the image histogram calculated as

$$q_k = \frac{\{\text{\# of pixel values } |I_{ij}|\} \in [k-1, k]}{KM} \tag{11a}$$

$$k \in [1, 256]. \tag{11b}$$

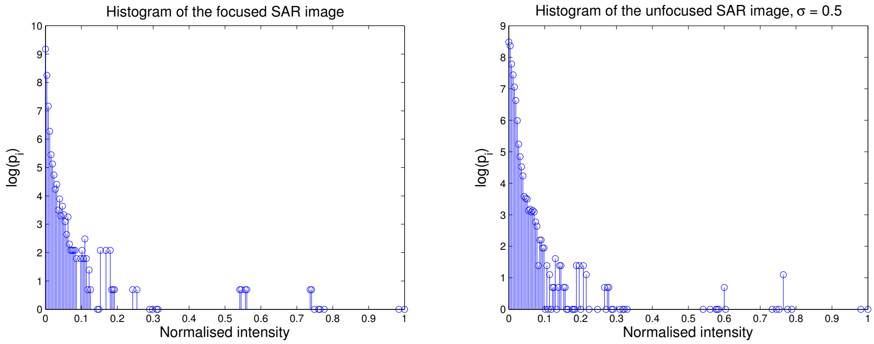
The more focused the image is, the lower the entropy is, see for example Morrison and Munson (2002) or Xi et al. (1999). Histograms for the images in Figure 3 are given in Figure 5. Note the log-scale on the y-axis. An alternative definition of entropy, and more commonly used in the SAR context, is, Yegulalp (1999); Xi et al. (1999); Morrison and Munson (2002),

$$E_2(I) = -\sum_{i=1}^{K}\sum_{j=1}^{M} q_{ij} \ln(q_{ij}) \tag{12a}$$

$$q_{ij} = \frac{|I_{ij}|^2}{\sum_{i=1}^{K}\sum_{j=1}^{M} |I_{ij}|^2}. \tag{12b}$$

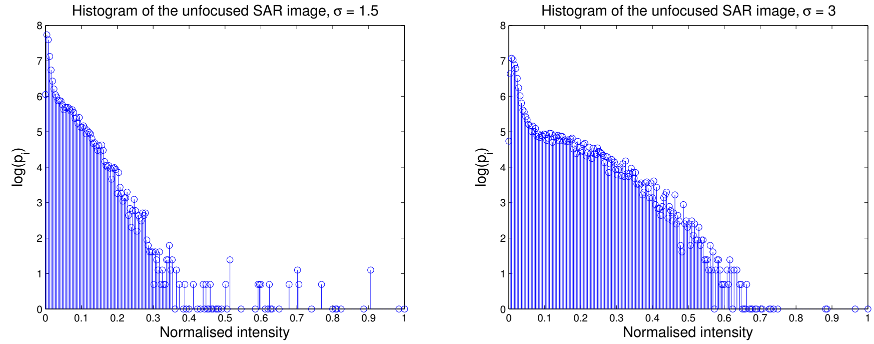
4.2 Focus Measure Performance

Entropy 1 and 2 focus measures are tested and compared on a SAR image according to Figure 6a and the results are depicted in Figure 7 where standard deviations $1-\sigma$, $2-\sigma$ and $3-\sigma$ are also drawn. These images are chosen since they are more informative than the image in Figure 3 and they represent both structured and unstructured scene. In this simulation setup the state noise in model (8) is set to zero, i.e. the trajectory is completely deterministic. This is done in order to illustrate the focus measure functions F_i in a two dimensional plot, since the trajectory, and consequently the focus measure, is then only dependent on the initial values. In these figures it can be seen that entropy 2 has a convex and pretty nice behaviour around the true value of the initial state. However it looks very flat along the velocity direction which indicates that it is very difficult to estimate that particular state. The entropy 1 measure has, on the other hand, a sharp minimum for the correct value of the initial state but many local minima. This means that the two entropy measures complement each other perfectly, and can be used in combination to obtain the global minimum of the focus measure.



***(a)** Histogram of the focused image.*

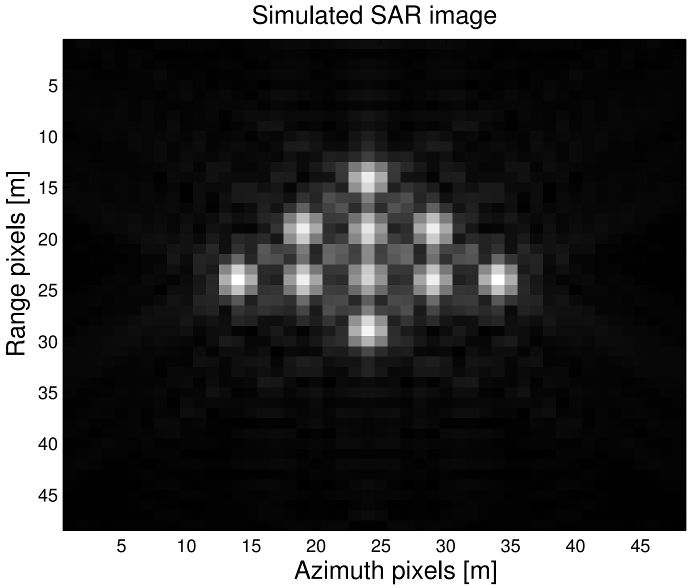
***(b)** Histogram of the unfocused image with* $\sigma = 0.5$.



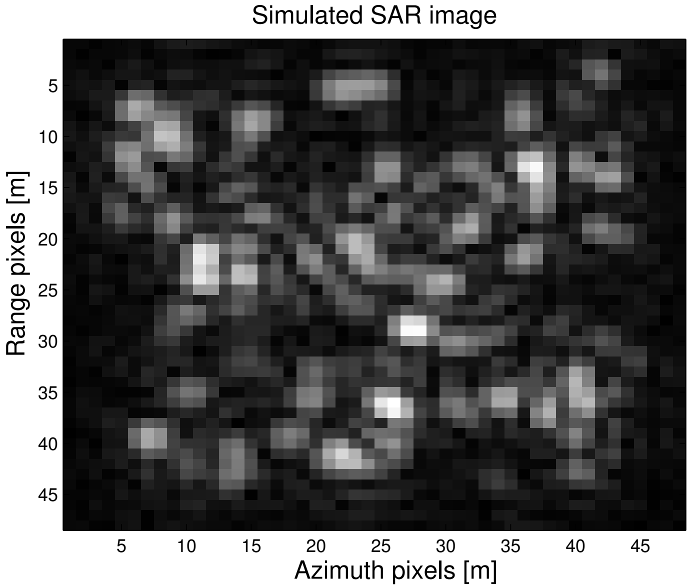
***(c)** Histogram of the unfocused image with* $\sigma = 1.5$.

***(d)** Histogram of the unfocused image with* $\sigma = 3$.

***Figure 5:** Histograms for the images in Figure 3.*

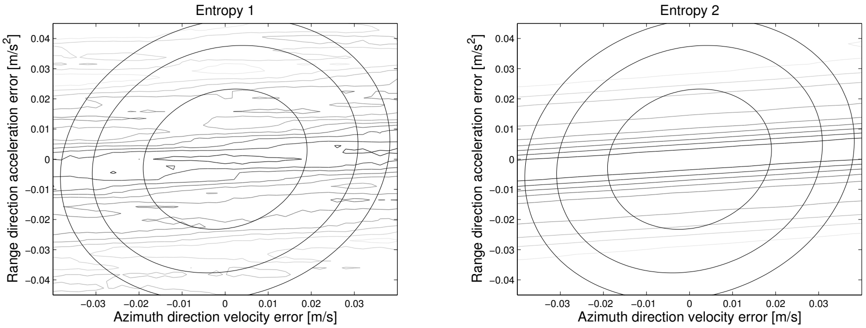


***(a)** SAR image of the structured scene.*



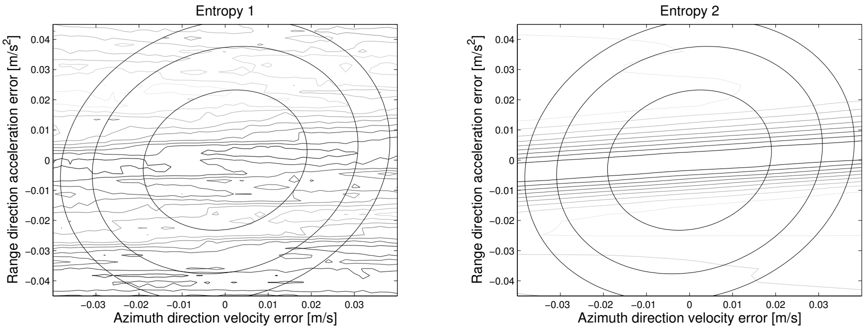
***(b)** SAR image of the unstructured scene.*

***Figure 6:** SAR images with a more informative scene than in Figures 2 and 3.*



***(a)** Entropy 1 focus measure for the structured scene.*

***(b)** Entropy 2 focus measure for the structured scene.*



***(c)** Entropy 1 focus measure for the unstructured scene.*

***(d)** Entropy 2 focus measure for the unstructured scene.*

***Figure 7:** Focus measures for the image in Figure 6 with standard deviation ellipses.*

5 Search Methods

As demonstrated in Section 4.2, the entropy 2 measure can be used as a course first step in the optimisation to come close to the global minimum, and then entropy 1 can be used to obtain the global minimum. Note that a special structure of the problem (8) allows for unconstrained solution of the problem. This is due to the fact that the constraints representing the trajectory can be taken into account while calculating the gradient of the cost function as will be demonstrated in Section 5.3.

5.1 Joint Optimisation of Trajectory and Focus

Since the focus of the image depends on the unknown trajectory, one solution is to solve the following minimisation problem

$$\hat{\theta} = \arg\min_{\theta} \; \gamma_F E_i(x_{0:N}) + \gamma_s \left(\sum_{t=1}^{N} \|y_t - h(x_t))\|^2_{R_t^{-1}} + \|w_t\|^2_{Q_t^{-1}} \right) \tag{13a}$$

$$\textbf{s. t. } \theta = [x_0^T, \; w_{1:N}^T]^T \tag{13b}$$

$$x_{t+1} = f(x_t, w_t) \tag{13c}$$

where γ_F and γ_s are the weights (and $\gamma_F + \gamma_s = 1$) and measurement equation $h(x)$ and system dynamics $f(x, w)$ are defined as in Section 3. In Equation (13a), $E_i(x_{0:N})$, $i \in \{1, 2\}$, is a function of the SAR image I created from the radar measurements and is of the type "how focused is the image?" according to Section 4.

5.2 Gradient Search

Gradient search methods will be exemplified here with a couple of examples with different trajectories and errors in them.

Only two states and their initial values are considered (v_0^X and a_0^Y), for illustrative purposes. In general, the minimisation should be applied for these states for all or at least some of the time instants along the trajectory. Such an example will be studied later.

A gradient search can, for the general problem $\min_\theta g(\theta)$, be formulated as

$$\theta^{k+1} = \theta^k + \mu_k H(\theta^k)^{-1} \nabla g(\theta^k) \tag{14a}$$

$$\nabla g(\theta) = \frac{\partial}{\partial \theta} g(\theta) \tag{14b}$$

where μ_k is step size with $\mu_0 = 1$, $g(\theta)$ is the loss function as in (13a) and $H(\theta)$ is some (positive definite) matrix. The initial estimate, θ^0, can be taken as the usual estimate from the navigation system. In the simplest case H can be chosen as the identity matrix and the procedure becomes a pure gradient search. The disadvantage of such procedure is the slow convergence, especially if the function to be minimised is ridge-like like entropy 1 focus measure. If H is chosen as the Hessian of g, the procedure becomes a Newton search. The Newton search has a fast convergence, and is to prefer if the Hessian is available. In many cases the Hessian is either not available or very difficult to obtain, as in the case considered here, and some approximate methods must be applied. One option is a quasi-Newton method, and BFGS in particular, where the Hessian is approximated by utilising gradients of the function during the search, see Nocedal and Wright (2006). The general gradient search procedure is summarised in Algorithm 1.

In all these procedures it is essential to obtain the gradient of the loss function. Because of the special structure of the focus measure function and the SAR processing algorithm, the complete analytical gradient is hard to obtain. For example, for the Entropy 1 measure it is hard to differentiate a histogram of the image. In this case numerical methods must be used. However, for the Entropy 2 measure it is possible to obtain analytical expressions for most part of the gradient, enabling very efficient algorithms, and this will be described in the next subsection.

Algorithm 1 Gradient search procedure

```
Input: Initial value of the optimisation parameters $\theta^0$, raw radar data, tolerance thresholds $\varepsilon_1, \varepsilon_2, \varepsilon_3$
Output: Solution $\hat{\theta}$, focused SAR image

$k := 0$
repeat
    Calculate gradient of the cost function, $\nabla g(\theta^k)$
    Calculate (approximate) Hessian, $H(\theta^k)$
    $\mu_k := 1$
    repeat
        $\theta^{k+1} := \theta^k - \mu_k H(\theta^k)^{-1} \nabla g(\theta^k)$
        $\mu_k := \mu_k / 2$
    until $g(\theta^{k+1}) < g(\theta^k)$
    $k := k + 1$
until $\|\theta^k - \theta^{k-1}\|^2 < \varepsilon_1$ or $\nabla g(\theta^{k-1}) < \varepsilon_2$ or $\|g(\theta^k) - g(\theta^{k-1})\|^2 < \varepsilon_3$
```

5.3 Calculating the Gradient

The calculations to obtain an analytical gradient of the entropy 2 function will be presented. The key to doing this is the chain rule for gradient calculation,

$$\frac{\partial E_2}{\partial x} = \frac{\partial E_2}{\partial q} \frac{\partial q}{\partial |I|} \frac{\partial |I|}{\partial R} \frac{\partial R}{\partial x}. \tag{15}$$

In order to apply the chain rule, first the decomposition chain of the entropy 2 focus measure will be considered and then, all partial derivatives will be presented.

The first factor to be differentiated is entropy 2 focus measure

$$E_2 = -\sum_{i=1}^{K} \sum_{j=1}^{M} q_{ij} \ln q_{ij} = -\sum_{i=1}^{KM} q_i \ln q_i \tag{16}$$

where last equality is simply reformulation of the double sum by vectorising the image.

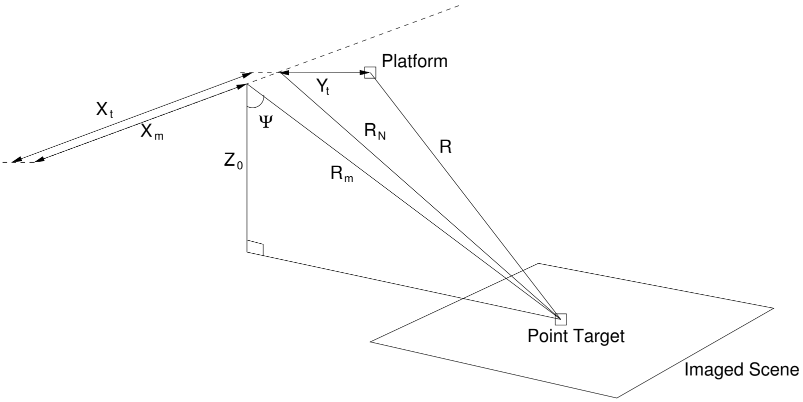


Figure 8: *SAR geometry. The figure is not to scale.*

In the second factor, each q_i is obtained by

$$q_i = \frac{|I_i|^2}{\sum_j |I_j|^2} \tag{17}$$

meaning that q_i is a function of the absolute value of the complex-valued SAR image.

For the third factor, we need to obtain the derivative $\partial|I|/\partial R$. In the creation of the image, a back-projection sum is evaluated and all partial images are summed up. Each partial image is a function of one column in the RAR image and the range from the platform to each pixel in the SAR image, see Section 2. Unfortunately it is not easy, if not impossible, to obtain analytical expression for the derivative $\partial|I|/\partial R$. However, this value can simply be obtained during image creation by means of numerical derivation. The cost for that procedure is memory demand and execution time which both are doubled. But this increase in cost is independent of the state, and thus constant no matter how many parameters optimisation is performed over. A straightforward numerical gradients would give a cost that is increasing linearly with the number of parameters.

The last factor that needs to be calculated is the gradient of the range as a function of the states, $R(x_t)$. To calculate an analytical expression of this function some SAR geometry preliminaries are needed. In order to express range as a function of the states, the geometry setup as in Figure 8 can be considered. From the figure it can be seen that the range R_t can, with help from Pythagoras theorem, be expressed as

$$R_t = \sqrt{(X_m - X_t)^2 + (R_g - Y_t)^2 + (Z_0 - Z_t)^2} \tag{18a}$$

$$R_g = \sqrt{R_m^2 - Z_0^2} \tag{18b}$$

i.e., as a function of the trajectory. Note that Z_t, the altitude of the platform, is assumed to be known here. This can be achieved by measuring it with, for example, barometric sensors which is always done in the aircraft applications. Here the exact expression for the range along the trajectory is used, unlike in most of the SAR literature, where approximate and linearised expressions are used, see for example Xing et al. (2009). This is due to the fact that in low frequency SAR application, as the one considered here, the ratio between range and trajectory length is not negligible due to the lobe width. If approximate expressions are used, too large errors would be introduced in the beginning and the end of the trajectory.

Next, the dynamical model (8) can be used to express the position states used in the range expression above as a function of any other state by using that

$$x_t = F^{t-k} x_k, \quad t > k \tag{19}$$

Note that the state noise term, w_t, is neglected in the following since it is equivalent to optimise over noise and over acceleration states, a_t, so the latter one is used here to simplify the expressions and reduce the amount of variables in the problem. This gives that positions can be expressed as

$$X_t = X_0 + T_s(t-k)v_k^X + \frac{T_s^2(t-k)^2}{2}a_k^X \tag{20a}$$

$$Y_t = Y_0 + T_s(t-k)v_k^Y + \frac{T_s^2(t-k)^2}{2}a_k^Y \tag{20b}$$

If these expressions are used in (18a), we can easily obtain partial derivatives of the range with respect to the velocities and accelerations in arbitrary time points.

Now, we have everything needed to calculate the gradient of the focus measure with respect to the trajectory states. The partial derivatives are, in turn (for $t > k$),

$$\frac{\partial E_2}{\partial q_i} = -\ln q_i - 1 \tag{21a}$$

$$\frac{\partial q_i}{\partial |I_j|} = \begin{cases} \frac{2|I_j|\sum|I|^2 - 2|I_j|^2|I_i|}{(\sum|I|^2)^2}, & i = j \\ -\frac{2|I_j|^2|I_i|}{(\sum|I|^2)^2}, & i \neq j \end{cases} \tag{21b}$$

$$\frac{\partial R_t}{\partial v_k^X} = -\frac{(X_m - X_t)T_s(t-k)}{R_t} \tag{21c}$$

$$\frac{\partial R_t}{\partial v_k^Y} = -\frac{(R_g - Y_t)T_s(t-k)}{R_t} \tag{21d}$$

$$\frac{\partial R_t}{\partial a_k^X} = -\frac{(X_m - X_t)T_s^2(t-k)^2}{2R_t} \tag{21e}$$

$$\frac{\partial R_t}{\partial a_k^Y} = -\frac{(R_g - Y_t)T_s^2(t-k)^2}{2R_t} \tag{21f}$$

and $\partial|I|/\partial R$ is numerically calculated during image formation. Now, at least for entropy 2 focus measure, we can calculate the gradient (semi-) analytically and use it in the minimisation procedure. The second term in (13a) is easy to differentiate, since it is a quadratic form and $h(x)$ is a linear function in this case.

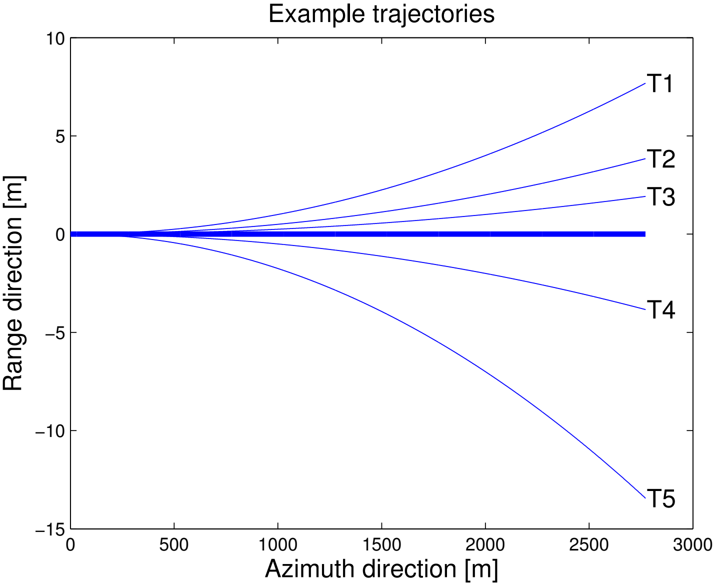


Figure 9: *Trajectory examples for different initial values of the velocity,* v_0^X *and acceleration,* a_0^Y.

6 Numerical Examples for Simulated Images

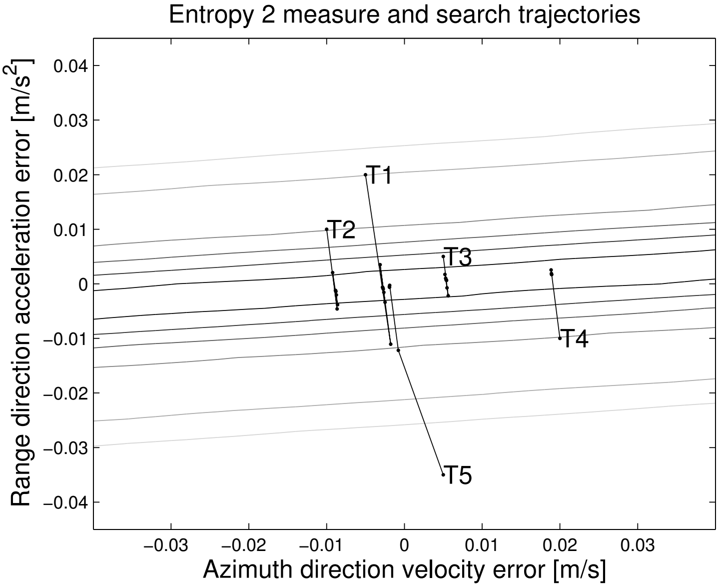
In order to demonstrate the behaviour of the gradient search for this setup, the SAR image from Figure 6a is used in two different experiments.

6.1 Two-Dimensional Optimisation

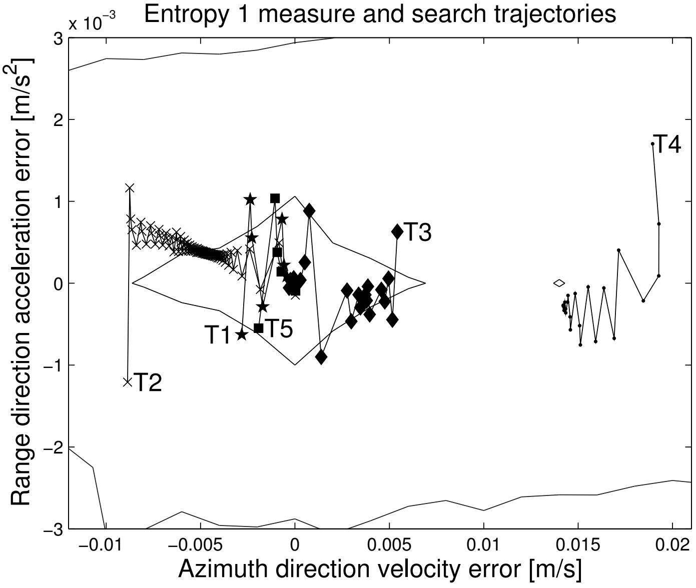
To be able to illustrate the convergence of the solution, only two optimisation variables are considered here, $\theta = [v_0^X,\ a_0^Y]^T$ and the algorithm is initiated with random starting points θ^0 based on the stationary covariance of the states in the system. Those initial values are $[100.005,\ 0.005]^T$, $[99.99,\ 0.01]^T$, $[99.995,\ 0.02]^T$, $[100.02,\ -0.01]^T$ and $[100.005,\ -0.035]^T$.

The trajectories generated with these initial values are illustrated in Figure 9. In Figure 10a, the gradient search based entropy 2 measure is illustrated and we can see that the solutions converge to the flat ridge-like area close to the correct acceleration, but not necessarily to the correct velocity. In Figure 10b, the gradient search where the entropy 1 measure is used is depicted. In this case the algorithm is initiated with the solution from the entropy 2 search. It can be seen that this minimisation strategy works pretty well, although one solution is stuck in a local minimum. In that case the velocity error is the largest one of all errors. Note also that only the focus measure is used to find estimate of the states i.e., γ_s is set to zero while γ_F is set to one in Equation (13a).

It is interesting to see how the image created with the solution that is stuck in the local



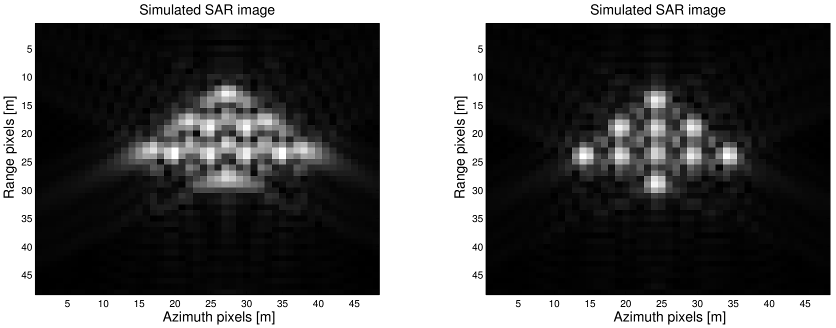
(a) *Search trajectory for five different values of* x^0 *using entropy 2 focus measure.*



(b) *Search trajectory for entropy 1 focus measure with* x^0 *given by the entropy 2 gradient search.*

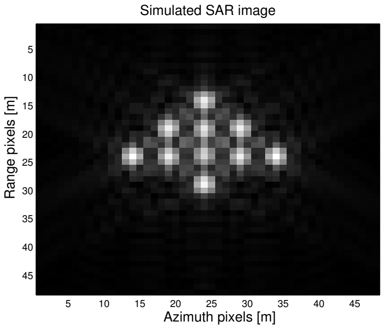
Figure 10: *Search trajectory for five different values of* x^0 *using two different entropy measures.*

minimum of the entropy 1 measure looks like compared to the unfocused image that is initialised with. As illustrated in Figure 11, it can be seen that the image created with values from the minimisation procedure is very close to the focused image and much better than the unfocused images that are initialised with. The probable explanation for this comes from the fact that small azimuth direction velocity errors do not influence the final image much due to the quantisation effects. However the estimate of the navigation states is not correct.



(a) Image created with error in velocity of 0.02 m/s and in acceleration of $-0.01\,m/s^2$.

(b) Image created with error in velocity of 0.014 m/s and in acceleration of $-0.0003\,m/s^2$.



(c) Focused image as a reference.

Figure 11: *Resulting images from the minimisation procedure with starting point* $[100.02,\ -0.01]^T$.

6.2 High-Dimensional Optimisation

In the second example a more realistic setup is done. The optimisation problem to be solved is

$$\hat{\theta} = \arg\min_{\theta} \gamma_F E_{1,2}(x_{0:N}) + \gamma_s \sum_{t=1}^{N} \|a_t^{mY} - a_t^Y\|^2_{R_t^{-1}} \tag{22a}$$

s. t.

$$\theta = [v_0^X,\, a_0^Y,\, a_{\lfloor N/4 \rfloor}^Y,\, a_{\lfloor N/2 \rfloor}^Y,\, a_{\lfloor 3N/4 \rfloor}^Y]^T \tag{22b}$$

$$\gamma_F = 0.99 \tag{22c}$$

$$\gamma_s = 0.01 \tag{22d}$$

$$x_{t+1} = Fx_t \tag{22e}$$

$$\begin{bmatrix} X_0 \\ Y_0 \\ v_0^Y \\ a_0^X \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{22f}$$

$$a_t^X = 0,\ t \in \{0 : N\} \tag{22g}$$

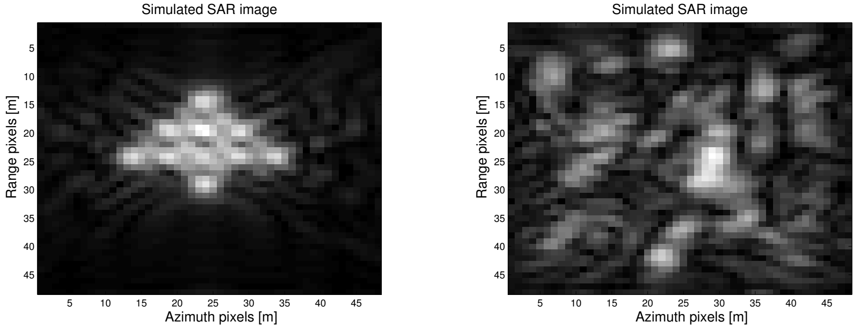
$$a_t^Y = \begin{cases} a_0^Y,\ t \in \{1 : \lfloor N/4 \rfloor - 1\} \\ a_{\lfloor N/4 \rfloor}^Y,\ t \in \{\lfloor N/4 \rfloor + 1 : \lfloor N/2 \rfloor - 1\} \\ a_{\lfloor N/2 \rfloor}^Y,\ t \in \{\lfloor N/2 \rfloor + 1 : \lfloor 3N/4 \rfloor - 1\} \\ a_{\lfloor 3N/4 \rfloor}^Y,\ t \in \{\lfloor 3N/4 \rfloor + 1 : N\} \end{cases} \tag{22h}$$

$$P_0 = \infty \cdot I_2 \tag{22i}$$

where a^{mY} is the measured acceleration in Y-direction with additive white Gaussian noise with $R_t = 0.0022\ \mathrm{m}^2/\mathrm{s}^4$. $E_{1,2}(x_{0:N})$ is either entropy 2 or entropy 1, exactly as in the previous example. Here it is assumed that a change in Y-direction acceleration will behave in a step like manner only a few times during the SAR image generation and that the amplitude of the step is arbitrary. It is also assumed that the acceleration in X-direction will vary slowly due to the platforms inherited inertia in this direction, so it can be assumed to be zero. The meaning of the constraint in (22i) is that there is no prior information about the initial values of the trajectory. Another spline-like interpretation of the setup in (22) is to find a best trajectory by fitting the second order polynomials between four points evenly spaced along the trajectory. Both scenes from Figure 6 are used and 30 Monte Carlo simulations are performed in order to evaluate the performance of the estimation procedure.

The resulting RMSE of the parameters and the mean value of the error image power are presented in Table 2 and Table 3 for both structured and unstructured scene. Here, the actual acceleration is presented instead of the process noise value, since it is more physically interesting. It can be noticed that the improvement of the RMSE after further minimisation with entropy 1 is not very big, it is in the magnitude of 10^{-5}. It suggests that the extra step of minimisation with entropy 1 can be skipped if a faster procedure is sought.

In Figure 12, a noisy position (one of the 30 noise realisations) is used for the image



(a) *SAR image of the structured scene created with noisy position data.*

(b) *SAR image of the unstructured scene created with noisy position data.*

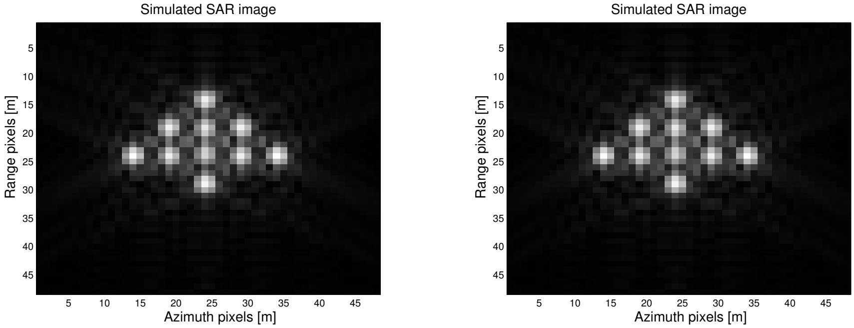
Figure 12: *SAR images created with noisy position data.*

Parameter	RMSE (opt. with E_2)	RMSE (opt. with E_1)
$\hat{v}_0^X$	$7.05 \cdot 10^{-3}$ m/s	$7.04 \cdot 10^{-3}$ m/s
$\hat{a}_0^Y$	$9.94 \cdot 10^{-4}$ m/s^2	$9.15 \cdot 10^{-4}$ m/s^2
$\hat{a}_{\lfloor N/4 \rfloor}^Y$	$6.51 \cdot 10^{-4}$ m/s^2	$6.34 \cdot 10^{-4}$ m/s^2
$\hat{a}_{\lfloor N/2 \rfloor}^Y$	$6.89 \cdot 10^{-4}$ m/s^2	$6.84 \cdot 10^{-4}$ m/s^2
$\hat{a}_{\lfloor 3N/4 \rfloor}^Y$	$6.02 \cdot 10^{-4}$ m/s^2	$6.03 \cdot 10^{-4}$ m/s^2
Mean value of the error image power	149.6	126.9

Table 2: *RMSE for the estimated parameters and the mean value for the error image power for the structured scene.*

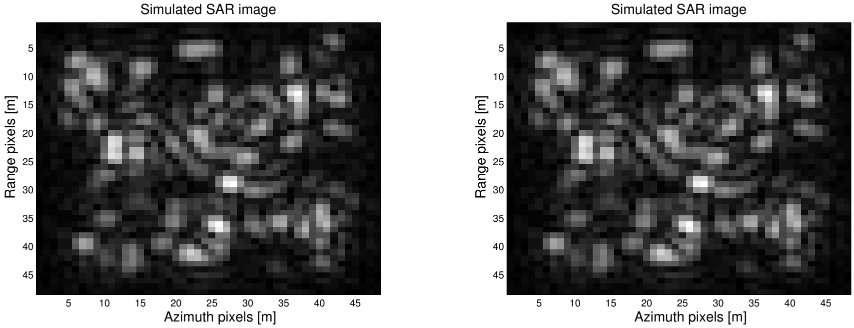
Parameter	RMSE (opt. with E_2)	RMSE (opt. with E_1)
$\hat{v}_0^X$	$11.2 \cdot 10^{-3}$ m/s	$11.2 \cdot 10^{-3}$ m/s
$\hat{a}_0^Y$	$11.61 \cdot 10^{-4}$ m/s^2	$10.98 \cdot 10^{-4}$ m/s^2
$\hat{a}_{\lfloor N/4 \rfloor}^Y$	$6.63 \cdot 10^{-4}$ m/s^2	$6.52 \cdot 10^{-4}$ m/s^2
$\hat{a}_{\lfloor N/2 \rfloor}^Y$	$9.31 \cdot 10^{-4}$ m/s^2	$8.86 \cdot 10^{-4}$ m/s^2
$\hat{a}_{\lfloor 3N/4 \rfloor}^Y$	$7.77 \cdot 10^{-4}$ m/s^2	$7.58 \cdot 10^{-4}$ m/s^2
Mean value of the error image power	1348	1242

Table 3: *RMSE for the estimated parameters and the mean value for the error image power for the unstructured scene.*



(a) *Image of the structured scene after minimisation with Entropy 2 as focus measure.*

(b) *Image of the structured scene after minimisation with Entropy 1 as focus measure.*



(c) *Image of the unstructured scene after minimisation with Entropy 2 as focus measure.*

(d) *Image of the unstructured scene after minimisation with Entropy 1 as focus measure.*

Figure 13: *Resulting images from the gradient search minimisation.*

generation. We see that both images are unfocused and the image of the unstructured scene is pretty bad, all the dominant targets are blurred. In Figure 13 the images after minimisation with entropy 2 and 1 are depicted (for the same noise realisation as above). Here it can be seen that any improvement in the image with extra minimisation with entropy 1 is impossible to see with the naked eye, i.e., the improvement of the navigation states does not visibly improve the images. This could be expected from the results from MC simulations.

The resulting estimates of the parameters and error image power after entropy 1 minimisation for the two scenes and this particular realisation of the noise are presented in Table 4.

Parameter	Structured scene	Unstructured scene
$\hat{v}_0^X$	$9.236 \cdot 10^{-3}$ m/s	$10.13 \cdot 10^{-3}$ m/s
$\hat{a}_0^Y$	$-3.057 \cdot 10^{-4}$ m/s^2	$-1.707 \cdot 10^{-4}$ m/s^2
$\hat{a}_{\lfloor N/4 \rfloor}^Y$	$-0.553 \cdot 10^{-4}$ m/s^2	$-1.733 \cdot 10^{-4}$ m/s^2
$\hat{a}_{\lfloor N/2 \rfloor}^Y$	$11.15 \cdot 10^{-4}$ m/s^2	$10.25 \cdot 10^{-4}$ m/s^2
$\hat{a}_{\lfloor 3N/4 \rfloor}^Y$	$1.384 \cdot 10^{-4}$ m/s^2	$1.234 \cdot 10^{-4}$ m/s^2
Error image power	51.37	53.12

Table 4: *Error in the estimated parameters for the two scenes after entropy 1 minimisation procedure.*

7 Example with Real SAR Image

Here, we illustrate the estimation results using data from the CARABAS II system (Hellsten et al., 1996) collected in western Sweden. The trajectory and the SAR image obtained by the proposed estimation method are compared to the GPS trajectory, which is assumed to be the ground truth. Part of the SAR image used for the estimation is illustrated in Figure 14 where the GPS based trajectory is used to generate the image.

For the real data case, the optimisation problem to be solved is formulated according to

$$\hat{\theta} = \arg\min_{\theta} \; \gamma_F E_2(x_{0:N}) + \gamma_s \sum_{t=1}^{N} \|a_t^m - a_t\|^2_{R_t^{-1}} \tag{23a}$$

s. t.

$$\theta = [v_0^X,\, v_0^Y,\, a_0^X,\, a_0^Y,\, a_i^X,\, a_i^Y]^T \tag{23b}$$

$$i \in \lfloor \{1 : 199\} N/200 \rfloor \tag{23c}$$

$$\gamma_F = 0.36 \tag{23d}$$

$$\gamma_s = 0.64 \tag{23e}$$

$$x_{t+1} = F x_t \tag{23f}$$

$$\begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{23g}$$

$$a_i = \begin{cases} a_0,\; i \in \{1 : \lfloor N/200 \rfloor - 1\} \\ a_{\lfloor N/200 \rfloor},\; i \in \{\lfloor N/200 \rfloor + 1 : \lfloor N/100 \rfloor - 1\} \\ \quad\vdots \\ a_{\lfloor 199N/200 \rfloor},\; i \in \{\lfloor 199N/200 \rfloor + 1 : N\} \end{cases} \tag{23h}$$

$$P_0 = \infty \cdot I_4 \tag{23i}$$

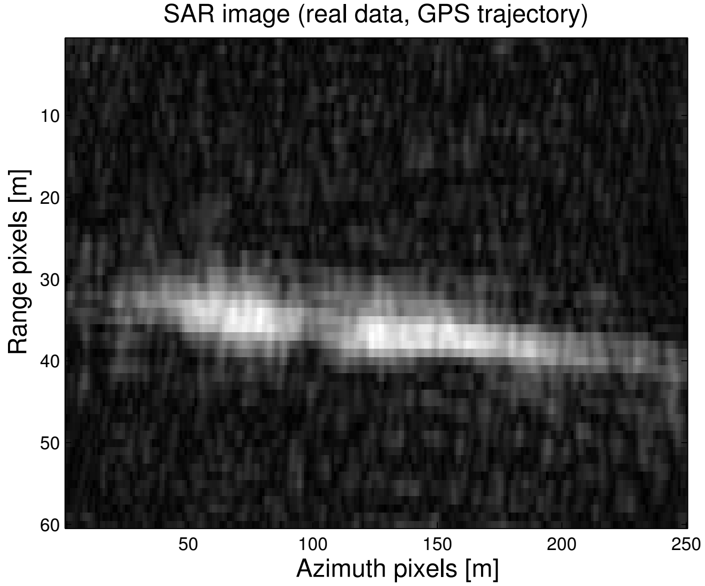


Figure 14: *SAR image for the real data case created with the GPS based trajectory (assumed to be ground truth).*

where the variables are defined as

$$a_t = [a_t^X,\ a_t^Y]^T \tag{24a}$$

$$a_t^m = [a_t^{mX},\ a_t^{mY}]^T \tag{24b}$$

$$R_t = \text{diag}\{R_t^X,\ R_t^Y\} = \text{diag}\{0.1,\ 0.1\}[\text{m}^2/\text{s}^4] \tag{24c}$$

Note that in this case only the entropy 2 focus measure is used due to the computational load to calculate the numerical gradient for the entropy 1 measure. However, according to the results in Section 6, the improvement of the estimates by using additional optimisation with entropy 1 is small and therefore it is omitted here. Also, the weights and the covariance of the acceleration measurements are seen as tuning parameters.

Results from the optimisation procedure, which takes five steps to converge, is illustrated in Figure 15, where the estimated trajectory is used to generate the image and Figure 16 where error between GPS and estimated trajectory is shown. That error can be compared to the error in the trajectory with the initial values of the parameters, θ^0, shown in Figure 17. It can be seen from these two plots that improvement in Y-direction is much less than improvement in X-direction. Only the total loss function for entropy 2 is depicted in Figure 18 and Figure 19. The image resulting form the estimated trajectory is hard do distinguish from the GPS based trajectory image, except that it is shifted in range direction. This ambiguity is, unfortunately, unobservable in the auto-focusing process, i.e., the method is invariant to the translation of the image.

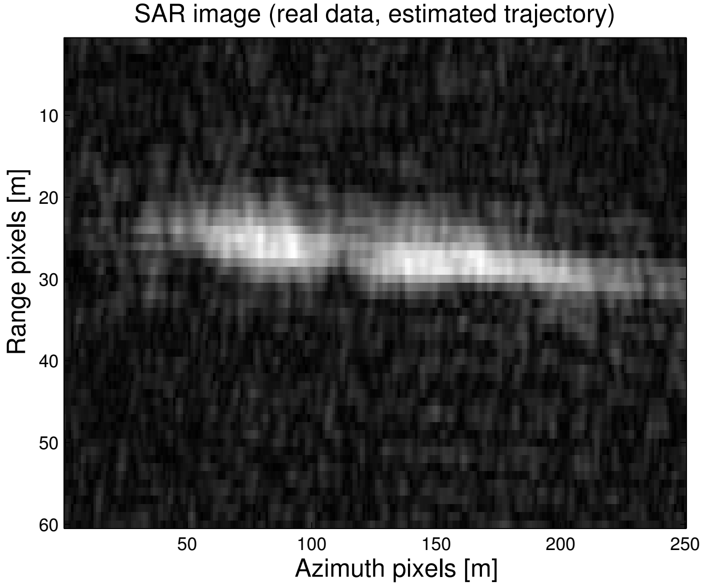


Figure 15: *SAR image for the real data case obtained with the optimisation procedure.*

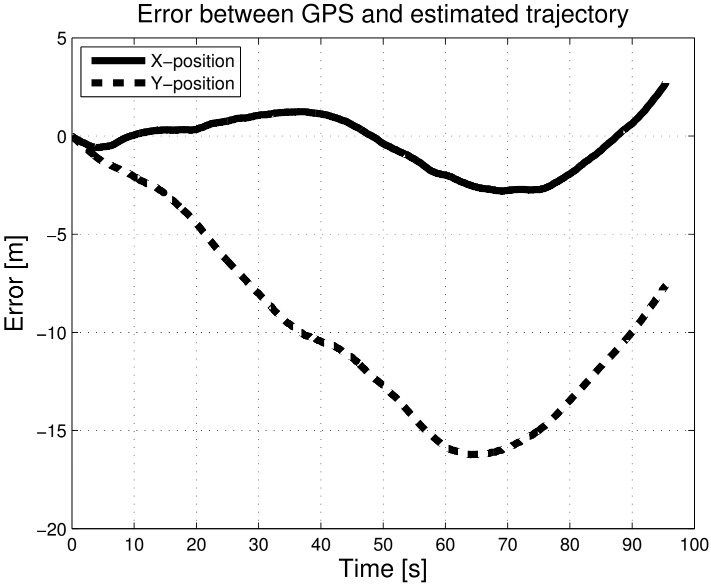


Figure 16: *Error in position between GPS and estimated trajectory.*

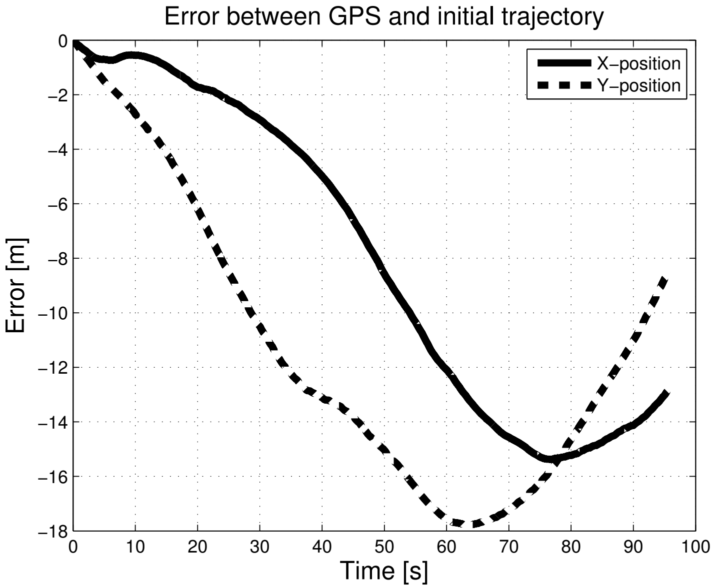


Figure 17: *Error in position between GPS and initial trajectory.*

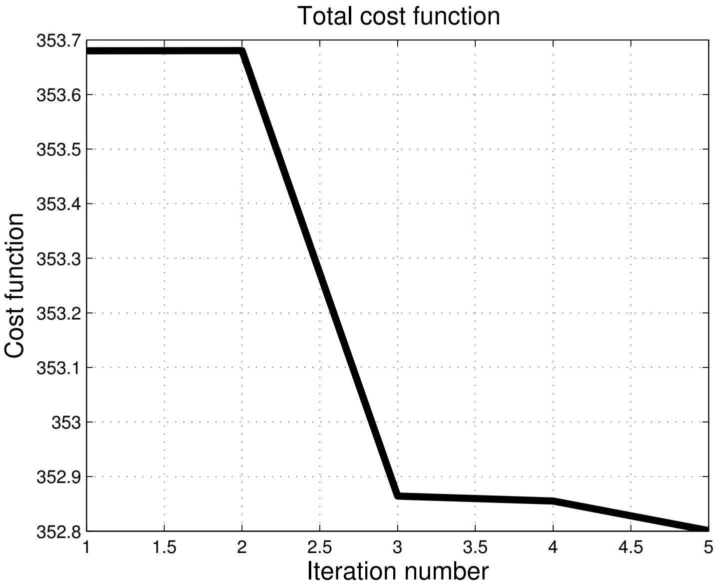


Figure 18: *Total cost function as a function of the iteration number.*

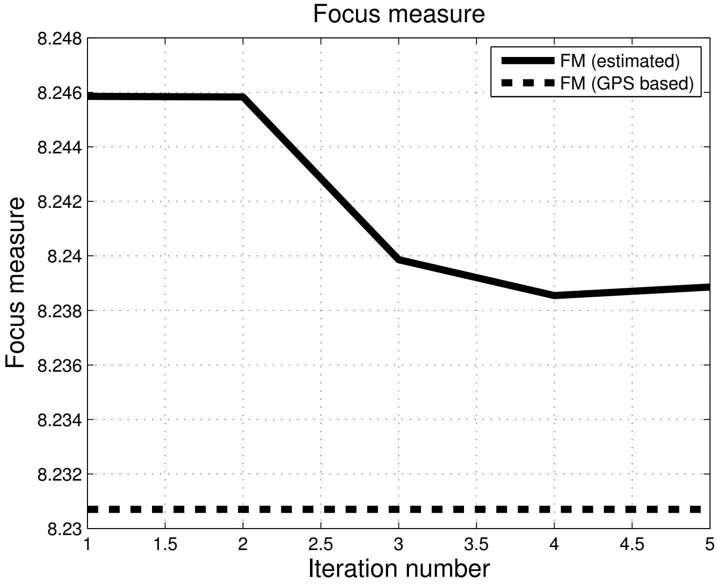


Figure 19: *Entropy 2 value as a function of the iteration number. Entropy 2 value for the GPS-based image is also shown for a comparison.*

8 Conclusions

A method is presented based on a decentralised sensor fusion framework, which is intended to provide better focused SAR images on future cheap small SAR platforms. The approach is based on jointly optimising a focus measure and the error in the navigation states. As was concluded from simulation examples of the simple scene and real SAR data, the method works fairly well, although not all states are observable. Nevertheless, even if not the whole navigation state vector can be corrected, the resulting SAR image is much more focused after optimisation than the original one. An important theoretical contribution, to reach the requirements on computation complexity, is an analytical expression for the gradient of a focus measure. The result enables an implementation of gradient search algorithms that add only marginal complexity to the SAR imaging process.

Bibliography

L. E. Andersson. On the Determination of a Function from Spherical Averages. *SIAM Journal on Mathematical Analysis*, 19(1):214–232, 1988. doi: 10.1137/0519016. URL `http://link.aip.org/link/?SJM/19/214/1`.

T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, Septemeber 2006.

C. Cafforio, C. Prati, and F. Rocca. SAR data focusing using seismic migration techniques. *IEEE Transactions on Aerospace and Electronic Systems*, 27(2):194 –207, March 1991. ISSN 0018-9251. doi: 10.1109/7.78293.

L. J. Cutrona, W. E. Vivian, E. N. Leith, and G. O. Hall. A high-resolution radar combat-surveillance system. *IRE Transactions on Military Electronics*, MIL-5(2):127–131, April 1961. ISSN 0096-2511. doi: 10.1109/IRET-MIL.1961.5008330.

H. Durrant-Whyte and T. Bailey. Simultaneous Localization and Mapping: Part I. *IEEE Robotics & Automation Magazine*, 13(12):99–110, June 2006.

J. Farrell and M. Barth. *The global positioning system and inertial navigation*. McGraw-Hill Professional, 1999.

J. A. Fawcett. Inversion of N-Dimensional Spherical Averages. *SIAM Journal on Applied Mathematics*, 45(2):336–341, 1985. ISSN 00361399. URL `http://www.jstor.org/stable/2101820`.

J. R. Fienup. Phase Error Correction by Shear Averaging. In *Signal Recovery and Synthesis*, pages 134–137. Optical Society of America, June 1989.

H. Hellsten and L. E. Andersson. An inverse method for the processing of synthetic aperture radar data. *Inverse Problems*, 3(1):111, 1987. URL `http://stacks.iop.org/0266-5611/3/i=1/a=013`.

H. Hellsten, L. M. Ulander, A. Gustavsson, and B. Larsson. Development of VHF CARABAS II SAR. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 2747 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 48–60, June 1996.

A. S. Milman. SAR Imaging by Omega-K Migration. *International Journal of Remote Sensing*, 14(10):1965–1979, 1993.

R. L. Jr. Morrison and D. C. Jr. Munson. An experimental study of a new entropy-based SAR autofocus technique. In *Proceedings of International Conference on Image Processing, ICIP 2002*, volume 2, pages II–441–4, September 2002. doi: 10.1109/ICIP.2002.1039982.

F. Natterer. *The Mathematics of Computerised Tomography*. New York: Wiley, 1986.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2006.

C. Oliver and S. Quegan. *Understanding Synthetic Aperture Radar Images*. The SciTech Radar and Defense Series. SciTech, 2004. ISBN ISBN 1-891121-31-6.

F. Rocca. Synthetic Aperture Radar: a New Application for Wave Equation Techniques. *Stanford Exploration Project SEP-56*, pages 167–189, 1987. URL `http://sepwww.stanford.edu/oldreports/sep56/56_13.pdf`.

Z. Sjanic and F. Gustafsson. Simultaneous Navigation and Synthetic Aperture Radar Focusing. *Provisionally accepted to IEEE Transactions on Aerospace and Electronic Systems*, August 2013.

L. M. H. Ulander, H. Hellsten, and G. Stenstrom. Synthetic-aperture radar processing using fast factorized back-projection. *Aerospace and Electronic Systems, IEEE Transactions on*, 39(3):760–776, July 2003. ISSN 0018-9251. doi: 10.1109/TAES.2003.1238734.

D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and C. V. Jr. Jakowatz. Phase gradient autofocus - a robust tool for high resolution SAR phase correction. *IEEE Transactions on Aerospace and Electronic Systems*, 30(3):827–835, July 1994. ISSN 0018-9251. doi: 10.1109/7.303752.

L. Xi, L. Guosui, and J. Ni. Autofocusing of ISAR images based on entropy minimization. *IEEE Transactions on Aerospace and Electronic Systems*, 35(4):1240–1252, October 1999. ISSN 0018-9251. doi: 10.1109/7.805442.

M. Xing, R. Jiang, X.and Wu, F. Zhou, and Z. Bao. Motion Compensation for UAV SAR Based on Raw Radar Data. *IEEE Transactions on Geoscience and Remote Sensing*, 47 (8):2870–2883, August 2009. ISSN 0196-2892. doi: 10.1109/TGRS.2009.2015657.

A. F. Yegulalp. Minimum entropy SAR autofocus. In *7th Adaptive Sensor Array Processing Workshop*, March 1999.

Paper B

Navigation and SAR focusing with Map Aiding

Authors: Zoran Sjanic and Fredrik Gustafsson

Edited version of the paper:

Z. Sjanic and F. Gustafsson. Navigation and SAR focusing with Map Aiding. *Submitted to IEEE Transactions on Aerospace and Electronic Systems*, June 2013b.

Navigation and SAR focusing with Map Aiding

Zoran Sjanic and Fredrik Gustafsson

Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
{zoran,fredrik}@isy.liu.se

Abstract

A method for fusing Synthetic Aperture Radar (SAR) images with optical aerial images is presented. This is done in a navigation framework, where the absolute position and orientation of the flying platform, as computed from the inertial navigation system, is corrected based on the aerial image coordinates taken as ground truth. The method is suitable for new low-price SAR systems for small unmanned vehicles. The primary application is remote sensing, where the SAR image provides one further "colour" channel revealing reflectivity to radio waves.

The method is based on first applying an edge detection algorithm to the images and then optimising the most important navigation states by matching the two binary images. To get a measure of the estimation uncertainty, we embed the optimisation in a least squares framework, where an explicit method to estimate the (relative) size of the errors is presented. The performance is demonstrated on real SAR and aerial images, leading to an error of only a few pixels.

1 Introduction

A radar mounted on a flying platform, like an aircraft or a satellite, can be used to get an image of the surroundings by taking intensity (or radar cross section) of the reflections and map it to pixels. This kind of image would be of pretty bad quality, since the resolution will be decided by the radar lobe width which in turn is decided by the antenna length and the frequency of the radar. For realistic antenna lengths found on the flying platforms, this resolution is in range of several tenths of meters or more. By taking many images over same area by moving the radar antenna and in this way creating a large synthetic antenna, images with much higher resolution can be created. This is the basics of the Synthetic Aperture Radar (SAR) imaging, (Cutrona et al., 1961). For more detailed description of SAR and SAR images see e.g., Oliver and Quegan (2004). With modern SAR systems the resolutions in images can be as good as a couple of decimetres, giving very detailed images of the scene. The knowledge of the flown trajectory is very important in the image creation principle and errors in the trajectory will lead to defocused SAR images. A process to correct for these image defects is called autofocusing. There exist many autofocusing methods, of which some are based on the raw radar data and others on the already processed SAR image, (Oliver and Quegan, 2004; Wahl et al., 1994; Xing et al., 2009; Fienup, 1989; Morrison and Munson, 2002; Xi et al., 1999). All these methods use only SAR images without any prior information of the scene to perform the focusing.

The goal of this work is to match SAR images with optical images or map information, e.g., Google Maps. The fusion of information from these sources is then utilised for autofocusing and correction of the navigation trajectory. The assumptions are that the most focused images will also match the optical map in the best way and in turn correspond to the best possible trajectory (giving the best focus) and best possible absolute position on the map (giving the global navigation ability). Traditionally, SAR images are usually used for surveillance and remote sensing purposes, but some cases where they are used for navigation purposes have also been studied, see e.g., Greco et al. (2011). The method can be useful as an alternative to high precision navigation aids, such as Global Navigation Satellite System (GNSS), of which GPS NavStar is the most famous one, to stabilise inertial based navigation systems which are known to be prone for long term drift. The method has many similarities to the e.g., terrain aided navigation (Hostetler and Andreas, 1983), where an altitude database of the terrain is used to support navigation. other similar methods are visual odometry, (Scaramuzza and Fraundorfer, 2011), and a method of aided navigation where optical cameras and maps are used to navigate by matching the camera images and the map, see e.g., Lindsten et al. (2010); Grelsson et al. (2013). However, the fusion of the SAR and optical map images is not as trivial task as to match optical camera images to the map images, since the SAR images have quite different properties than the optical images. The SAR images show the reflectivity of the scene for radar frequencies instead of visible light frequencies. This implies that completely different information can be contained in the SAR images compared to the optical images, although some of the features in the images are clearly very similar. This makes the fusion of SAR and optical images a promising method for remote sensing applications. As a navigation tool, SAR is not sensitive to occlusions from clouds like optical sensors are, giving a less weather sensitive position sensor.

As means for extracting useful information from the images, an edge detector (Canny edge detector) and a modified image matching method (Chamfer matching) will be used in order to match SAR images to the optical map images. The results of the matching and focusing method will be illustrated on real SAR and optical images that are depicted in Figure 1 and Figure 2. This work is an extension of Sjanic and Gustafsson (2012), where only image matching was considered with the assumption that autofocusing has already been performed.

The paper is organised as follows, Section 1 introduces the work, Section 2 explains the SAR imaging principle. Section 3 introduces the navigation models and defines the basic SAR geometry that relates the image and the flying platform while Section 4 introduces the image matching approaches. In Section 5, estimation of the kinematic trajectory parameters is explained and in Section 6 the results from the matching and kinematic estimation are shown. In Section 7, conclusions are given and some future work is discussed.

Figure 1: *SAR image of Washington D.C. Image: Sandia National Laboratories.*

Figure 2: *Optical image of Capitol Hill, Washington D.C. with surroundings. Image: Google Maps.*

2 SAR Imaging Principle

SAR imaging is based on a moving platform that passes the scene that shall be imaged. During the movement, the platform transmits radar pulses which hit the scene and return to the platform with a certain time delay which is proportional to the range to the scene. This returned signal is filtered with a matched filter and then sampled. Each reflector in the scene will contribute with its reflected power which will then be placed in the appropriate range bin. The range is determined as a product between signal propagation speed (usually speed of light) and delay time. In this way a single scene transfer function is obtained, denoted $g(R)$. Now this process can be repeated during platform movement, and all the stored transfer functions are stored in a two-dimensional array $g_t(R)$. Basically, this raw data, $g_t(R)$, is an example of a real aperture radar (RAR). The resolution in such radar system is proportional to the radar lobe width and is usually quite poor. One important thing to notice is that lobe width is inversely proportional to the antenna size, i.e., the larger antenna the smaller lobe we can obtain. The idea behind SAR is to artificially synthesise a big antenna by moving the platform. Traditionally, this operation is performed in the frequency domain using FFT like methods, e.g., the Fourier-Hankel method (Fawcett, 1985; Hellsten and Andersson, 1987; Andersson, 1988) or the ω-K migration methods (Cafforio et al., 1991; Rocca, 1987; Milman, 1993). The common denominator of these methods is that they assume that the aircraft's (or antenna's) flown path is linear and that is generally not the case in practice. If the trajectory is not linear the integration will result in an unfocused image. It is possible to partly correct for the deviation from the nonlinear trajectory but then the methods become computationally inefficient. Another method that can be used is so called global back-projection method that will be outlined below.

Given the raw (possibly complex) data $z_t(R)$ we can back-project each radar echo on the image giving the subimage I_t and each reflector will create a circle in each subimage. The total image can then be created by summing up all the subimages along the synthetic aperture, (Natterer, 1986), (i.e., solving the back-projection integral in discrete time)

$$I = \sum_{t=1}^{N} I_t \tag{1}$$

Another way of creating the image is to integrate the raw data for each pixel in the image I_{ij} as

$$I_{ij} = \sum_{t=1}^{N} z_t(R_t^{ij}) \tag{2a}$$

$$R_t^{ij} = \|p_t - s^{ij}\|_2 \tag{2b}$$

where p_t is the position of the platform and s^{ij} is the position in the scene which corresponds to the pixel (i, j). This method is schematically illustrated in Figure 3 for a simple scene with only one point target and where only a few platform positions are considered, here $N = 5$. In SAR practice, N is in the order of a few thousand. The complexity of this operation is proportional to $\mathcal{O}(M^2N)$ for an $M \times M$ image and N time points. However, by means of coordinate transformation, an approximation to exact back-projection can be performed, which is called Fast Factorised Back-projection, see Ulander et al. (2003).

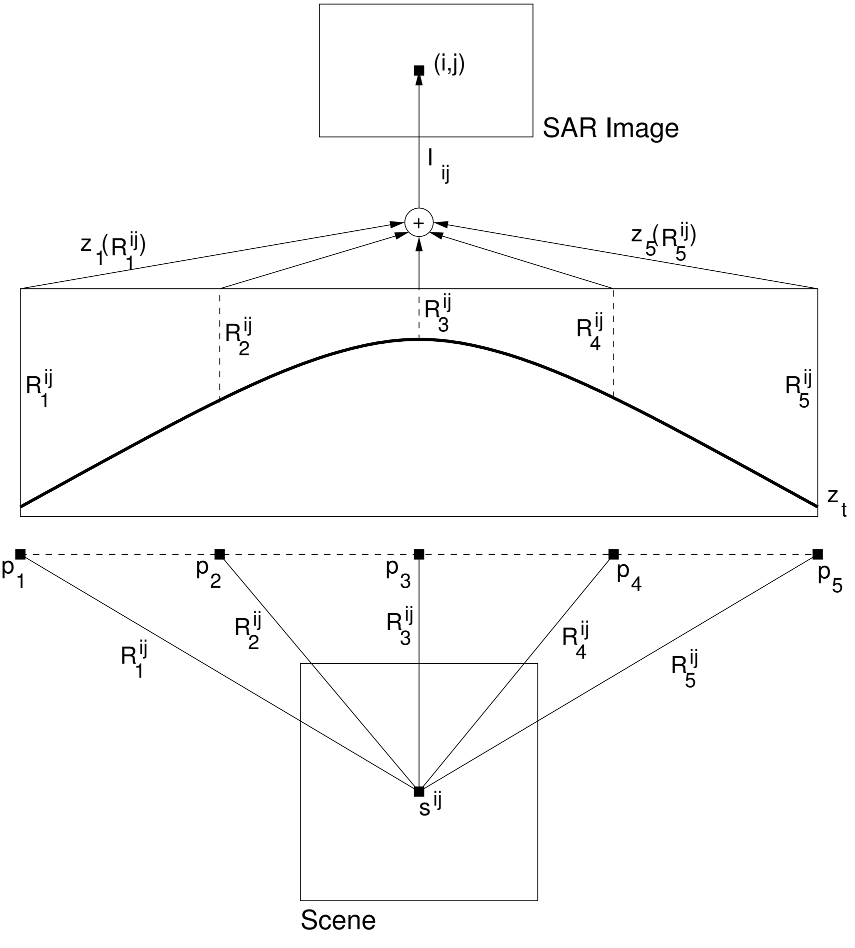


Figure 3: *The global backprojection method for creating SAR images. The scene consists of only one point target in this illustration. The figure is not to scale.*

The complexity of this algorithm is proportional to $\mathcal{O}(M^2 \log N)$ operations, which for large N implies an important saving. With this faster algorithm it should be possible to create images in real time, possibly in dedicated hardware. This possibility opens up for applications where SAR images and navigation can be done concurrently.

3 Motion Models

Precise knowledge of the antenna position p_t in (2b) is apparently crucial. The on-board inertial navigation system (INS) provides a nominal trajectory $p_{1:N}$ that can be used to construct a first (un-focused) SAR image. Our approach to focus the SAR image is based on computing a refined trajectory $p_{1:N}$. In the sequel, we will implicitly only model the deviation from the nominal trajectory, so p_t denotes the difference of the true position and the INS position.

This error trajectory of the platform is assumed to follow a simple second order dynamics expressed in discrete time as

$$p_{t+1} = p_t + T_s v_t + \frac{T_s^2}{2} a_t \tag{3a}$$

$$v_{t+1} = v_t + T_s a_t \tag{3b}$$

$$a_{t+1} = a_t \tag{3c}$$

where $p_t = [X_t \; Y_t \; Z_t]^T$ is the platform's position relative to the beginning of the synthetic aperture and $v_t = [v_t^X \; v_t^Y \; v_t^Z]^T$ is its velocity. This model of the kinematics allows us to calculate the whole trajectory if all the initial states, p_0 and v_0, and acceleration sequence, $a_{0:N}$, are known. The initial position is basically arbitrary since the SAR image can be translated freely, but the initial velocity is not, since it is influencing the trajectory's shape. It is therefore no loss of generality to take $p_0 = 0_{3\times 1}$, to define the navigation frame. In that case, the trajectory is related to this zero-frame. However, this frame can be translated and rotated and the trajectory will follow as a rigid body. Usually, the accelerations are measured by the onboard inertial measurement unit (IMU), but these are not perfect, and that will also cause an error in the trajectory. These errors will in turn cause the SAR image to become out of focus as mentioned before. So the best focus, i.e., the sharpest image should be produced if correct kinematic states are used. The main approach in this work is basically to use optical images or maps to match the SAR image to these in order to determine the initial states, i.e., both trajectory parameters and global position. This can be done by minimising some criterion that depends on parameters for SAR image's global position and orientation and kinematic states from the model in Equation (3). These parameters are collected into a vector θ, which can for example be $\theta = [r \; a \; \chi \; v_0^X \; v_0^Y \; a_0^X \; a_0^Y]^T$, where r, a and χ are the pixel positions and orientation of the SAR image relative to the optical map image and the other ones are the interesting kinematic states. Since these will be treated in a slightly different manner, we divide θ into an image part, θ_I, and a kinematic part, θ_K, as $\theta = [\theta_I^T \; \theta_K^T]^T$. One reason is the fact that the image matching procedure to estimate θ_I is the difficult part of the procedure. The variation of the kinematic part of the parameters, θ_K, will just create different SAR images and each such image must be matched to the optical image. In Section 4 the

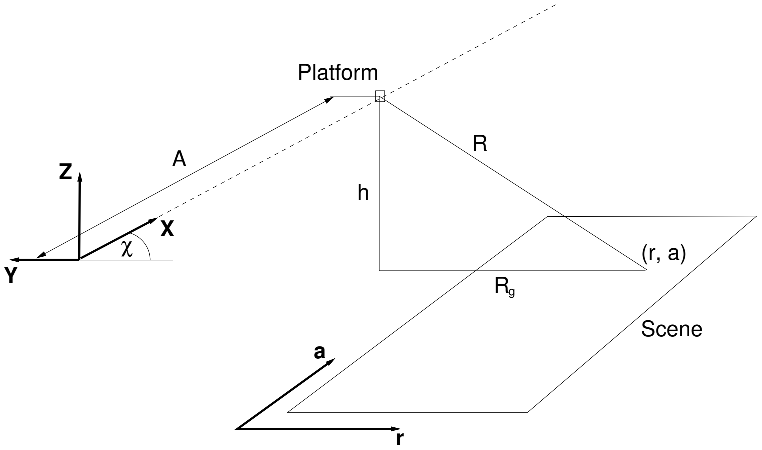


Figure 4: *Basic SAR geometry. Figure is not to scale.*

image matching approach (the one that estimates θ_I) will be described in detail. The main assumption is that the best focused (sharpest) image will give the best matching criterion and in that way global position of the platform can be resolved and the most focused image created.

3.1 SAR Geometry

Aerial images and maps are orthorectified, and for matching purposes the SAR image also needs to be orthorectified. This section describes the geometrical transformation to get a SAR image in the horizontal plane.

The basic SAR geometry is illustrated in Figure 4. SAR images use a coordinate system consisting of azimuth direction which is parallel to antenna's moving direction (in most cases the same as the platform's moving direction), here called X, and range direction which is perpendicular to antenna's moving direction, here called Y. Range direction can either be slant range or ground range coordinates. SAR images are naturally slant range images and the advantage of using those is that there is no need for transforming these to ground range images. But if we want to match these to the map images, the map must be transformed to the slant range image.

In order to relate the navigation frame and the SAR image pixels some basic geometry must be defined. For any pixel coordinate in the SAR image, (r, a), we can calculate its position in the navigation frame, $s^{ra} = [R_g \; A]^T$, as (see Figure 4 for geometry)

$$A = a\Delta_A \tag{4a}$$

$$R_g = \sqrt{R^2 - h^2} = \sqrt{(r\Delta_R + R_0)^2 - h^2} \tag{4b}$$

where

- A is the position of the pixel in the azimuth direction (X).
- R_g is the position of the pixel in the range direction (Y).

- h is the pixel's altitude below the navigation frame (Z).
- R is the slant range from the X-axis of the navigation frame to the pixel point on the ground (parallel with the Y-axis).
- r is the pixel's coordinate in the image's range direction.
- Δ_R is the resolution in the SAR image's range direction.
- R_0 is minimum slant range in the image (first row in the SAR image is on the range R_0).
- a is the pixel's coordinate in the image's azimuth direction.
- Δ_A is the resolution in the SAR image's azimuth direction.

Assuming we have a successful matching of the SAR image to the optical image, then a correspondence between the SAR image pixels and the optical image pixels is obtained giving the true geographical positions of the SAR image pixels, (r_G, a_G), since the optical image pixels have known geographical positions. The platform's average direction of the flight, χ, i.e., the angle between the North-axis and the X-axis of the navigation frame, is part of the matching results and is obtained directly, since this is the rotation of the SAR image relative to the map image. With these given, it is now possible to calculate the navigations frame's true position as

$$\hat{p}_0 = \begin{bmatrix} \hat{X}_0 \\ \hat{Y}_0 \\ \hat{Z}_0 \end{bmatrix} = \begin{bmatrix} a_G \\ r_G \\ 0 \end{bmatrix} + \begin{bmatrix} \sin(\chi) & \cos(\chi) & 0 \\ -\cos(\chi) & \sin(\chi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_g \\ A \\ h \end{bmatrix} \tag{5}$$

where $[\hat{X}_0\ \hat{Y}_0\ \hat{Z}_0]^T$ is the navigation frame's position in the global coordinates. Note that Z-coordinate is the same as altitude h, which we in general can obtain directly from the barometric measurements. If these measurements are not available or have bad performance, the approach from Sjanic and Gustafsson (2012) can be used to obtain an altitude estimate.

Since the kinematic parameters are also part of the matching results, any position in the trajectory can now be calculated by using Equation (3). Note that these calculations are valid under the flat earth approximation, which is valid if the SAR image is fairly close to the platform. This is true for most aircraft (but not satellites). The procedure described above is summarised in Algorithm 2.

4 Image Matching Approach

In step 4 of Algorithm 2, we simply defined a matching step which directly delivers the interesting parameters. In this section, one such matching algorithm will be described in more detail since it is a prerequisite in order to obtain the parameters used to calculate a platform's position and orientation. The matching between those images can be obtained in many ways, for example by simple correlation or by using image point features extracted by some point feature detectors, like Harris corner detector (Harris and Stephens,

Algorithm 2 SAR-OptMAP Global Position Calculation

```
Require: $(r_i, a_i)$, $i = 1, \dots, N$, $R_0$, $\Delta_R$, $\Delta_A$, $[\bar{X}_0\ \bar{Y}_0\ \bar{Z}_0]^T$, $\bar{\chi}$ (information from navigation system), Optical image or map with known geographical coordinates
Ensure: $[\hat{X}_0\ \hat{Y}_0\ \hat{Z}_0]^T$
for $i = 1 : N$ do
    Calculate $A_i$ with $a_i$, $a_{\text{last}}$ and $\Delta_A$ from (4a)
    Calculate $R_i$ with $r_i$, $R_0$ and $\Delta_R$ from (4b)
    Match the SAR image to the map image to obtain $r_{G,i}$, $a_{G,i}$, $\chi_i$ and kinematic states with $[\bar{X}_0\ \bar{Y}_0\ \bar{Z}_0]^T$ and $\bar{\chi}$ as a prior for initialisation
end for
Calculate $[\hat{X}_{0,i}\ \hat{Y}_{0,i}\ \hat{Z}_0]^T$ with $h$, $A_i$, $R_{g,i}$, $\chi_i$, $r_{G,i}$ and $a_{G,i}$ from (5) for all $i$
Calculate $[\hat{X}_0\ \hat{Y}_0\ \hat{Z}_0]^T$ from $[\hat{X}_{0,i}\ \hat{Y}_{0,i}\ \hat{Z}_0]^T$ for all $i$ as e.g., weighted mean with weights determined by the covariance of the $[\hat{X}_{0,i}\ \hat{Y}_{0,i}\ \hat{Z}_0]^T$
(Optional) Calculate any trajectory position in the global coordinates, $\hat{p}_t$, by using model 3 and estimated kinematic parameters
```

1988) or SIFT detector (Lowe, 2004). However, although the SAR and optical map images can share many similarities, in particular over man-made structured environments, they can be very different in their structure and appearance. For example, structures like rooftops can have completely different intensities, very bright in the SAR images and very dark in the optical images, and the above-mentioned methods might not work satisfactory. In this case it might be better to increase the feature complexity one level and use the lines (edges) in the images. See Wagner (2007) or Taylor and Kriegman (1995) for examples where edges are used as image features. Exactly as for point features, there are several well known edge detectors, where Sobel, Prewitt and Canny (Canny, 1986) are maybe the most known ones. Since the Canny edge detector is quite robust to noise, it is suggested as the detector in the approach described here. By applying this detector to SAR and optical images, two binary edge images are obtained. The next problem to be solved is to match these SAR and optical map binary edge images to each other. One well known method for parametric matching of templates to the image is so called Chamfer matching method, see Barrow et al. (1977), Borgefors (1988) or Ericsson and Thid (2006). Since this is quite a robust matching method, it will be the basis of the approach proposed here. Next, a short description of the Canny edge detector will be given as well as an introduction to Chamfer matching and the modifications we propose for this particular application.

4.1 Edge Detector

The Canny edge detector uses image gradient and thresholding to detect edges in the images like many other detectors. Its main advantage is better robustness to the noise in the images. This is obtained by using hysteresis with two thresholds, one high and one low. This avoids the problem of broken edges, or streaking, which is almost always present in detectors with only one threshold. The higher threshold is used to detect edges, just as in any detector, while the lower one is used to implement hysteresis and keep an edge even if the gradient response would fall under the higher threshold. The general problem of threshold tuning still remains. Individual thresholds for different images must be found on a case by case basis. In this work we are using an existing Canny edge detector implemented in the Image Processing Toolbox in MATLAB.

4.2 Chamfer Image Matching

The basics of the Chamfer image matching is the distance transform of the edge image to which the template image is to be matched. In this context the template image is considered to be the edge pixels of the binary edge image. The distance transform is calculated by assigning the pixels in the binary image a value of the distance to the closest nonzero pixel. The distance metric is usually Euclidean, but also Manhattan distance (1-norm) or even maximum norm can be used. As an illustrative example, consider a simple binary 7×7 image represented as a matrix

$$I = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The distance transform of this image using Euclidean distance is

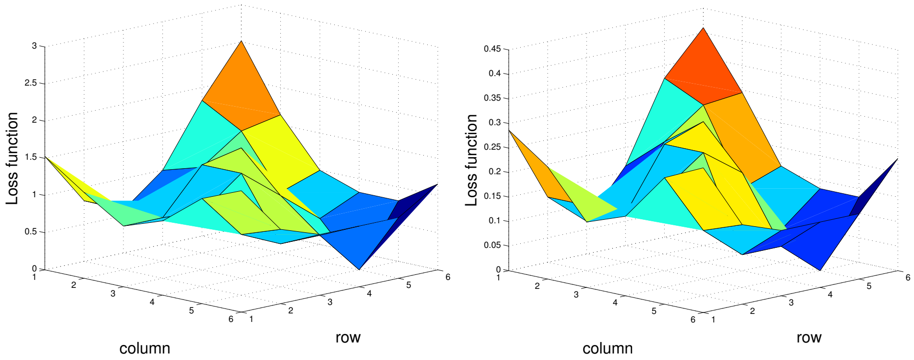
$$D = \begin{bmatrix} \sqrt{5} & 2 & \sqrt{2} & 1 & \sqrt{2} & 2 & \sqrt{5} \\ \sqrt{2} & 1 & 1 & 0 & 1 & 1 & \sqrt{2} \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 2 & 1 & 0 & 1 \\ \sqrt{2} & 1 & \sqrt{2} & 1 & 1 & 0 & 1 \\ \sqrt{5} & 2 & 1 & 0 & 0 & 0 & 1 \\ \sqrt{10} & \sqrt{5} & \sqrt{2} & 1 & 1 & 1 & \sqrt{2} \end{bmatrix}$$

Now the idea in Chamfer matching is to overlay the edge pixels of the binary template image, T, on the distance image for different translation, rotation and scaling values and calculate some loss function as some metric based on the values in the distance transform image that are hit by the template edge pixels, for example, the total sum of the values. From the implementation point of view this is equivalent to taking a whole binary template image as a matrix and element-wise multiplying it with the distance transform image. The reason is simply the fact that edge pixels have value 1 and non-edge ones have value 0. This can be written as

$$\eta(\theta_I) = D \odot \widetilde{T}(\theta_I) \tag{6a}$$

$$C(\theta_I) = f(\eta(\theta_I)) \tag{6b}$$

where $\eta(\theta_I)$ is the matrix resulting from the element-wise product ($\odot$) of the extended template image, $\widetilde{T}$, and the distance transform image, D. In the general case $\theta_I = [r\ a\ \chi\ s_r\ s_c]^T$, where we introduced image scaling parameters s_r and s_c. Sometimes it is possible to take the subset of the θ_I if, for example, some of the parameters are known



(a) *Chamfer matching loss function, $C(r, a)$, of the matching example.*

(b) *Nonlinear least squares loss function, $V(r, a)$, of the matching example.*

Figure 5: *Different loss functions for the matching example.*

or not estimated. The extended template image, $\widetilde{T}$, has been created by first rotating the original template image with χ degrees and scaling it s_r and s_c times in row and column directions respectively. The binary image created from this template is then extended with zeros to the size of D in such a way that the upper left corner of the template image is on the coordinate (r, a). Here, $f : \mathbf{R}^{\text{size}(D)} \to \mathbf{R}_+$ is some positive and monotonously increasing function. This means that for correct matching parameters, the loss function $C(\theta_I)$ would obtain its minimum value and the parameter estimates are obtained as

$$\hat{\theta}_I = \arg\min_{\theta} \, C(\theta_I) \tag{7}$$

If the template which is to be matched to the image above is

$$T = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

and only translation is considered, i.e., $\theta_I = [r \ a]^T$, the surf plot of the resulting loss function, $C(\theta_I)$, is according to Figure 5a. It can be seen that the minimum value is obtained for the translation parameters $r = 5$ (row) and $a = 5$ (column) which is the best possible match. The function f used here is the RMSE value of the element-wise product of the distance transform image and the extended template image

$$C(r, a) = \sqrt{\frac{1}{N_{\text{nz}}} \sum_{k=1}^{7} \sum_{l=1}^{7} \eta_{k,l}(r, a)^2}, \tag{8}$$

where $\eta_{k,l}(r, a)$ is the matrix of values according to (6a), except that rotation and scaling are not considered. Here, N_{nz} is the amount of nonzero elements in the extended template image, here $N_{\text{nz}} = 3$.

4.3 Modified Matching Approach

In this work, a slightly modified loss function is proposed, which bears more similarity to the well known least squares approach. The reason is that we need an uncertainty measure to the position estimate, otherwise higher level fusion with the on-board navigation system would be problematic. To get a statistically correct measure of covariance is a complicated problem, but at least we get a matrix that has the most essential properties of a covariance matrix: it is a positive definite symmetric matrix, it reveals lack of excitation by having a high condition number, and it shows relative size of estimation errors by having different size of the diagonal elements.

First, the distance transformed binary map image, D, is transformed as

$$\widetilde{D} = \exp(-D) \tag{9}$$

where $\exp(\,\cdot\,)$ function acts element-wise. This will basically "invert" the distance transform making zero valued pixels become ones and high valued pixels become low valued. For the example above $\widetilde{D}$ looks like

$$\widetilde{D} = \begin{bmatrix} 0.11 & 0.14 & 0.24 & 0.37 & 0.24 & 0.14 & 0.11 \\ 0.24 & 0.37 & 0.37 & 1 & 0.37 & 0.37 & 0.24 \\ 0.37 & 1 & 0.37 & 0.37 & 0.37 & 1 & 0.37 \\ 0.37 & 1 & 0.37 & 0.14 & 0.37 & 1 & 0.37 \\ 0.24 & 0.37 & 0.24 & 0.37 & 0.37 & 1 & 0.37 \\ 0.11 & 0.14 & 0.37 & 1 & 1 & 1 & 0.37 \\ 0.04 & 0.11 & 0.24 & 0.37 & 0.37 & 0.37 & 0.24 \end{bmatrix}$$

Let $\xi(\theta_I)$ be the $N_{\text{nz}} \times 1$ vector of values from $\widetilde{D}$ hit by the translated, rotated and scaled edge pixels of the binary template, $T(\theta_I)$. Then we have the following relation

$$\mathbf{1}_{N_{\text{nz}}} = \xi(\theta_I) + e \tag{10}$$

where $\mathbf{1}_{N_{\text{nz}}}$ is the $N_{\text{nz}} \times 1$ vector of ones and e is some noise. This relation can be interpreted as a measurement equation which is a function of parameter vector θ_I, and then the minimisation criterion can be written as

$$\hat{\theta}_I = \arg\min_{\theta_I} V(\theta_I), \tag{11a}$$

$$V(\theta_I) = \frac{\|\mathbf{1}_{N_{\text{nz}}} - \xi(\theta_I)\|_2^2}{2N_{\text{nz}}} = \frac{1}{2N_{\text{nz}}} \sum_{k=1}^{N_{\text{nz}}} (1 - \xi_k(\theta_I))^2 \tag{11b}$$

which is a nonlinear least squares formulation. The modified loss function, $V(r, a)$, for the example is depicted in Figure 5b. This loss function has a very similar shape as the original one, $C(r, a)$, but it is a little bit steeper close to the minimum. Since both of these loss functions are defined on a grid of discrete values, the minimisation procedure can be performed as a global grid search.

Besides the parameter values, $\hat{\theta}_I$, it is also desirable to estimate the covariance which in turn can be used to estimate the covariance of the estimated navigation parameters, position of the navigation frame, $[\hat{X}_0\ \hat{Y}_0\ \hat{Z}_0]^T$, and track angle, $\hat{\chi}$. These covariances can then be used as for weighting purpose in Algorithm 2. The covariance can be estimated by assuming a locally quadratic function around the minimum value of the loss function, $V(\hat{\theta}_I)$, and estimating the Hessian matrix, H. This can be done by solving the overdetermined linear system of equations originating from the following relation

$$V(\hat{\theta}_I + \Delta) \approx V(\hat{\theta}_I) + \Delta^T H \Delta \tag{12}$$

where a Taylor expansion around $\hat{\theta}_I$ is performed for some Δ assuming that the gradient is zero (since $V(\hat{\theta}_I)$ is a stationary point, it is the minimum value). Then the covariance of the parameter estimates can be estimated as

$$\mathrm{Cov}(\hat{\theta}_I) = \hat{\lambda} H^{-1} \tag{13}$$

where $\hat{\lambda} = V(\hat{\theta}_I)$, see Gustafsson et al. (2010). Note that in the example above we obtain the covariance which equals zero for both parameters and it is natural since the template fits perfectly, and there is no uncertainty. In the general case, however, the template will not fit perfectly and there will always be some uncertainty in the estimates.

5 Kinematic Parameter Estimation

The procedure described in Section 4 considers how to estimate the image parameters θ_I given a SAR image produced with some trajectory $\hat{p}_{0:N}$. By varying the values of the kinematic part of the parameter vector, θ_K, and using the model (3) and possibly accelerations measured by the onboard IMU, different SAR images $I(\theta_K)$ can be obtained. Each of these images can now be matched according to the solution of Equation (11) which produces another loss function,

$$J(\theta_K, \hat{\theta}_I) = V_{\theta_K}(\hat{\theta}_I), \tag{14}$$

where $V_{\theta_K}(\hat{\theta}_I)$ is the value of the image matching loss function obtained for a SAR image created with θ_K as kinematic parameters. The loss function J is, exactly as V, a non-convex function with many local minima, implying that a grid search is the best option to find a solution according to

$$\hat{\theta}_K = \arg\min_{\theta_K} J(\theta_K, \hat{\theta}_I). \tag{15}$$

This will give the total solution $\hat{\theta} = [\hat{\theta}_I^T \; \hat{\theta}_K^T]^T$, with the best focus in this metric, and an accurate global position. The covariance estimation approach from Equations (12) and (13) can be used here as well in order to obtain covariance of the kinematic parameters.

6 Results

In this section we will present both the results for the image matching approach and the kinematic parameter estimation. The image matching approach will be presented in more detail on two example patches from the SAR image assuming the focused image. For the kinematic parameter estimation, a low-frequency SAR simulation environment, CARABAS II in particular, (Hellsten et al., 1996), is used where it is possible to vary

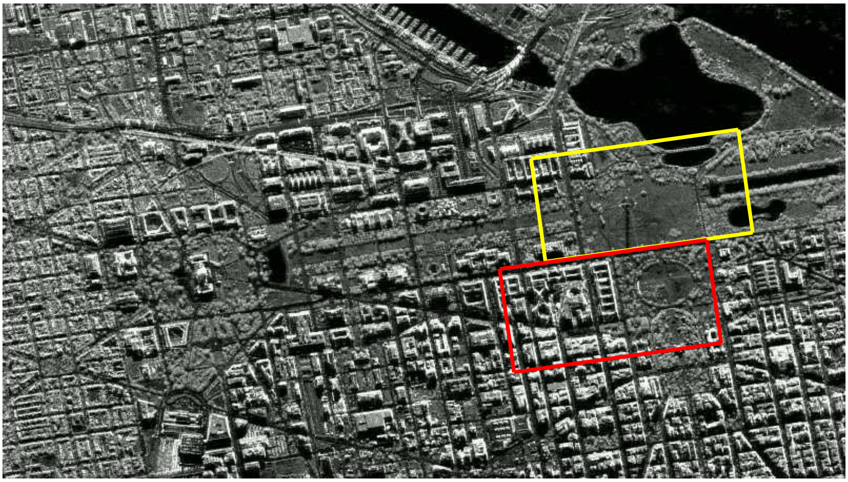


Figure 6: *Zoomed part of the SAR image with the three patches used in the Chamfer matching procedure.*

Parameter	Error	Standard Deviation
r [pixels]	0, 2	5.73, 7.73
a [pixels]	0, -3	6.82, 7.40
χ [degrees]	1, 1	5.26, 6.49

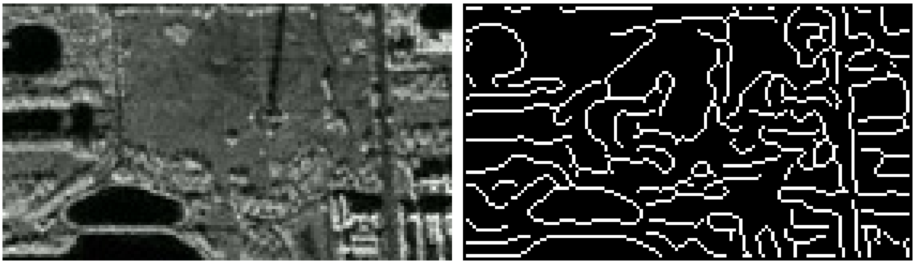
Table 1: *Errors and standard deviations of the parameter estimates for the two different example SAR patches.*

trajectories and create different SAR images. This environment makes it possible to create realistic SAR images of the scene and to evaluate the focusing results in a controlled manner.

6.1 Results of the Image Matching Approach

In order to show the results of the matching procedure described in Section 4, two patches from the SAR image in Figure 1 are matched to the optical image in Figure 2. These two patches are depicted in Figure 6. Parameters that are optimised over are translations and rotation, and the scaling is fixed beforehand in order to minimise the parameter space and speed up the search. Notice that it is the optical image that has been fixed north up and the SAR image that has been rotated. In that case the flight direction angle, χ, is directly obtained. It should also be pointed out that in the search for the matching parameters a prior from the navigation system is used to narrow down the search space and in that way prune possible false solutions due to a possibly too similar environment. The results are presented both graphically, where SAR image patches are overlaid on the optical image, and in a table with an error and a standard deviation of the estimates.

The SAR patches rotated with the angle $\hat{\chi}$ obtained in the optimisation are depicted in



***(a)** SAR image patch.* ***(b)** Binary edge image of the SAR patch.*

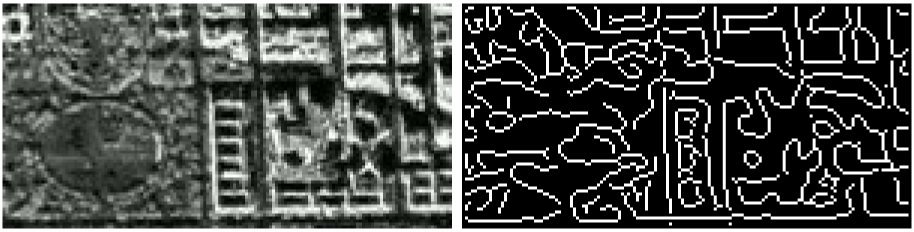
***(c)** Optical map with SAR image patch overlaid.* ***(d)** Optical map with the frame placed on the matching pixels.*

***Figure 7:** Example 1: SAR image patch and its edge image and optical image map with the SAR patch overlaid on the pixels given by the solution to the matching.*

Figure 7a and Figure 8a. The result of the matching is depicted in Figure 7c and Figure 8c by overlaying the SAR image patch on the optical image on the solution pixels, $(\hat{r}, \hat{a})$. The errors and the standard deviations of the estimates for these two cases are presented in Table 1.

6.2 Results for the Kinematic Parameters Estimation

For the kinematic parameter estimation results, one patch from the SAR image is chosen, see Figure 9. This patch is produced with the CARABAS II simulation environment, and it can be seen that it is quite realistic. First, a simple case where initial velocity in X-direction is unknown while all the other parameters are known is examined. In this case, $\theta_K = v_0^X$, and the error in speed was varied, first between -4% and 4%. The resulting loss function $J(\theta_K, \hat{\theta}_I)$ is depicted in Figure 10. Here it can be seen that the minimum value is obtained for the correct initial velocity, and furthermore there is no matching error. Note that grid has higher resolution in the middle of the plot. Another simple case that is examined concerns variation of initial acceleration in the Y-direction, while all the other parameters are known. In this case $\theta_K = a_0^Y$ and the resulting loss function is



***(a)** SAR image patch.*

***(b)** Binary edge image of the SAR patch.*

***(c)** Optical map with SAR image patch overlaid.*

***(d)** Optical map with the frame placed on the matching pixels.*

***Figure 8:** Example 2: SAR image patch and its edge image and optical image map with the SAR patch overlaid on the pixels given by the solution to the matching.*

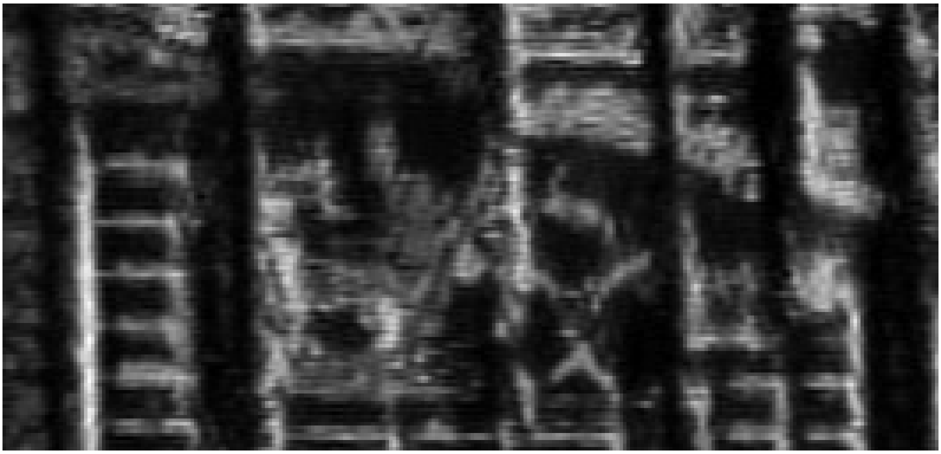


***Figure 9:** SAR image patch used for the evaluation of the kinematic parameters estimation.*

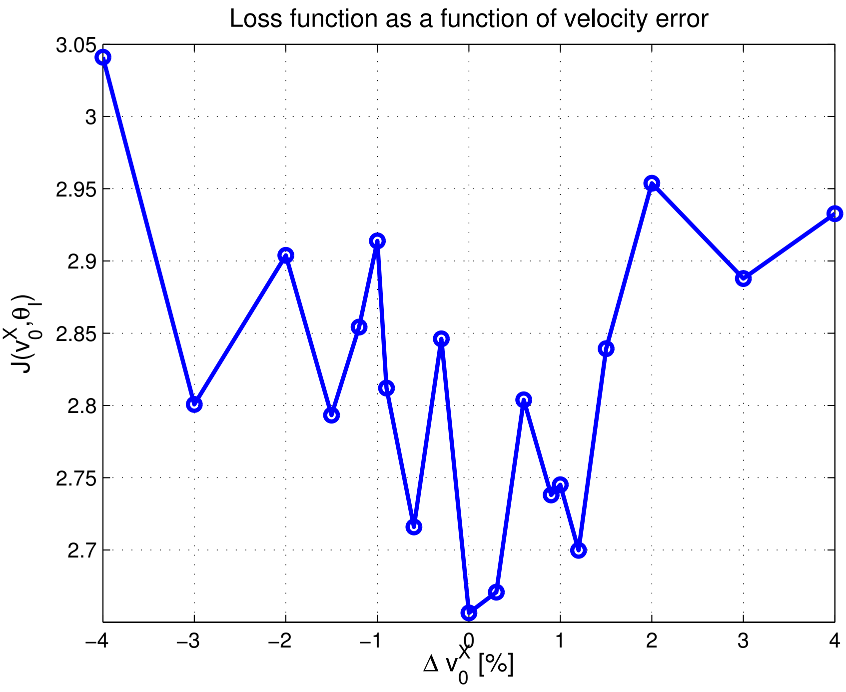


Figure 10: *The value of the loss function as a function of a percentual error in initial velocity in X-direction.*

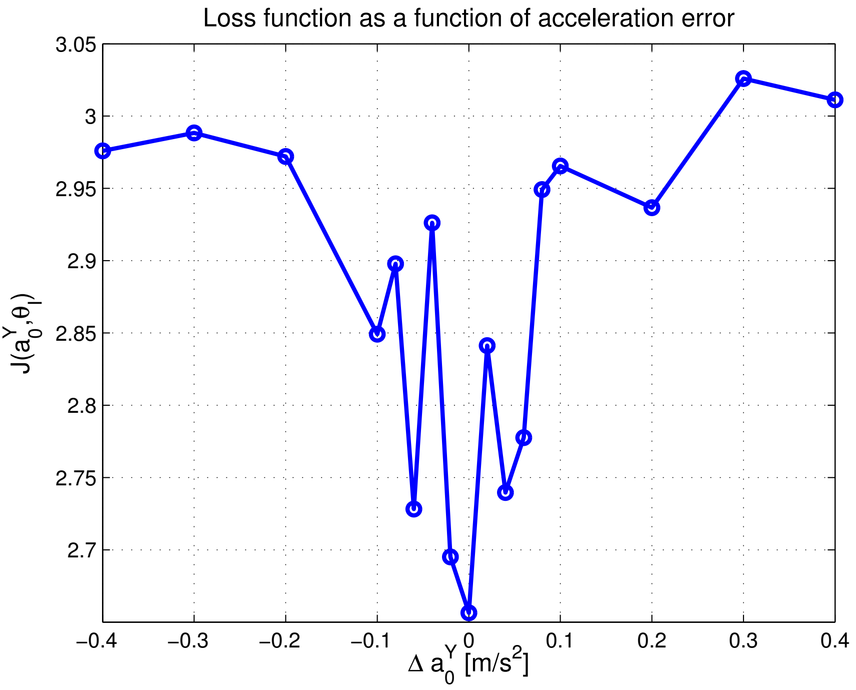
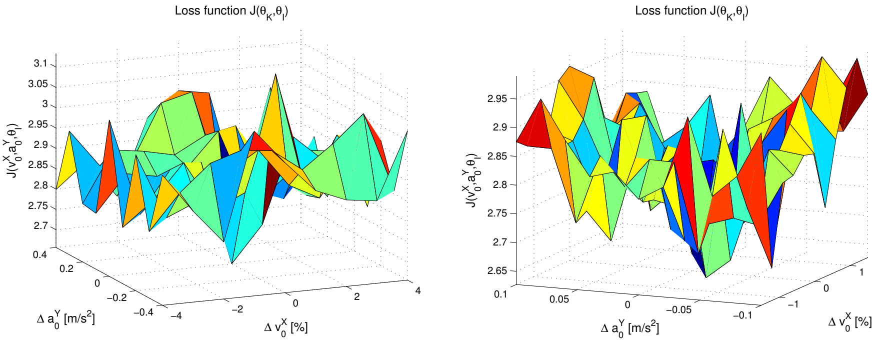


Figure 11: *The value of the loss function as a function of an error in initial acceleration in Y-direction.*

presented in Figure 11. Even here the minimum value is obtained for the correct value of the acceleration, and there was no matching error in this case either. In both cases it can be seen that loss function has much more irregular behaviour close to the optimum and that some of the loss function values are close to the value for the correct acceleration. If both initial velocity in X-direction and acceleration in Y-direction are set as parameters, i.e., $\theta_K = [v_0^X \; a_0^Y]^T$, the loss function is according to Figure 12 for both large and small error in states. In this case the global minimum of the loss function is obtained for a correct value if the error was large, i.e., the grid resolution was coarse, while a non-correct value of the parameters is obtained for the small error case, $\hat{\theta}_K = [-0.30 \; -0.02]^T$ ($[\% \; \mathrm{m/s^2}]$) although the correct value was the second smallest. This error is, however, quite small giving a trajectory RMSE error of about $6 \; \mathrm{m}$ and the SAR image patch resulting from this trajectory is shown in Figure 13. The matching errors in this case are 1 and 2 pixels in range and azimuth directions, respectively, and the error in rotation of the patch is $0.5°$. We see that the actual difference in the image quality is hard to distinguish with the naked eye, and that the navigation parameter estimates are also quite good. Note also that in all cases it is obvious that the loss function is highly non-convex and that a grid based search is necessary. This also implies that the grid resolution will set the accuracy limit, and the number of operations grows exponentially with the number of grid points. However, it is possible to evaluate each grid point individually, which suits parallel computation



(a) *Loss function for the large initial velocity and acceleration error.*

(b) *Loss function for the small initial velocity and acceleration error.*

Figure 12: *The value of the loss function as a function of an error in initial velocity in X-direction and acceleration in Y-direction.*

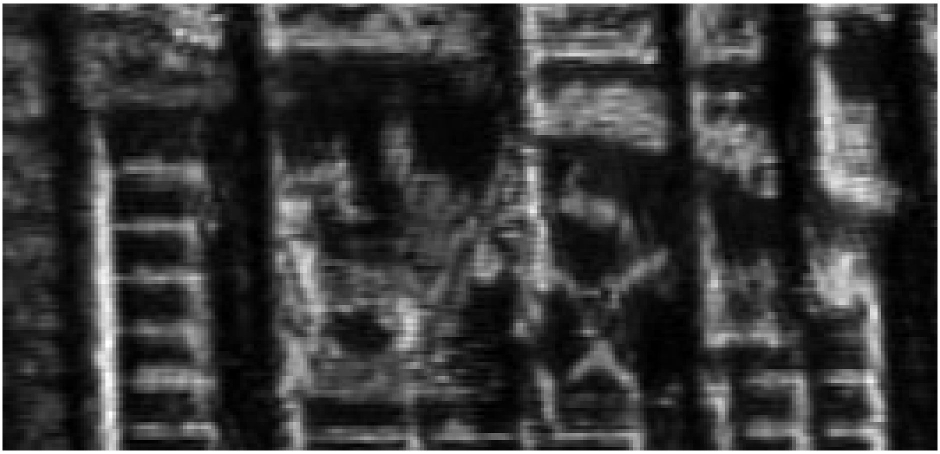


Figure 13: *SAR image patch obtained with the estimated trajectory given the values from the small error case.*

architectures.

6.3 Discussion

In this section a discussion on the method's results and performance and some possible improvements is provided. The performance of the method for both patch matching and kinematic parameter estimation is highly dependent on different quantisation effects. Both optical and SAR images have finite pixel resolution. This sets the limit on the performance of the edge detector and on the matching performance. Furthermore, the standard implementation of the Canny detector uses greyscale images which implies that additional quantisation is present. All these effects will put the limit on matching and rotation estimation performance. The obtained performance for the examples studied here is fairly good and in the magnitude of what can be expected.

For the kinematic parameters, the grid resolution will naturally set the limit on the performance. The finer grid, the better possibility to get good performance. But if the resolution is too small, the difference in the trajectories created with neighboring grid parameter values will not be enough to make SAR images different enough from the focusing point of view. Then, in practice, only the numerical accuracy and their effects will dominate. The grid size and resolution set also the limit on the execution speed. Therefore the grid resolution and size are seen as tuning parameters.

In Sjanic and Gustafsson (2013a), an auto-focusing approach based on the SAR image only is exploited using image entropy as a focus measure. The entropy is here defined as

$$E = -\sum_{i=1}^{M}\sum_{j=1}^{K} q_{ij} \log q_{ij} \tag{16a}$$

$$q_{ij} = \frac{|I_{ij}|^2}{\sum_k \sum_l |I_{kl}|^2}. \tag{16b}$$

where I_{ij} is the complex-valued pixel (i, j) in the SAR image. Note that E is a function of θ_K. Then, we could combine the entropy and the loss function $J(\theta)$ around the global minimum to improve the estimation results. In the case used above, the combined loss function $J + E$ is shown in Figure 14. We see that for this function a correct value of the parameters is obtained.

7 Conclusions and Future Work

A method of matching SAR images and optical images is presented, for the primary purpose of autofocusing and adding the radar reflectance image to ordinary images as another 'color' channel, which can be useful in remote sensing applications. The method is based on the pattern matching algorithm called Chamfer matching, which is modified to resemble a least squares formulation and a grid based optimisation of the kinematic parameters. For both cases a statistical performance measure, covariance, of the estimates can also be obtained. The evaluation of the results is performed on the SAR image and optical map image, and both matching performance and autofocusing performance is evaluated based on a couple of SAR image patches. The obtained results on the real SAR images

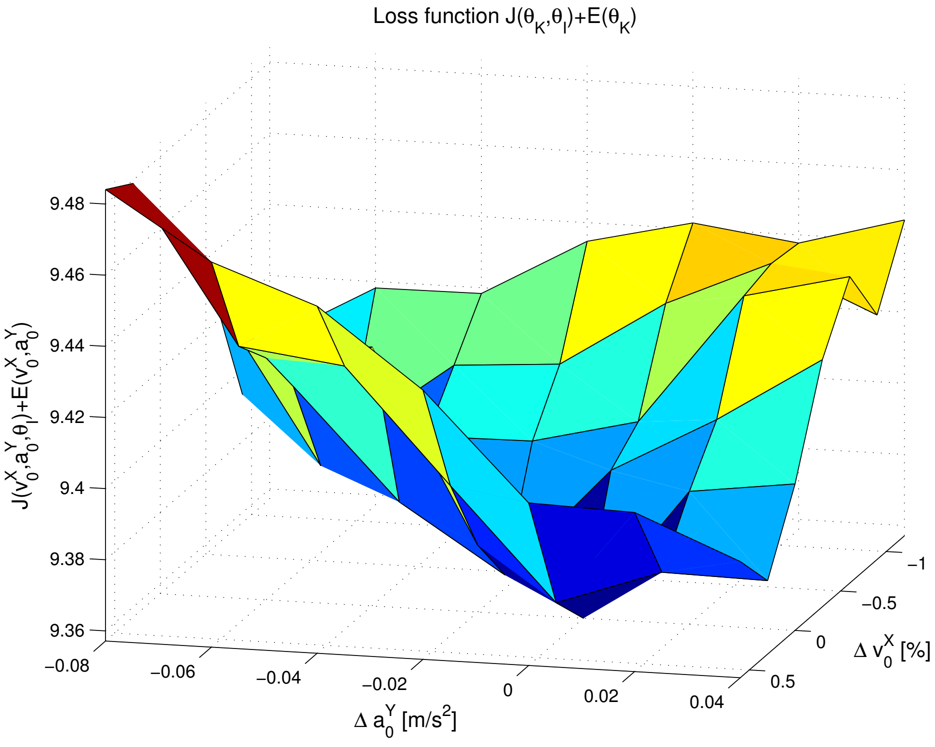


Figure 14: *Combined entropy, E, and loss function, J, around the global minimum of the loss function.*

and very simple optical map images from Google Maps show that the performance of the matching and autofocusing methods is fairly good, with small errors and variance, even with these simple means. However, it should also be pointed out that this method assumes a variation in the scene in order to work. The environment where edge features are hard to extract or missing will of course give much poorer results.

As a future extension of this work, as preliminary results show in the discussion above, entropy measure can be incorporated in the total cost function to eventually improve the results of the kinematic parameters estimation and in turn the autofocusing performance. As a future application, the methodology can also be used as an all-weather GNSS-like support and backup for the inertial navigation system.

Bibliography

L. E. Andersson. On the Determination of a Function from Spherical Averages. *SIAM Journal on Mathematical Analysis*, 19(1):214–232, 1988. doi: 10.1137/0519016. URL `http://link.aip.org/link/?SJM/19/214/1`.

H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: two new techniques for image matching. In *Proceedings of the 5th international joint conference on Artificial intelligence - Volume 2*, pages 659–663, San Francisco, CA, USA, 1977. Morgan Kaufmann Publishers Inc. URL `http://dl.acm.org/citation.cfm?id=1622943.1622971`.

G. Borgefors. Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10:849–865, November 1988. ISSN 0162-8828. doi: 10.1109/34.9107. URL `http://dl.acm.org/citation.cfm?id=56917.56924`.

C. Cafforio, C. Prati, and F. Rocca. SAR data focusing using seismic migration techniques. *IEEE Transactions on Aerospace and Electronic Systems*, 27(2):194 –207, March 1991. ISSN 0018-9251. doi: 10.1109/7.78293.

J. Canny. A Computational Approach to Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, November 1986. ISSN 0162-8828. doi: 10.1109/TPAMI.1986.4767851.

L. J. Cutrona, W. E. Vivian, E. N. Leith, and G. O. Hall. A high-resolution radar combat-surveillance system. *IRE Transactions on Military Electronics*, MIL-5(2):127–131, April 1961. ISSN 0096-2511. doi: 10.1109/IRET-MIL.1961.5008330.

J. Ericsson and A. Thid. Automatic SAR Image to Map Registration, January 2006. Master's Thesis, Chalmers, Göteborg, Sweden.

J. A. Fawcett. Inversion of N-Dimensional Spherical Averages. *SIAM Journal on Applied Mathematics*, 45(2):336–341, 1985. ISSN 00361399. URL `http://www.jstor.org/stable/2101820`.

J. R. Fienup. Phase Error Correction by Shear Averaging. In *Signal Recovery and Synthesis*, pages 134–137. Optical Society of America, June 1989.

M. Greco, G. Pinelli, K. Kulpa, P. Samczynski, B. Querry, and S. Querry. The study on SAR images exploitation for air platform navigation purposes. In *Proceedings of the International Radar Symposium (IRS), 2011*, pages 347 –352, sept. 2011.

B. Grelsson, M. Felsberg, and F. Isaksson. Efficient 7D Aerial Pose Estimation. In *IEEE Workshop on Robot Vision 2013, Clearwater Beach, Florida, USA, January 16-17, 2013*, 2013.

F. Gustafsson, L. Ljung, and M. Millnert. *Digital Signal Processing.* Studentliteratur, Lund, 2010.

C. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

H. Hellsten and L. E. Andersson. An inverse method for the processing of synthetic aperture radar data. *Inverse Problems*, 3(1):111, 1987. URL `http://stacks.iop.org/0266-5611/3/i=1/a=013`.

H. Hellsten, L. M. Ulander, A. Gustavsson, and B. Larsson. Development of VHF CARABAS II SAR. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 2747 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 48–60, June 1996.

L. Hostetler and R. Andreas. Nonlinear Kalman filtering techniques for terrain-aided navigation. *Automatic Control, IEEE Transactions on*, 28(3):315–323, 1983. ISSN 0018-9286. doi: 10.1109/TAC.1983.1103232.

F. Lindsten, J. Callmer, H. Ohlsson, D. Törnqvist, T.B. Schön, and F. Gustafsson. Georeferencing for UAV navigation using environmental classification. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1420 –1425, May 2010. doi: 10.1109/ROBOT.2010.5509424.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, November 2004. ISSN 0920-5691.

A. S. Milman. SAR Imaging by Omega-K Migration. *International Journal of Remote Sensing*, 14(10):1965–1979, 1993.

R. L. Jr. Morrison and D. C. Jr. Munson. An experimental study of a new entropy-based SAR autofocus technique. In *Proceedings of International Conference on Image Processing, ICIP 2002*, volume 2, pages II–441–4, September 2002. doi: 10.1109/ICIP.2002.1039982.

F. Natterer. *The Mathematics of Computerised Tomography*. New York: Wiley, 1986.

C. Oliver and S. Quegan. *Understanding Synthetic Aperture Radar Images*. The SciTech Radar and Defense Series. SciTech, 2004. ISBN ISBN 1-891121-31-6.

F. Rocca. Synthetic Aperture Radar: a New Application for Wave Equation Techniques. *Stanford Exploration Project SEP-56*, pages 167–189, 1987.

D. Scaramuzza and F. Fraundorfer. Visual Odometry [Tutorial]. *Robotics Automation Magazine, IEEE*, 18(4):80 –92, dec. 2011. ISSN 1070-9932. doi: 10.1109/MRA.2011.943233.

Z. Sjanic and F. Gustafsson. Fusion of Information from SAR and Optical Map Images for Aided Navigation. In *Proceedings of 15th International Conference on Information Fusion*, Singapore, July 2012.

Z. Sjanic and F. Gustafsson. Simultaneous Navigation and Synthetic Aperture Radar Focusing. Technical Report LiTH-ISY-R-3063, Department of Electrical Engineering, Linköping University, June 2013a.

Z. Sjanic and F. Gustafsson. Navigation and SAR focusing with Map Aiding. *Submitted to IEEE Transactions on Aerospace and Electronic Systems*, June 2013b.

C. J. Taylor and D. J. Kriegman. Structure and motion from line segments in multiple images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(11): 1021 –1032, nov. 1995. ISSN 0162-8828. doi: 10.1109/34.473228.

L. M. H. Ulander, H. Hellsten, and G. Stenstrom. Synthetic-aperture radar processing using fast factorized back-projection. *Aerospace and Electronic Systems, IEEE Transactions on*, 39(3):760–776, July 2003. ISSN 0018-9251. doi: 10.1109/TAES.2003. 1238734.

J. D. Wagner. Automatic Fusion of SAR and Optical Imagery, September 2007. Master's Thesis, Lebniz Universität, Hannover, Germany.

D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and C. V. Jr. Jakowatz. Phase gradient autofocus - a robust tool for high resolution SAR phase correction. *IEEE Transactions on Aerospace and Electronic Systems*, 30(3):827–835, July 1994. ISSN 0018-9251. doi: 10.1109/7.303752.

L. Xi, L. Guosui, and J. Ni. Autofocusing of ISAR images based on entropy minimization. *IEEE Transactions on Aerospace and Electronic Systems*, 35(4):1240–1252, October 1999. ISSN 0018-9251. doi: 10.1109/7.805442.

M. Xing, R. Jiang, X.and Wu, F. Zhou, and Z. Bao. Motion Compensation for UAV SAR Based on Raw Radar Data. *IEEE Transactions on Geoscience and Remote Sensing*, 47 (8):2870–2883, August 2009. ISSN 0196-2892. doi: 10.1109/TGRS.2009.2015657.

Paper C

Navigation and SAR Auto-focusing Based on the Phase Gradient Approach

Authors: Zoran Sjanic and Fredrik Gustafsson

Edited version of the paper:

Z. Sjanic and F. Gustafsson. Navigation and SAR Auto-focusing Based on the Phase Gradient Approach. In *Proceedings of 14th International Conference on Information Fusion*, Chicago, USA, July 2011.

Navigation and SAR Auto-focusing Based on the Phase Gradient Approach

Zoran Sjanic and Fredrik Gustafsson

Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
{zoran,fredrik}@isy.liu.se

Abstract

Synthetic Aperture Radar (SAR) equipment is an all-weather radar imaging system that can be used to create high resolution images of the scene by utilising the movement of the flying platform. It is therefore essential to accurately estimate the platform's trajectory in order to get good and focused images. Recently, both real time applications and smaller and cheaper platforms have been considered. This, in turn, leads to unfocused images since cheaper platforms, in general, have navigation systems with poorer performance. At the same time the radar data contain information about the platform's motion that can be used to estimate the trajectory and get more focused images. Here, a method of utilising the phase gradient of the SAR data in a sensor fusion framework is presented. The method is illustrated on a simulated example with promising results. At the end a discussion about the obtained results and future work is covered.

1 Introduction

The method of creating high-resolution radar images by utilising the relative motion between imaged scene and a platform that carries the radar, usually an aircraft, is known as Synthetic Aperture Radar (SAR). Typically, during flight, radar echos are collected along the trajectory and saved, giving the Real Aperture Radar (RAR) image with low resolution, see Figure 1. The partial sub images from the RAR image are then integrated to obtain the SAR image with much higher resolution, (Cutrona et al., 1961). This integration can be performed in the frequency domain with Fast Fourier transforms with its advantage of fast processing, (Fawcett, 1985; Hellsten and Andersson, 1987; Andersson, 1988; Rocca, 1987; Cafforio et al., 1991; Milman, 1993). However, these methods have one major shortcoming, they assume a straight flying trajectory in order to work. In reality, the trajectory will newer be straight, and some other method must be applied, or SAR images will be distorted and back-projection is one of the most known ones, (Natterer, 1986). In the back-projection procedure each saved radar echo, which is one dimensional, is back-projected onto a two dimensional area. In this way a low resolution image of

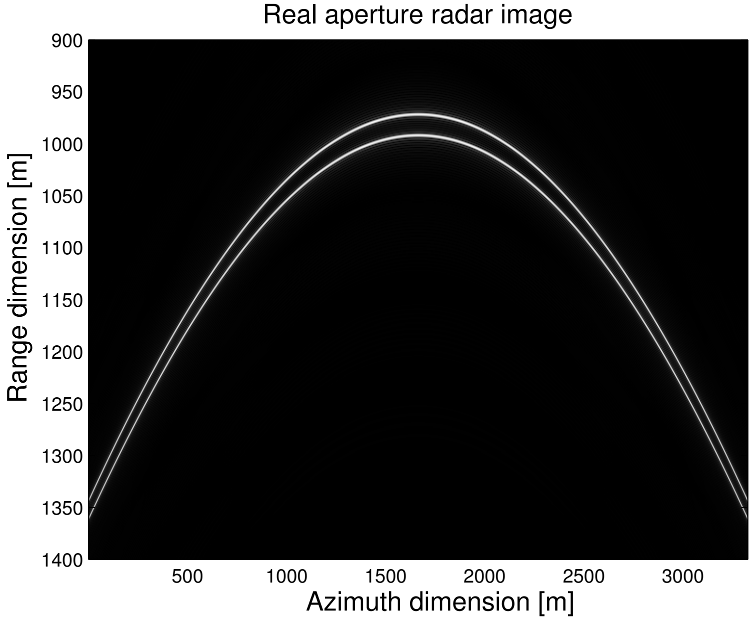


Figure 1: *Real aperture radar image (magnitude of the raw radar data) of the two point targets.*

the scene is obtained. Now we can sum up all these back-projected images in order to obtain the full SAR image. Figure 2 describes this procedure in a schematic way. It is now clear that in order to perform the back-projection operation the trajectory of the platform must be known and deviations from the real trajectory will cause the back-projected sub images to be shifted. The summation operation of the sub images will then cause defocusing, which is one of the most common SAR image distortions. This is illustrated in Figure 3, where Figure 3a is created using the true trajectory, resulting in a perfectly focused image, while Figure 3b is created using a trajectory where Gaussian white noise $w_t \sim \mathcal{N}(0, \sigma^2)$, $\sigma = 1.5$, is added to the cross-track position of the platform.

The process of correcting for this distortion is called auto-focusing and much effort has been spent on it during recent years, see for example Oliver and Quegan (2004); Yegulalp (1999); Wahl et al. (1994); Xi et al. (1999); Morrison and Munson (2002); Xing et al. (2009); Fienup (1989). Traditionally, these methods are open-loop type, meaning that the image is created with assumptions of linear flight trajectory and focusing is done afterwards in an open-loop way discarding possible flight trajectory information. This is a consequence of the off-line image generating process where the trajectory is no longer interesting. In the setup where SAR images are generated on-line, an idea is to use information from the image defocusing and navigation system together. The approach is to fuse this information in a sensor fusion framework and to try to obtain the best possible solution to both navigation states and image focusing simultaneously.

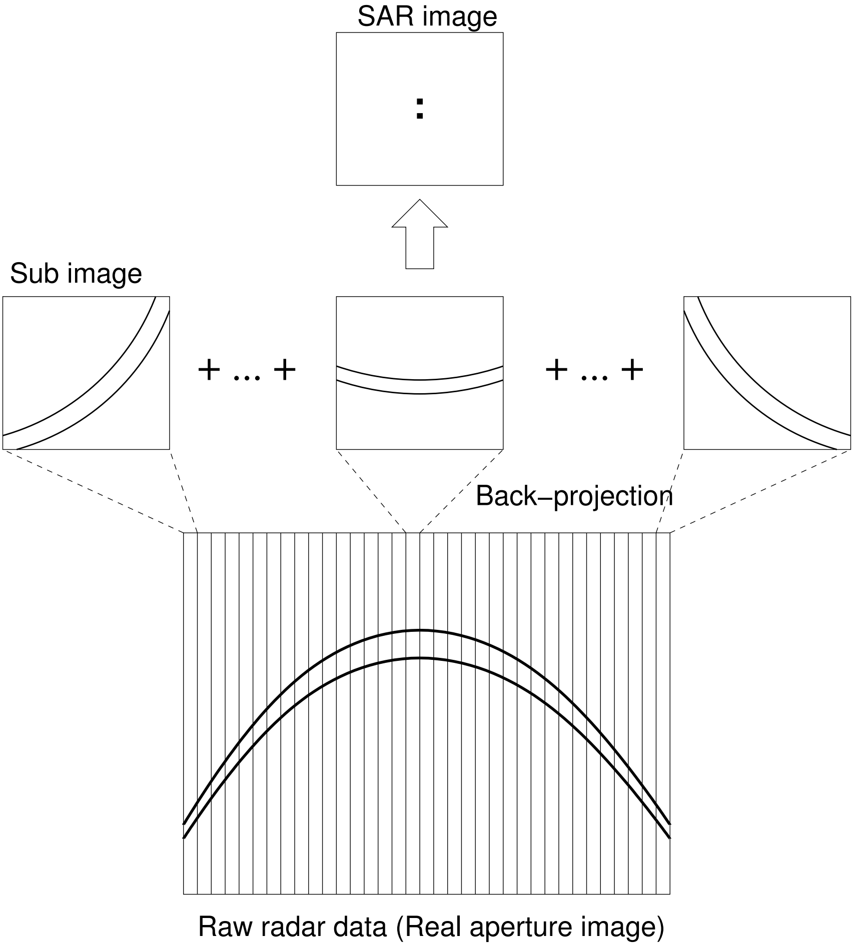
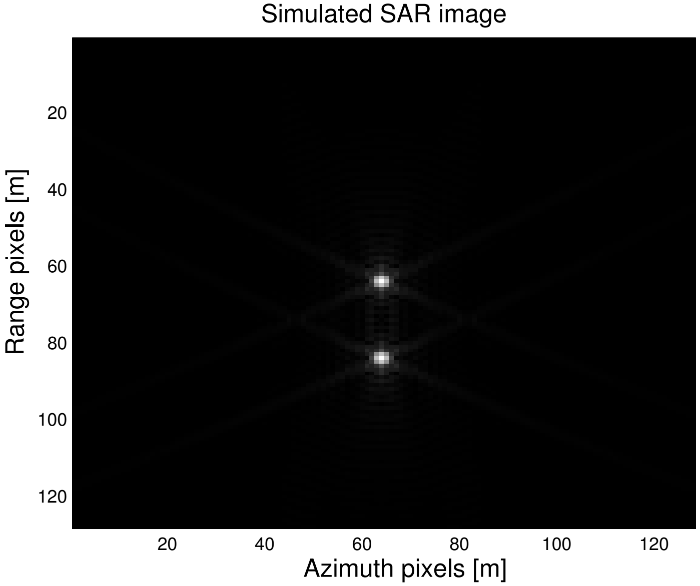
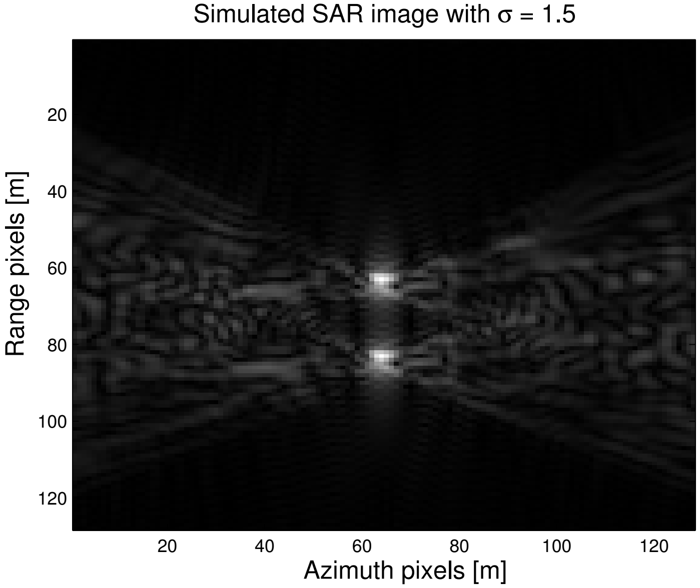


Figure 2: *Back-projection operation schematically described.*



***(a)** Focused SAR image of two point targets.*



***(b)** Unfocused SAR image of two point targets with* $\sigma = 1.5$.

***Figure 3:** Example SAR images without and with perturbed trajectory.*

One approach to simultaneous navigation and auto-focusing is to use complete SAR images to obtain a focus measure as a function of the platform trajectory. This off-line approach is examined in Sjanic and Gustafsson (2010) with good results. Since essentially the same information is contained in the raw SAR measurements (RAR images), these can be used for the same goal, i.e., auto-focus and extracting the platform's trajectory, but in the on-line setup. Raw SAR data contains phase delay information (which is discarded during image formation), and it is that information that is mainly used for auto-focusing. The methods based on phase information are called Phase Gradient (PG) or Phase Difference (PD) methods. An example of these methods can be found in Fienup (1989), where phase corrections are done by estimating the phase error by shear averaging of the raw radar data. Slightly different and expanded approach can be found in Wahl et al. (1994) and Yegulalp (1999) where the phase gradient is estimated after the preprocessing of the complete SAR images. Recent work based on phase gradient auto-focus is found in Xing et al. (2009) where SAR images are focused with means of the motion compensation obtained from the phase gradient estimate and parametric straight line fitting. One drawback of these approaches is that they are all off-line and basically developed for high frequency SAR systems, where some approximations simplify the focusing process, see Section 3 for further discussion.

In this work, an on-line trajectory estimation and auto-focusing method for a low frequency SAR system, CARABAS (Hellsten et al., 1996), is proposed. The main approach is to fuse the phase information from RAR images and inertial measurement system in the sensor fusion framework and obtain good estimate of the platform's trajectory and in turn focused SAR images. The information from the RAR images is essentially the phase gradient while the inertial information is the acceleration measurements. Note that high precision navigation aid, like GPS, is not used in this setup. However, the proposed method makes it possible to include such measurements as well.

2 Sensor Fusion Framework

A standard approach when using sensor fusion framework is to define a discrete time dynamical model of the system whose states are to be estimated,

$$x_{t+1} = f(x_t, w_t) \tag{1a}$$

$$y_t = h(x_t; \theta) + e_t \tag{1b}$$

where x_t are states of the system, w_t is system noise with variance Q_t, e_t is measurement noise with variance W_t, y_t are measurements, $f(\,\cdot\,)$ describes dynamics of the system and $h(\,\cdot\,)$ is the measurement equation that relates measurements and states of the system. θ is used here to explicitly denote the parametrisation of the measurement equation. In most cases, the dynamics (1a) corresponds to 6-DOF aircraft dynamics. In order to present the main ideas of the proposed approach, the much simpler 2-DOF model will be used here,

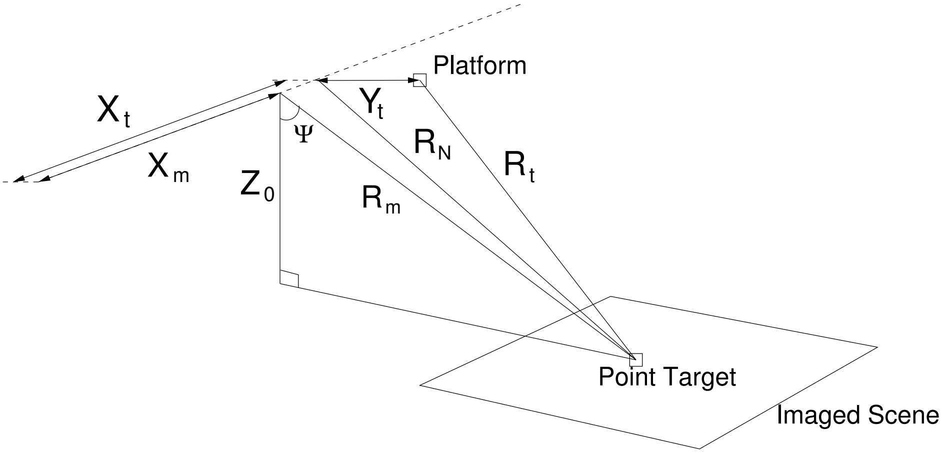


Figure 4: *SAR geometry. The figure is not to scale.*

(Farrell and Barth, 1999),

$$x_{t+1} = Fx_t + Gw_t \tag{2a}$$

$$F = \begin{bmatrix} I_2 & T_sI_2 & \frac{T_s^2}{2}I_2 \\ 0_{2\times 2} & I_2 & T_sI_2 \\ 0_{2\times 2} & 0_{2\times 2} & I_2 \end{bmatrix} \tag{2b}$$

$$G = \begin{bmatrix} \frac{T_s^3}{6}I_2 \\ \frac{T_s^2}{2}I_2 \\ T_sI_2 \end{bmatrix} \tag{2c}$$

$$x_t = [X_t \; Y_t \; v_t^X \; v_t^Y \; a_t^X \; a_t^Y]^T \tag{2d}$$

$$w_t = [w_t^X \; w_t^Y]^T \tag{2e}$$

where T_s is the sampling time, X is the position in azimuth direction and Y is the position in range direction, v^X and v^Y are the velocities in the X- and Y-directions respectively and a^X and a^Y are the accelerations in X- and Y-directions respectively. w^X and w^Y are the unknown forces acting in the azimuth (X) and range direction (Y) respectively. This is just the discretised double integrator dynamics. However, the proposed methods are easily extended to the full 6-DOF model with expense of increased complexity. Note that there are no states for the rotation of the platform because the position and orientation of the global frame is arbitrary and here it is assumed to be aligned with the flight direction.

In order to apply the framework, an equation of the form (1b) is also needed. To obtain such equation, the ideas from PG method can be used.

3 Phase Gradient Auto-focusing Method

3.1 Basics of the PG method

The basis for this approach is the fact that the phase delay of the radar echo data is proportional to the range to the imaged scene, which will vary hyperbolically as a function of time, see Figure 4,

$$\varphi_t = -\frac{4\pi}{\lambda} R_t \tag{3}$$

where λ is the wavelength of the radar carrier. It is seen that phase delay and range are proportionally related to each other. That means that phase and range can be used equivalently, it is only the factor $-4\pi/\lambda$ that differs. We can now calculate the time derivative of the phase delay and obtain

$$\dot{\varphi}_t = -\frac{4\pi}{\lambda} \dot{R}_t \tag{4}$$

The range derivative, $\dot{R}_t$, can be calculated by taking the time derivative of the range R_t that can be expressed as a function of the states, x_t, according to the setup in Figure 4. From the figure it can be seen that the range R_t can, with help from cosine theorem, be expressed as

$$R_t = \sqrt{R_N^2 + Y_t^2 - 2R_N \sin(\Psi) Y_t} \tag{5a}$$

$$R_N = \sqrt{R_m^2 + (X_m - X_t)^2} \tag{5b}$$

By using the chain rule we obtain

$$\dot{R}_t = \frac{-(X_m - X_t)v_t^X + Y_t v_t^Y - R_N \sin(\Psi) v_t^Y}{R_t} + \frac{Y_t \sin(\Psi)(X_m - X_t)v_t^X}{R_N R_t} \tag{6}$$

In the SAR applications where radar frequency is high or antenna size is large, the lobe is narrow and an approximate expression for the range and its gradient can be used without much loss of accuracy. The source of the approximation in this case is the fact that the synthetic aperture length, $2X_m$, and deviation Y_t are much shorter than the range to the middle of the scene, R_m. The range can then be approximated with the Taylor expansion as

$$R_t \approx R_m + \frac{(X_m - X_t)^2 + Y_t^2}{2R_m} - \sin(\Psi) Y_t \tag{7}$$

where all the terms with R_m^2 in the denominator are neglected. Now the range gradient can be calculated from this approximate expression. By using $X_t = v_0^X t$ and $Y_t = v_0^Y t + 0.5\, a_0^Y t^2$, i.e., the acceleration a_t^X is zero and velocity in X-direction and acceleration in Y-direction are constant (reasonable assumptions for short aperture times) the gradient becomes

$$\dot{R}_t \approx \frac{-(X_m - v_0^X t)v_0^X + (v_0^Y t + 0.5 a_0^Y t^2)(v_0^Y + a_0^Y t)}{R_m} - \sin(\Psi)(v_0^Y + a_0^Y t) \approx \tag{8a}$$

$$\approx -\frac{X_m v_0^X}{R_m} - \sin(\Psi) v_0^Y + \left(\frac{(v_0^X)^2 + (v_0^Y)^2}{R_m} - \sin(\Psi) a_0^Y \right) t \tag{8b}$$

where terms in t^2 and t^3 has been neglected in (8b). The PG methods try to estimate the slope and constant term of this affine function in t from the raw data in order to compensate for the phase delay error caused by the platform's unknown motion, as in e.g., Oliver and Quegan (2004) or Xing et al. (2009). This compensation is then applied during image formation in order to focus the image. For the time domain image formation approach, this is equivalent to estimating the unknown motion. The assumption made above about narrow radar lobe is not applicable to the SAR systems which operate with low frequency. This will imply that range gradient cannot be approximated with the linear function except in the narrow band around zero phase delay. Due to this, the method of fitting a linear function in order to compensate for the phase delay error described above will not work satisfactory for low frequency SAR. That is why the exact expression for the range is used here.

If we look at the equation (6) we see that the right hand side consists of states of the dynamical model, i.e., positions, velocities and some (known) constants. The left hand side of the equation is the entity that can be estimated from the SAR data (either raw or partially processed) which will be explained in the next subsection. This implies that we have a measurement equation from the standard sensor fusion framework in the form (7a) with $\theta = [R_m \; X_m \; \Psi]^T$, and some standard sensor fusion methods, such as the Extended Kalman Filter (EKF), can be applied.

3.2 Estimating the Phase/Range Gradient

In order to create measurements, y_t, in (7a), we need an estimate of the phase delay gradient. The phase delay gradient estimation kernel, proposed in Fienup (1989), is (superindex m stands for measured)

$$\dot{\varphi}_t^m = \frac{1}{T_s} \arg \left\{ \sum_R I_{\text{RAR}}(R,t) I_{\text{RAR}}^*(R, t - T_s) \right\} \tag{9}$$

where $I_{\text{RAR}}(R,t)$ is the complex RAR image, i.e., raw radar data, see Figure 1, I^* denotes the complex conjugate, R is the range dimension, t is the azimuth (or time) dimension and T_s is the time between radar pulses which is the inverse of the pulse repetition frequency (here used as a sampling time). The motivation for this approach is the fact that the complex raw radar data, $I_{\text{RAR}}(R,t)$, can be written in polar coordinates as

$$I_{\text{RAR}}(R,t) = |I_{\text{RAR}}(R,t)| e^{i\varphi_t} \tag{10}$$

where φ_t is defined in (3), see even Oliver and Quegan (2004). The estimate (9) can be then interpreted as the first order difference approximation of the derivative $\dot{\varphi}_t$. This estimate can be calculated in a sequential manner, pulse by pulse and be interpreted as a measurement y_t for each time instance and used in the sensor fusion framework. This estimation kernel is developed for the high frequency SAR systems where phase gradient

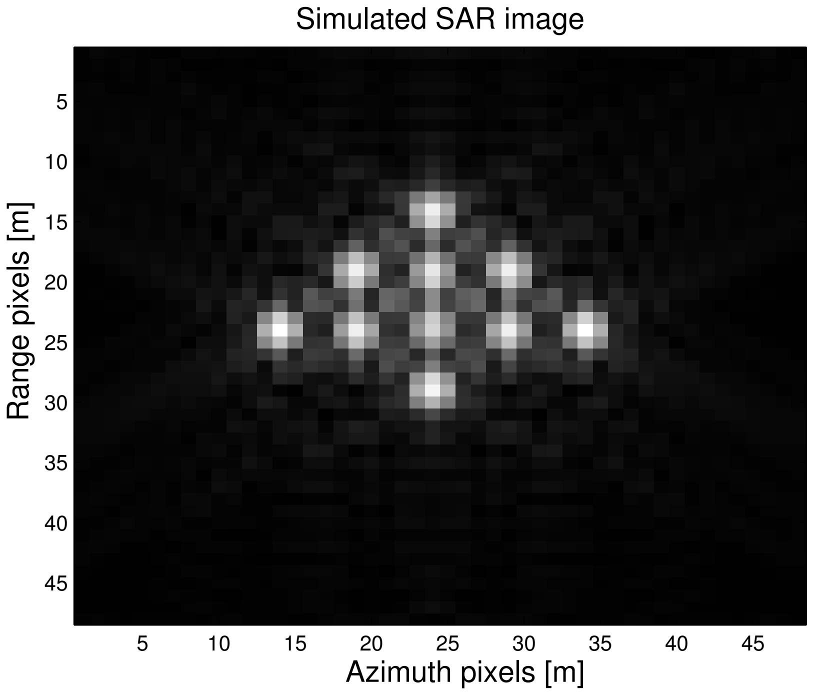
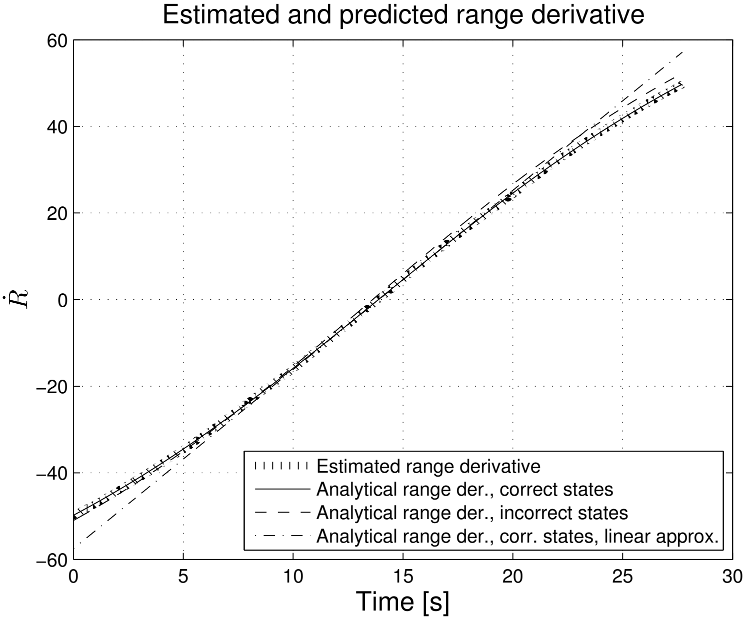


Figure 5: *Simulated SAR image of the scene with 10 point targets.*

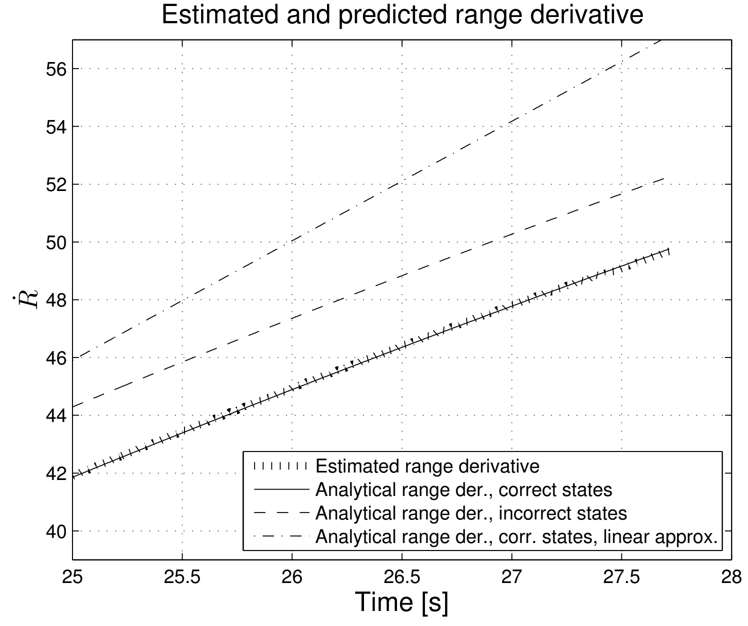
can be approximated with a linear function. Since this is not the case in the low frequency SAR system some other method must be applied. If seen in the context of back-projection and sub image processing described in Section 1, if each radar echo is used to generate a sub image, the complete SAR image is then simply the sum of all sub images. Each sub image is created with some assumption about range from the platform to the scene, either correct or perturbed, and the raw data contain information about the correct range. This in turn means that each (complex valued) pixel in the back-projected sub image will contain information about the range between the platform and the scene. Nevertheless, due to the phase wrapping effect in the complex number arithmetic, the absolute value of the range cannot be obtained. However, it is possible to estimate the range derivative from the consecutive sub images in a manner similar to (9) in the following way

$$\dot{\varphi}_t^m = \frac{1}{T_s M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \arg\left\{ \tilde{I}_{ij}(t) \tilde{I}_{ij}^*(t - T_s) \right\} \tag{11}$$

where $\tilde{I}_{ij}(t)$ is the complex $M \times M$ sub image generated from radar echo t, see Figure 2, and T_s is the time between radar pulses as before. This is basically an estimate of the average range to the centre of the imaged scene. The sub image is created with an assumption of straight and constant velocity trajectory with nominal state values. Note that it is only raw data used for the image generation that is summed here, not the whole range dimen-



(a) Range derivative, the whole trajectory.



(b) Range derivative, only the last few seconds of the trajectory.

Figure 6: *Range derivative estimated from data (dotted line) and calculated with the analytical expression (solid, dashed and dot-dashed lines). Solid line curve is generated with correct trajectory, while dashed is not. Dot-dashed line is generated with approximated expression in* (8b).

sion as in (9). This is important, if the image to be created is not the same size as the raw data. Range gradient estimated with (11) for the image in Figure 5 is illustrated in Figure 6 (plotted with dotted line) together with analytically calculated range gradients based on (6) (plotted with solid and dashed lines). The one plotted with solid line is based on the correct trajectory, i.e., data are collected with the same trajectory used for the analytical expression. For the range gradient plotted with dashed line another trajectory, different from the one used for collecting data, is used. As a comparison, range gradient obtained with the approximated expression (8b) is plotted with dot-dashed line. It is clearly seen that despite correct states are used, the values of this analytical expression deviate from the estimated range gradient in the beginning and the end of the trajectory. This further motivates use of the complete expression for the range gradient, (6). All this gives that the estimate of the range gradient (11) can be interpreted as a measurement y_t and together with (6) as $h(x_t;\theta)$ a standard EKF can be applied.

4 EKF Auto-focusing and Evaluation of the Performance

4.1 Extended Kalman Filter

Given the dynamical and the measurement models of the system as in (1), the EKF is defined by the following recursive steps, (Kailath et al., 2000; Gustafsson, 2010),

$$\hat{x}_{t+1|t} = f(\hat{x}_{t|t}, 0) \tag{12a}$$

$$P_{t+1|t} = F_t P_{t|t} F_t^T + G_t Q_t G_t^T \tag{12b}$$

$$F_t = \frac{\partial}{\partial x} f(x, w)\Big|_{x=\hat{x}_{t|t},\ w=0} \tag{12c}$$

$$G_t = \frac{\partial}{\partial w} f(x, w)\Big|_{x=\hat{x}_{t|t},\ w=0} \tag{12d}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - h(\hat{x}_{t|t-1})) \tag{12e}$$

$$P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1} \tag{12f}$$

$$H_t = \frac{\partial}{\partial x} h(x)\Big|_{x=\hat{x}_{t|t-1}} \tag{12g}$$

$$K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1} \tag{12h}$$

Here $\hat{x}_{t|t}$ is the estimate of the states in the time t given all the measurements up to the time t and $\hat{x}_{t|t-1}$ is the estimate of the states at time t given all the measurements up to the time $t-1$. $P_{t|t}$ and $P_{t|t-1}$ are their respective covariances. Q_t and R_t are considered tuning parameters.

4.2 Measurement Model

The measurements are the range gradient $\dot{R}_t^m$ obtained from (11) and scaled with factor $-4\pi/\lambda$, and the accelerations in X- and Y-direction. The measurement equation becomes

$$\underbrace{\begin{bmatrix} \tilde{a}_t^X \\ \tilde{a}_t^Y \\ \dot{R}_t^m \end{bmatrix}}_{y_t} = \underbrace{\begin{bmatrix} a_t^X \\ a_t^Y \\ \dot{R}_t \end{bmatrix}}_{h(x_t;\theta)} + \underbrace{\begin{bmatrix} e_t^{a^X} \\ e_t^{a^Y} \\ e_t^{\dot{R}} \end{bmatrix}}_{e_t} \tag{13}$$

where $\tilde{a}_t$ is the measured acceleration, $\dot{R}_t$ is defined in (6) and e_t is white Gaussian noise with covariance matrix $W = \text{diag}\{W_a,\ W_a,\ W_{\dot{R}}\}$. In order to use EKF, the Jacobian of the measurement equation with respect to the states, $\partial h/\partial x$ is needed. For the measurement equation above, the Jacobian is (omitting time index for readability)

$$H = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial h_1}{\partial X} & \frac{\partial h_1}{\partial Y} & \frac{\partial h_1}{\partial v^X} & \frac{\partial h_1}{\partial v^Y} & \frac{\partial h_1}{\partial a^X} & \frac{\partial h_1}{\partial a^Y} \\ \frac{\partial h_2}{\partial X} & \frac{\partial h_2}{\partial Y} & \frac{\partial h_2}{\partial v^X} & \frac{\partial h_2}{\partial v^Y} & \frac{\partial h_2}{\partial a^X} & \frac{\partial h_2}{\partial a^Y} \\ \frac{\partial h_3}{\partial X} & \frac{\partial h_3}{\partial Y} & \frac{\partial h_3}{\partial v^X} & \frac{\partial h_3}{\partial v^Y} & \frac{\partial h_3}{\partial a^X} & \frac{\partial h_3}{\partial a^Y} \end{bmatrix} \tag{14}$$

and the nonzero elements are (with $\sin(\Psi) = s_\Psi$)

$$\frac{\partial h_1}{\partial a^X} = 1 \tag{15a}$$

$$\frac{\partial h_2}{\partial a^Y} = 1 \tag{15b}$$

$$\frac{\partial h_3}{\partial X} = \frac{v^X + \frac{(X_m - X)v^Y s_\Psi}{R_N} - \frac{Y v^X s_\Psi}{R_N} + \frac{(X_m - X)^2 Y v^X s_\Psi}{R_N^3}}{R} - \frac{L(X_m - X)\left(1 - \frac{Y s_\Psi}{R_N}\right)}{R^3} \tag{15c}$$

$$\frac{\partial h_3}{\partial Y} = \frac{v^Y + \frac{(X_m - X)v^X s_\Psi}{R_N}}{R} + \frac{L\left(Y - R_N s_\Psi\right)}{R^3} \tag{15d}$$

$$\frac{\partial h_3}{\partial v^X} = -\frac{X_m - X - \frac{(X_m - X)Y s_\Psi}{R_N}}{R} \tag{15e}$$

$$\frac{\partial h_3}{\partial v^Y} = \frac{Y - R_N s_\Psi}{R} \tag{15f}$$

$$L = (X_m - X)v^X - Y v^Y + v^Y R_N s_\Psi - \frac{(X_m - X)v^X Y s_\Psi}{R_N} \tag{15g}$$

4.3 Numerical Results

Using the dynamic model (2) and the measurement model (13), an EKF has been applied to two cases, one where range gradient measurement has not been used and one where it has. In order to simulate somewhat more realistic acceleration measurements, a bias of $5 \cdot 10^{-3}$ m/s^2 is added to the X-direction and $-5 \cdot 10^{-3}$ m/s^2 to the Y-direction in addition to the i.i.d. Gaussian noise, $e_t^a \sim \mathcal{N}(0, 3.6 \cdot 10^{-3})$. The bias and measurement noise values are chosen as representative for the inertial sensors in the interesting performance class. The performance is then assessed in terms of Root Mean Square Error (RMSE) of the trajectory and mean of the error SAR image power. The RMSE is defined as

$$\text{RMSE}(\hat{x}_t) = \sqrt{\frac{\sum_{k=1}^{N}(\hat{x}_t^k - x_t)^2}{N}} \tag{16}$$

where $\hat{x}_t = [\hat{x}_t^1, \ldots, \hat{x}_t^N]^T$, are unbiased estimates of the scalar parameter x_t (which is a function of time). The error SAR image power is defined as

$$E_{\hat{I}} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}|\hat{I}_{ij} - I_{ij}|^2}{MN} \tag{17}$$

where $\hat{I}$ is the $M \times N$ complex SAR image obtained with the estimation procedure and I is the perfect focused SAR image, i.e., created with the true trajectory.

The RMSE for the position and velocity based on the 30 Monte Carlo EKF runs as a function of time for cases with and without range gradient measurement is depicted in Figure 7a and Figure 7b. The mean of the SAR error image power obtained from these 30 Monte Carlo runs is 384.1 for the case with range gradient included as a measurement and 2690 for the case with only inertial measurements. The resulting images created with one of the 30 estimated trajectories are depicted in Figure 8. The error image power is 2620 for the image in Figure 8a and 323 for the image in Figure 8b. These images clearly show that addition of the range gradient measurement improves the image focus and the estimate of the navigation states, especially compared to the pure inertial estimates.

As a comparison, an approximate expression of the range derivative (8a) is used as $h(x_t; \theta)$ in a measurement equation and EKF is run. In order to obtain the measurement equation as a function of the states, $X_t = v_0^X t$, $Y_t = v_0^Y t + 0.5 a_0^Y t^2$ and $v_t^Y = v_0^Y + a_0^Y t$ are used giving the nonzero elements of the Jacobian (14) as

$$\frac{\partial h_1}{\partial a_t^X} = 1 \tag{18a}$$

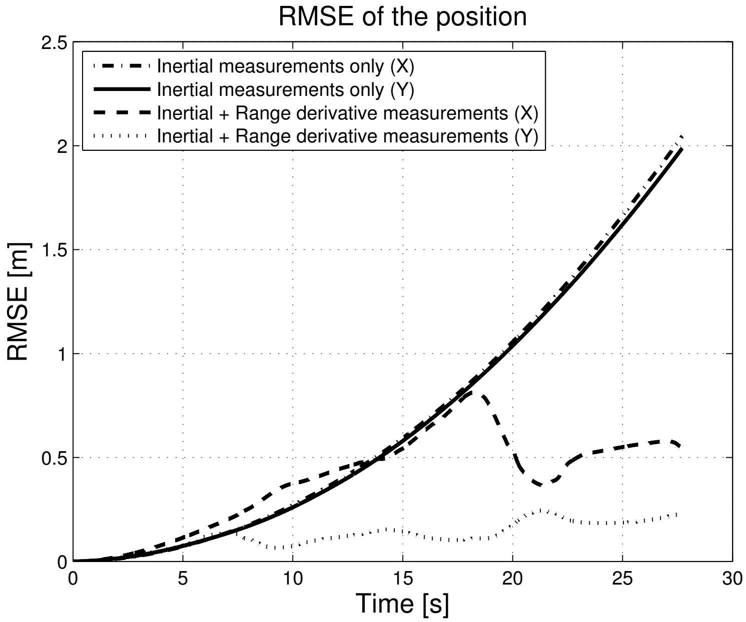
$$\frac{\partial h_2}{\partial a_t^Y} = 1 \tag{18b}$$

$$\frac{\partial h_3}{\partial X_t} = \frac{v_0^X}{R_m} \tag{18c}$$

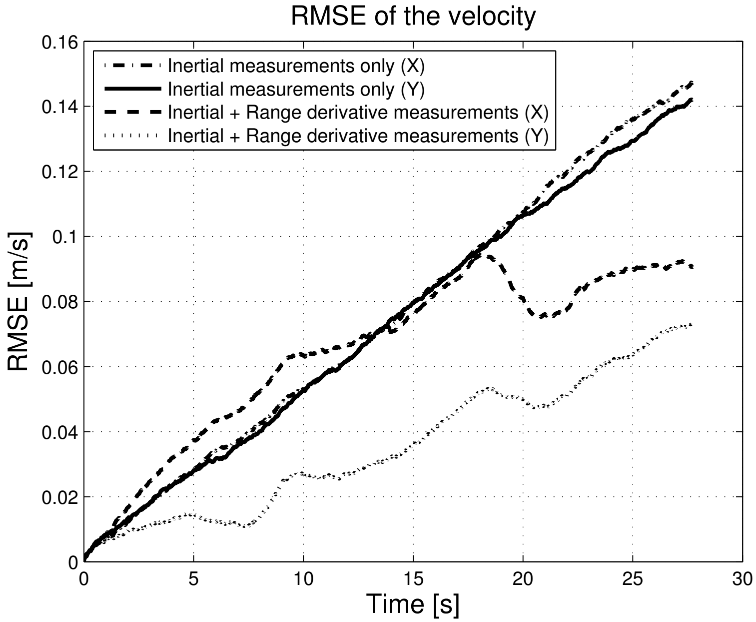
$$\frac{\partial h_3}{\partial Y_t} = \frac{v_t^Y}{R_m} \tag{18d}$$

$$\frac{\partial h_3}{\partial v_t^Y} = \frac{Y_t}{R_m} - \sin(\Psi) \tag{18e}$$

The results from this simulation are presented in Figure 9. It can be seen that estimate with exact expression has much better RMSE value than the one with approximate expression which is even worse than pure inertial estimate. Here, the tuning of the measurement noise, R_t, was changed to eight times higher value in order to get somewhat comparable values (same tuning produces RMSE values in order of 20 m). These results clearly demonstrate

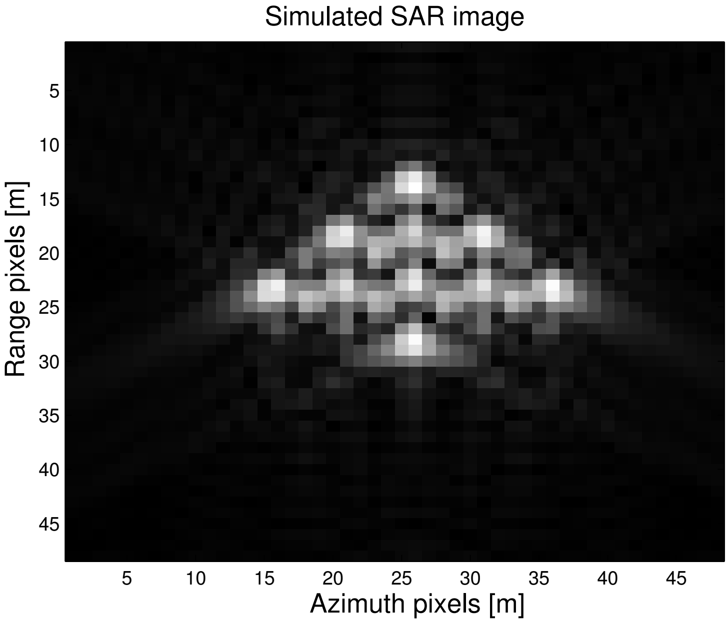


*(**a**) RMSE for the EKF estimated position for the example scene.*

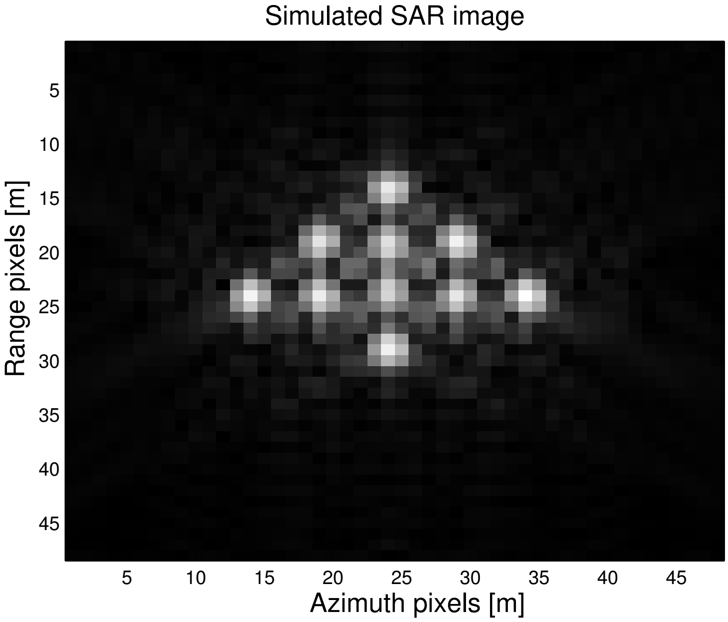


*(**b**) RMSE for the estimated velocity for the example scene.*

Figure 7: *RMSE for the estimated position and velocity for the example scene. X is the azimuth (or along track) dimension and Y is the range (or cross track) dimension.*



*(**a**) Image of the example scene created with trajectory from the pure inertial estimate.*



*(**b**) Image of the example scene created with trajectory from the estimate with range gradient as additional measurement.*

Figure 8: *Images of the example scene created with estimated trajectories with and without range gradient measurement.*

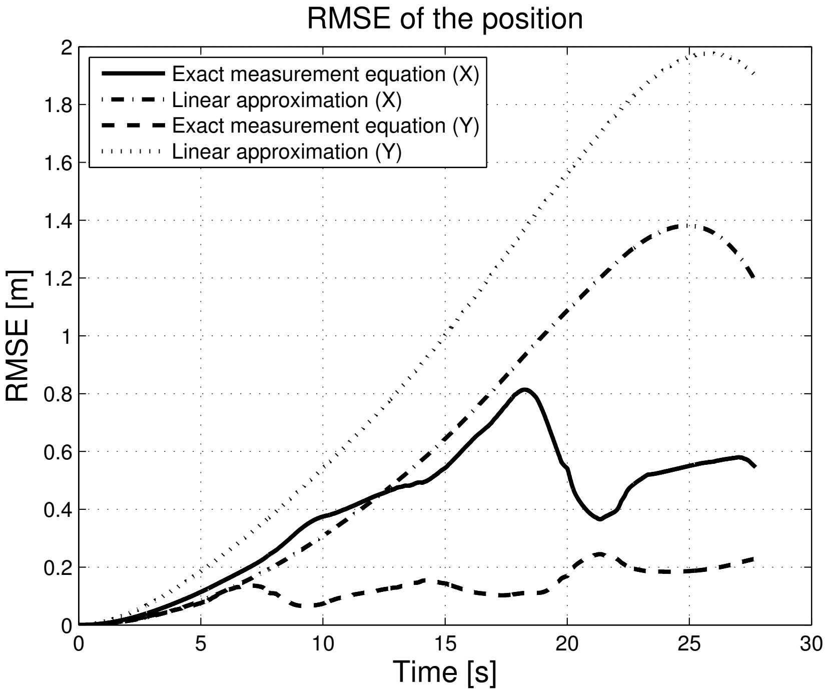


Figure 9: *RMSE for the estimated position with approximate and exact range derivative expression. X is the azimuth (or along track) dimension and Y is the range (or cross track) dimension.*

the importance of the exact expression for the range derivative.

5 Conclusions and Future Work

Here, a method of simultaneous navigation and low frequency SAR auto-focusing based on a sensor fusion framework is presented. The basis for the approach is the gradient of the range between flying platform and the scene which is used as a measurement in a nonlinear filtering setup. It has also been shown that it is important to use the exact expression for the range gradient instead of the linear approximation common in the high frequency SAR where the lobe is narrow. The average performance in terms of the trajectory RMSE and the focused images is significantly improved compared to the ones based on pure dead-reckoning, without high precision navigation aids like GPS. The RMSE of the position has been largely reduced, as well as of the velocity. The image created with the range gradient supported estimate is hardly distinguishable from the correct image, see Figure 8b and Figure 5.

The evaluation of the proposed methods is, at the moment, only done with the simulated SAR and inertial data. In this way, the ground truth is readily available and the evaluation of the performance easily done. However, in order to fully assess the method, the real SAR and inertial data shall be used, and this is the next step in this work.

Bibliography

L. E. Andersson. On the Determination of a Function from Spherical Averages. *SIAM Journal on Mathematical Analysis*, 19(1):214–232, 1988. doi: 10.1137/0519016. URL `http://link.aip.org/link/?SJM/19/214/1`.

C. Cafforio, C. Prati, and F. Rocca. SAR data focusing using seismic migration techniques. *IEEE Transactions on Aerospace and Electronic Systems*, 27(2):194 –207, March 1991. ISSN 0018-9251. doi: 10.1109/7.78293.

L. J. Cutrona, W. E. Vivian, E. N. Leith, and G. O. Hall. A high-resolution radar combat-surveillance system. *IRE Transactions on Military Electronics*, MIL-5(2):127–131, April 1961. ISSN 0096-2511. doi: 10.1109/IRET-MIL.1961.5008330.

J. Farrell and M. Barth. *The global positioning system and inertial navigation.* McGraw-Hill Professional, 1999.

J. A. Fawcett. Inversion of N-Dimensional Spherical Averages. *SIAM Journal on Applied Mathematics*, 45(2):336–341, 1985. ISSN 00361399. URL `http://www.jstor.org/stable/2101820`.

J. R. Fienup. Phase Error Correction by Shear Averaging. In *Signal Recovery and Synthesis*, pages 134–137. Optical Society of America, June 1989.

F. Gustafsson. *Statistical Sensor Fusion.* Studentliteratur, Lund, 2010.

H. Hellsten and L. E. Andersson. An inverse method for the processing of synthetic aperture radar data. *Inverse Problems*, 3(1):111, 1987. URL `http://stacks.iop.org/0266-5611/3/i=1/a=013`.

H. Hellsten, L. M. Ulander, A. Gustavsson, and B. Larsson. Development of VHF CARABAS II SAR. In *Proceedings of Radar Sensor Technology*, volume 2747, pages 48–60. SPIE, April 1996. doi: 10.1117/12.243087. URL `http://link.aip.org/link/?PSI/2747/48/1`.

T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation.* Prentice-Hall, Upper Saddle River, New Jersey, 2000.

A. S. Milman. SAR Imaging by Omega-K Migration. *International Journal of Remote Sensing*, 14(10):1965–1979, 1993.

R. L. Jr. Morrison and D. C. Jr. Munson. An experimental study of a new entropy-based SAR autofocus technique. In *Proceedings of International Conference on Image Processing, ICIP 2002*, volume 2, pages II–441–4, September 2002. doi: 10.1109/ICIP.2002.1039982.

F. Natterer. *The Mathematics of Computerised Tomography.* New York: Wiley, 1986.

C. Oliver and S. Quegan. *Understanding Synthetic Aperture Radar Images.* The SciTech Radar and Defense Series. SciTech, 2004. ISBN ISBN 1-891121-31-6.

F. Rocca. Synthetic Aperture Radar: a New Application for Wave Equation Techniques. *Stanford Exploration Project SEP-56*, pages 167–189, 1987. URL `http://sepwww.stanford.edu/oldreports/sep56/56_13.pdf`.

Z. Sjanic and F. Gustafsson. Simultaneous Navigation and SAR Auto-focusing. In *Proceedings of 13th International Conference on Information Fusion*, Edinburgh, UK, July 2010.

Z. Sjanic and F. Gustafsson. Navigation and SAR Auto-focusing Based on the Phase Gradient Approach. In *Proceedings of 14th International Conference on Information Fusion*, Chicago, USA, July 2011.

D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and C. V. Jr. Jakowatz. Phase gradient autofocus - a robust tool for high resolution SAR phase correction. *IEEE Transactions on Aerospace and Electronic Systems*, 30(3):827–835, July 1994. ISSN 0018-9251. doi: 10.1109/7.303752.

L. Xi, L. Guosui, and J. Ni. Autofocusing of ISAR images based on entropy minimization. *IEEE Transactions on Aerospace and Electronic Systems*, 35(4):1240–1252, October 1999. ISSN 0018-9251. doi: 10.1109/7.805442.

M. Xing, R. Jiang, X.and Wu, F. Zhou, and Z. Bao. Motion Compensation for UAV SAR Based on Raw Radar Data. *IEEE Transactions on Geoscience and Remote Sensing*, 47 (8):2870–2883, August 2009. ISSN 0196-2892. doi: 10.1109/TGRS.2009.2015657.

A. F. Yegulalp. Minimum entropy SAR autofocus. In *7th Adaptive Sensor Array Processing Workshop*, March 1999.

Paper D

A Nonlinear Least-Squares Approach to the SLAM Problem

Authors: Zoran Sjanic, Martin A. Skoglund, Thomas B. Schön and Fredrik Gustafsson

Edited version of the paper:

Z. Sjanic, M. A. Skoglund, T. B. Schön, and F. Gustafsson. A Nonlinear Least-Squares Approach to the SLAM Problem. In *Proceedings of 18th IFAC World Congress*, Milano, Italy, August/Septemeber 2011.

A Nonlinear Least-Squares Approach to the SLAM Problem

Zoran Sjanic, Martin A. Skoglund, Thomas B. Schön and Fredrik Gustafsson

Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
`{zoran,ms,schon,fredrik}@isy.liu.se`

Abstract

In this paper we present a solution to the simultaneous localisation and mapping (SLAM) problem using a camera and inertial sensors. Our approach is based on the maximum a posteriori (MAP) estimate of the complete SLAM problem. The resulting problem is posed in a nonlinear least-squares framework which we solve with the Gauss-Newton method. The proposed algorithm is evaluated on experimental data using a sensor platform mounted on an industrial robot. In this way, accurate ground truth is available, and the results are encouraging.

1 Introduction

In this paper we present an optimisation based solution to the simultaneous localisation and mapping (SLAM) problem formulated as nonlinear least-squares, and solved with the Gauss-Newton method. The method aims at providing high quality SLAM estimates which could e.g., be used as priors for computing detailed terrain maps.

SLAM is the problem of estimating a map of the surrounding environment from a moving platform, while simultaneously localising the platform. These estimation problems usually involve nonlinear dynamics and nonlinear measurements of a high dimensional state space.

In Dellaert and Kaess (2006) a nonlinear least-squares approach to SLAM, called square root Smoothing and Mapping ($\sqrt{\text{SAM}}$) is presented. We extend this approach by considering a full 6 DOF platform, 3 DOF landmarks, inputs using inertial sensors and camera measurements. The resulting algorithm is evaluated on experimental data from a structured indoor environment and compared with ground truth data.

For more than twenty years SLAM has been a popular field of research and is considered an important enabler for autonomous robotics. An excellent introduction to SLAM is given in the two part tutorial by Durrant-Whyte and Bailey (2006); Bailey and Durrant-Whyte (2006) and for a thorough overview of visual SLAM Chli (2009) is highly recommended. In the seminal work of Smith et al. (1990) the idea of a stochastic map was presented and was first used in Moutarlier and Chatila (1989), where the estimate is computed with an Extended Kalman Filter (EKF). There are by now quite a few examples of successful EKF SLAM implementations, see e.g., Guviant and Nebot (2001); Leonard et al. (2000). Another popular approach is the FastSLAM method (Montemerlo et al., 2002, 2003) which uses particle filters. These are known to handle nonlinearities very well. Both EKF SLAM and FastSLAM suffer from inconsistencies due to poor data association, linearisation errors (Bailey et al., 2006a) and particle depletion (Bailey et al., 2006b).

Some impressive work where the SLAM problem is solved solely with cameras can be found in Davison et al. (2007); Davison (2003); Eade (2008); Klein and Murray (2007). The camera only SLAM methods have many similarities with bundle adjustment techniques, (Hartley and Zisserman, 2004; Triggs et al., 2000), and the stochastic map estimation problem can be seen as performing structure from motion estimation (Fitzgibbon and Zisserman, 1998; Taylor et al., 1991). Without any other sensors measuring the platform dynamics, the image frame rate and the visual information contents in the environment are limiting factors for the ego motion estimation, and hence the map quality.

Recent years' increase in computational power has made smoothing an attractive option to filtering. One of the first SLAM related publications, where the trajectory is not filtered out to a single estimate is Eustice et al. (2006), where the whole time history is estimated with a so called *delayed state information filter*. Other, more optimisation like approaches are Dellaert and Kaess (2006); Kaess et al. (2008); Bibby and Reid (2007); Bryson et al. (2009), which all optimise over the whole trajectory and a feature based map.

2 Problem Formulation

We assume that the dynamic model and the measurements are on the following form

$$x_t = f(x_{t-1}, u_t) + \underbrace{B_w w_t}_{\tilde{w}_t}, \tag{1a}$$

$$m_t = m_{t-1}, \tag{1b}$$

$$y_{t_k} = h(x_{t_k}, m_{t_k}) + e_{t_k}, \tag{1c}$$

where x_t and m_t are vehicle and landmark states, respectively, and the inertial measurements can be modelled as inputs u_t. The meaning of y_{t_k} is a measurement relative to landmark m_{t_k} at time t_k, and this is because the measurements and the dynamic model deliver data in different rates. If we assume that all the measurements and the inputs for $t = \{0 : N\}$ and $k = \{1 : K\}$ $(K \ll N)$ are available and the noise is independent and identically distributed (i.i.d.), then the joint probability density of (1) is

$$p(x_{0:N},\, m_N\, |y_{1:K},\, u_{1:N}) = \\ p(x_0) \prod_{t=1}^{N} p_{\tilde{w}_t}(x_t|x_{t-1}, u_t) \prod_{k=1}^{K} p_{e_{t_k}}(y_{t_k}|x_{t_k}, m_{t_k}). \tag{2}$$

Note that the map, m_N, is static and the estimate is given for the last time step only.

Furthermore, the initial platform state x_0 is fixed to the origin without uncertainty. This is a standard SLAM approach and x_0 is therefore treated as a constant. The smoothed maximum a posteriori (MAP) estimate of $x_{0:N}$ and m_N is then

$$\begin{aligned}[][x^*_{0:N},\, m^*_N] = & \operatorname*{arg\,max}_{x_{0:N},\, m_N} p(x_{0:N},\, m_N\, |y_{t_1:t_K},\, u_{1:N}) = \\ & \operatorname*{arg\,min}_{x_{0:N},\, m_N} -\log p(x_{0:N},\, m_N\, |y_{t_1:t_K},\, u_{1:N}). \end{aligned} \tag{3}$$

If the noise terms $\tilde{w}_t$ and e_{t_k} are assumed to be Gaussian and white, i.e., $e_{t_k} \sim \mathcal{N}(0, R_{t_k})$ and $\tilde{w}_t \sim \mathcal{N}(0, \widetilde{Q}_t)$, (3) then becomes

$$[x^*_{0:N},\, m^*_N] = \operatorname*{arg\,min}_{x_{0:N},\, m_N} \sum_{t=1}^{N} \|x_t - f(x_{t-1}, u_t)\|^2_{\widetilde{Q}_t^{-1}} + \sum_{k=1}^{K} \|y_{t_k} - h(x_{t_k}, m_{t_k})\|^2_{R^{-1}_{t_k}}, \tag{4}$$

which is a nonlinear least-squares formulation.

3 Models

Before we introduce the details of the dynamic model some coordinate frame definitions are necessary:

- Body coordinate frame (b), moving with the sensor and with origin fixed in the IMU's inertial centre.
- Camera coordinate frame (c), moving with the sensor and with origin fixed in the camera's optical centre.
- Earth coordinate frame (e), fixed in the world with its origin arbitrary positioned.

When the coordinate frame is omitted from the states it is assumed that they are expressed in the earth frame e.

3.1 Dynamics

The dynamic model used in this application has 10 states consisting of the position and velocity of the b frame expressed in the e frame, $p^e = [X^e\ Y^e\ Z^e]^T$ and $v^e = [v_x\ v_y\ v_z]^T$, respectively. The orientation is described using a unit quaternion $q^{be} = [q_0\ q_1\ q_2\ q_3]^T$ defining the orientation of the b frame expressed in the e frame. The IMU measurements are treated as inputs, reducing the state dimension needed, and we denote the specific force $u^b_a = [a^b_x\ a^b_y\ a^b_z]^T$ and denote the angular rate $u^b_\omega = [\omega^b_x\ \omega^b_y\ \omega^b_z]^T$. The dynamics of the sensor in (1a) is then

$$\underbrace{\begin{bmatrix} p_t^e \\ v_t^e \\ q_t^{be} \end{bmatrix}}_{x_t} = \underbrace{\begin{bmatrix} I_3 & T_sI_3 & 0 \\ 0 & I_3 & 0 \\ 0 & 0 & I_4 \end{bmatrix} \begin{bmatrix} p_{t-1}^e \\ v_{t-1}^e \\ q_{t-1}^{be} \end{bmatrix} + \begin{bmatrix} \frac{T_s^2}{2}I_3 & 0 \\ T_sI_3 & 0 \\ 0 & \frac{T_s}{2} \end{bmatrix} \begin{bmatrix} \mathsf{R}(q_{t-1}^{be})^T u_{a,t}^b + g^e \\ S(u_{\omega,t}^b)q_{t-1}^{be} \end{bmatrix}}_{f(x_{t-1},u_t)} +$$

$$\underbrace{\begin{bmatrix} \frac{T_s^2}{2}I_3 & 0 \\ T_sI_3 & 0 \\ 0 & \frac{T_s}{2}\widetilde{S}(q_{t-1}^{be}) \end{bmatrix}}_{B_w(x_{t-1})} \underbrace{\begin{bmatrix} w_{a,t}^b \\ w_{\omega,t}^e \end{bmatrix}}_{w_t}, \tag{5}$$

where

$$w_{a,t}^b \sim \mathcal{N}(0, Q_a),\ Q_a = \sigma_a I_3, \tag{6a}$$

$$w_{w,t}^e \sim \mathcal{N}(0, Q_w),\ Q_w = \sigma_w I_3, \tag{6b}$$

$$S(u_{\omega,t}^b) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix},\ \widetilde{S}(q_t^{be}) = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}, \tag{6c}$$

and $\mathsf{R}(q_t^{be}) \in SO(3)$ is the rotation matrix parametrised using the unit quaternion and $\mathsf{R}(q_t^{be})^T u_{a,t}^b + g^e$ is the specific force input expressed in the e frame. Vector $g^e = [0\,0\,g]^T$, where $g \approx -9.82$, compensates for the earth gravitational field.

3.2 Landmark State Parametrisation

Landmark states are encoded in the Inverse Depth Parametrisation (IDP) (Civera et al., 2008). The first three states, X^e, Y^e and Z^e, represent the 3D position of the camera when the landmark was first observed. The last three states describe a vector to the landmark in spherical coordinates parametrised with azimuthal angle φ^e, elevation angle θ^e and inverse distance ρ^e, giving $m^e = [x\ y\ z\ \theta\ \phi\ \rho]^T$. The angles φ^e, θ^e and the inverse distance ρ^e are expressed in the right handed earth coordinate frame e with Z^e-axis pointing upwards. This means that a landmark m, with earth fixed coordinates $[X_m^e\ Y_m^e\ Z_m^e]^T$ is parametrised as

$$\begin{bmatrix} X_m^e \\ Y_m^e \\ Z_m^e \end{bmatrix} = \begin{bmatrix} X^e \\ Y^e \\ Z^e \end{bmatrix} + \frac{1}{\rho^e}\gamma(\varphi^e, \theta^e), \tag{7a}$$

$$\gamma(\varphi^e, \theta^e) = \begin{bmatrix} \cos\varphi^e \sin\theta^e \\ \sin\varphi^e \sin\theta^e \\ \cos\theta^e \end{bmatrix}. \tag{7b}$$

Since the camera is calibrated, as in Zhang (2000) using the toolbox (Bouguet, 2010), the landmark states can be introduced using normalised pixel coordinates $[u\ v]^T$ according to

$$p^e = \begin{bmatrix} X^e \\ Y^e \\ Z^e \end{bmatrix}, \tag{8a}$$

$$l^e = \begin{bmatrix} l_x^e \\ l_y^e \\ l_z^e \end{bmatrix} = \mathsf{R}(q_t^{be})^T\, \mathsf{R}(q^{bc}) \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \tag{8b}$$

$$\varphi^e = \arctan 2(l_y^e, l_x^e), \tag{8c}$$

$$\theta^e = \arctan 2\left(\|[l_x^e\ l_y^e]^T\|_2, l_z^e\right), \tag{8d}$$

$$\rho^e = \frac{1}{d_0^e}. \tag{8e}$$

Here, q^{bc} is the unit quaternion describing the fixed rotation from the camera frame to the body frame. Furthermore, p^e is the camera position when the landmark is observed and d_0 is the initial depth for the landmark. Finally, $\theta = \arctan 2(\,\cdot\,)$ is the four-quadrant arc-tangent, $\theta \in [-\pi, \pi]$. The complete landmark vector is of the dimension $6 \times n_{\text{landmarks}}$ and $n_{\text{landmarks}}$ will vary depending on when new landmarks are initiated.

3.3 Camera Measurements

The measurements are sub-pixel coordinates in the images given by the SIFT feature extractor (Lowe, 1999). The dimension of the measurement vector y_{t_k} is $2 \times n_{\text{af}}$, where n_{af} denotes the number of associated features. The measurements are expressed in normalised pixel coordinates. The camera measurement equation relating states and measurements has the form

$$y_{t_k} = \underbrace{h(x_{t_k}, m_{t_k})}_{y_{t_k}^c} + e_{t_k}, \tag{9}$$

where

$$e_{t_k} \sim \mathcal{N}(0, R_{t_k}),\ R_{t_k} = \sigma_{\text{features}} I_{2 \times n_{\text{af}}}. \tag{10}$$

Using the IDP, (7) and (8), for a single landmark j and omitting time dependency, the measurement (9) is calculated as

$$m_j^c = \begin{bmatrix} m_{x,j}^c \\ m_{y,j}^c \\ m_{z,j}^c \end{bmatrix} = \mathsf{R}(q^{bc})^T\, \mathsf{R}(q^{be}) \left(\rho_j^e \left(p^e - p_j^e - \mathsf{R}(q^{be})^T r^{bc}\right) + \gamma(\varphi_j, \theta_j)\right), \tag{11a}$$

$$y_j^c = \frac{1}{m_{z,j}^c} \begin{bmatrix} m_{x,j}^c \\ m_{y,j}^c \end{bmatrix}, \tag{11b}$$

where p_j^e and ρ_j^e are defined in (8a) and (8e), respectively. The translation r^{bc} and orientation $\mathsf{R}(q^{bc})$ defines the constant relative pose between the camera and the IMU. The parameters in r^{bc} and $\mathsf{R}(q^{bc})$ were estimated in the previous work by Hol et al. (2010).

4 Solution

The proposed solution starts with an initialisation of the states using EKF SLAM and the initial states are then smoothed using nonlinear least squares.

4.1 Initialisation

The nonlinear least-squares algorithm needs an initial estimate $x_{0:N}^0$, m_N^0, which is obtained using EKF SLAM. The time update is performed with the model (5) in a standard EKF, for details, see e.g., Kailath et al. (2000). The the landmark states (1b) are stationary and will therefore only be corrected in the measurement update.

The measurement update needs some further explanation. Each time an image is available (which in our experiments is 8 times slower than the specific force and the angular rate inputs) a measurement update is made.

The measurement update needs an association between the features extracted from the current image and the landmarks present in the state vector. The associations computed during EKF SLAM are found in the following way; first, all landmarks are projected into the image according to (9) and the most probable landmarks are chosen as the nearest neighbours inside a predefined region. Second, the SIFT feature descriptors for the landmarks and features inside the region are matched. In this way a data association sequence is created for each image, relating the measurements and the landmarks in the state vector. To enhance the feature tracking we discard unstable features (i.e., those that are only measured once or twice) and features are proclaimed usable only if they are found at least three times.

4.2 Nonlinear Least-Squares Smoothing

The nonlinear problem (4) is in our approach solved using the Gauss-Newton method, i.e., at each iteration we solve the linearised version of the problem.

In order to formulate the linearised least-squares smoothing problem for our specific setup we first need some definitions:

$$F_t \triangleq \left.\frac{\partial f(x,u)}{\partial x}\right|_{(x,u)=(x_{t-1}^0,u_t)}, \tag{12}$$

is the Jacobian of the motion model and

$$H_{t_k}^j \triangleq \left.\frac{\partial h(x,m)}{\partial x}\right|_{(x,m)=(x_{t_k}^0,m_j^0)}, \tag{13}$$

is the Jacobian of the measurement k at time t_k with respect to the vehicle states.

The IDP gives a special structure to the equations since the measurements of the features are related to the pose where the features where initialised. Therefore, the landmark Jacobian is split into two parts. The first part is

$$J^j_{x_{t_k}} \triangleq \left.\frac{\partial h(x,m)}{\partial x}\right|_{(x,m)=(x^0_{t_k},m^0_j)}, \tag{14}$$

which is the Jacobian of measurement k at time t_k, with respect to the position where landmark j was initialised. The second part is the Jacobian of measurement k at time t_k of the states ϕ_j, θ_j and ρ_j of landmark j

$$J^j_{t_k} \triangleq \left.\frac{\partial h(x,m)}{\partial m}\right|_{(x,m)=(x^0_{t_k},m^0_j)}. \tag{15}$$

From the initialisation, Section 4.1, a trajectory $x^0_{0:N}$ and a landmark m^0_N estimate is given and is therefor treated as a constant. The linearised process model at time t is then

$$x^0_t + \delta x_t = F_t(x^0_{t-1} + \delta x_{t-1}) + Bu_t + B_w(x^0_{t-1})w_t. \tag{16}$$

The linearised measurement equations are given by

$$y^j_{t_k} = h(x^0_{t_k}, m^0_j) + H^j_{t_k}\delta x_{t_k} + J^j_{x_{t_k}}\delta x_{t_k} + J^j_{t_k}\delta m_j + e^j_{t_k}. \tag{17}$$

The linearised least-squares problem for the prediction and measurement errors is then

$$\begin{aligned}[\delta x^*_t,\ \delta m^*_j] = \operatorname*{arg\,min}_{\delta x_t, \delta m_j} & \sum_{t=1}^{N} \|F_t\delta x_{t-1} - I\delta x_t - a_t\|^2_{\widetilde{Q}_t^{-1}} + \\ & \sum_{k=1}^{K} \|H^j_{t_k}\delta x_{t_k} + J^j_{x_{t_k}}\delta x_{t_k} + J^j_{t_k}\delta m_j - c^j_{t_k}\|^2_{R^{-1}_{t_k}}\end{aligned} \tag{18}$$

where $a_t = x^0_t - F_t x^0_{t-1} - Bu_t$ and $c^j_{t_k} = y^j_{t_k} - h(x^0_{t_k}, m^0_j)$. Here a_t and $c^j_{t_k}$ are the prediction errors of the linearised dynamics around x^0_t and the innovations, respectively. The stacked version of the problem (18) can be solved iteratively according to

$$\eta^{i+1} = \operatorname*{arg\,min}_{\eta} \|A(\eta^i)\eta - b(\eta^i)\|^2_2, \quad \eta^0 = 0, \tag{19}$$

where we define $\eta = [\delta x_t,\ \delta m_j]$, and $A(\eta)$ and $b(\eta)$ is the matrix part and the vector part of (18), respectively. The structure of the A matrix is perhaps best explained using an example:

$$A(\eta) = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix} = \left[\begin{array}{cccccccc|cc}
-I & & & & & & & & & \\
F_2 & -I & & & & & & & & \\
 & F_3 & \ddots & & & & & & & \\
 & & \ddots & -I & & & & & & \\
 & & & F_6 & \ddots & & & & & \\
 & & & & \ddots & -I & & & & \\
 & & & & & F_{10} & \ddots & & & \\
 & & & & & & \ddots & -I & & \\
\hline
J^1_{x_5} & & & H^1_5 & & & & & J^1_5 & \\
J^2_{x_5} & & & H^2_5 & & & & & & J^2_5 \\
J^2_{x_9} & & & & & H^2_9 & & & & J^2_9 \\
J^1_{x_{13}} & & & & & & & H^1_{13} & J^1_{13} &
\end{array}\right], \tag{20}$$

$t_k = 1:$ Two landmarks are seen for the first time giving the landmarks' initialisation positions, i.e., the columns where the Jacobians (14) are placed.

$t_k = 5:$ The second camera measurement arrives, landmarks 1 and 2 are observed and the first two block rows of A_{21} and A_{22} are added.

$t_k = 9:$ Camera measurement 3 arrives, landmark 2 is observed and the third block row of A_{21} and A_{22} is added.

$t_k = 13:$ Camera measurement 4 arrives, landmark 1 is observed and the fourth block row of A_{21} and A_{22} is added.

A single iteration of the nonlinear least-squares smoothing algorithm can be summarised in pseudo code as seen in Algorithm 3.

The least-squares problem is weighted, so it is assumed that all of the terms in (18) are multiplied with the corresponding matrix square root of the inverse of the covariance matrices for the process and the measurement noise, respectively. Note that the covariance matrix of the process noise, $\widetilde{Q}_t = B_w(x_t^0)Q_tB_w(x_t^0)^T$, is singular rendering the use of normal inversion impossible. In order to overcome this, we simply regularise the problem by adding a diagonal matrix ΔI to the covariance matrix, with Δ being a small number, rendering the covariance matrix invertible. Furthermore, it is assumed that the associations from the initialisation is good enough and that we do not have to compute new associations after each iterate.

5 Experiments

The implementation is done in MATLAB, except for the SIFT binaries, where we use a C code library from Hess (2010).

Algorithm 3 Nonlinear Least-Squares Smoothing for SLAM

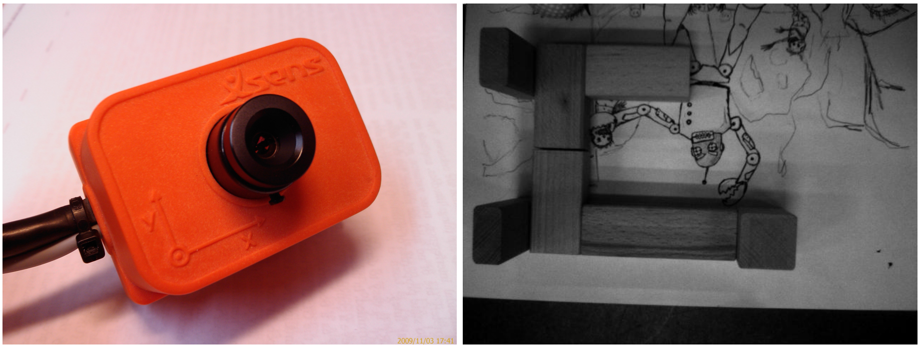
```
Input: $x^0, m^0$ (trajectory and map from previous iteration), $u$ (inputs), data association

Output: $x^s, m^s$ (smoothed estimate of the trajectory and the map)
N = # IMU measurements
$A = [\,],\ a = [\,], c = [\,]$
for $i$ = 1 to N do
    predict states, $x_i = f(x_{i-1}^0, u_i)$
    if image available then
        use the data association and calculate $h(x_i^0, m_i^0)$
        calculate $A_{11} = [A_{11}\ A_{11}^i]^T$,
        $A_{21} = [A_{21}\ A_{21}^i]$ and
        $A_{22} = [A_{22}\ A_{22}^i]$ according to (12) - (20)
        calculate $a_i = x_i^0 - x_i$
        calculate $c_i = y_i - h(x_i^0, m_i^0)$
        set $a = [a^T\ a_i^T]^T$
        set $c = [c^T\ c_i^T]^T$
    else
        calculate $A_{11} = [A_{11}\ A_{11}^i]^T$
        calculate $a_i = x_i^0 - x_i$
        set $a = [a^T\ a_i^T]^T$
    end if
end for
Assemble up $A$ according to (20) and $b = [a^T\ c^T]^T$
solve the least squares problem (19)
calculate $[{x^s}^T,\ {m^s}^T]^T = [{x^0}^T,\ {m^0}^T]^T + \eta$
```

5.1 Experimental Setup

For the purpose of obtaining high quality ground truth motion data we used an IRB 1400 industrial robot from ABB. In an industrial robot the rotation and translation of the end tool can be logged with high accuracy. This gives an excellent performance evaluation possibility, which is otherwise difficult. The actual robot trajectory was not possible to acquire during the experiment. However, since the industrial robot is very accurate the actual output of the robot will be very close to the programmed trajectory.

We constructed a small synthetic environment with known topography to obtain realistic ground truth map data, see Figure 1b. We use a combined IMU/camera sensor unit, shown in Figure 1a. The sensor unit is mounted at the end tool position of the industrial robot. The IMU measurements are sampled at 100 Hz and images of size 640 × 480 pixels are sampled at 12.5 Hz.



*(**a**) The combined strap down IMU and camera system.*

*(**b**) An image from the camera during the experiment.*

Figure 1: *The IMU/camera sensor unit used in the experiments and an image from the camera over-viewing the synthetic environment.*

5.2 Results

The resulting trajectories and map obtained with the data from an experiment are presented in Figure 2. The Ground truth trajectory is a reference trajectory for the robot. From these plots it is clearly visible that the smoothed estimate is closer to the true trajectory than the initial estimate. The improvement is also visible if the initial estimate and the final smoothed landmark estimate are compared as in Figure 3. Note that some landmark positions are already quite accurately estimated since the change is small after the smoothing. The smoothed estimate also has a more accurate universal scale.

Both the smoothed horizontal speed of the platform, defined as $\|[v_t^x \; v_t^y]^T\|_2$, and resulting estimate from the initialisation are plotted in Figure 4. We see that the smoothed speed is much closer to 0.1 m/s, which is the true speed.

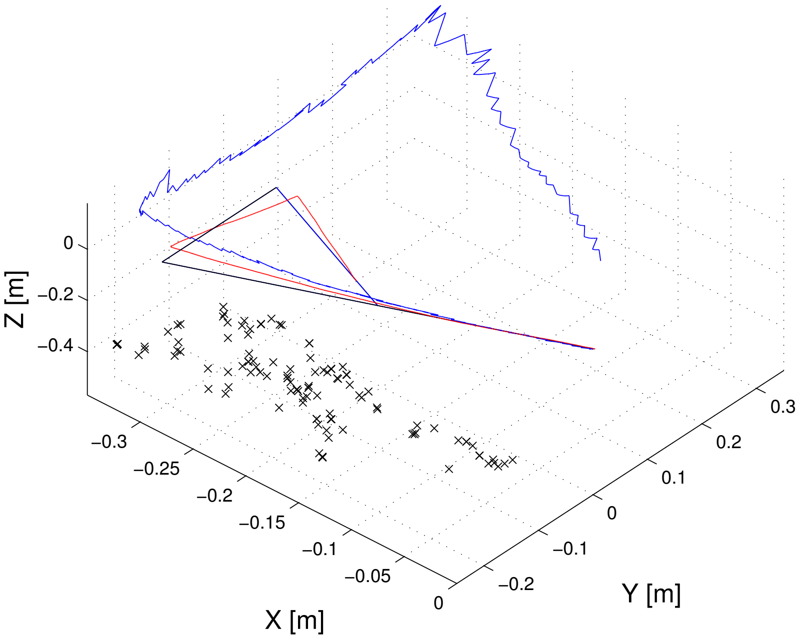


Figure 2: *The smoothed trajectory in red, the initial EKF trajectory in blue and the ground truth trajectory in black. The black crosses are the smoothed landmark estimates.*

6 Conclusions and Future Work

In this work we have presented the SLAM problem formulated as nonlinear least-squares. For evaluation we have used a combined camera and IMU sensor unit mounted at the manipulator of an industrial robot which gives accurate ground truth.

The experimental results in Section 5.2 show that the nonlinear least-squares trajectory, Figure 2, and the speed estimate, Figure 4, show a significant improvement of the initial estimate. The sparse point cloud in Figure 3, illustrating the initial landmark estimate and final smoothed estimate, shows also an improvement. The universal scale of the environment is improved since the landmarks have moved towards more probable positions.

For a long-term solution another initialisation procedure is necessary, since EKF SLAM is intractable for large maps. A possible alternative is to use IMU supported visual odometry to get a crude initial estimate. This approach needs a supporting global data association scheme.

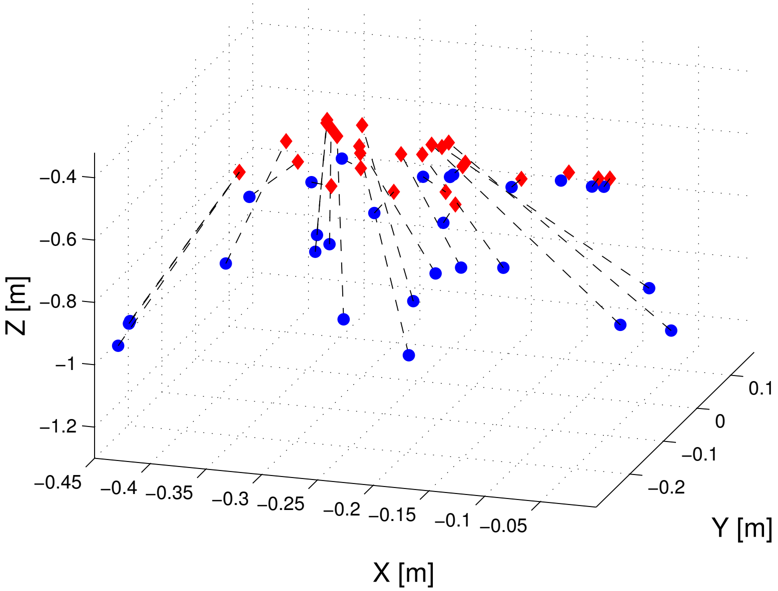


Figure 3: *The initial landmark estimates given by the EKF in blue bullets and the final smoothed estimate in red diamonds, where the black dashed line illustrate the relative displacement.*

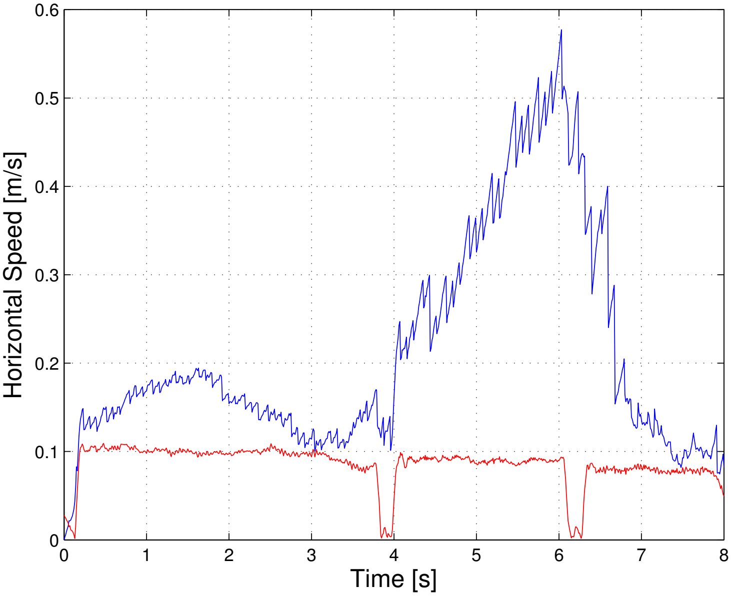


Figure 4: *The smoothed horizontal speed of the camera in red and EKF in blue. The true speed is* 0.1 *m/s except for when the robot stops and changes direction, this happens at about 4 seconds and 6 seconds.*

Bibliography

T. Bailey and H. Durrant-Whyte. Simultaneous Localization and Mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, September 2006.

T. Bailey, J. Nieto, J. E. Guviant, M. Stevens, and E. M. Nebot. Consistency of the EKF-SLAM algorithm. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 3562–3568, Beijing, China, 2006a. doi: 10.1109/IROS.2006.281644. URL `http://dx.doi.org/10.1109/IROS.2006.281644`.

T. Bailey, J. Nieto, and E. M. Nebot. Consistency of the FastSLAM algorithm. In *International Conference on Robotics and Automation (ICRA)*, pages 424–429, Orlando, Florida, USA, 2006b.

C. Bibby and I. Reid. Simultaneous Localisation and Mapping in Dynamic Environments (SLAMIDE) with reversible data association. In *Proceedings of Robotics: Science and Systems (RSS)*, Atlanta, GA, USA, June 2007.

J.-Y. Bouguet. Camera Calibration Toolbox for Matlab, 2010. URL `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

M. Bryson, M. Johnson-Roberson, and S. Sukkarieh. Airborne smoothing and mapping using vision and inertial sensors. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 3143–3148, Kobe, Japan, 2009. IEEE Press. ISBN 978-1-4244-2788-8.

M. Chli. *Applying Information Theory to Efficient SLAM*. PhD thesis, Imperial College London, 2009.

J. Civera, A.J. Davison, and J. Montiel. Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, October 2008. ISSN 1552-3098. doi: 10.1109/TRO.2008.2003276.

A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the 9th IEEE International Conference on computer vision*, pages 1403–1410, 2003.

A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29 (6):1052–1067, 2007.

F. Dellaert and M. Kaess. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *International Journal of Robotics Research*, 25(12):1181–1203, 2006. ISSN 0278-3649. doi: 10.1177/0278364906072768. URL `http://dx.doi.org/10.1177/0278364906072768`.

H. Durrant-Whyte and T. Bailey. Simultaneous Localization and Mapping: Part I. *IEEE Robotics & Automation Magazine*, 13(12):99–110, June 2006.

E. Eade. *Monocular Simultaneous Localisation and Mapping*. PhD thesis, Cambridge University, 2008.

R. M. Eustice, H. Singh, and J. J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transactions on Robotics*, 22(6):1100–1114, 2006.

A. W. Fitzgibbon and A. Zisserman. Automatic Camera Recovery for Closed or Open Image Sequences. In Hans Burkhardt and Bernd Neumann, editors, *ECCV (1)*, volume 1406 of *Lecture Notes in Computer Science*, pages 311–326. Springer, 1998. ISBN 3-540-64569-1.

J. E. Guviant and E. M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, June 2001. ISSN 1042-296X.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. ISBN 0-521-54051-8.

R. Hess. OpenSIFT, 2010. URL `http://robwhess.github.io/opensift/`.

J. Hol, T. B. Schön, and F. Gustafsson. Modeling and Calibration of Inertial and Vision Sensors. *The International Journal of Robotics Research*, 29(2), February 2010.

M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental Smoothing and Mapping. *IEEE Transansactions on Robotics*, 24(6):1365–1378, December 2008.

T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice-Hall, Upper Saddle River, New Jersey, 2000.

G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 225–234, Nara, Japan, 2007.

J. J. Leonard, H. Jacob, and S. Feder. A Computationally Efficient Method for Large-Scale Concurrent Mapping and Localization. In *Proceedings of the Ninth International Symposium on Robotics Research*, pages 169–176, Salt Lake City, Utah, 2000. Springer-Verlag.

D. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, pages 1150–1157, Corfu, Greece, 1999.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003.

P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modelling. In *5th International Symposium on Robotics Research*, pages 207–216, Tokyo, Japan, 1989.

Z. Sjanic, M. A. Skoglund, T. B. Schön, and F. Gustafsson. A Nonlinear Least-Squares Approach to the SLAM Problem. In *Proceedings of 18th IFAC World Congress*, Milano, Italy, August/Septemeber 2011.

R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990. ISBN 0-387-97240-4.

C. J. Taylor, D. Kriegman, and P. Anandan. Structure and Motion in Two Dimensions from Multiple Images: A Least Squares Approach. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 242 –248, Princeton, NJ , USA, October 1991.

B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000.

Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, November 2000. ISSN 0162-8828. doi: 10.1109/34.888718.

Paper E

Initialisation and Estimation Methods for Batch Optimisation of Inertial/Visual SLAM

Authors: Martin A. Skoglund, Zoran Sjanic and Fredrik Gustafsson

Edited version of the paper:

M. A. Skoglund, Z. Sjanic, and F. Gustafsson. Initialisation and Estimation Methods for Batch Optimisation of Inertial/Visual SLAM. *Submitted to IEEE Transactions on Robotics*, September 2013.

Preliminary version:

Technical Report LiTH-ISY-R-3065, Dept. of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden.

Initialisation and Estimation Methods for Batch Optimisation of Inertial/Visual SLAM

Martin A. Skoglund, Zoran Sjanic and Fredrik Gustafsson

Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
{ms,zoran,fredrik}@isy.liu.se

Abstract

Simultaneous Localisation and Mapping (SLAM) denotes the problem of jointly localising a moving platform and mapping the environment. This work studies the SLAM problem using a combination of inertial sensors, measuring the platform's accelerations and angular velocities, and a monocular camera observing the environment. We formulate the SLAM problem on a nonlinear least squares (NLS) batch form, whose solution provides a smoothed estimate of the motion and map. The NLS problem is highly nonconvex in practice, so a good initial estimate is required. We propose a multi-stage iterative procedure, that utilises the fact that the SLAM problem is linear if the platform's rotations are known. The map is initialised with camera feature detections only, by utilising feature tracking and clustering of feature tracks. In this way, loop closures are automatically detected. The initialisation method and subsequent NLS refinement is demonstrated on both simulated and real data.

1 Introduction

The goal in Simultaneous Localisation and Mapping (SLAM) is to estimate a map of the surrounding environment from a moving platform, while simultaneously localising the platform, or more generally, to estimate the state of the platform including both position and orientation. For more than twenty years SLAM has been a popular field of research and is considered an important enabler for autonomous robotics. An excellent introduction to SLAM is given in the two part tutorial by Durrant-Whyte and Bailey (2006); Bailey and Durrant-Whyte (2006).

There are many types of sensors used in SLAM applications and the laser range scanners are probably the most popular while cameras have increased in popularity in recent years, see Chli (2009) for a thorough overview of visual SLAM. Bundle Adjustment (BA), (Hartley and Zisserman, 2004; Triggs et al., 2000; Agarwal et al., 2009), belongs to these methods. The idea is to minimise the reprojection error for all camera poses and the structure as a (potentially) large nonlinear least-squares (NLS) problem. The online (filtering) meth-

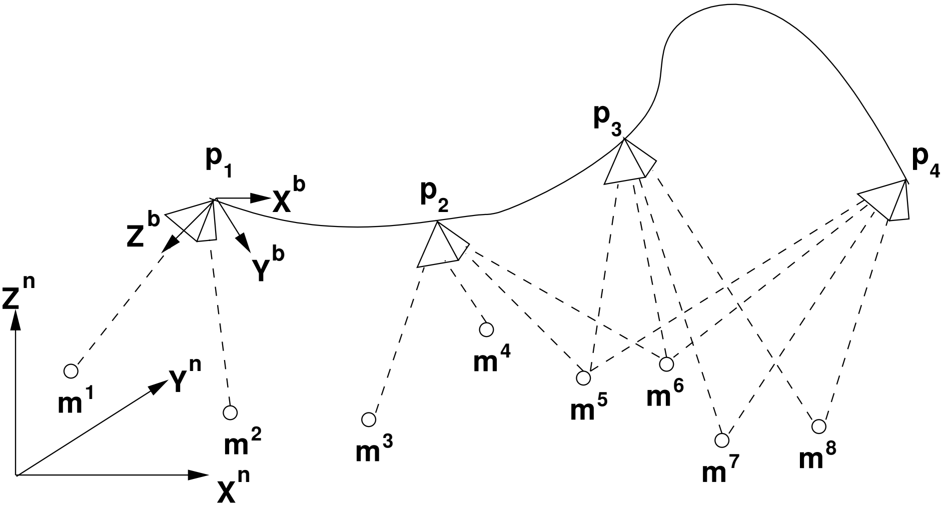


Figure 1: *The setup with inertial and visual sensors. The camera is observing the environment represented by point landmarks,* $m^1, \ldots, m^8$*, and the inertial sensors are measuring accelerations and angular rates in the body coordinate system,* (b)*, which moves together with the camera. Also, a global, fixed navigation coordinate system,* (n)*, is drawn.*

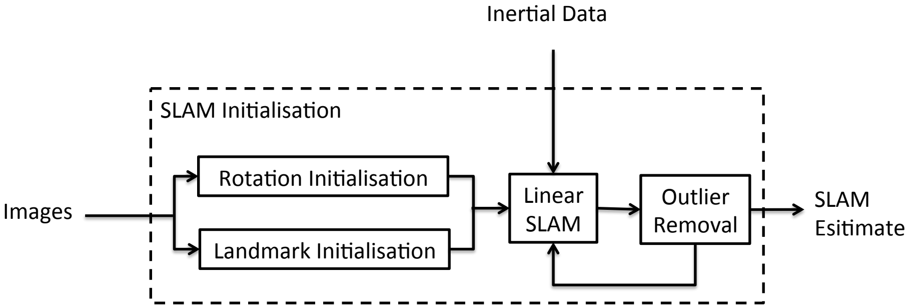
ods, which handle sequential data, are known as Structure from Motion (SfM) (Fitzgibbon and Zisserman, 1998; Taylor et al., 1991). A typical pipeline combining SfM and BA initialises the BA optimiser with a solution obtained from running SfM on the complete data. Without any other sensors measuring the platform's dynamics, the trajectory and map can only be estimated up to a universal scale. It is therefore necessary to enforce a scale (typically unit) when solving a BA system, since otherwise all coordinates will converge to a point. The underlying idea in visual/inertial SLAM is to implicitly estimate the scale using the inertial measurements. By adding an Inertial Measurement Unit (IMU) measuring accelerations and angular rates (up to an unknown bias and noise), this, otherwise unknown, scale can be resolved, see e.g., Kneip et al. (2011b); Martinelli (2012); M. Bryson and M. Johnson-Roberson and S. Sukkarieh (2009); Lupton and Sukkarieh (2012). Inertial/visual SLAM is a branch of research with many applications in entertainment, augmented reality and autonomous robotics. There are both on-line and batch solutions to the SLAM problem. EKF-SLAM (P. Moutarlier and R. Chatila, 1989) and FastSLAM(Montemerlo et al., 2002) belong to the class of on-line algorithms. They are quite efficient for ground robotics, which was the driving application when the SLAM problem was originally formulated. However, these algorithms scale badly with the dimension of the map and platform state and are difficult to apply in their standard formulations. Batch algorithms can potentially overcome this limitation. The SLAM problem is easily formulated in a NLS framework, but the NLS cost function is highly non-convex. Thus, proper initialisation is needed. Although there are many batch formulations of the SLAM problem in literature, see for instance Dellaert and Kaess (2006); Thrun and Montemerlo (2006); Grisetti et al. (2007); Jung and Taylor (2001), initialisation is not discussed in detail. This paper is entirely devoted to the initialisation problem for a combination of monocular camera and inertial sensors. See Figure 1 for an illustration of this setup.

A multi-stage procedure for estimation of initial motion, map and data association based on images-only as well as combined visual/IMU methods is devised in this work. The initialisation is utilising a reformulation of the standard reprojection error given that platform's rotations are known, resulting in an almost linear formulation of the SLAM problem. That is, without measurement noise the problem would be completely linear, see Sinha et al. (2010); Martinelli (2012); Kneip et al. (2011b), but in our case the noise is parameter dependent and is estimated in an iterative fashion. Moreover, another important issue related to the visual part is the data association between camera measurements and map landmarks. These associations should be as accurate as possible since erroneous labeling can cause inconsistency in the estimate. The correspondence problem is fundamental in both SLAM and Bundle Adjustment. It is however somewhat easier with sequential data because features can be tracked in order to find the relative displacements of the platform and to predict positions of the tracked features. While locally consistent correspondences are rather easy to obtain, consistent loop closures are more difficult and there are no standard methods. In the proposed approach data association is based on data clustering, see for instance Hastie et al. (2009), of feature tracks. The feature tracks, see e.g., Thormählen et al. (2008), are estimated as a linear program formulation of the assignment problem. Outliers are efficiently eliminated using an iterative procedure on the reprojection errors using the IMU data. This initial estimate of the motion and map together with the data association can be used for warm starting for example a NLS SLAM procedure or other nonlinear estimators where all the parameters, including rotations, are treated as unknown. The initialisation procedure suggested in this paper provides such a value and leads to a better total estimate than for example naïve initialisation based on measurements only.

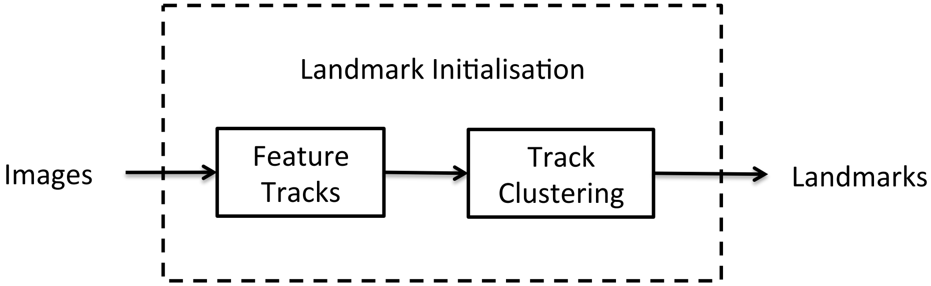
The block diagrams in Figure 2a serve as illustrative overview of the proposed method. The initialisation procedure computes a set of landmarks with its corresponding measurements (this includes the loop closure candidates), an estimate of the trajectory and velocity using IMU data and a camera-only rotation estimate. The flow of this procedure is illustrated in Figure 2b. The landmark initialisation in Figure 2b represents the image processing operations such as feature tracking and track clustering. Feature tracks are extracted from matching correspondence pairs in the image sequence. The feature tracks are then clustered based upon their average feature descriptor in order to find loop closure candidates. The clustered feature tracks are then used to initialise 3D point landmarks.

The paper is organised as follows; Section 2 describes the models of the different sensors used in the formulation of the problem. Section 3 handles the initialisation procedure based on the almost linear formulation of the visual/inertial SLAM problem. Here, all steps for the initialisation of the trajectory, orientation, landmarks and data association are described. Each subsection here represents a particular block in Figure 2b and Figure 2c. In Section 4 a nonlinear refinement method is used where the initial point is given by the proposed initialisation procedure. In Section 5 some comments and discussion that motivate the linear initialisation method are given. Finally, results on both simulated and real data are shown in Section 6 and Section 7 respectively, and some conclusions and

(a) *High level abstraction of the computational flow.*



(b) *Camera and inertial data initialisation procedure.*



(c) *Image processing procedure.*

Figure 2: *An overview of the proposed method. Most blocks corresponds to a subsection in Section 3 and Section 4 with the same name*

future work are discussed in Section 8.

2 Models

The sensors of interest are monocular camera and 6-DOF inertial sensors, gyroscopes and accelerometers, contained in a single unit. A standard Cartesian 3D point landmark parametrisation is used and its measurement is given by the pinhole projection model. In this work we assume that both the camera and the relative pose of the camera optical center with respect to the center of the IMU are calibrated. The camera calibration implies that image pixel coordinates can be transformed to metric coordinates and all the inertial measurements can be assumed to measure the camera's rotation and acceleration.

2.1 Position and Orientation

Given the accelerations, $a = [a^x, a^y, a^z]^T$, and angular velocities, $\omega = [\omega^x, \omega^y, \omega^z]^T$, of a moving and rotating object expressed in the non-moving frame, the so called navigation (or world or earth) frame. The the position, velocity and orientation (parametrised as unit quaternion $q = [q_0, q_1, q_2, q_3]^T$, $q^T q = 1$) of the object in the navigation frame, $[p, v, q]$, can be written as a discrete time dynamic model as

$$p_t = p_{t-1} + T_s v_{t-1} + \frac{T_s^2}{2} a_t \tag{1a}$$

$$v_t = v_{t-1} + T_s a_t \tag{1b}$$

$$q_t = \exp\left(\frac{T_s}{2} S_\omega(\omega_t)\right) q_{t-1} \tag{1c}$$

where T_s is the sampling time, the skew-symmetric matrix

$$S_\omega(\omega) = \begin{bmatrix} 0 & -\omega^x & -\omega^y & -\omega^z \\ \omega^x & 0 & \omega^z & -\omega^y \\ \omega^y & -\omega^z & 0 & \omega^x \\ \omega^z & \omega^y & -\omega^x & 0 \end{bmatrix}, \tag{2}$$

parametrises the quaternion dynamics and here $\exp(\,\cdot\,)$ denotes the matrix exponential.

2.2 IMU Measurements

The IMU measures the specific force and rotation speed in a frame attached to the IMU body frame, denoted b. Usually these measurements are imperfect and contain both biases and measurement noise. The biases are assumed constant and this is usually only a good approximation for a short period of time since in practice biases will vary due to e.g., temperature. Under these assumptions the measurements can then be described as

$$y_t^a = \mathsf{R}^{be}(q_t)(a_t^e - g^e) + b_a + e_t^a \tag{3a}$$

$$y_t^\omega = \omega_t + b_\omega + e_t^\omega \tag{3b}$$

where $g^e = [0, 0, -g]$ is the local gravity vector expressed in the navigation frame, and

$g \approx 9.82$, $\mathsf{R}^{be}(q)$ is the rotation matrix parametrisation of the quaternion and the measurement noises are assumed i.i.d. Gaussian with zero mean and time invariant covariances R_a and R_ω, i.e., $e_t^a \sim \mathcal{N}(0, R_a)$ and $e_t^\omega \sim \mathcal{N}(0, R_\omega)$.

2.3 Camera Measurements

The monocular camera is modeled as a standard pinhole camera, see cf. Hartley and Zisserman (2004). The camera calibration matrix and lens distortion need to be estimated prior to usage. Since the calibration and distortion are known the distorted pixels can be pre-multiplied with the inverse of the camera matrix and distortion can be compensated for. Thus, the camera then works as a projective map in Euclidean space, $P : \mathbb{R}^3 \to \mathbb{R}^2$. The projection is defined as $P([X, Y, Z]^T) = [X/Z, Y/Z]^T$ and normalised camera measurement $y_t^m = [u_t, v_t]^T$ of a landmark, m, at time t is then

$$y_t^m = P(\mathsf{R}^{ce}(q_t)(m - p_t)) + e_t^m \tag{4}$$

which relates the absolute pose of the camera w.r.t. the 3D location of the point. The measurement noise is assumed i.i.d. Gaussian, $e_t^m \sim \mathcal{N}(0, R_m)$. The correspondence variables at time t, c_t^i, encodes the measurement-landmark assignment, $y_t^i \leftrightarrow m^j$, which gives a subset of all M landmarks at time t, $\mathsf{M}_t = \{m^j\}$, $j \in \{1, \dots, M \mid c_t^i = j\}$. At time t the stacked measurement equation is then

$$\underbrace{\begin{bmatrix} u_t^1 \\ v_t^1 \\ \vdots \\ u_t^{\mathsf{N}_y} \\ v_t^{\mathsf{N}_y} \end{bmatrix}}_{y_t} = \underbrace{\begin{bmatrix} P(\mathsf{R}^{ce}(q_t)(m^{c_t^1} - p_t)) \\ \vdots \\ P(\mathsf{R}^{ce}(q_t)(m^{c_t^{\mathsf{N}_y}} - p_t)) \end{bmatrix}}_{h_t(x_t, \theta)} + e_t^m, \tag{5}$$

where c_t^i denotes the index of the corresponding landmark and N_y is the number of measurements, which of course varies over time. Methods for estimation of correspondence variables are discussed in Section 3.2.

3 SLAM Initialisation

In this section a method intended for initialisation of monocular visual/inertial SLAM from sequential data is described. The output of the method is a landmark map and the motion of the platform. It also establishes local correspondences via assignment variables using image features descriptors. Classical algorithms that solve assignment problems are the Hungarian (Munkres) algorithm (Kuhn, 1955) and the popular Auction algorithm (Bertsekas, 1991). Here a slightly different approach is adopted which results in a sequence of linear optimisation problems. In the landmark initialisation procedure we use appearance based correspondence matching, see e.g., Cummins and Newman (2010); Ho and Newman (2006). It aims at finding similar features corresponding to the same physical object in different images. Appearance based matching relies on feature descriptors that are distinctive and holds some invariance properties. For instance, image intensity invariance can be important in outdoor environments where lighting conditions may change and matching over large baselines requires invariance against scale, rotation and possibly invariance against change of viewpoint is desirable. In the following subsections we will describe the total initialisation procedure in detail and provide algorithms that implement these steps.

3.1 Feature Tracks

Feature tracks are established from the appearance of correspondences over multiple views by a matching scheme. Feature descriptor vectors, $\boldsymbol{f}$, from the popular Scale-Invariant Feature Transform (SIFT) (Lowe, 1999) are used for establishing correspondences. Given a sequence of images $I_t,\ t \in \{1, \dots, K\}$, the feature matching problem consists of assigning a subset of feature measurements from image $I_t,\ \boldsymbol{f}_t^i,\ i \in \{1, \dots, N_t\}$, to a subset of feature measurements from the next image $I_{t+1},\ \boldsymbol{f}_{t+1}^j,\ j \in \{1, \dots, M_{t+1}\}$, such that each measurement gets assigned to exactly one, unique, other measurement. In a manner similar to measurement-landmark assignment described before these assignments are also encoded by correspondence variables (which are binary in this case), $c_t^{ij} \in \{0, 1\}$, which are collected into c_t and the assignments for all images are collected into $\boldsymbol{c}$. Furthermore, each assignment is associated with a matching cost G_t^{ij} as

$$\mathsf{G}_t^{ij} = -\|\boldsymbol{f}_t^i - \boldsymbol{f}_{t+1}^j\|_2^{-1}, \tag{6}$$

which is the negative inverse Euclidean distance between the feature descriptor vectors. The costs are used to construct a matrix and to find pairwise matches in the image sequence. This is done by solving the assignment problem which can be formulated as the following binary program (BP)

$$\begin{aligned} \widehat{c}_t = \arg\min_{c_t^{ij}} & \sum_{i=1}^{N_t} \sum_{j=1}^{M_{t+1}} \mathsf{G}_t^{ij}\, c_t^{ij} \\ \textbf{s. t.} & \sum_{i=1}^{N_t} c_t^{ij} \le 1, \forall j \\ & \sum_{j=1}^{M_{t+1}} c_t^{ij} \le 1, \forall i \\ & c_t^{ij} \in \{0, 1\} \end{aligned} \tag{7}$$

which is typically hard to solve. A standard method is to relax the binary constraints $c_t^{ij} \in \{0, 1\}$ to $0 \le c_t^{ij} \le 1$. This relaxation gives that (7) becomes a linear program (LP) which is much easier to solve. A compact representation of the relaxed BP assignment problem (7) on matrix form is (omitting time index for readability)

$$\begin{aligned} \widehat{\bar{c}} = \arg\min_{\bar{c}} \ & \bar{\mathsf{G}}^T \bar{c} \\ \textbf{s. t. } & A\bar{c} \le \mathbf{1}_{(N+M)\times 1} \\ & \mathbf{0}_{NM\times 1} \le \bar{c} \le \mathbf{1}_{NM\times 1} \end{aligned} \tag{8}$$

where $\bar{\mathsf{G}}$ and $\bar{c}$ are the vectorised versions of the matrices G and c where columns are stacked on top of each other. The matrices $\mathbf{1}_{i\times j}$ and $\mathbf{0}_{i\times j}$ are the $i \times j$ matrix of ones and zeros respectively. Matrix A, which has dimension $(N+M)\times NM$, has a specific structure as follows: The first M rows look like

$$A_1 = I_M \otimes \mathbf{1}_{1\times N} \tag{9a}$$

and the last N rows look like

$$A_2 = \mathbf{1}_{1\times M} \otimes I_N \tag{9b}$$

and $A = [A_1^T \; A_2^T]^T$. $\otimes$ represents the Kronecker's matrix product. This constraint matrix A is totally unimodular, that is, all possible square sub-matrices are unimodular i.e., having determinant equal to ± 1. An important observation is that the matrix of (relaxed) constraints is unimodular. This means that the LP problem is integral, i.e., its optimum has an integer value corresponding to the optimum of the original BP problem (Papadimitriou and Steiglitz, 1982). This means the assignment problems are simple since good and fast LP solvers are readily available. In this work Gurobi Optimizer (Gurobi Optimization Inc, 2013) is used. The computational bottleneck for these problems is creating the cost matrix G since each element must be calculated.

The solution to the assignment problem will always use all the measurements from the smaller set no matter how bad the fit is. The reason is because the cost will always decrease by assigning one more variable, no matter how small the decrease is. This is not a desired behavior since these matches can in principle be arbitrarily bad. It is therefore necessary to model features which are unique for each measurement such that they do not end up being assigned. One way of doing this is to add a regularisation term to the assignment cost as

$$\mathsf{G}_t^{ij} = -\|\boldsymbol{f}_t^i - \boldsymbol{f}_{t+1}^j\|_2^{-1} + \eta, \tag{10}$$

where $\eta > 0$ is a tuning parameter which controls the rejection of the excess assignments that are bad. Thus, η will force the cost of certain, unlikely, assignments to become positive, implying that those assignments will never be chosen since they would increase the total cost.

3.2 Track Clustering

Feature tracks are defined as a time sequence of pairwise matching feature correspondences $\mathsf{C}_{t:t+s} = [\boldsymbol{f}_t, \boldsymbol{f}_{t+1}, \ldots, \boldsymbol{f}_{t+s}]$. The minimum length of a track is then a pair because a feature without a match is not useful. The length of the tracks has a twofold interpretation; a feature descriptor is unique with respect to others in the sequence, i.e., the feature has a unique surrounding, and the other case is when the camera is stationary and thus the scene has been observed for a long time. However, in case of a moving camera, feature tracks may be lost due to e.g., occlusion or change of perspective. Therefore, new feature tracks may represent previously initiated tracks. To cater for this a track clustering scheme is employed joining tracks that may represent observations of the same feature. For simplicity of calculation, each track is represented by the mean value, $\bar{\mathsf{C}}$, of all the descriptors that constitute that track. The distance between tracks used for clustering is

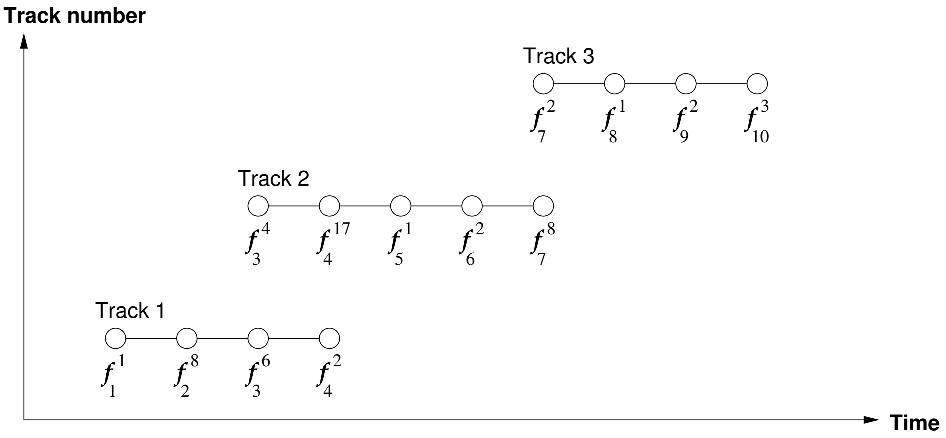


Figure 3: *Example of time-disjoint track clustering where only Track 1 and Track 3 are allowed to be clustered. Subscript figures on the descriptors, f, are the time indices and superscript figures are the enumeration of features at each time instant.*

then

$$d_{ij} = \|\bar{\mathsf{C}}^i - \bar{\mathsf{C}}^j\|_2. \tag{11}$$

Since tracks have a temporal meaning and each feature can be measured only once in each image, valid clusters of tracks contain only time-disjoint tracks. That is, $\mathsf{C}_{i:j}$ and $\mathsf{C}_{k:l}$ are allowed to be clustered together only if $\{i : j\} \cap \{k : l\} = \emptyset$. Time-disjoint clustering is illustrated in Figure 3. Solving for these constraints can be done by simply removing time-overlapping tracks within clusters. Since the amount of landmarks is unknown, a clustering method where the number of clusters is not explicitly given is used. One such method is single-linkage clustering where the clustering stops when some condition on the between-cluster distance is fulfilled, (Hastie et al., 2009). This distance is viewed as a tuning parameter. Another benefit of the single-linkage clustering is its speed, since there are good implementations available. Furthermore, data reduction and automatic loop closure detection is obtained since loops are defined by clusters containing more than one track. This implies that the between-cluster distance used for termination of the clustering controls the quality of the loop closures; if stopping too soon, there will be many small clusters and some loop closures will be missed and if stopping too late the risk of clustering wrong tracks together is increased. The track clustering algorithm is summarised in Algorithm 4. In this way, a set of landmarks has been obtained representing the initial map of the environment. Errors introduced, outlier measurements, in the clustering and in the feature tracks should be removed. This will be done according to Algorithm 6.

3.3 Rotation Initialisation

From sets of correspondences, estimated as in Section 3.1, for each image pair in the image sequence the relative transformations (up to a scale of the translation) can be obtained. This is done with the Eight-Point Algorithm, see e.g., Hartley and Zisserman (2004), re-

Algorithm 4 Track Clustering

```
Input: The set of all tracks $\{\mathsf{C}^i\}_{i=1}^{N_\mathsf{C}}$, where $\mathsf{C}^i = [\boldsymbol{f}_k, \boldsymbol{f}_{k+1}, \dots, \boldsymbol{f}_{k+l}]$, $l - k > 1$.
Output: Track clusters $\mathbf{C}$
Compute track means:
$\bar{\mathsf{C}}^i = \frac{1}{l-k}\sum_{t=k}^{l} \boldsymbol{f}_t,\ i = 1, \dots, N_\mathsf{C}$
Cluster the tracks:
$\tilde{\mathbf{C}} = Cluster\_data(\bar{\mathsf{C}}^i),\ i = 1, \dots, N_\mathsf{C}$
for all clusters do
    if tracks within the cluster are time disjoint then
        keep the cluster
    end if
end for
Remaining clusters are $\mathbf{C}$
```

sulting in a rotation sequence

$$\mathbf{R}^c = \{\mathsf{R}_i^c\}_{i=1}^{K-1}, \tag{12}$$

where R_i^c is the relative rotation of the cameras, (c), from time i to $i+1$. The global rotation, from world to camera, (ce), for any time, j, can be calculated by the matrix product

$$\mathsf{R}_j^{ce} = \prod_{i=j}^{0} \mathsf{R}_i^c. \tag{13}$$

Note that rotation matrices do not generally commute meaning that the product must be done in the reverse time order. Also note that the rotations are initialised with camera only and the gyro is not used. This is because the dominating error from the gyro is bias giving drifting rotation estimates. Errors from camera estimated rotations have more random like behaviour resulting in the random walk errors. There may also be correspondence errors in the feature tracks resulting in bad rotations. The sensitivity to initial rotation errors are further analysed in Section 6.2.

3.4 Linear SLAM

Methods of 3D structure estimation using linear methods and image point correspondences are well known. The basic idea is to form an overdetermined triangulation problem, which is linear in the unknowns, and solve it by linear least squares. This is essentially the Direct Linear Transformation (DLT) (Abdel-Aziz and Karara, 1971) which may work well in practice. However, instead of minimising the discrepancy between measured image coordinates and the back projection of points, an algebraic error without good geometrical interpretation is minimised (Hartley and Zisserman, 2004). It is therefore common to proceed with a nonlinear optimisation over the reprojection residuals with the linear method as a starting point. Given the correct weighting of the measurements, then the linear method minimises the reprojection error, see for instance Zhang and Kanade (1998). This weighting does however depend on the unknown depth of the points but iterative re-weighting often improves the linear solution.

The projection is a convex operation and this can be exploited in many ways. Optimal approaches to reconstruction consider reprojection errors under the L_∞ norm (Hartley and Schaffalitzky, 2004; Kahl and Henrion, 2007) since it preserves (quasi)-convexity, see Olsson and Kahl (2010); Kahl (2005), and have a single optimum which is typically not the case for the L_2 cost. These approaches assume outlier free measurements because otherwise the maximum error will correspond to an outlier and by using this insight an iterative method for outlier removal using L_∞ was proposed (Sim and Hartley, 2006), however, it does not scale to large problems (Agarwal et al., 2008). In common, it is assumed that rotations are known beforehand and in many situations this is a reasonable alternative since rotations may be estimated from point correspondences. Known (or error-free) orientation was considered for fusion of vision and inertial sensors in Martinelli (2012) and for the case of visual odometry with inertial measurements in Kneip et al. (2011a,b). The assumption of known orientation will also be used here which results in an almost linear method. Assuming known rotations it is possible to rewrite (4) in the following way (omitting time index)

$$\delta = \mathsf{R}^{ce}(q)(m - p) \tag{14}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{\delta_x}{\delta_z} \\ \frac{\delta_y}{\delta_z} \end{bmatrix} + \begin{bmatrix} e_u \\ e_v \end{bmatrix} \Rightarrow \tag{15}$$

$$\begin{bmatrix} u\delta_z \\ v\delta_z \end{bmatrix} = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} + \begin{bmatrix} e_u\delta_z \\ e_v\delta_z \end{bmatrix} \tag{16}$$

where $\delta(p, m) = [\delta_x,\ \delta_y,\ \delta_z]^T$ is the difference between landmark and camera positions expressed in the camera coordinate system. Equation (16) is linear in the unknown parameters m and p, but has noise that is dependent on the parameters. With δ explicit (16) becomes

$$\mathsf{R}_{3,:}(m - p) \begin{bmatrix} u - e_u \\ v - e_v \end{bmatrix} = \begin{bmatrix} \mathsf{R}_{1,:}(m - p) \\ \mathsf{R}_{2,:}(m - p) \end{bmatrix}, \tag{17}$$

where $\mathsf{R}_{i,:}$ denotes the i:th row of the rotation matrix R^{ce}. The accelerometer measurements with bias are as in Section 2.2

$$y_t^a = \mathsf{R}_t^{ce}(a_t - g^e) + b_a + e_t^a \tag{18}$$

where g^e, b_a are assumed constant and $e_t^a \sim \mathcal{N}(0, R_a)$. Usually, the sampling rate of an IMU is faster than a camera which can be handled in a straightforward fashion e.g., by averaging accelerations between the camera samples

$$y_t^a = \frac{1}{(S_t - S_{t-1})} \sum_{s=S_{t-1}}^{S_t} y_s^a, \tag{19}$$

where S_t maps the time indices between the camera and IMU. Treating the accelerations, its bias, initial velocity and landmarks as unknown parameters and defining $\theta =$

Algorithm 5 Iterative Reweighted Least Squares

Input: IMU measurements, $y^a_{1:N}, (\omega_{1:N})$, feature measurements, $y^m_{1:N}$, rotations $\mathsf{R}^{ec}_{1:N}$, data associations $\boldsymbol{c}$, initial parameters $\theta_0 = [a_{1:N}, v_0, b_a, (b_\omega), m]^T$, number of iterations K

Output: Parameter estimates θ_{init}

```
for $k := 1 \dots K$ do
    Create the WLS problem according to (21) with $\tilde{R}_m = \delta_z(\theta_{k-1})^2 R_m$
    Solve the WLS problem giving $\theta_k$
end for
$\theta_{\text{init}} := \theta_k$
```

$[a_{1:N}, v_0, b_a, m]^T$, the following formulation is proposed

$$\theta_{\text{init}} = \arg\min_{\theta} \sum_{t=1}^{N} \|y^a_t - \mathsf{R}^{ce}_t(a_t - g^e) - b_a\|^2_{R_a^{-1}} + \|y^m_t \delta_z(p_t, m) - \delta_{x,y}(p_t, m)\|^2_{\tilde{R}_m^{-1}}$$

$$\textbf{s. t.} \begin{bmatrix} p_t \\ v_t \end{bmatrix} = F^t \begin{bmatrix} p_0 \\ v_0 \end{bmatrix} + \sum_{i=1}^{t} F^{i-1} B a_i, \qquad (20)$$

$$F = \begin{bmatrix} I_3 & T_s I_3 \\ 0 & I_3 \end{bmatrix}, \; B = \begin{bmatrix} \frac{T_s^2}{2} I_3 \\ T_s I_3 \end{bmatrix},$$

where $\tilde{R}_m = \delta_z(p^0_t, m^0)^2 R_m$, and superscript 0 indicates that the parameters used for weighting are fixed for each iteration. The constraints represent the second order linear dynamics introduced in (1). In turn, this can also be interpreted as a second order interpolation of the trajectory. The formulation in (20) would be a constrained weighted linear least squares (WLS) problem if the measurement noise in the camera did not depend on the landmark. Now since any p_t can be expressed as a (linear) function of v_0 and $a_{1:t}$ the constraint can be directly substituted into the cost function resulting in the following unconstrained problem

$$\theta_{\text{init}} = \arg\min_{\theta} \sum_{t=1}^{N} \|y^a_t - \mathsf{R}^{ce}_t(a_t - g^e) - b_a\|^2_{R_a^{-1}} + \\ \|y^m_t \delta_z(v_0, a_{1:t}, m) - \delta_{x,y}(v_0, a_{1:t}, m)\|^2_{\tilde{R}_m^{-1}}, \qquad (21)$$

where $\tilde{R}_m = \delta_z(v_0, a_{1:t}, m)^2 R_m$. The only difference between this problem and usual WLS is the parameter dependent noise for the landmark measurements. This can be treated in an iterative fashion where δ_z is used for weighting the noise covariance which is evaluated using the parameter values from the previous iteration. This approach is also known as Iterative Reweighted Least Squares (IRLS) which is a well known method, see e.g., Björck (1996). The procedure is described in Algorithm 5. This approach usually converges after a few iterations and in our implementation three ($K = 3$) iterations were a suitable choice.

The problem (21) can be augmented with linear terms for initial gyro bias estimation,

Algorithm 6 Iterative Outlier Rejection

Input: IMU measurements, $y^a_{1:N}$, feature measurements, $y^m_{1:N}$, rotations $\mathsf{R}^{ec}_{1:N}$, data associations $\boldsymbol{c}$, initial parameters $a_{1:N}$, v_0, b_a, m, rejection threshold λ
Output: Data associations $\boldsymbol{c}$

```
terminate := false
$k := 0$
$\boldsymbol{c}^k := \boldsymbol{c}$
while not terminate do
    Solve the problem according to Algorithm 5 given the assignments $\boldsymbol{c}^k$ and all the parameters and produce a set of landmark residuals for each image $i$, $\varepsilon^i_m$
    for each image $i$ do
        $\bar{\varepsilon} := \varepsilon^i_m \setminus \max(\varepsilon^i_m)$
        if all $\max(\varepsilon^i_m) \oslash \bar{\varepsilon} > \lambda$ then
            remove the assignment that is associated with $\max(\varepsilon^i_m)$ from $\boldsymbol{c}^k$
        end if
    end for
    if no assignments removed from $\boldsymbol{c}^k$ then
        terminate := true
    else
        $k := k + 1$
    end if
end while
$\boldsymbol{c} := \boldsymbol{c}^k$
```

$\|q_t - \frac{T_s}{2} S_\omega(\omega_t + b_\omega) q_{t-1}\|^2_{R_q^{-1}}$, which is the first order approximation of (1c). Using the bilinear relation $S_\omega(b)q = \tilde{S}_q(q)b$, this can be written as $\|q_t - \frac{T_s}{2} S_\omega(\omega_t) q_{t-1} - \frac{T_s}{2} \tilde{S}_q(q_{t-1}) b_\omega\|^2_{R_q^{-1}}$, which is also a linear function of b_ω, since rotations are assumed to be known. These terms are decoupled from the rest of the parameters and the WLS problem defined by these can be solved separately if required.

3.5 Iterative Outlier Removal

The landmark initialisation produced by Algorithm 4 will introduce erroneous associations due to the unavoidable ambiguity of the feature descriptors. These associations should be considered outliers. It is difficult to discriminate outliers based on descriptors alone. However, given the IMU data, which describe the motion independently of cluster appearance, a strategy for inertial based outlier rejection can be devised according to the pseudo-code in Algorithm 6. This procedure will terminate when all of the residuals are of similar size, where similar is defined here by the rejection threshold λ. The operator $\oslash$ denotes element-wise division.

4 Nonlinear Least-Squares SLAM

Accelerations, initial velocity, landmarks and biases estimated with the linear method described in Section 3.4 are used as an initial value for NLS. Here the reprojection error is formulated in its original form, that is

$$y_t^m = P(\mathsf{R}^{ce}(q_t)(m - p_t)) + e_t \tag{22}$$

where the operator P is defined as in (4). The second addition is that rotations are not fixed any more, but are estimated together with the rest of the parameters. This is done by adding the angular velocities $\omega_{1:N}$ to the parameter set. Now, the measurement relation from Section 2.2 can be used

$$y_t^\omega = \omega_t + b_\omega + e_t^\omega. \tag{23}$$

and the rotations can be calculated by using the relation (1c) as

$$q_t = \left[\prod_{k=1}^{t} \exp\left(\frac{T_s}{2} S_\omega(\omega_k)\right)\right] q_0, \tag{24}$$

where q_0 is assumed given. These two modifications give the new parameter vector $\theta = [a_{1:N}, \omega_{1:N}, v_0, b_a, b_\omega, m]^T$ and the new nonlinear least squares (NLS) formulation of the problem according to

$$\begin{aligned}\hat{\theta} = \arg\min_{\theta} \sum_{t=1}^{N} & \|y_t^a - \mathsf{R}_t^{ce}(a_t - g^e) - b_a\|^2_{R_a^{-1}} + \\ & \|y_t^\omega - \omega_t - b_\omega\|^2_{R_\omega^{-1}} + \\ & \left\| y_t^m - \frac{\delta_{x,y}(v_0, a_{1:t}, \omega_{1:t}, m)}{\delta_z(v_0, a_{1:t}, \omega_{1:t}, m)} \right\|^2_{R_m^{-1}}\end{aligned} \tag{25}$$

and R_t^{ce} is now a function of $\omega_{1:t}$ and δ is defined in (14). This problem can be solved efficiently with e.g., a standard Levenberg-Marquardt solver, (Nocedal and Wright, 2006).

5 Heuristic Motivation of the Linear Initialisation

In tracking and navigation the measurement models are often nonlinear and so are most SLAM systems stemming from e.g., transformations between reference frames by rotations, perspective divide, among others. In practice this means that there exist local minima which should be avoided. In order to reach the global minimum the initialisation point should be in the proximity of the global minimum or at least the function should be monotone between the initial point and the global minimum and even better is of course if it is also convex along this direction. In fact, it is sufficient that the function is convex on a path that the minimisation procedure will take in order to end up in the global minimum. Here we propose an initialisation procedure based on the almost linear method. We will use a simple heuristics to motivate that the initial point created in this manner is better than an initial point created using available measurements.

The definition of convex function $f(\theta) : \mathbb{R}^n \to \mathbb{R}$ is

$$f(\lambda\theta_1 + (1-\lambda)\theta_2) \leq \lambda f(\theta_1) + (1-\lambda)f(\theta_2), \tag{26a}$$

$$0 \leq \lambda \leq 1, \tag{26b}$$

$$\forall \theta_1, \theta_2 \in \textbf{dom } f \subset \mathbb{R}^n. \tag{26c}$$

Geometrically, this is interpreted as the hyperplane that lies between points $(\theta_1, f(\theta_1))$ and $(\theta_2, f(\theta_2))$ is always above the function f. If this is fulfilled for all θ then the function is convex. Convex functions have a property that there is only one global minimum, so any minimisation procedure can be used to obtain that. For example linear least squares problems fall into this category. However, many functions, although non-convex on the whole domain, are convex on a subset of the domain, usually in the proximity of the local minima. This can be motivated with the fact that Taylor expansion around the local minimum will be quadratic function plus a rest term of a higher degree. If this rest term is not dominating over the quadratic term, the function is (locally) convex in this region. As stated before, in order to apply a local minimisation procedure, and successfully obtain the global minimum, the path between the initial point and minimum should also fulfill the convexity property (26).

The main idea is to check the local convexity of the NLS cost function given the initial point produced with the linear initialisation procedure. This can be done approximately by using a dense sampling of the cost function along the search direction as given by the initialisation and then evaluate if the path is convex according to (26). In addition, we also require that the search direction in the initial point, p_0, is well aligned with the direction from the initial point to the true solution, $\theta^* - \theta_0$. The intuition behind this is that an initial search direction is crucial for convergence to a good solution. This is determined with the angle between these two directions defined as

$$\gamma = \arccos\left(\frac{p_0^T(\theta^* - \theta_0)}{\|p_0\|_2 \, \|\theta^* - \theta_0\|_2}\right), \tag{27a}$$

$$p_0 = -(J_0^T J_0)^{-1} J_0^T \varepsilon_0, \tag{27b}$$

where J_0 is a Jacobian matrix of the cost function evaluated in θ_0 and ε_0 is the residual in θ_0. These criteria will be evaluated in a Monte Carlo fashion for the linear initialisation procedure and compared to other initialisation approaches, for example using measurements only.

6 Monte Carlo Simulations

Monte Carlo (MC) simulations are used to evaluate the whole initialisation approach. That is, to find out if the linear method gives a good starting point for the nonlinear optimisation and if the outlier rejection procedure seems reasonable. For the initialisation of the landmarks, i.e., the clustering approach, only real data are used since it is complicated to make simulated SIFT features and these results are presented in Section 7.

Initialisation method	Linear	Naïve
Average initial angle γ [°]	17.7	66.6
Average error $\|\hat{\theta}-\theta^*\|/\mathbf{dim}(\theta^*)$	$1.6 \cdot 10^{-3}$	$221 \cdot 10^{-3}$
# of non convex paths	0	0

Table 1: *MC simulation results for the initialisation method, see Section 6.1, where the measurement noise was varying. The angle γ is defined in* (27a), *$\hat{\theta}$ is defined in* (25) *and θ^* is the true solution.*

6.1 Efficiency of the Linear Initialisation

Here, the method according to Section 3.4 is evaluated. In the first set of simulations a trajectory and a set of landmarks are created and different white Gaussian noise is simulated and added to the measurements for each MC simulation. No outliers are present. Initial rotation used for the linear optimisation is also randomly perturbed but fixed together with the trajectory and the scene. The linear solution produced in this way is compared with the naïve guess where the initial point is created from the measurements and randomised landmark positions.

The first set of MC simulations consist of 50 simulations performed on a data set with 20 landmarks and 30 time points. Landmarks were placed in a general configuration, with varying distance to the camera. The noise standard deviation was fixed and is set to $\sigma_a = 10^{-3}\ \mathrm{m/s^2}$, $\sigma_m = 10^{-4}\ \mathrm{m}$ and $\sigma_\omega = .5°/\mathrm{s}$. The results are listed in Table 1. Both the proposed initialisation method and the naïve one have convex paths between the initial and the true point, but the initial angle between the search direction and the direction to the true solution is much larger (approximately four times) for the measurement initialised optimisation. This causes the average error of the solution to be much larger (approximately 140 times) for the naïve initialisation.

6.2 Sensitivity to Initial Rotation Errors

In the second set, consisting of 3 times 25 realisations, the trajectory and the scene are fixed as before, no noise is added to the measurements and no outliers are present. Here the initial rotations are varied randomly with different perturbation magnitude 0.1, 1 and 5 $[°/\mathrm{s}]$. The solutions to the linear initialisations given different rotation perturbations are then used to solve the non-linear optimisation problem and the results are in Table 2. Here it can be seen that both error in the linear solution as well as initial angle grow with the magnitude of the error in rotation. As a consequence the number of non-convex paths also increases with the perturbation magnitude and the average error of the estimate is large. The reason for the very large average for the perturbation of 1 degree per second is the presence of 4 really bad solutions. With these removed the value is 0.45 which is more reasonable. This evaluation shows the importance of the initial estimation of the rotations.

Perturbation magnitude [°/s]	0.1	1	5
Average initial angle γ [°]	1.64	40.2	84.0
Average error $\|\hat{\theta} - \theta^*\|/\mathbf{dim}(\theta^*)$	$8.34 \cdot 10^{-9}$	69.1	1.0
Average error $\|\theta_{\text{init}} - \theta^*\|/\mathbf{dim}(\theta^*)$	0.08	0.63	0.78
# of non convex paths	0	3	18

Table 2: *MC simulation results for the initialisation method, see Section 6.2, where the initial rotation error was varying. The angle γ is defined in (27a), the NLS estimate $\hat{\theta}$ is defined in (25), the initial estimate θ_{init} is defined in (21) and θ^* is the true solution.*

6.3 Iterative Outlier Removal

The third set is used to evaluate the performance of the outlier rejection method proposed in Algorithm 6. Here the outlier measurement rate is varied while the trajectory, scene and the initial rotations are fixed. The performance is evaluated according to the amount of outlier measurements that are left and the amount of inlier measurements that are removed.

Results are presented in Table 3, in this case the scene is fixed to 30 landmarks in 21 images and the outlier rate is varied, both as a number of landmark outliers and as a number of images where outliers are present. The outliers are created by randomly flipping a pair of associations in an image. For example, if the number of landmarks to be outliers are two, then in all images where outliers are present the measurements from these two landmarks are flipped with two other randomly chosen landmarks. This strategy is chosen because it is the behaviour of the data association method employed here. Three different rejection thresholds are used, $\lambda = \{3, 5, 8\}$, in the experiments. The result shows that, as a general trend and as expected, the rejection threshold governs the amount of outliers that are left and number of inliers that are rejected. It is of general interest to reject as many outliers as possible and to keep as many inliers as possible. The lower threshold means better outlier rejection, but the price is that more inliers are also rejected. This also emphasises the importance of having many landmark measurements in order to be resilient to removing inliers. Note that this statistics is conservative in the way that only measurements that are created as outliers are considered as true outliers. In many cases all measurements for a landmark that is creating outliers are removed, implying that this landmark is no longer deemed as usable and it can be excluded from the statistics. This means that results in Table 3 would be somewhat better if these are taken into account.

7 Real Data Experiments

In the real data experiments, a sensor unit, see Figure 4, equipped with monocular monochrome VGA camera (Pointgrey firefly) and three axis inertial sensors (gyroscopes and accelerometers) was used. It contains also magnetometers and a temperature sensor which is used for internal calibration. In one of the experiments the sensor unit was mounted at the tool position of an IRB-1400 industrial robot from ABB for the purpose of an accu-

$\lambda = 3$	Outliers left [%]			Inliers removed [%]		
IO \ LO	0.07	0.13	0.2	0.07	0.13	0.2
0.05	0	0	0	13	22	22
0.10	0	0	0	18	23	31
0.14	0	0	0	17	26	41
0.19	0	6.3	0	29	36	40
0.24	0	0	0	20	20	52
0.29	0	0	0	20	46	48
0.33	0	0	2.4	23	44	52
0.38	6.3	0	2.1	42	50	54

$\lambda = 5$	Outliers left [%]			Inliers removed [%]		
IO \ LO	0.07	0.13	0.2	0.07	0.13	0.2
0.05	0	50	0	9.1	12	14
0.10	0	0	0	11	21	32
0.14	0	8.3	5.6	17	25	42
0.19	0	0	0	21	34	38
0.24	0	5	0	19	25	54
0.29	0	0	0	16	36	49
0.33	7	0	0	21	34	51
0.38	6.3	0	0	39	35	54

$\lambda = 8$	Outliers left [%]			Inliers removed [%]		
IO \ LO	0.07	0.13	0.2	0.07	0.13	0.2
0.05	0	0	8.3	8.1	9.2	15
0.10	0	0	0	17	20	27
0.14	0	0	11	16	18	26
0.19	0	0	0	15	25	35
0.24	0	5	10	22	34	58
0.29	8.3	0	0	16	25	42
0.33	0	0	2.4	27	42	38
0.38	6.3	0	8.3	34	31	66

Table 3: *Outlier rejection simulation results. IO is the fraction of images in which outliers are present. LO is the fraction of the landmark set in which outliers are present. Three different rejection thresholds* λ *are used.*

Figure 4: *Sensor unit containing both camera and inertial sensors.*

rately known ground truth. Also, a small scene with objects of known size was created so that the estimated scene could be compared with respect to its size. In the second experiment, a free-hand movement of the camera is used in a room, and no accurate ground truth is available. In this case the accuracy is evaluated based on the approximate measurements in the room and approximately "knowing" where camera was. Prior to usage, the camera was calibrated using the toolbox (Bouguet, 2010) and the relative pose of the camera centre with respect to the IMU centre was calibrated as described in Hol et al. (2010). A open source SIFT implementation from Vedaldi and Fulkerson (2008) was used.

In Figure 6 the trajectory estimate from the linear initialisation and the nonlinear refinement are plotted together with the ground truth trajectory for the first data set. An image with the feature measurements is illustrated in Figure 5. Velocity in x, y-plane is shown in Figure 7. Since the true rotations are known, errors in quaternions and Euler angles are depicted in Figure 8 and Figure 9 respectively. Landmark estimates for initial and nonlinear estimation are shown in Figure 10.

For the second data set (free-hand run), an example image together with plotted features is showed in Figure 11. The resulting trajectory estimate is shown in Figure 12. The x, y-plane velocity, quaternions, Euler angles and landmark estimates are plotted in Figure 13, Figure 14, Figure 15 and Figure 16 respectively.

In both cases the nonlinear refinement improves the initial estimate for both rotations and kinematic parameters. In the first data set the landmark estimate is much better than the initial estimate, and for the second data set this is harder to assess, but the positions of the landmarks look reasonable given the environment. For example we see that there are three levels in the bookshelf with distinct features, which can be seen in both Figure 11 and Figure 16.

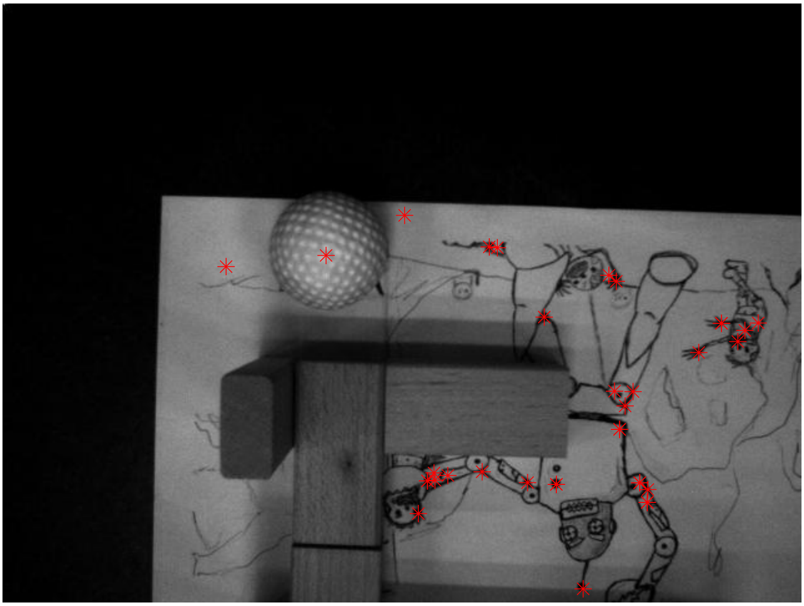


Figure 5: Example image from the robot run with extracted features shown as red stars.

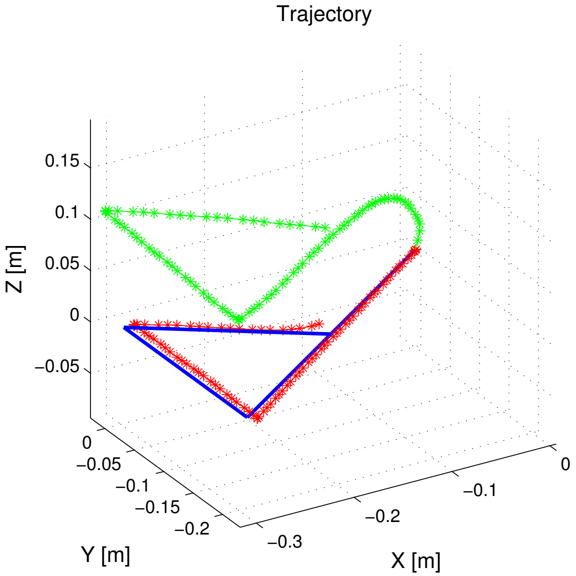


Figure 6: Trajectory from the linear estimation (green), from nonlinear refinement (red) and ground truth (blue).

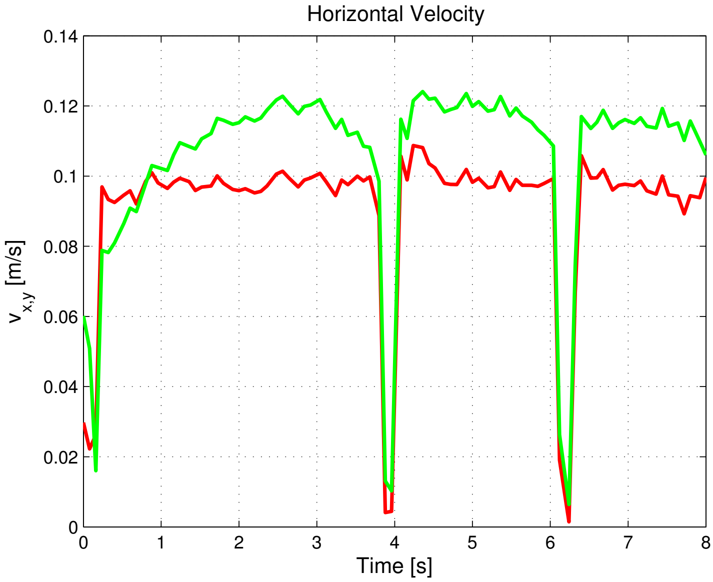


Figure 7: *Estimated horizontal velocity from the linear estimation (green) and from the nonlinear refinement (red). True velocity is* $0.1\ \mathrm{m/s}$ *except in three time points when it is zero.*

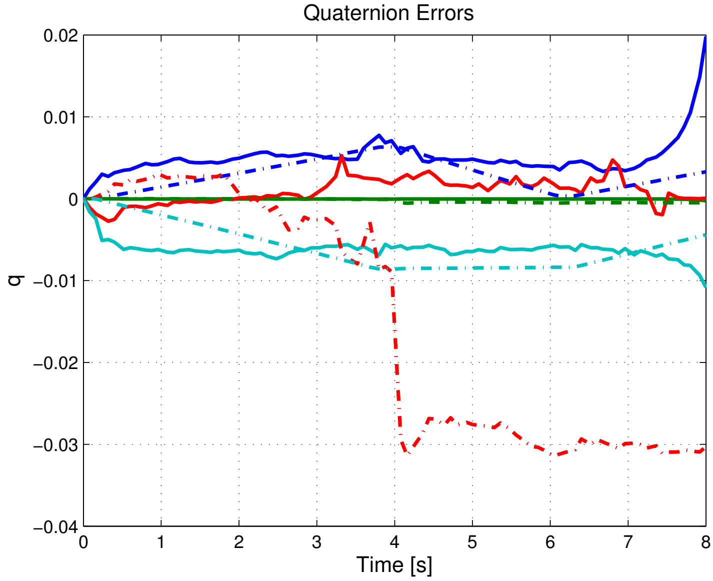


Figure 8: *Estimated quaternion error from camera (dash-dotted) and after nonlinear refinement (solid).*

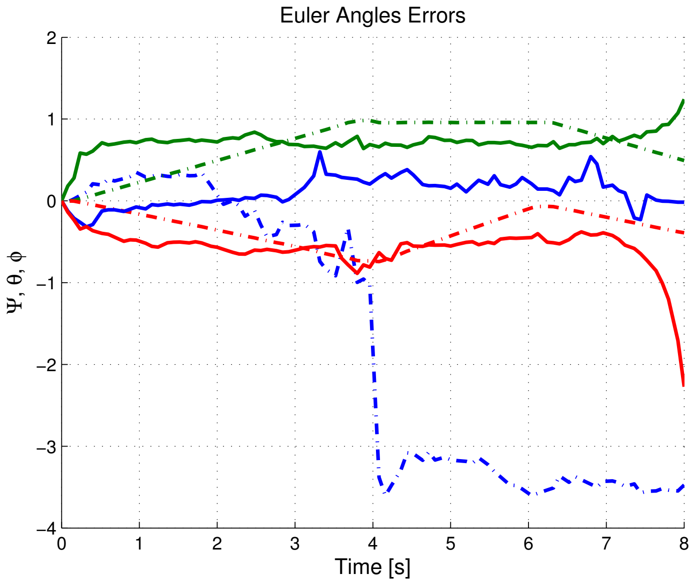


Figure 9: *Estimated Euler angles error in degrees from camera (dash-dotted) and after nonlinear refinement (solid). Blue is the yaw, green is the pitch and red is the roll angle.*

7.1 Clustering Results

For the landmark initialisation approach (track clustering) a helicopter data set made with a Yamaha Rmax helicopter is used. In this data set only the images were available, i.e., no IMU, implying that no outlier rejection nor complete SLAM could be done and this data set is only used to evaluate the clustering performance. The flight is performed in a circle, meaning that the helicopter visits the same place. The total length of one such loop was 325 images taken in 4 Hz giving the total time of 81.25 s. An example of the loop-closure is illustrated in Figure 17 where two tracks, one from the beginning and one from the end of the loop, are clustered together. With this cluster tuning the amount of loop-closures is about 4%. This should be compared to the amount of images showing the overlapping environment which were 7% of the total amount of images. For the other two data sets this number is higher, especially for the free-hand run data set, where large parts of the environment were visible the whole time. In general, most of the clusterings are local loops where for example a feature is lost for a few images and is then tracked again. One such example can be seen in Figure 18.

Even some outliers are introduced in the clustering process, which is, as explained earlier, expected, and one such is depicted in Figure 19. The total amount of outlier landmark was about 10%, which is similar for the other two data sets. Many of the outliers are caused by the too similar environment, for the example in Figure 19 the road edge looks similar to SIFT and the descriptors are too similar. This will cause the erroneous clustering.

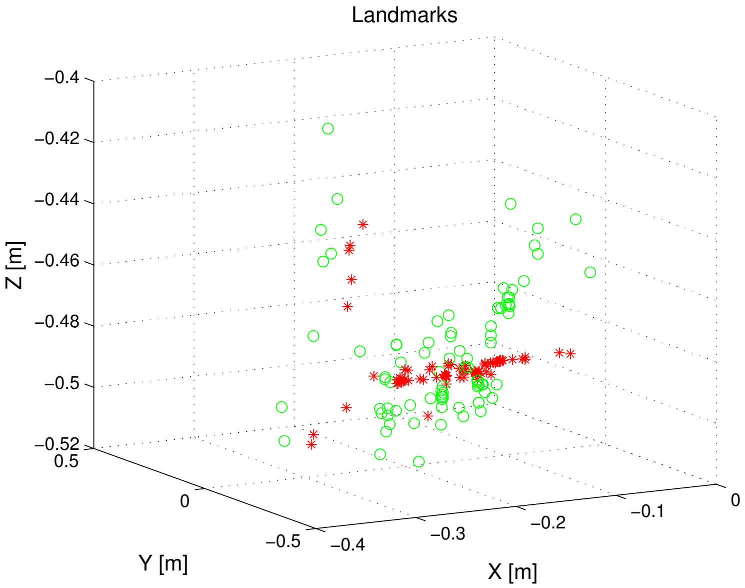


Figure 10: *Landmark estimates from the linear estimation (green) and from non-linear refinement (red). Most of the landmarks should lie on the* -0.5 m *plane and some should be higher up.*

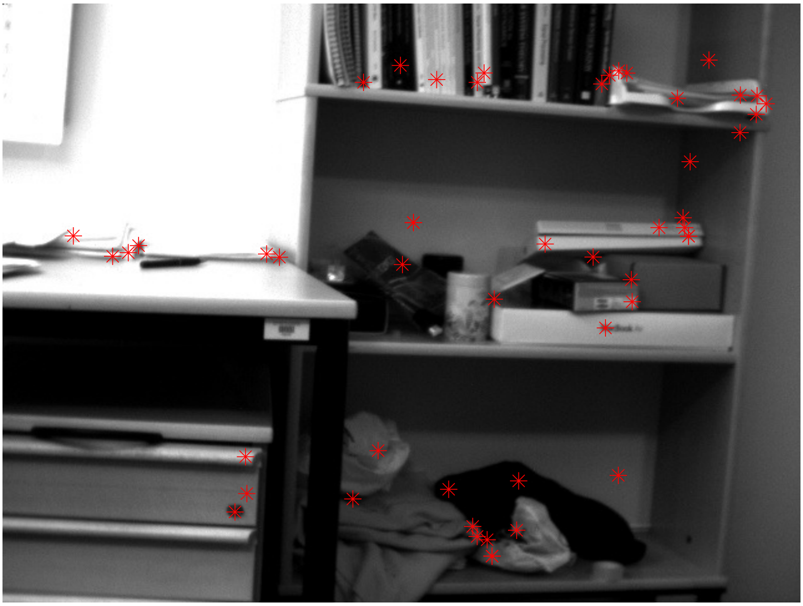


Figure 11: *Example image from the free-hand run with extracted features shown as red stars.*

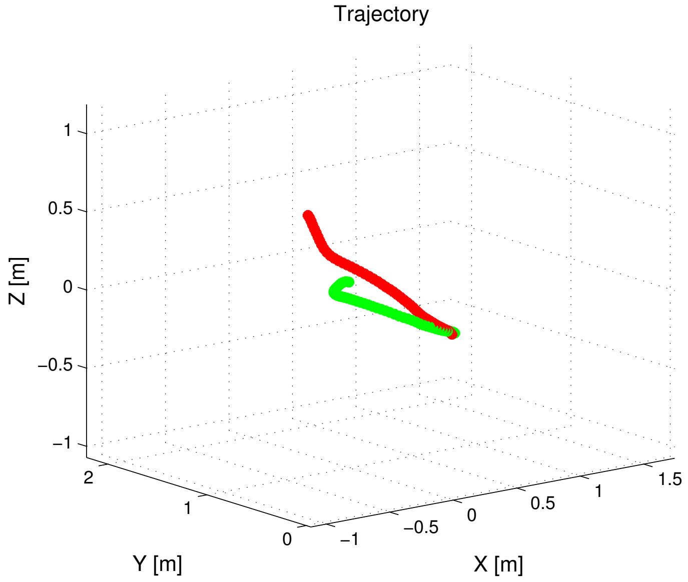


Figure 12: *Trajectory from the linear estimation (green), from nonlinear refinement (red).*

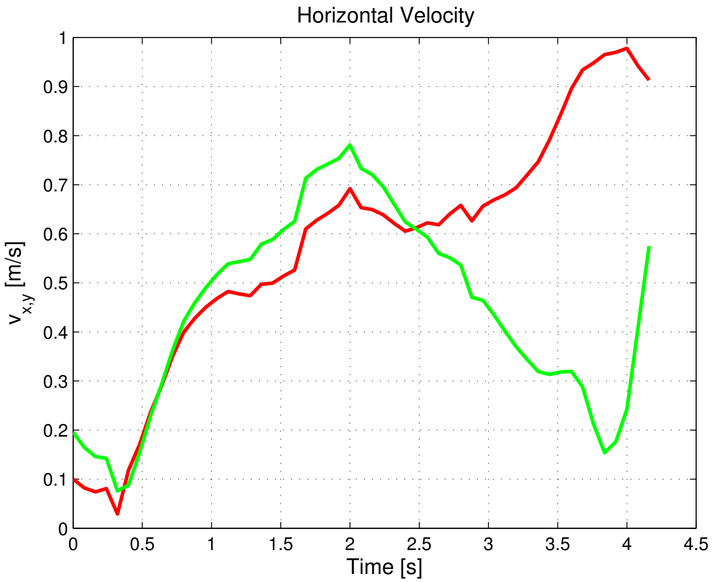


Figure 13: *Estimated horizontal velocity from the linear estimation (green) and from the nonlinear refinement (red).*

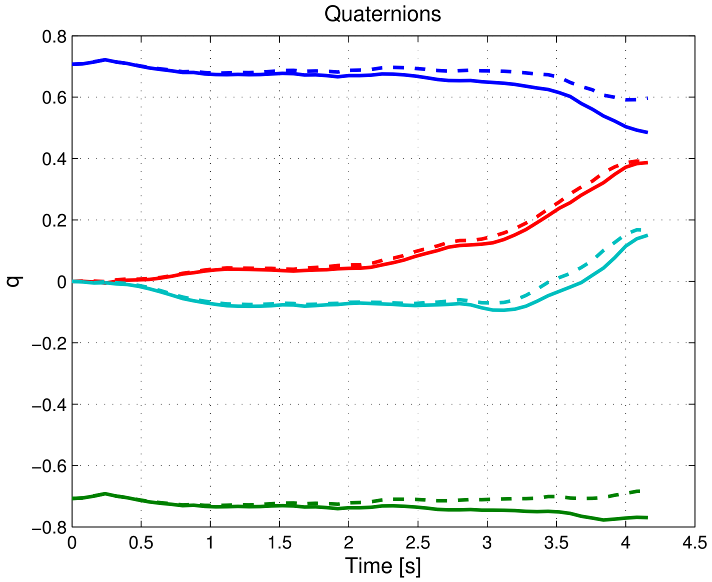


Figure 14: *Estimated rotations from camera (dashed) and after nonlinear refinement (solid).*

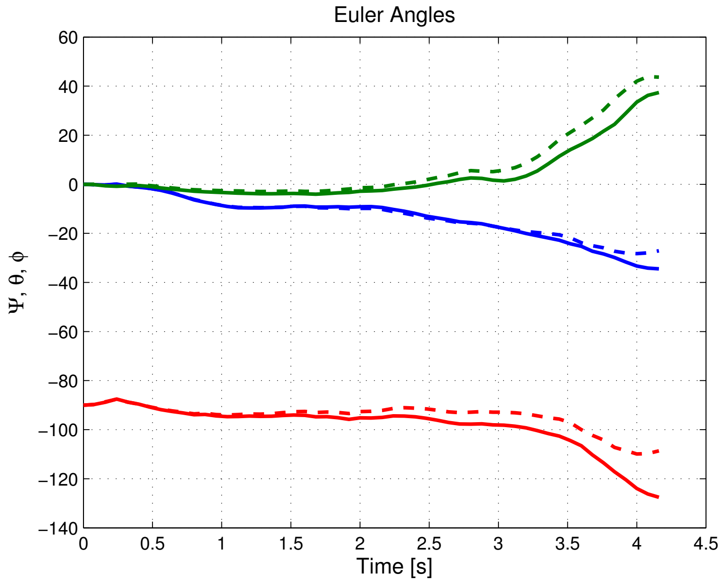


Figure 15: *Estimated Euler angles in degrees from camera (dash-dotted) and after nonlinear refinement (solid). Blue is the yaw, green is the pitch and red is the roll angle.*

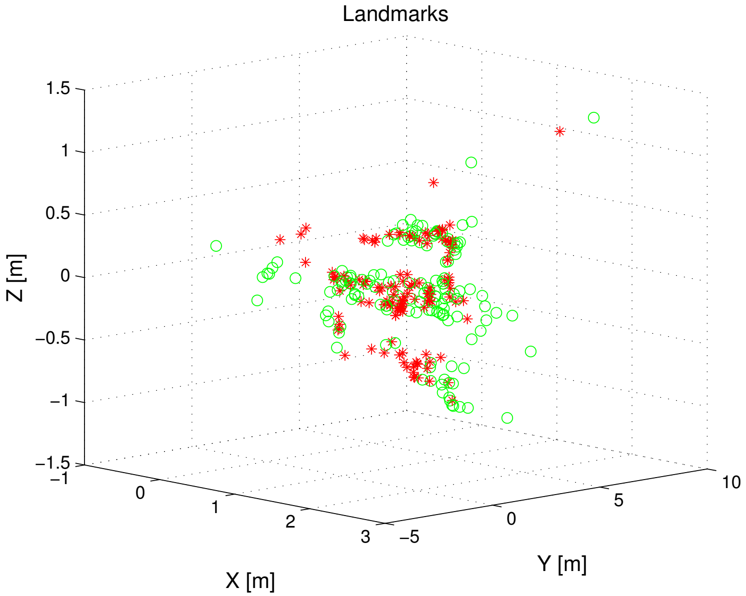


Figure 16: *Landmark estimates from the linear estimation (green) and from nonlinear refinement (red). Three distinct layers can be recognised which basically come from the bookshelf's levels.*

8 Conclusions and Future Work

In this work we presented a method for initialisation of optimisation based visual/inertial SLAM on batch form. This sensor combination makes it possible to obtain full Euclidian reconstruction of the environment and trajectory. The method is based on a multistage strategy where visual methods, such as the Eight-Point Algorithm, feature extraction and clustering of feature tracks, are used for rotation and landmark initialisation. Inertial data are used for data association including outlier rejection and initialisation of trajectory and landmark location parameters. The method exploits the conditional linearity of visual/inertial SLAM. The experiments done on the simulated and real data sets show that the initialisation method gives better starting point for the subsequent full nonlinear optimisation than naïve initialisation with measurements only.

Also, the landmark initialisation method based on clustering of the tracked features gives quite promising results where many possible loop-closures are identified while the amount of wrong associations is rather low. This allows for the iterative outlier rejection method with aid from the inertial data. Even this method shows good results with efficient outlier removal while keeping the inlier amount relatively high.

It must be pointed out that this approach requires a large amount of landmark measurements in order to produce good results, i.e., the equation system must be highly overdetermined. On top of that, since a camera is a bearings-only sensor, there is also a demand

(a) *Image 7.* ***(b)*** *Image 8.*

(c) *Image 9.* ***(d)*** *Image 321.*

(e) *Image 322.* ***(f)*** *Image 323.*

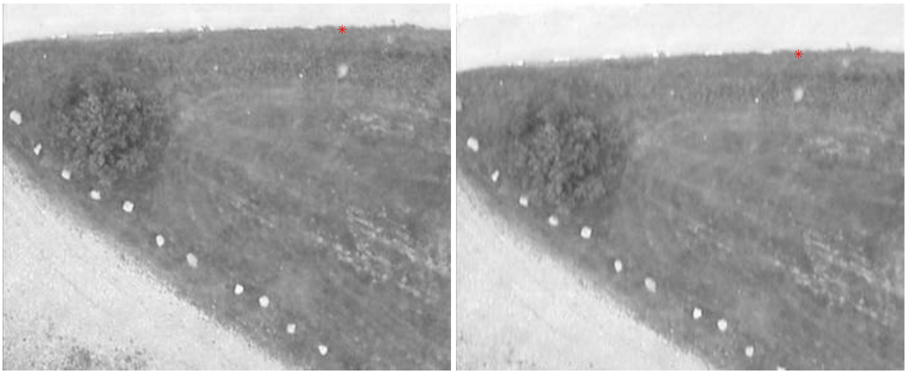
(g) *Image 324.* ***(h)*** *Image 325.*

Figure 17: *Example of successful loop-closure clustering with the helicopter data. The tracked feature is marked with the red star.*

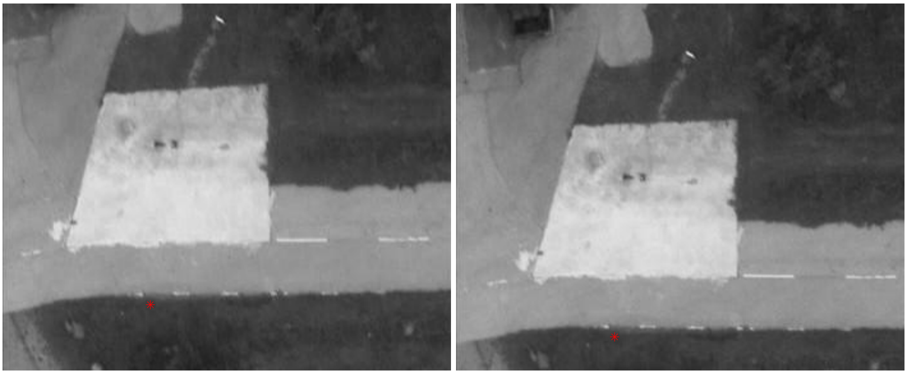
(a) *Image 3.* ***(b)*** *Image 4.*

(c) *Image 12.* ***(d)*** *Image 13.*

Figure 18: *Example of clustering creating local loops with the helicopter data. The tracked feature is marked with the red star.*



(a) *Image 299.* ***(b)*** *Image 300.*



(c) *Image 314.* ***(d)*** *Image 315.*

Figure 19: *Example of erroneous clustering causing an outlier with the helicopter data. The tracked feature is marked with the red star.*

for sufficient viewpoint change, also known as parallax, in order to accurately estimate landmark position. The trajectory estimation can be compensated with the inertial data, but even there a good SNR is required.

In the future it can be interesting to use proper constrained clustering algorithm instead of discarding clusters with overlapping times, as it is done now. Also alternative feature detectors might be used to see if descriptors from those have different behaviour compared to SIFT.

Bibliography

Y. Abdel-Aziz and H. Karara. Direct Linear Transformation from Comparator to Object Space Coordinates in Close-Range Photogrammetry. In *Proceedings of the American Society of Photogrammetry Symposium on Close-Range Photogrammetry*, pages 1–18, Falls Church, VA, USA, 1971.

S. Agarwal, N. Snavely, and S. M. Seitz. Fast algorithms for L_∞ problems in multiview geometry. In *Computer Vision and Pattern Recognition*, pages 1–8, 2008. doi: 10.1109/CVPR.2008.4587713.

S. Agarwal, N. Snavely, I. Simon, S.M. Seitz, and R. Szeliski. Building Rome in a Day. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 72 –79, oct 2009. doi: 10.1109/ICCV.2009.5459148.

T. Bailey and H. Durrant-Whyte. Simultaneous Localization and Mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, September 2006.

D. P. Bertsekas. An Auction Algorithm for Shortest Paths. *SIAM Journal on Optimization*, 1(4):425–447, 1991. ISSN 10526234. doi: DOI:10.1137/0801026. URL `http://dx.doi.org/doi/10.1137/0801026`.

Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, 1996. ISBN 0-89871-360-9.

J.-Y. Bouguet. Camera Calibration Toolbox for Matlab, 2010. URL `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

M. Chli. *Applying Information Theory to Efficient SLAM*. PhD thesis, Imperial College London, 2009.

M. Cummins and P. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, 2010. doi: 10.1177/0278364910385483. URL `http://ijr.sagepub.com/content/early/2010/11/11/0278364910385483.abstract`.

F. Dellaert and M. Kaess. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *International Journal of Robotics Research*, 25(12):1181–1203, 2006. ISSN 0278-3649. doi: 10.1177/0278364906072768. URL `http://dx.doi.org/10.1177/0278364906072768`.

H. Durrant-Whyte and T. Bailey. Simultaneous Localization and Mapping: Part I. *IEEE Robotics & Automation Magazine*, 13(12):99–110, June 2006.

A. W. Fitzgibbon and A. Zisserman. Automatic Camera Recovery for Closed or Open Image Sequences. In *ECCV (1)*, pages 311–326, 1998.

G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard. Efficient estimation of accurate maximum likelihood maps in 3D. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3472–3478, 2007.

Gurobi Optimization Inc. Gurobi optimizer reference manual, 2013. URL `http://www.gurobi.com`.

R. Hartley and F. Schaffalitzky. L_∞; minimization in geometric reconstruction problems. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–504 – I–509 Vol.1, june-2 july 2004. doi: 10.1109/CVPR.2004.1315073.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. ISBN 0-521-54051-8.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2 edition, 2009. ISBN 0387952845.

K. L. Ho and P. M. Newman. Loop closure detection in SLAM by combining visual and spatial appearance. *Robotics and Autonomous Systems*, 54(9):740–749, 2006.

J. Hol, T. B. Schön, and F. Gustafsson. Modeling and Calibration of Inertial and Vision Sensors. *The international journal of robotics research*, 29(2):231–244, 2010.

S.-H. Jung and C. J. Taylor. Camera trajectory estimation using inertial sensor measurements and structure from motion results. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.*, volume 2, pages II–732–II–737 vol.2, 2001. doi: 10.1109/CVPR.2001.991037.

F. Kahl. Multiple view geometry and the L_∞-norm. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1002 – 1009 Vol. 2, oct. 2005. doi: 10.1109/ICCV.2005.163.

F. Kahl and D. Henrion. Globally Optimal Estimates for Geometric Reconstruction Problems. *Int. J. Comput. Vision*, 74(1):3–15, August 2007. ISSN 0920-5691. doi: 10.1007/s11263-006-0015-y. URL `http://dx.doi.org/10.1007/s11263-006-0015-y`.

L. Kneip, M. Chli, and R. Siegwart. Robust Real-Time Visual Odometry with a Single Camera and an IMU. In *Proceedings of the British Machine Vision Conference*, pages 16.1–16.11. BMVA Press, 2011a. ISBN 1-901725-43-X. http://dx.doi.org/10.5244/C.25.16.

L. Kneip, A. Martinelli, S. Weiss, D. Scaramuzza, and R. Siegwart. Closed-Form Solution for Absolute Scale Velocity Determination Combining Inertial Measurements and a Single Feature Correspondence. In *International Conference on Robotics and Automation*, Shanghai, China, May 2011b. URL `http://hal.inria.fr/hal-00641772`.

H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. ISSN 1931-9193. doi: 10.1002/nav.3800020109. URL `http://dx.doi.org/10.1002/nav.3800020109`.

D. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, pages 1150–1157, Corfu, Greece, 1999.

T. Lupton and S. Sukkarieh. Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions. *Robotics, IEEE Transactions on*, 28 (1):61 –76, feb. 2012. ISSN 1552-3098. doi: 10.1109/TRO.2011.2170332.

M. Bryson and M. Johnson-Roberson and S. Sukkarieh. Airborne smoothing and mapping using vision and inertial sensors. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 3143–3148, Kobe, Japan, 2009. IEEE Press. ISBN 978-1-4244-2788-8.

A. Martinelli. Vision and IMU Data Fusion: Closed-Form Solutions for Attitude, Speed, Absolute Scale, and Bias Determination. *Robotics, IEEE Transactions on*, 28(1):44 –60, feb. 2012. ISSN 1552-3098. doi: 10.1109/TRO.2011.2160468.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006. ISBN 978-0-387-40065-5.

C. Olsson and F. Kahl. Generalized Convexity in Multiple View Geometry. *Journal of Mathematical Imaging and Vision*, 38:35–51, 2010. ISSN 0924-9907. URL `http://dx.doi.org/10.1007/s10851-010-0207-5`. 10.1007/s10851-010-0207-5.

P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modelling. In *5th International Symposium on Robotics Research*, pages 207–216, Tokyo, Japan, 1989.

C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: algorithms and complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982. ISBN 0-13-152462-3.

K. Sim and R. Hartley. Removing Outliers Using The L_∞ Norm. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 485 – 494, june 2006. doi: 10.1109/CVPR.2006.253.

S. N. Sinha, D. Steedly, and R. Szeliski. A Multi-staged Linear Approach to Structure from Motion. In *In RMLE-ECCV workshop, 2010.*, 2010.

M. A. Skoglund, Z. Sjanic, and F. Gustafsson. Initialisation and Estimation Methods for Batch Optimisation of Inertial/Visual SLAM. *Submitted to IEEE Transactions on Robotics*, September 2013.

C. J. Taylor, D. Kriegman, and P. Anandan. Structure and Motion in Two Dimensions from Multiple Images: A Least Squares Approach. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 242 –248, Princeton, NJ, USA, October 1991.

T. Thormählen, N. Hasler, M. Wand, and H.-P. Seidel. Merging of Feature Tracks for Camera Motion Estimation from Video. In *5th European Conference on Visual Media Production (CVMP 2008)*, pages 1 – 8, Hertfordshire, UK, jan 2008. IET.

S. Thrun and M. Montemerlo. The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *International Journal on Robotics Research*, 25(5): 403–430, 2006.

B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000.

A. Vedaldi and B. Fulkerson. VLFeat: An Open and Portable Library of Computer Vision Algorithms, 2008. URL `http://www.vlfeat.org/`.

Z. Zhang and T. Kanade. Determining the Epipolar Geometry and its Uncertainty: A Review. *International Journal of Computer Vision*, 27:161–195, 1998.

Paper F

Expectation-Maximisation Maximum Likelihood Estimation for Inertial/Visual SLAM

Authors: Zoran Sjanic, Martin A. Skoglund and Fredrik Gustafsson

Edited version of the paper:

Z. Sjanic, M. A. Skoglund, and F. Gustafsson. Expectation-Maximisation Maximum Likelihood Estimation for Inertial/Visual SLAM. *Submitted to IEEE Transactions on Robotics*, September 2013b.

Expectation-Maximisation Maximum Likelihood Estimation for Inertial/Visual SLAM

Zoran Sjanic, Martin A. Skoglund and Fredrik Gustafsson

Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
{zoran,ms,fredrik}@isy.liu.se

Abstract

The general Simultaneous Localisation and Mapping (SLAM) problem aims at estimating the state of a moving platform simultaneously with building a map of the local environment. There are essentially three classes of algorithms. EKF-SLAM and FastSLAM solve the problem on-line, while Nonlinear Least Squares (NLS) is a batch method. All of them scales badly with either the state dimension, the map dimension or the batch length. We investigate the EM algorithm for solving a generalised version of the NLS problem. This EM-SLAM algorithm solves two simpler problems iteratively, hence it scales much better with dimensions. The iterations switch between state estimation, where we propose an Extended Rauch-Tung-Striebel smoother, and map estimation, where a quasi-Newton method is suggested. The proposed method is evaluated in real experiments and also in simulations on a platform with a monocular camera attached to an intertial measurement unit. It is demonstrated to produce lower RMSE than with a standard Levenberg-Marquardt solver of NLS problem, at a computational cost that increases considerably slower.

1 Introduction

The aim in Simultaneous Localisation and Mapping (SLAM) is to estimate a moving platform's position and orientation while mapping the observed environment. In the seminal work of Smith et al. (1990) the idea of a stochastic map was presented and was first used in P. Moutarlier and R. Chatila (1989), where the estimate is computed with an Extended Kalman Filter (EKF) and some later ones are J. E. Guviant and E. M. Nebot (2001); J. J. Leonard and H. Jacob and S. Feder (2000). Another popular approach is the FastSLAM method, (Montemerlo et al., 2002, 2003), which uses particle filters which are known to handle nonlinearities very well. However, these approaches have some downsides, like constantly increasing covariance matrix size for the EKF-SLAM, making it computationally infeasible for large data sets. Additionally, linearisation errors can contribute to the inconsistencies in the estimates, (Bailey et al., 2006a). However, FastSLAM estimates can also be inconsistent due to particle depletion (Bailey et al., 2006b) and FastSLAM also scales badly with the state dimension of the moving platform. Recently, some approaches based on optimisation, and Nonlinear Least-Squares (NLS) in particular, have been proposed. These methods solve batch problems, i.e., a smoothed estimate is obtained, but for online applications a moving horizon strategy could be applied. A downside of NLS is the quadratic scaling in batch length. Popular approaches applying optimisation are e.g., GraphSLAM, (Thrun et al., 2005) and Square Root Smoothing and Mapping, $\sqrt{\text{SAM}}$, (Dellaert and Kaess, 2006). A common denominator of these approaches is that they solve some form of Maximum Likelihood (ML) problem. Also it is usual that the sparsity of the problems is utilised enabling fast computations.

Here, we propose a formulation of the SLAM problem in a ML form and use an approach based on the Expectation Maximisation algorithm, (Dempster et al., 1977). In an EM setting, so called latent, or hidden, variables are introduced in order to solve ML problems that can be difficult. This is achieved by splitting the problem into two simpler problems, one where expectation with respect to the conditional density of the latent variables has to be calculated and one where a certain function needs to be maximised with respect to the parameters. These two steps are then repeated until convergence. This motivates the name of the method. In EM-SLAM, the map is viewed as the unknown parameter and the platform states, such as position and orientation, are considered to be the latent variables. As a simplified and intuitive motivation for this separation we can consider two simpler problems; one with known map and the other one with known trajectory and orientation. The first problem is then simply the navigation problem with known landmarks. The second problem is known as the triangulation problem, i.e., finding the landmark positions given the known platform positions and camera observations. See e.g., Hartley and Sturm (1997) for an example of triangulation application. Each of these problems are rather straightforward to solve separately but hard to solve combined. By separating the variables in the proposed way we basically split the SLAM problem into the above-mentioned two simpler problems. In the first of these problems some approximations are necessary in order to implement the algorithm. In the conditional expectation step the latent variables are assumed to have Gaussian distribution and that they can be well approximated with an Extended Rauch-Tung-Striebel (E-RTS) smoother, (Rauch et al., 1965). The maximisation step is solved using a quasi-Newton method. The proposed method is compared with the NLS formulation which can be seen as a straightforward ML formulation where both the state sequence and the map are seen as parameters. This reference method is solved using the Levenberg-Marquardt algorithm, (Nocedal and Wright, 2006). The comparison is done for both the performance of the estimation and for the complexity of each approach.

SLAM is a general class of problems where the combination of sensors vary and one sensor which have gained in popularity is the camera. Methods based on camera only has been known in the computer vision society as the structure from motion (SfM) problem for quite some time, see e.g., Fitzgibbon and Zisserman (1998); Taylor et al. (1991). The structure and motion recovered from SfM will have unknown universal scale, since the camera suffers from the depth ambiguity problem. In other words, given a motion of the camera, we cannot say if our velocity was large and the scene was far away or if the velocity was low and the scene was close to the camera. One way to solve this problem and to resolve the universal scale is to add some kind of velocity measurement, and the inertial measurement unit (IMU) measuring accelerations and angular velocities is one such way (M. Bryson and M. Johnson-Roberson and S. Sukkarieh, 2009; Kneip et al., 2011a,b; Martinelli, 2012; Lupton and Sukkarieh, 2012). The combination of inertial sensors measuring linear accelerations and angular velocities and monocular camera for observation of the environment is particular challenging from a complexity point of view. For that reason, both the simulations and real data experiments are based on a monocular camera and inertial sensors.

The paper is organised as follows; in Section 2 the Expectation Maximisation algorithm is explained more detailed and application to SLAM is described in Section 3. The dynamical and measurement models specific to visual/inertial SLAM are introduced in Section 4 and an alternative method of solving ML SLAM problem, NLS, is explained in Section 5. Comparison between EM-SLAM and NLS-SLAM is discussed in Section 6, and a brief explanation about obtaining an initial estimate for the landmarks is given in Section 7. Finally, results, conclusions and future work are discussed in Section 8 and Section 9.

2 Expectation Maximisation

Maximum Likelihood in its basic form is a batch method which takes a set of observations $\mathcal{Y} = \{y_1, \ldots, y_N\}$, where the index denotes time, and aims at finding the maximum likelihood (ML) estimate of the parameters θ from the measurement likelihoods as

$$\hat{\theta}^{ML} = \arg\max_{\theta} p_\theta(\mathcal{Y}), \tag{1}$$

which can be solved by considering minimising the sum of the negative measurement log-likelihoods. Naturally, $p_\theta(\mathcal{Y})$ is the probability density function parametrised by the unknown θ. Often, the maximisation of (1) can be very difficult and the key idea with Expectation Maximisation is to consider the joint density $p_\theta(\mathcal{Y}, \mathcal{X})$, where $\mathcal{X} = \{x_1, \ldots, x_N\}$ are latent variables. Then, by splitting this density into two coupled, and hopefully easier, problems the parameters and the latent variables can be solved for in an iterative manner. The first step is the *Expectation step*, commonly denoted *E-step*, where the expectation of the joint log-likelihood, $\log p_\theta(\mathcal{Y}, \mathcal{X})$, with respect to the density of the latent variable conditioned on all the measurements, $p_{\theta_k}(\mathcal{X}|\mathcal{Y})$, is computed. The expectation $\mathbf{E}_{\theta_k}\{\log p_\theta(\mathcal{Y}, \mathcal{X})|\mathcal{Y}\}$ will be a function, called $\mathcal{Q}(\theta, \theta_k)$, of the parameter vector θ as

$$\mathcal{Q}(\theta, \theta_k) = \int p_{\theta_k}(\mathcal{X}|\mathcal{Y}) \log p_\theta(\mathcal{Y}, \mathcal{X})\, \mathrm{d}\mathcal{X}. \tag{2}$$

Note that the conditional density of the latent variables, $p_{\theta_k}(\mathcal{X}|\mathcal{Y})$, is computed using the previous estimate of the parameters, θ_k, which is also emphasised in the notation. In the *Maximisation step* or *M-step*, the $\mathcal{Q}$ function obtained in the E-step is maximised with respect to the parameter θ obtaining new estimate θ_{k+1}. These two steps are repeated until some convergence criterion is met, usually when the change in the parameter or likelihood value is below a certain threshold. For an explanation of the EM algorithm applied to dynamical systems see e.g., Schön (2009), and how it can be used in system identification is exemplified in Wills et al. (2010), where a particle smoother is used to calculate the conditional expectation in the E-step. In Ghahramani and Roweis (1999); Duncan and Gyongy (2006) nonlinear dynamical models are treated using EM where the E-step is calculated using an Extended Kalman smoother, which is the same approach that will be used here. All these EM variants are formulated as batch methods, but there are also online EM methods which typically use sequential Monte Carlo and stochastic approximation methods (Ozkan et al., 2012; Le Corff et al., 2011). However in these online approaches, certain criterion must be fulfilled on the models involved in order to have working estimators. As will be shown, in our case these are not met which will require different approximations to be applied.

3 EM-SLAM

A common way of formulating visual/inertial SLAM problem is to define a state space model as

$$x_t = f(x_{t-1}, u_t, w_t), \tag{3a}$$

$$y_t = h_t(x_t, \theta) + e_t, \tag{3b}$$

where the measurement noise, e_t, is considered white and Gaussian with mean zero and covariance R while the process noise, w_t, is considered white with mean zero and covariance Q. f describes the state transition function and u_t are considered to be inputs given by the inertial sensors from which pose and velocity, $x = [p, v, q]^T$, are computed. The measurements y_t here are the camera measurements i.e., features extracted from images and h is the measurement function relating measurements, states and parameters. The parameter vector θ, consists of landmark coordinates in three dimensions. These models will be defined in detail in Section 4. The most significant difference, as opposed to traditional SLAM state space model formulation, is that the map is seen as a parameter which parametrises the measurement equation and in turn, the measurement likelihood function. This formulation is quite natural since the map is time independent and is naturally seen as a parameter and not part of the state vector. Furthermore, the conditional expectation step is assumed to be well approximated by an Extended Rauch-Tung-Striebel (E-RTS) smoother. E-RTS is a straightforward modification of the standard RTS smoother, (Rauch et al., 1965), by using the Extended Kalman Filter instead of the Kalman Filter in the forward filtering step, while the backward smoothing step is the same as in the original RTS smoother.

The state space formulation above constitutes the basis for the ML formulation that is naturally put into EM setting, i.e., it is straightforward to define the joint likelihood $p_\theta(\mathcal{Y}, \mathcal{X})$. Here the platform states, $\mathcal{X}$, are considered to be latent variables. By using the Markov properties this density can be written as

$$p_\theta(\mathcal{Y}, \mathcal{X}) = \prod_{t=1}^{N} p_\theta(y_t | x_t) p(x_t | x_{t-1}). \tag{4}$$

Notice that the process model does not depend explicitly on the parameter θ, which will simplify the calculations significantly as it will be shown in the next section.

Next, both E-step and M-step will be explained in detail with all the derivations and approximations used.

3.1 E-step

Given the joint likelihood from (4) the expectation step gets the following form

$$\mathcal{Q}(\theta,\theta_k) = \mathbf{E}_{\theta_k}\left\{\log\left[\prod_{t=1}^{N} p_\theta(y_t|x_t)p(x_t|x_{t-1})\right]\middle|\mathcal{Y}\right\}, \tag{5}$$

where the measurement likelihood is given by the PDF

$$p_\theta(y_t|x_t) = p_\theta(e_t) = p_\theta(y_t - h_t(x_t,\theta)), \tag{6}$$

and the state transition density, $p(x_t|x_{t-1})$, does not depend on θ. Assuming that the likelihood has Gaussian distribution the expectation (5) becomes

$$\begin{aligned}\mathcal{Q}(\theta,\theta_k) &= \text{const.} - \mathbf{E}_{\theta_k}\left\{\sum_{t=1}^{N}\frac{1}{2}\|y_t - h_t(x_t,\theta)\|^2_{R_t^{-1}} + \log p(x_t|x_{t-1})\middle|\mathcal{Y}\right\}\\ &= \text{const.} \ -\sum_{t=1}^{N}\mathbf{E}_{\theta_k}\left\{\frac{1}{2}\|y_t - h_t(x_t,\theta)\|^2_{R_t^{-1}}\middle|\mathcal{Y}\right\}\end{aligned} \tag{7}$$

where all the terms not depending on θ are lumped into a constant term, which will not affect the optimisation in the subsequent step. Due to the nonlinear nature of the measurement function, see Section 4.2, there is no closed form solution. Thus, some approximations are necessary and one such is

$$\begin{aligned}\mathcal{Q}(\theta,\theta_k) \approx \text{const.} - \frac{1}{2}\sum_{t=1}^{N}\Big(&\|y_t - h_t(\hat{x}_{t|N},\theta)\|^2_{R^{-1}} +\\ &\text{Tr}(R^{-1}\nabla_x h_t(\hat{x}_{t|N},\theta)P^s_{t|N}(\nabla_x h_t(\hat{x}_{t|N},\theta))^T)\Big)\end{aligned} \tag{8}$$

Here, $\hat{x}_{t|N}$ is the smoothed estimate of the latent variable and $P^s_{t|N}$ is its covariance. The smoothed estimate is obtained with an E-RTS smoother which is summarised in Algorithm 7. The trace term can be thought of as a regularisation term to compensate for the usage of the estimated latent variables instead of the true ones. If the true ones have been used that term would vanish and only the nonlinear least squares part had to be solved. See Appendix for derivation of (8).

3.2 M-step

Maximisation of the $\mathcal{Q}$-function can be done using standard optimisation software. Optimisation software usually assumes that the cost function should be minimised, which can easily be obtained by defining a new function as $-\mathcal{Q}(\theta,\theta_k)$. In the continuation the minimisation of the function will be considered. An important special case is linear systems since then the minimisation step can be solved by linear least-squares. As for our particular setting, the function to be minimised is a nonlinear function of the parameters and nonlinear methods need to be used. We use a quasi-Newton method called BFGS, (Nocedal and Wright, 2006) since it is quite efficient but other choices are also possible. In this method, the Hessian of the function to be optimised is recursively approximated using

Algorithm 7 Extended Rauch-Tung-Striebel Smoother (E-RTS)

```
Input: measurements $\mathcal{Y} = \{y_1, \dots, y_N\}$, inputs $U = \{u_1, \dots, u_N\}$, covariance matrices $Q$ and $R$, parameter estimate $\theta_k$
Output: smoothed state estimates $\hat{x}^s = \{\hat{x}_{1|N}, \dots, \hat{x}_{N|N}\}$, covariances $P^s_{1:N|N}$
Run a forward Extended Kalman filter (EKF) where measurement equation uses fixed value of the parameter $\theta = \theta_k$, and store time and measurement updates for states, $\hat{x}_{t|t},\ \hat{x}_{t|t-1}$, the covariances $P_{t|t},\ P_{t|t-1}$ and the Jacobians of the dynamics, $F_{t-1} = \frac{\partial}{\partial x} f(x_{t-1}, u_t, w_t)|_{x_{t-1}=\hat{x}_{t-1|t-1}, w_t=0}$, defined in (3a).
$P^s_{N|N} := P_{N|N}$
for $t = N : 2$ do
    $S_{t-1} := P_{t-1|t-1} F^T_{t-1} P^{-1}_{t|t-1}$
    $\hat{x}_{t-1|N} := \hat{x}_{t-1|t-1} + S_{t-1}(\hat{x}_{t|N} - \hat{x}_{t|t-1})$
    $P^s_{t-1|N} := P_{t-1|t-1} + S_{t-1}(P^s_{t|N} - P_{t|t-1}) S^T_{t-1}$
end for
```

the gradient information. The BFGS algorithm is summarised in Algorithm 8.

4 Models

In this section the models in Equation (3) will be specified. The sensors of interest are monocular camera and 6-DOF inertial sensors, gyroscopes and accelerometers, contained in a single sensor package, Inertial Measurement Unit (IMU). To reduce the state and parameter space the inertial sensors are considered as inputs to a process model. A minimal 3D point landmark parametrisation is used and its measurement is given by a pinhole projection model. Also, since, in general, an IMU has a higher sampling rate than a camera, a multirate system model is obtained. Basically, it implies that several state updates are performed, using the process model, between the measurements. This however poses no limitations for the methods presented here, since the E-RTS can easily handle multirate models.

4.1 IMU Parametrisation

The models for the gyroscopes and accelerometers are simple as they are only considered to be inputs to the process model. The gyroscope signals are denoted $u^\omega = [u^\omega_x, u^\omega_y, u^\omega_z]^T$ where the subscript refers to each axis of the body frame. Similarly the accelerometer signals are denoted $u^{\mathrm{a}} = [u^{\mathrm{a}}_x, u^{\mathrm{a}}_y, u^{\mathrm{a}}_z]^T$ which are also given in the sensor body frame. A discretised process model for the position velocity and rotation, $[p, v, q]$, in the local, inertial, navigation frame is then,

Algorithm 8 M-step (Quasi-Newton minimisation method with BFGS Hessian update)

Input: smoothed states $\hat{x}^s$, measurements $\mathcal{Y}$, initial parameters θ_k, inverse Hessian approximation $B_0 \approx \left(\nabla_\theta^2 \mathcal{Q}(\theta_k, \theta_k)\right)^{-1}$, termination threshold ε.

Output: θ_{k+1}.

```
$i := 0$
terminate := false
$\theta^i := \theta_k$
while not terminate do
    Compute search direction:
    $p_i := -B_i \nabla_\theta \mathcal{Q}(\theta^i, \theta_k)$
    Update the parameter:
    $\theta^{i+1} := \theta^i + \alpha_i p_i$
    where $\alpha_i$ is the step length computed by line search ensuring decrease in cost
    Compute:
    $s_i = \theta_{i+1} - \theta_i$
    $r_i = \nabla_\theta \mathcal{Q}(\theta^{i+1}, \theta_k) - \nabla_\theta \mathcal{Q}(\theta^i, \theta_k)$
    Update the inverse Hessian
    $B_{i+1} := \left(I - \frac{s_i r_i^T}{r_i^T s_i}\right) B_i \left(I - \frac{r_i s_i^T}{r_i^T s_i}\right) + \frac{s_i s_i^T}{r_i^T s_i}$
    if $\|\nabla_\theta \mathcal{Q}(\theta^{i+1}, \theta_k)\| < \varepsilon$ then
        terminate := true
    else
        $i := i + 1$
    end if
end while
$\theta_{k+1} := \theta^{i+1}$
```

$$p_t = p_{t-1} + T_s v_{t-1} + \frac{T_s^2}{2}\,\mathsf{R}^T(q_{t-1})\left(u_t^{\mathsf{a}} + g^b + w_t^{\mathsf{a}}\right) \tag{9a}$$

$$v_t = v_{t-1} + T_s\,\mathsf{R}^T(q_{t-1})\left(u_t^{\mathsf{a}} + g^b + w_t^{\mathsf{a}}\right) \tag{9b}$$

$$q_t = \exp\left(\frac{T_s}{2} S_\omega(u_t^\omega + w_t^\omega)\right) q_{t-1} \tag{9c}$$

where the T_s denotes the sampling interval, $\mathsf{R}(q)$ is a rotation matrix parametrisation of the unit quaternion $q = [q_0, q_1, q_2, q_3]^T$ which describes the rotation from navigation to body frame, $g^b = \mathsf{R}(q)g^n$, is the gravity expressed in the body frame, $g^n = [0, 0, -g]$ is the local gravity vector expressed in the inertial frame where $g \approx 9.82$ and $\exp(\,\cdot\,)$ is here considered as the matrix exponential. The noise terms are assumed Gaussian and independent $[(w_t^{\mathsf{a}})^T, (w_t^\omega)^T]^T = w_t \sim \mathcal{N}(0, diag\,(Q_{\mathsf{a}}, Q_\omega))$. The skew-symmetric matrix

$$S_\omega(u^\omega) = \begin{bmatrix} 0 & -u_x^\omega & -u_y^\omega & -u_z^\omega \\ u_x^\omega & 0 & u_z^\omega & -u_y^\omega \\ u_y^\omega & -u_z^\omega & 0 & u_x^\omega \\ u_z^\omega & u_y^\omega & -u_x^\omega & 0 \end{bmatrix}, \tag{10}$$

parametrises the quaternion dynamics. This parametrisation is very similar to reduced-dimension observers in Rugh (1996).

4.2 Camera Measurements

The monocular camera is modeled as a standard pinhole camera, see e.g., Hartley and Zisserman (2004). The camera calibration matrix and lens distortion was estimated prior to usage. Since the calibration and distortion are known the undistorted pixels can be pre-multiplied with the inverse of the camera matrix, thus the camera then works as a projective map in Euclidean space, $P : \mathbb{R}^3 \to \mathbb{R}^2$. The projection is defined as $P([X, Y, Z]) = [X/Z, Y/Z]$ and the Z coordinate is assumed positive and non-zero since otherwise the point would be behind the camera. Then a normalised camera measurement $y_t^m = [u_t, v_t]^T$ of a landmark, m, at time t is

$$y_t^m = P(\mathsf{R}(q_t)(m - p_t)) + e_t^m \tag{11}$$

which relates the pose (position and orientation) of the camera to the 3D location of the point. The measurement noise is assumed i.i.d. Gaussian, $e_t^m = [e_t^u, e_t^v]^T \sim \mathcal{N}(0, R_m)$. The correspondence variables at time t, c_t^i, encode the measurement-landmark assignment, $y_t^i \leftrightarrow m^j$, which gives a subset of all M landmarks at time t, $\mathsf{M}_t = \{m^j\},\ j \in \{1, \ldots, M |\, c_t^i = j\}$. At time t the stacked measurement equation is then

$$\underbrace{\begin{bmatrix} u_t^1 \\ v_t^1 \\ \vdots \\ u_t^{\mathsf{N}_y} \\ v_t^{\mathsf{N}_y} \end{bmatrix}}_{y_t^m} = \underbrace{\begin{bmatrix} P(\mathsf{R}(q_t)(m^{c_t^1} - p_t)) \\ \vdots \\ P(\mathsf{R}(q_t)(m^{c_t^{\mathsf{N}_y}} - p_t)) \end{bmatrix}}_{h_t(x_t, \theta)} + e_t^m, \tag{12}$$

where c_t^i denotes the index of the corresponding landmark and N_y is the number of measurements, which of course varies over time. In this paper the correspondences assumed correctly solved for in the initialisation step (see Section 7) but in practice there will always be outliers of some kind. This is a strong assumption which should be treated carefully since faulty associations will bias the SLAM estimate. Interesting approaches to data association was exploited in e.g., Bibby and Reid (2007); Dellaert et al. (2003) which both make use of the EM algorithm to estimate correspondences.

5 Nonlinear Least-Squares

Another way of solving the ML SLAM problem is to consider all the interesting parameters explicitly instead of having position, velocity and orientation as hidden variables. In this case the parameter vector θ will consist of all unknown parameters, that is landmarks, accelerations in navigation frame and rate gyros. The dynamics for the velocity and position is in this case used as explicit constraints. In this setting it is also possible to include biases for accelerations and angular rates as parameters, which was avoided in the EM formulation. This is because the problem greatly simplifies if the parameters affect only the measurement relation, as already explained in Section 3. Note however, that these extra terms can be put in the state vector.

The measurement models for accelerations and angular rates are then

$$y_t^a = \mathsf{R}(q_t)(a_t - g^e) + b_a + e_t^a \tag{13a}$$

$$y_t^\omega = \omega_t + b_\omega + e_t^\omega \tag{13b}$$

and camera measurements are defined as in Equation (12)

$$y_t^m = h_t(p_t, q_t, \mathsf{M}_t) + e_t^m. \tag{14}$$

The unknown parameters are then accelerations, $a_{1:N}$, angular rates, $\omega_{1:N}$, initial velocity v_0, acceleration bias, b_a, angular velocity bias, b_ω, and landmark positions m. Under the assumption that all noises are Gaussian and white, i.e., $e_t^i \sim \mathcal{N}(0, R_i)$, the corresponding negative log-likelihood becomes

$$-\log p_\theta(\mathcal{Y}) = \sum_{t=1}^{N} \|y_t^a - \mathsf{R}(q_t)(a_t - g^e) - b_a\|^2_{R_a^{-1}} + \|y_t^\omega - \omega_t - b_\omega\|^2_{R_\omega^{-1}} + \|y_t^m - h_t(p_t, q_t, \mathsf{M}_t)\|^2_{R_m^{-1}}. \tag{15}$$

where $\theta = [a_{1:N}^T, \omega_{1:N}^T, v_0^T, b_a^T, b_\omega^T, m^T]^T$, q_t is a function of $\omega_{1:t}$ and p_t is a function of v_0 and $a_{1:t}$. Maximum likelihood minimisation problem can now be formulated as

$$\hat{\theta}^{ML} = \arg\min_\theta \; -\log p_\theta(\mathcal{Y}) \tag{16a}$$

$$\textbf{s. t.} \quad \begin{bmatrix} p_t \\ v_t \end{bmatrix} = F^t \begin{bmatrix} p_0 \\ v_0 \end{bmatrix} + \sum_{i=1}^{t} F^{i-1} B a_i, \tag{16b}$$

$$F = \begin{bmatrix} I_3 & T_s I_3 \\ 0 & I_3 \end{bmatrix}, \quad B = \begin{bmatrix} \frac{T_s^2}{2} I_3 \\ T_s I_3 \end{bmatrix},$$

$$q_t = \left[\prod_{k=1}^{t} \exp\left(\frac{T_s}{2} S_\omega(\omega_k) \right) \right] q_0 \tag{16c}$$

The constraints can actually be removed by expanding them and substituting them into the cost function giving an unconstrained problem. This problem is solved with e.g., standard Levenberg-Marquardt solver. The estimate obtained in this way will be used to compare to the estimate obtained with the EM-SLAM method.

6 Computation Complexity

The main difference between NLS and EM approach is the number of parameters. While NLS has both landmarks and platform's motion as parameters, EM considers the motion as latent variables. Seen the other way around, the ML problem in (1) can be considered as a marginalised version of (15), where motion is integrated out. As such, the complexity of the EM approach is actually lower than the NLS approach. To see this, consider the problem sizes of the two approaches, given N time instances where landmarks are observed, M landmarks and N_m landmark measurements, the NLS problem will have $6(N-1) + 3M + 3$ variables (6 more if biases are also included) and $6(N-1) + 2N_m$ measurements. So, the size of the problem grows both with the number of landmarks and time and it grows quadratically in time. For the EM approach, the E-step is realised as an E-RTS smoother with constant size state vector (unlike traditional EKF-SLAM for example), meaning that the complexity increases linearly with the amount of time points, even if a multirate model is considered, since it is the measurement update which is dominating. For the M-step, the size of the problem to be solved is $3M$ variables and $2N_m$ measurements, although the problem is slightly more difficult than NLS. However, since the BFGS approach is quite efficient, it has superlinear convergence properties, see Nocedal and Wright (2006), this is not a problem. This basically means that each step in the EM-SLAM will eventually be cheaper in total when the number of time steps grows.

7 Obtaining an Initial Estimate

Both EM and NLS-SLAM need an initial value of the parameters in order to do the iterations. This initial value is also important for the performance of the methods, since both formulations are non-linear and non-convex. The initialisation can be performed by simply randomising parameter values but that can lead to solutions that are stuck in local minima. A better estimate of the initial values can be obtained by noting that the NLS-SLAM problem, defined in Equation (16), is actually almost linear if rotations are fixed, (Martinelli, 2012). In that case Equation (13b) is not needed any more and Equation (13a) is linear in parameters. For the landmark measurements consider the projection according to Equation (11) which for fixed rotations can be rewritten as

$$\begin{bmatrix} [u_t \, \mathsf{R}_{3,:}(q_t) - \mathsf{R}_{1,:}(q_t)](m - p_t) \\ [v_t \, \mathsf{R}_{3,:}(q_t) - \mathsf{R}_{2,:}(q_t)](m - p_t) \end{bmatrix} = \begin{bmatrix} \mathsf{R}_{3,:}(q_t)(m - p_t)e_t^u \\ \mathsf{R}_{3,:}(q_t)(m - p_t)e_t^v \end{bmatrix} \tag{17}$$

where $\mathsf{R}_{i,:}(q_t)$ denotes the i:th row of the rotation matrix. The only thing that makes this equation non-linear is the parameter dependent noise term. However this formulation leads to a well known Iterative Reweighted Least Squares (IRLS) method which is solved efficiently, see e.g., Björck (1996). The accuracy of the estimate obtained in this way is dependent of the fixed rotations, but it still constitutes a much better initial value for the EM and NLS-SLAM then simply random values, see Skoglund et al. (2013) for more details. The initial rotations can be obtained in several ways, for example simply by integrating rate gyros using Equation (9c), or by some camera based method like 8-point, see e.g., Hartley and Zisserman (2004). The first method works quite fine if the gyro bias is small, while the latter one demands that the scene geometry is beneficial.

8 Results

Evaluation of the proposed method is carried out on both simulated and experimental data.

8.1 Simulations

Simulations give the ability to choose noise levels, correspondences, the true parameters and the true accuracy of the method. Monte Carlo (MC) simulations with 30 different measurement noise realisations have been performed in order to evaluate the performance of the proposed method and to compare EM-SLAM with the NLS-SLAM. In Figure 1 the setup used for the simulations is illustrated. The true trajectory is in black and true landmarks are represented with red circles. One of the resulting trajectory (magenta) and landmark (blue stars) estimates is also plotted. Table 1 shows the average of the landmark estimation error for the two methods, while in Figure 2 the RMSE of the trajectory, for both methods, is plotted. In general it can be seen that the EM-SLAM method performs slightly better in average than the NLS-SLAM method.

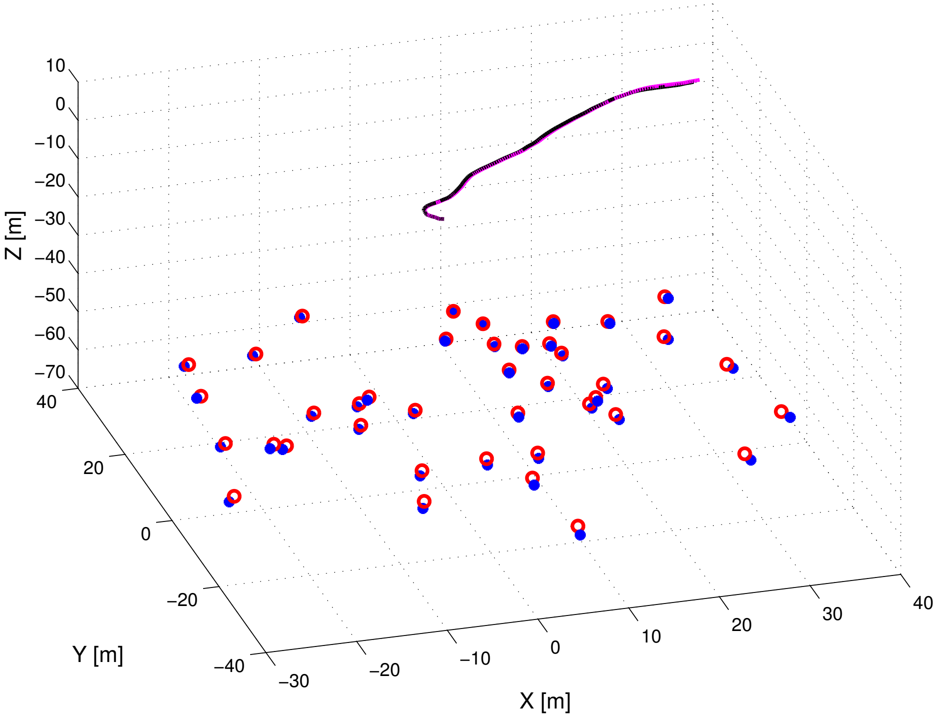


Figure 1: *The setup used in the MC simulations. True trajectory is in black and true landmarks are red circles. One of 30 simulation results is depicted as magenta trajectory and blue landmarks.*

Method	EM-SLAM	NLS-SLAM
Mean $\|\hat{\theta} - \theta^*\|/\mathbf{dim}(\theta^*)$	0.11	0.19

Table 1: *MC simulation results for the varying measurement noise (30 realisations). Note that θ contains only landmarks in this case.*

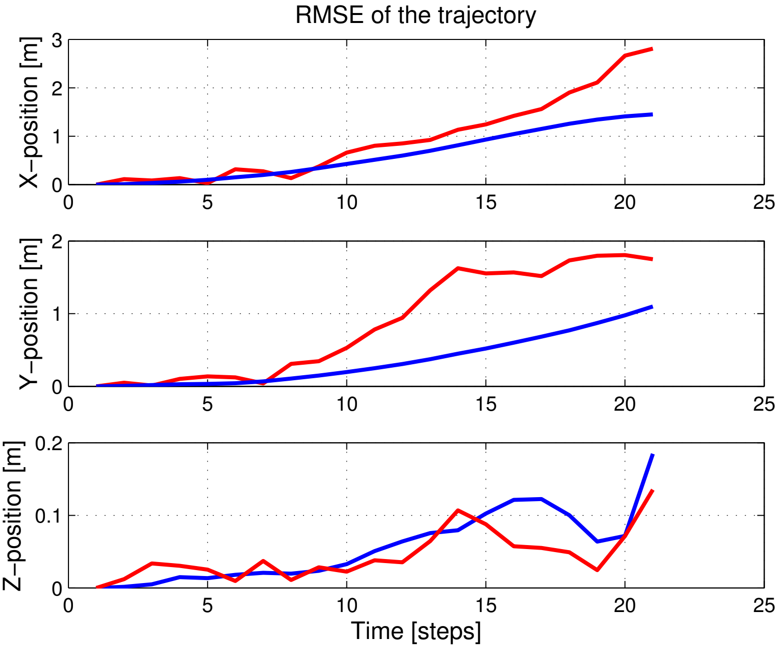


Figure 2: *RMSE of the EM-SLAM and NLS-SLAM estimated trajectories, EM in blue, NLS in red. 30 MC simulations are used.*

Figure 3: *Sensor unit containing both camera and inertial sensors.*

8.2 Real Data Experiments

In the experiments, a sensor unit, see Figure 3, equipped with monocular monochrome VGA camera (Pointgrey firefly) and three axis inertial sensors (gyroscopes and accelerometers) is used. The sensor unit also contains magnetometers, which are not used here, and a temperature sensor which is used for obtaining internal calibration of the inertial sensors. For the purpose of an accurately known ground truth the sensor unit was mounted at the tool position of an IRB-1400 industrial robot from ABB. Also, a small scene with objects of known size was created so that the estimated scene could be compared with respect to its size. Prior to usage, the camera was calibrated using the toolbox (Bouguet, 2010) and the relative pose of the camera centre with respect to the IMU centre was calibrated as described in Hol et al. (2010). A open source SIFT implementation from Vedaldi and Fulkerson (2008) was used to extract the features used as camera measurements. The results of the estimation are depicted in Figure 4, Figure 5 and Figure 6. We see that both NLS and EM methods have similar performance on this data set. EM performs somewhat worse for the rotations which might be explained by the lack of biases as opposed to NLS.

9 Conclusions and Future Work

In this work we presented a Maximum Likelihood method for solving inertial/visual SLAM problem based on the EM algorithm. The particular structure of the SLAM problem, where landmarks are seen as static parameters while the platform's motion is intro-

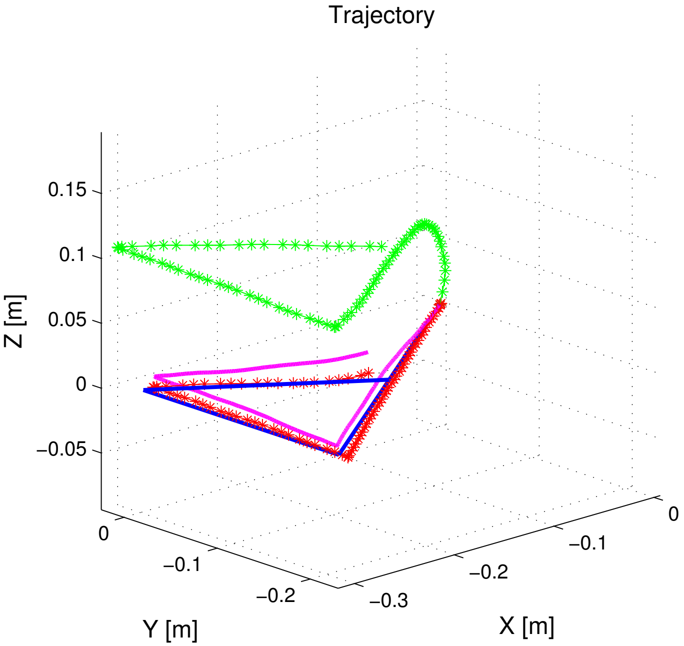


Figure 4: *Trajectory from the linear initialisation (green), EM (magenta), NLS (red) and ground truth (blue).*

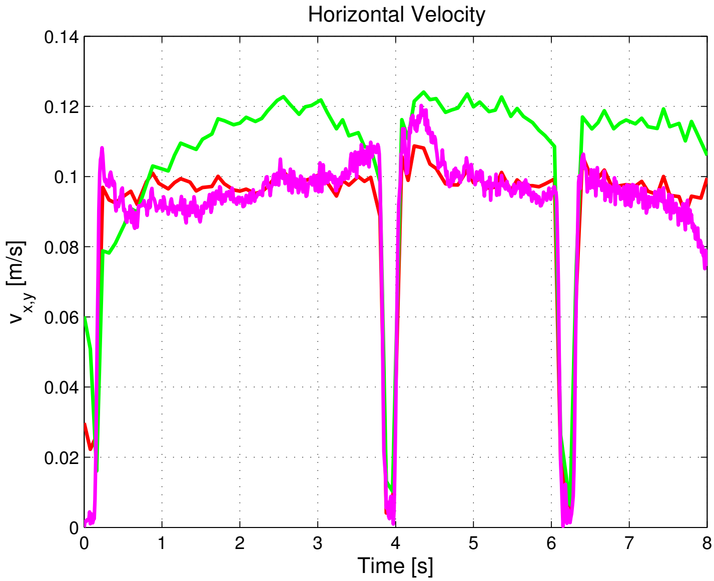


Figure 5: *Estimated horizontal velocity from the linear initialisation (green), EM (magenta) and NLS (red).*

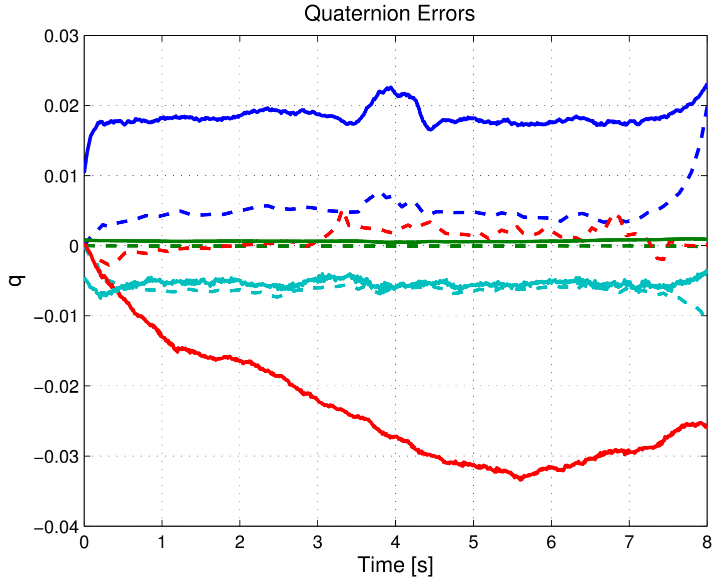


Figure 6: *Estimated quaternion error from EM (solid) and NLS (dashed).*

duced as latent variables, makes the EM scheme a natural way to formulate the problem. This gives better scaling of the problem compared to both FastSLAM, EKF-SLAM and NLS. In particular, a qualitative analysis of the computational complexity is made comparing EM-SLAM and NLS. Furthermore, it is also shown that somewhat better results are obtained for EM-SLAM than the straightforward formulation in the form of NLS. These results are demonstrated on both simulated and real data sets.

In the future work it would be interesting to change E-RTS smoother estimate of the states with the particle smoother estimate, since it may handle nonlinearities in the models better and see if the performance improves.

Appendix

Given a smoothed estimate of the latent variables, $\hat{x}^s = \hat{x}_{1:N|N}$ the measurement function $h(x, \theta)$ can be linearised around these as

$$h(x,\theta) \approx \underbrace{h(\hat{x}^s,\theta)}_{\hat{h}} + \underbrace{\nabla_x h(\hat{x}^s,\theta)}_{H}\underbrace{(x-\hat{x}^s)}_{\tilde{x}}. \tag{18}$$

Using this approximation and expanding the norm in (7) for one time instant, while dropping the time index for readability, we obtain

$$\begin{aligned}\|y-\hat{h}-H\tilde{x}\|^2_{R^{-1}} =&(y-\hat{h}-H\tilde{x})^T R^{-1}(y-\hat{h}-H\tilde{x}) = \\ & y^T R^{-1} y - y^T R^{-1}\hat{h} - y^T R^{-1} H\tilde{x} - \hat{h}^T R^{-1} y + \hat{h}^T R^{-1}\hat{h} + \\ & \hat{h}^T R^{-1} H\tilde{x} - (H\tilde{x})^T R^{-1} y + (H\tilde{x})^T R^{-1}\hat{h} + (H\tilde{x})^T R^{-1}(H\tilde{x})\end{aligned} \tag{19}$$

and taking the expected value

$$\begin{aligned}\mathbf{E}_{\theta_k}\{\|y-\hat{h}-H\tilde{x}\|^2_{R^{-1}}\} =& y^T R^{-1} y - y^T R^{-1}\hat{h} - \hat{h}^T R^{-1} y + \hat{h}^T R^{-1}\hat{h} + \\ & \mathbf{E}_{\theta_k}\{(H\tilde{x})^T R^{-1}(H\tilde{x})|\mathcal{Y}\},\end{aligned} \tag{20}$$

since all terms with only $\tilde{x}$ evaluate to zero under the assumption $(\tilde{x}|\mathcal{Y}) \sim \mathcal{N}(0, P^s)$. Because $(H\tilde{x})^T R^{-1}(H\tilde{x})$ is scalar, it is equal to its trace and by using the trace rule $\mathrm{Tr}(A^T BA) = \mathrm{Tr}(BAA^T)$ together with the linearity of the trace and expectation operators, the last term becomes

$$\begin{aligned}\mathbf{E}_{\theta_k}\{(H\tilde{x})^T R^{-1}(H\tilde{x})|\mathcal{Y}\} =& \mathbf{E}_{\theta_k}\{\mathrm{Tr}((H\tilde{x})^T R^{-1}(H\tilde{x}))|\mathcal{Y}\} = \\ & \mathbf{E}_{\theta_k}\{\mathrm{Tr}(R^{-1}H\tilde{x}\tilde{x}^T H^T)|\mathcal{Y}\} = \\ & \mathrm{Tr}(R^{-1}H\mathbf{E}_{\theta_k}\{\tilde{x}\tilde{x}^T|\mathcal{Y}\}H^T) = \\ & \mathrm{Tr}(R^{-1}HP^sH^T)\end{aligned} \tag{21}$$

which results in

$$\begin{aligned}\mathcal{Q}(\theta, \theta_k) \approx \text{const.} - \frac{1}{2}\sum_{t=1}^{N}\Big(\|y_t - \hat{h}_t\|^2_{R^{-1}} + \mathrm{Tr}(R^{-1}H_t P^s_{t|N} H_t^T)\Big) = \\ \text{const.} - \frac{1}{2}\sum_{t=1}^{N}\Big(\|y_t - h_t(\hat{x}_{t|N}, \theta)\|^2_{R^{-1}} + \\ \mathrm{Tr}(R^{-1}\nabla_x h_t(\hat{x}_{t|N}, \theta) P^s_{t|N} (\nabla_x h_t(\hat{x}_{t|N}, \theta))^T)\Big)\end{aligned} \tag{22}$$

which is the expression in (8).

Bibliography

T. Bailey, J. Nieto, J. E. Guviant, M. Stevens, and E. M. Nebot. Consistency of the EKF-SLAM Algorithm. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 3562–3568, Beijing, China, 2006a. doi: 10.1109/IROS.2006.281644. URL `http://dx.doi.org/10.1109/IROS.2006.281644`.

T. Bailey, J. Nieto, and E. M. Nebot. Consistency of the FastSLAM Algorithm. In *International Conference on Robotics and Automation (ICRA)*, pages 424–429, Orlando, Florida, USA, 2006b.

C. Bibby and I. Reid. Simultaneous localisation and mapping in dynamic environments (SLAMIDE) with reversible data association. In *In Proceedings of Robotics: Science and Systems*, 2007.

Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, 1996. ISBN 0-89871-360-9.

J.-Y. Bouguet. Camera Calibration Toolbox for Matlab, 2010. URL `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

F. Dellaert and M. Kaess. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *International Journal of Robotics Research*, 25(12):1181–1203, 2006. ISSN 0278-3649. doi: 10.1177/0278364906072768. URL `http://dx.doi.org/10.1177/0278364906072768`.

F. Dellaert, S.M. Seitz, C. Thorpe, and S. Thrun. EM, MCMC, and Chain Flipping for Structure from Motion with Unknown Correspondence. *Machine Learning*, 50(1-2): 45–71, 2003.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. ISSN 00359246. doi: 10.2307/2984875. URL `http://web.mit.edu/6.435/www/Dempster77.pdf`.

S. Duncan and M. Gyongy. Using the EM algorithm to estimate the disease parameters for smallpox in 17th century London. In *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, pages 3312 –3317, oct. 2006. doi: 10.1109/CACSD-CCA-ISIC.2006.4777169.

A. W. Fitzgibbon and A. Zisserman. Automatic Camera Recovery for Closed or Open Image Sequences. In *ECCV (1)*, pages 311–326, 1998.

Z. Ghahramani and S. T. Roweis. Learning Nonlinear Dynamical Systems using an EM Algorithm. In *Advances in Neural Information Processing Systems*, volume 11, pages 599–605. MIT Press, 1999.

R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146 – 157, 1997. ISSN 1077-3142. doi: 10.1006/cviu.1997.

0547. URL `http://www.sciencedirect.com/science/article/pii/S1077314297905476`.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. ISBN 0-521-54051-8.

J. Hol, T. B. Schön, and F. Gustafsson. Modeling and Calibration of Inertial and Vision Sensors. *The international journal of robotics research*, 29(2):231–244, 2010.

J. E. Guviant and E. M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, June 2001. ISSN 1042-296X.

J. J. Leonard and H. Jacob and S. Feder. A Computationally Efficient Method for Large-Scale Concurrent Mapping and Localization. In *Proceedings of the Ninth International Symposium on Robotics Research*, pages 169–176, Salt Lake City, Utah, 2000. Springer-Verlag.

L. Kneip, M. Chli, and R. Siegwart. Robust Real-Time Visual Odometry with a Single Camera and an IMU. In *Proceedings of the British Machine Vision Conference*, pages 16.1–16.11. BMVA Press, 2011a. ISBN 1-901725-43-X. http://dx.doi.org/10.5244/C.25.16.

L. Kneip, A. Martinelli, S. Weiss, D. Scaramuzza, and R. Siegwart. Closed-Form Solution for Absolute Scale Velocity Determination Combining Inertial Measurements and a Single Feature Correspondence. In *International Conference on Robotics and Automation*, Shanghai, China, May 2011b. URL `http://hal.inria.fr/hal-00641772`.

S. Le Corff, G. Fort, and E. Moulines. Online Expectation Maximization algorithm to solve the SLAM problem. In *Statistical Signal Processing Workshop (SSP), 2011 IEEE*, pages 225 –228, june 2011. doi: 10.1109/SSP.2011.5967666.

T. Lupton and S. Sukkarieh. Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions. *Robotics, IEEE Transactions on*, 28 (1):61 –76, feb. 2012. ISSN 1552-3098. doi: 10.1109/TRO.2011.2170332.

M. Bryson and M. Johnson-Roberson and S. Sukkarieh. Airborne smoothing and mapping using vision and inertial sensors. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 3143–3148, Kobe, Japan, 2009. IEEE Press. ISBN 978-1-4244-2788-8.

A. Martinelli. Vision and IMU Data Fusion: Closed-Form Solutions for Attitude, Speed, Absolute Scale, and Bias Determination. *Robotics, IEEE Transactions on*, 28(1):44 –60, feb. 2012. ISSN 1552-3098. doi: 10.1109/TRO.2011.2160468.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably

Converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003.

J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer, New York, 2nd edition, 2006. ISBN 978-0-387-40065-5.

E. Ozkan, C. Fritsche, and F. Gustafsson. Online EM algorithm for joint state and mixture measurement noise estimation. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 1935 –1940, july 2012.

P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modelling. In *5th International Symposium on Robotics Research*, pages 207–216, Tokyo, Japan, 1989.

H. E. Rauch, C. T. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965.

W. J. Rugh. *Linear System Theory.* Prentice Hall, Engelwood Cliffs, NJ, 2nd ed., 1996.

T. B. Schön. An Explanation of the Expectation Maximization Algorithm. Technical Report LiTH-ISY-R-2915, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, August 2009.

Z. Sjanic, M. A. Skoglund, and F. Gustafsson. Expectation-Maximisation Maximum Likelihood Estimation for Inertial/Visual SLAM. *Submitted to IEEE Transactions on Robotics*, September 2013.

M. A. Skoglund, Z. Sjanic, and F. Gustafsson. Initialisation and Estimation Methods for Batch Optimisation of Inertial/Visual SLAM. Technical Report LiTH-ISY-R-3065, Department of Electrical Engineering, Linköping University, September 2013.

R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990. ISBN 0-387-97240-4.

C. J. Taylor, D. Kriegman, and P. Anandan. Structure and Motion in Two Dimensions from Multiple Images: A Least Squares Approach. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 242 –248, Princeton, NJ, USA, October 1991.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents).* The MIT Press, 2005. ISBN 0262201623.

A. Vedaldi and B. Fulkerson. VLFeat: An Open and Portable Library of Computer Vision Algorithms, 2008. URL `http://www.vlfeat.org/`.

A. Wills, B. Ninness, and T. B. Schön. Estimating State-Space Models in Innovations Form using the Expectation Maximisation Algorithm. In *The 49th IEEE Conference on Decision and Control (CDC)*, Atlanta, USA, December 2010.

Paper G

Cellular Network Non-Line-of-Sight Reflector Localisation Based on Synthetic Aperture Radar Methods

Authors: Zoran Sjanic, Fredrik Gunnarsson, Carsten Fritsche and Fredrik Gustafsson

Edited version of the paper:

Z. Sjanic, F. Gunnarsson, C. Fritsche, and F. Gustafsson. Cellular Network Non-Line-of-Sight Reflector Localisation Based on Synthetic Aperture Radar Methods. *Submitted to IEEE Transactions on Antennas and Propagation*, September 2013a.

Cellular Network Non-Line-of-Sight Reflector Localisation Based on Synthetic Aperture Radar Methods

Zoran Sjanic, Fredrik Gunnarsson, Carsten Fritsche and Fredrik Gustafsson

Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
`{ms,zoran,carsten,fredrik}@isy.liu.se`

Abstract

The dependence of radio signal propagation on the environment is well known, and both statistical and deterministic methods have been presented in the literature. Such methods are either based on randomised or actual reflectors of radio signals. In this work, we instead aim at estimating the location of the reflectors based on geo-localised radio channel impulse response measurements and using methods from synthetic aperture radar (SAR). Radio channel data measurements from 3GPP E-UTRAN have been used to verify the usefulness of the proposed approach. The obtained images show that the estimated reflectors are well correlated with the aerial map of the environment. Also, which part of the trajectory contributed to different reflectors have been estimated with promising results.

1 Introduction

Radio signal propagation significantly depends on the environment through which the signal propagates. Free space propagation is different from line-of-sight (LOS) propagation close to the ground, which in turn is different from non-line-of-sight (NLOS) propagation where obstacles and reflectors have a significant impact on the propagation properties. Modeling radio signal propagation properties based on environment characteristics is therefore important to properly plan and analyse terrestrial wireless communication systems.

Numerous activities are focusing on propagation channel modeling featuring NLOS propagation, and the efforts can be separated into statistical models and ray-tracing models. The former can be motivated by theoretical modeling, or designed as parametric models that can be tuned and validated based on empirical data. The surveys Jensen and Wallace (2004); Almers et al. (2007) cover statistical models well. One key statistical model example is the 3GPP Spatial Channel Model (SCM) (3GPP, 2012), which has been empirically validated (Medbo et al., 2006). In brief, the model describes random scatterers that reflect incident radio waves toward the radio receiver. Ray-tracing models (Jensen and Wallace, 2004; Fügen et al., 2006) on the other hand, aim at modeling the properties of the physical multi-path propagation channel, including knowledge about the location of scatterers and reflectors. Such information can be derived from building databases, but will still be associated with uncertainty.

In order to model the NLOS propagation in a specific area, one either needs to tune the parameters of the statistical model so that it generates reflectors representative for the area, or to analyse building data to determine the reflectors and occlusions deterministically. In this paper, we take an intermediate approach, using geo-localised impulse response measurements to estimate the reflector locations. Furthermore, the ambition is also to estimate where in an area a particular reflector is active. Since the number of reflectors is unknown, a non-parametric method must be applied to get an initial map of the reflector locations that can be refined with some parametric method afterwards. One non-parametric method to get initial estimate of the map is based on the multistatic Synthetic Aperture Radar (SAR)-like technique, (Cutrona et al., 1961; Willis, 2007; Krishnan et al., 2010).

The paper is organised as follows; Section 2 describes SAR in general terms, and Section 3 the considered application based on measurements in 3GPP Long Term Evolution. In Section 4, the radio channel measurement campaign is described, while Section 5 adopts the SAR methods to enable reflector estimation and provides some results. Finally, Section 6 concludes the paper.

2 SAR and Multistatic SAR

The SAR imaging is based on a moving platform, moving along a scene that shall be imaged. During the movement, the platform transmits radar pulses which will hit the scene and return to the platform with a certain time delay, which is proportional to the range to the scene. This returned signal is filtered with a matched filter and then sampled. Each reflector in the scene will contribute with its reflected power, and will then be placed in an appropriate range bin. The range R is determined as a product between signal propagation speed (usually speed of light) and delay time. In this way a single scene transfer function is obtained, denoted $g(R)$. Now this process can be repeated during the platform movement at different time instances t, yielding a transfer function $g_t(R)$ which can be stored in a two-dimensional array. Basically, this raw data, $g_t(R)$, is an example of a real aperture radar or RAR. The resolution in such a radar system is proportional to the radar lobe width and is usually quite poor. One important thing to notice is that the lobe width is inversely proportional to the antenna size, i.e., the larger antenna the smaller lobe we can obtain.

The idea behind SAR is to artificially synthesise a large antenna by moving the platform, (Cutrona et al., 1961; Oliver and Quegan, 2004). One way that this can be done is by a *global back-projection* method, (Natterer, 1986; Andersson, 1988; Fawcett, 1985), that can be described in the following way; given the raw (possibly complex valued) data $g_t(R)$ we can back-project each radar echo on the image yielding the subimage I_t and each reflector will create a circle in each subimage. A total image I can then be created

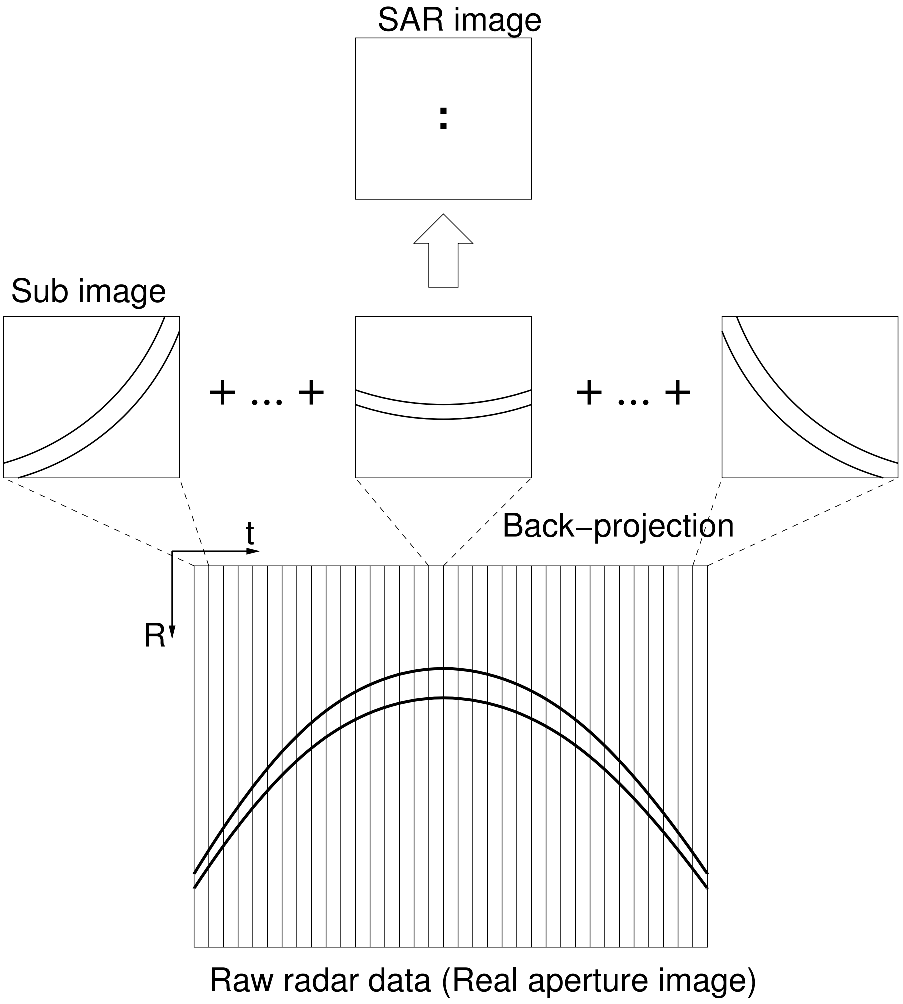


Figure 1: *Global back-projection method for creating SAR images. The scene consists of two point targets.*

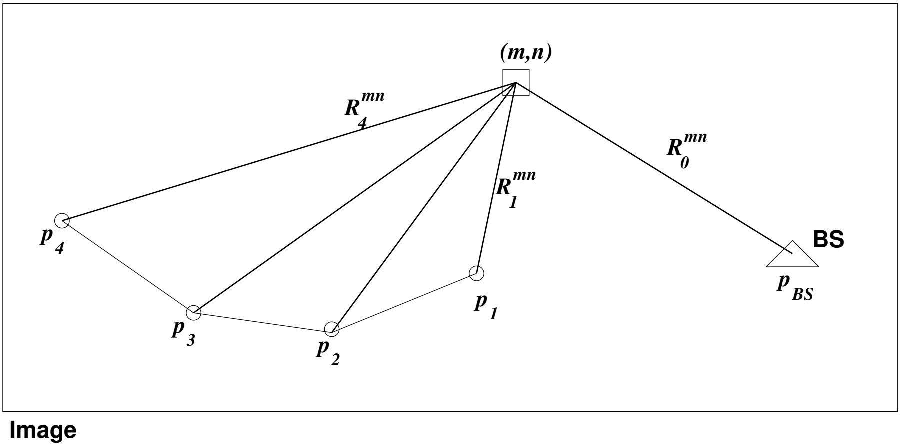


Figure 2: *Example of a bi-static SAR geometry in two dimensions.*

by summing up all the subimages along the synthetic aperture

$$I = \sum_{t=1}^{N} I_t \tag{1}$$

This method can schematically be described as in Figure 1. Another way of creating the image is to integrate the raw data for each pixel in the resulting image I_{mn} as

$$I_{mn} = \sum_{t=1}^{N} g_t(R_t^{mn}) \tag{2a}$$

$$R_t^{mn} = \|p_t - s^{mn}\|_2 \tag{2b}$$

where p_t is the position of the platform, s^{mn} is the position in the scene which corresponds to the pixel (m, n) and $||\cdot||_2$ is the usual vector 2-norm. The complexity of this operation is proportional to $\mathcal{O}(NM^2)$ for an $M \times M$ image and N time points. This describes the principle of the *mono-static* SAR in which transmitting and receiving antennas are co-located. The principle can be generalised with transmitting and receiving antennas at different positions to principles of *multi-static* SAR. Also, the specific case of *bi-static* SAR with one separate transmitting antenna and multiple receiving antennas or antenna positions, is highly relevant.

Figure 2 illustrates a case with one transmitter (or base station) in one position p_{BS} and a receiving antenna at positions p_1 to p_4. The only difference to the case in (2) is to add the distance R_0^{mn} between the transmitter and the pixel coordinate (m, n) to the total range.

$$I_{mn} = \sum_{t=1}^{N} g_t(R_0^{mn} + R_t^{mn}) \tag{3a}$$

$$R_0^{mn} = \|p_{BS} - s^{mn}\|_2 \tag{3b}$$

$$R_t^{mn} = \|p_t - s^{mn}\|_2 \tag{3c}$$

In this case each reflector will describe an ellipse in the image with focal points located in p_{BS} and p_t. It shall be noted that the integration above can be done with the complex valued raw data (coherent) or with the magnitude of the complex valued data (incoherent).

3 OFDM Signal and SAR Modeling

In this section, the considered OFDMA signals are modeled, and associations to the SAR theory are established.

Even though the methods described in the sequel apply to a general radio network, the modeling will adopt 3GPP LTE (Dahlman et al., 2011) nomenclature and describe broadcasted signals from base stations based on Orthogonal Frequency Division Multiplexing (OFDM). The transmitted signal consists of coded symbols which can be described as

$$d_t = \sum_{k=0}^{N-1} S_k e^{2\pi jk\frac{t}{T}}, \quad 0 \leq t < T \tag{4}$$

where S_k are the transmitted symbols, N is the number of symbols and T is the OFDM symbol time and the bandwidth needed is $B = N/T$. This baseband signal is then transformed to a passband signal centered at carrier frequency f_c

$$z_t = \Re\{e^{2\pi jf_c t} d_t\}, \quad 0 \leq t < T \tag{5}$$

where $\Re\{\cdot\}$ denotes the real part of a complex number. The received signal y_t should ideally be scaled (by A_0) and time delayed (by τ_0) version of the transmitted signal (i.e., LOS signal)

$$y_t = A_0 z_{t-\tau_0} + e_t \tag{6}$$

where e_t denotes some channel noise. However, because of the urban environment and presence of multipath signals, the actual received signal can be written as

$$y_t = \sum_{p=0}^{P} A_p z_{t-\tau_p} + e_t \tag{7}$$

where the number of the multipath reflectors is P. In general, the amplitudes A_p will be proportional to $(R_t^p)^{-\alpha} = (c_0\tau_p)^{-\alpha}$, but many of them, including the LOS amplitude, can also be zero, for example if there is an occlusion present. α is here defined as a path loss exponent.

The received signal is usually collected in the frequency domain since the matched filtering can be implemented with multiplications instead of convolutions, and the time domain signal is simply obtained as an inverse discrete Fourier transform (IDFT) of the frequency signal, $y_t = \mathcal{F}^{-1}\{Y_f\}$.

The received signal or impulse response y_t can be seen as the scene transfer function for one time instant during the data acquisition, i.e., $y_t = g_t(R)$ according to the SAR notation.

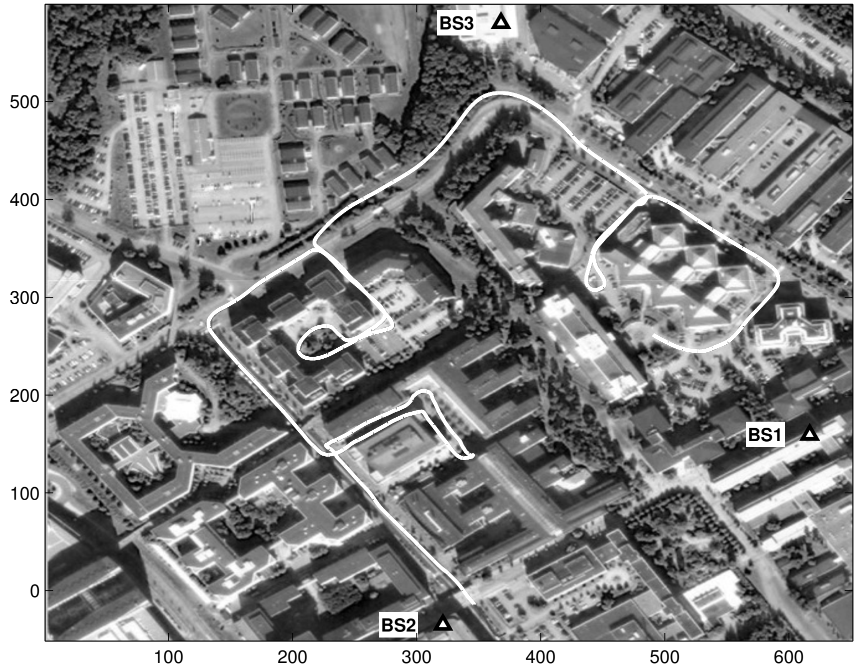


Figure 3: *Configurations of the three base stations used in the experiment in Kista, Stockholm. The driven trajectory is in white starting near base station 2 in the lower part of the image.*

4 Radio Channel Measurements

Radio channel data has been gathered during initial E-UTRAN trials in Kista Stockholm. More complete details can be obtained from Medbo et al. (2009). The base station configuration was according to Figure 3. Three base stations are used with the commercially available antennas having transmit power of about 35 dBm at the antenna ports. The four receiving antennas were mounted on the rooftop of a van and synchronisation of the transmitting and receiving signal is obtained by the rubidium clocks from Stanford Research Systems with the Allan standard deviation less than 10^{-12} s, which corresponds to an error in propagation of less than 1 m during an 8 min period.

In the actual experiment, the symbols are transmitted in the whole bandwidth $B \approx 20$ MHz on the carrier frequency $f_c = 2.66$ GHz. The frequency response of the channel is sampled every $\Delta t = 5.3$ ms with the frequency resolution of $\Delta f = 45$ kHz and saved in a memory with $L = 432$ samples. This gives the effective range resolution of $c_0/(L\Delta f) \approx 15$ m, where c_0 is the speed of light. The frequency response is transformed to the spatial domain with the IDFT for each time instant. On top of measuring the base station signals, the GPS based position of the receiver is saved for each time instant. The GPS position data are also synchronised to the signal receiving times with the high accuracy clock.

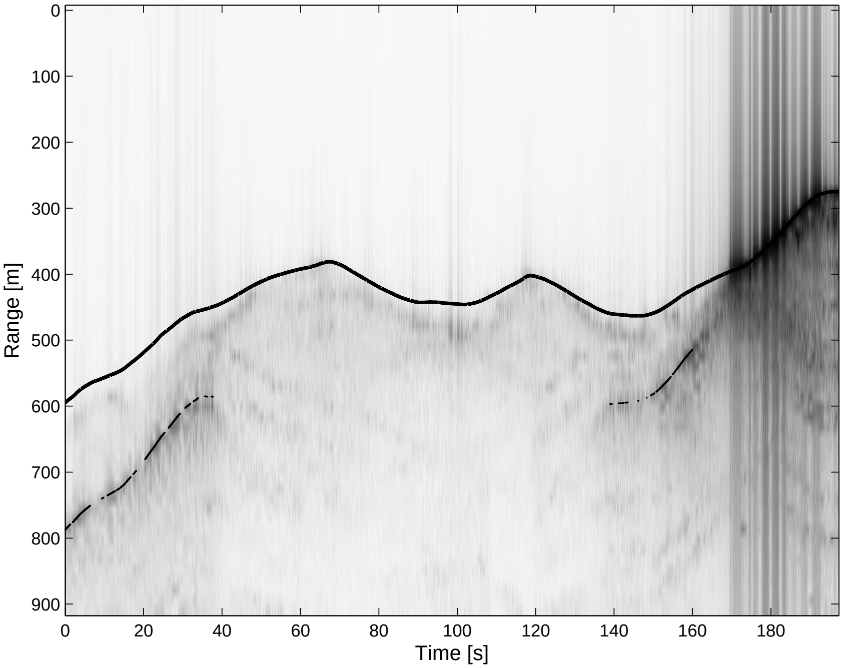


Figure 4: *Raw data (logarithm of the magnitude of the impulse response) from one base station. Solid black line is the range to the base station 3 (LOS range) based on the GPS trajectory. Non-continuous line is the NLOS range to the point (100, 300) in Figure 6.*

An example of the impulse response data collected from one base station (BS3) and averaged over all 4 receiving antennas is depicted in Figure 4. Also the LOS range to the base station is plotted with the solid black line. It can be seen in this figure that during certain times obvious occlusions happen, for example time interval between ca. 0 to 170 s. This time interval corresponds to the parts of the trajectory located at the farthest end from the base station 3, between coordinates ca. (350, 0) and (150, 280), see Figure 3, where high buildings are probable cause for the occlusions. The non-continuous line segments marks the parts of the impulse response with the most evident non-line of sight propagation. The line segments are further discussed in Section 5.

5 Results

The resulting image of the estimated reflectors can be seen in Figure 5. The estimated image, in grayscale, of the reflectors is overlaid on the image of the Kista area in Stockholm. To enhance the visibility, the level curves are also drawn. The trajectory used for the estimation is plotted as a thick black line, and three base stations, denoted BS1, BS2 and BS3, are shown as black triangles. The estimated reflectors seem to be well correlated with the aerial map of the environment, although their resolution is somewhat low.

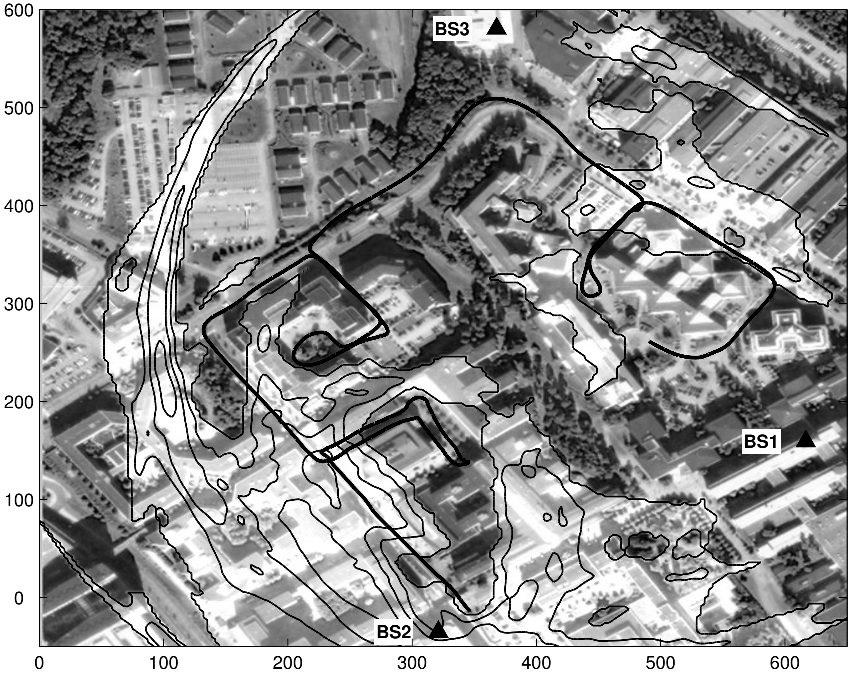


Figure 5: *Image of the reflectors estimated from all three base stations in light gray together with level curves overlaid on the test area image. The trajectory is drawn as a thick black line and base station positions are represented with black triangles.*

This is a consequence of the SAR processing where many time points (full aperture) are needed in order to obtain the full image resolution, see Section 2. In many cases here, the reflectors were visible only for a short periods of time causing the low resolution in the image.

The reflections can be further analysed by identifying parts of the trajectory where the identified reflector has been active. Figure 6 illustrates such an analysis with an identified reflector marked as a white square at coordinate (100, 300) in the image. Parts of the trajectory that contribute to that reflector point seen from base station BS3 are highlighted in black. The result is natural, since BS3 is not directly visible from that part of the trajectory due to the occlusion from the buildings. Instead, signals from BS3 propagates via the identified reflector to this part of the trajectory. These parts of the trajectory are also marked with corresponding NLOS range, via the identified reflector, in Figure 4. Beyond doubt, it is the identified reflector that is active in the NLOS propagation during these segments.

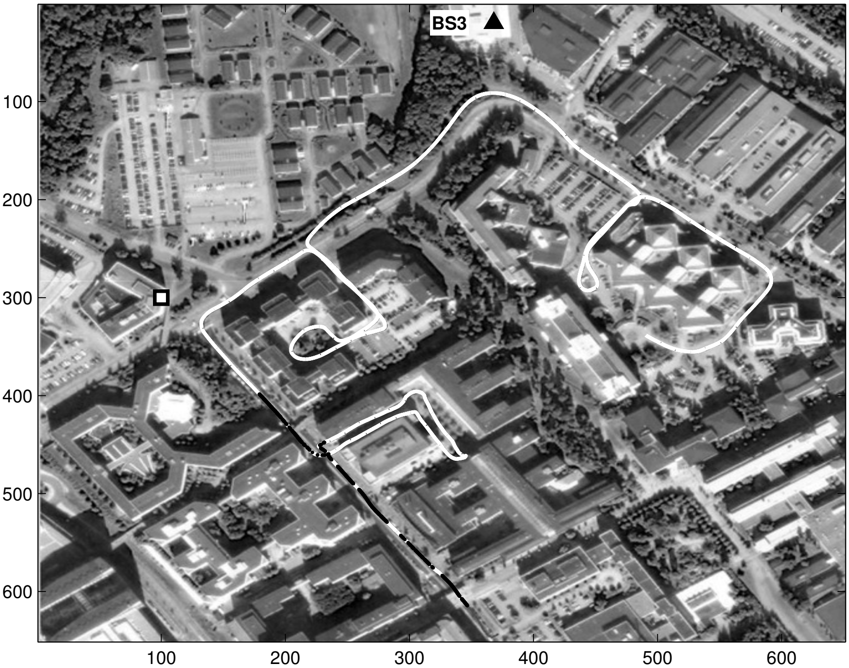


Figure 6: *Identified reflector marked with a white square, together with the parts of the trajectory (in black) that are associated to this reflector.*

6 Conclusions and Future Work

In this paper we devise a method for estimating a map of NLOS reflectors for a mobile radio network based on multistatic SAR imaging. We apply the back-projection principle for the image creation which is a well known method from computerised tomography and also conventional SAR imaging. The obtained images give promising results where the reflectors are well correlated with the large buildings detectable in the aerial map of the environment. Also, the possibility to extract which part of the trajectory contributed to different reflectors and from which base station is added. In this way a map where dominating reflectors are present can be built up.

The applicability of the estimated reflectors is still unexploited. One possible application area is to consider the estimated reflectors and occlusions in positioning estimation, (Gustafsson and Gunnarsson, 2005).

Bibliography

3GPP. Spatial channel model for Multiple Input Multiple Output (MIMO) simulations, (TS 25.996), 2012.

P. Almers, E. Bonek, A. Burr, N. Czink, M. Debbah, V. Degli-Esposti, H. Hofstetter, P. Kyosti, D. Laurenson, G. Matz, A. F. Molisch, C. Oestges, and H. Ozcelik. Survey of Channel and Radio Propagation Models for Wireless MIMO Systems. *EURASIP Journal on Wireless Communications and Networking*, 2007(1):019070, 2007. ISSN 1687-1499. doi: 10.1155/2007/19070. URL `http://jwcn.eurasipjournals.com/content/2007/1/019070`.

L. E. Andersson. On the Determination of a Function from Spherical Averages. *SIAM Journal on Mathematical Analysis*, 19(1):214–232, 1988. doi: 10.1137/0519016. URL `http://link.aip.org/link/?SJM/19/214/1`.

L. J. Cutrona, W. E. Vivian, E. N. Leith, and G. O. Hall. A High-Resolution Radar Combat-Surveillance System. *IRE Transactions on Military Electronics*, MIL-5(2): 127–131, April 1961. ISSN 0096-2511. doi: 10.1109/IRET-MIL.1961.5008330.

E. Dahlman, S. Parkvall, and J. Sköld. *4G LTE/LTE-Advanced for Mobile Broadband*. Academic Press. Academic Press, 2011. ISBN 9780123854896.

J. A. Fawcett. Inversion of N-Dimensional Spherical Averages. *SIAM Journal on Applied Mathematics*, 45(2):336–341, 1985. ISSN 00361399. URL `http://www.jstor.org/stable/2101820`.

T. Fügen, J. Maurer, T. Kayser, and W. Wiesbeck. Capability of 3-D Ray Tracing for Defining Parameter Sets for the Specification of Future Mobile Communications Systems. *IEEE Transactions on Antennas and Propagation*, 54(11):3127–3137, Nov. 2006.

F. Gustafsson and F. Gunnarsson. Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements. *Signal Processing Magazine, IEEE*, 22(4):41–53, 2005.

M. A. Jensen and J. W. Wallace. A Review of Antennas and Propagation for MIMO Wireless Communications. *IEEE Trans. on Antennas and Propagation*, 52(11):2810–2824, Nov. 2004.

V. Krishnan, J. Swoboda, C. E. Yarman, and B. Yazici. Multistatic Synthetic Aperture Radar Image Formation. *Image Processing, IEEE Transactions on*, 19(5):1290–1306, 2010. ISSN 1057-7149. doi: 10.1109/TIP.2009.2039662.

J. Medbo, M. Riback, and J.-E. Berg. Validation of 3GPP spatial channel model including WINNER wideband extension using measurements. In *In Proc. IEEE Vehicular Technology Conference*, 2006.

J. Medbo, Y. Jading, I. Siomina, and J. Furuskog. Propagation channel impact on LTE positioning accuracy: A study based on real measurements of observed time difference of arrival. In *Personal, Indoor and Mobile Radio Communications, 2009 IEEE International Symposium on*, pages 2213–2217, 2009. doi: 10.1109/PIMRC.2009.5450144.

F. Natterer. *The Mathematics of Computerised Tomography*. New York: Wiley, 1986.

C. Oliver and S. Quegan. *Understanding Synthetic Aperture Radar Images*. The SciTech Radar and Defense Series. SciTech, 2004. ISBN ISBN 1-891121-31-6.

Z. Sjanic, F. Gunnarsson, C. Fritsche, and F. Gustafsson. Cellular Network Non-Line-of-Sight Reflector Localisation Based on Synthetic Aperture Radar Methods. *Submitted to IEEE Transactions on Antennas and Propagation*, September 2013.

N. Willis. *Bistatic Radar*. SciTech, 2 edition, 2007.

PhD Dissertations
Division of Automatic Control
Linköping University

M. Millnert: Identification and control of systems subject to abrupt changes. Thesis No. 82, 1982. ISBN 91-7372-542-0.

A. J. M. van Overbeek: On-line structure selection for the identification of multivariable systems. Thesis No. 86, 1982. ISBN 91-7372-586-2.

B. Bengtsson: On some control problems for queues. Thesis No. 87, 1982. ISBN 91-7372-593-5.

S. Ljung: Fast algorithms for integral equations and least squares identification problems. Thesis No. 93, 1983. ISBN 91-7372-641-9.

H. Jonson: A Newton method for solving non-linear optimal control problems with general constraints. Thesis No. 104, 1983. ISBN 91-7372-718-0.

E. Trulsson: Adaptive control based on explicit criterion minimization. Thesis No. 106, 1983. ISBN 91-7372-728-8.

K. Nordström: Uncertainty, robustness and sensitivity reduction in the design of single input control systems. Thesis No. 162, 1987. ISBN 91-7870-170-8.

B. Wahlberg: On the identification and approximation of linear systems. Thesis No. 163, 1987. ISBN 91-7870-175-9.

S. Gunnarsson: Frequency domain aspects of modeling and control in adaptive systems. Thesis No. 194, 1988. ISBN 91-7870-380-8.

A. Isaksson: On system identification in one and two dimensions with signal processing applications. Thesis No. 196, 1988. ISBN 91-7870-383-2.

M. Viberg: Subspace fitting concepts in sensor array processing. Thesis No. 217, 1989. ISBN 91-7870-529-0.

K. Forsman: Constructive commutative algebra in nonlinear control theory. Thesis No. 261, 1991. ISBN 91-7870-827-3.

F. Gustafsson: Estimation of discrete parameters in linear systems. Thesis No. 271, 1992. ISBN 91-7870-876-1.

P. Nagy: Tools for knowledge-based signal processing with applications to system identification. Thesis No. 280, 1992. ISBN 91-7870-962-8.

T. Svensson: Mathematical tools and software for analysis and design of nonlinear control systems. Thesis No. 285, 1992. ISBN 91-7870-989-X.

S. Andersson: On dimension reduction in sensor array signal processing. Thesis No. 290, 1992. ISBN 91-7871-015-4.

H. Hjalmarsson: Aspects on incomplete modeling in system identification. Thesis No. 298, 1993. ISBN 91-7871-070-7.

I. Klein: Automatic synthesis of sequential control schemes. Thesis No. 305, 1993. ISBN 91-7871-090-1.

J.-E. Strömberg: A mode switching modelling philosophy. Thesis No. 353, 1994. ISBN 91-7871-430-3.

K. Wang Chen: Transformation and symbolic calculations in filtering and control. Thesis No. 361, 1994. ISBN 91-7871-467-2.

T. McKelvey: Identification of state-space models from time and frequency data. Thesis No. 380, 1995. ISBN 91-7871-531-8.

J. Sjöberg: Non-linear system identification with neural networks. Thesis No. 381, 1995. ISBN 91-7871-534-2.

R. Germundsson: Symbolic systems – theory, computation and applications. Thesis No. 389, 1995. ISBN 91-7871-578-4.

P. Pucar: Modeling and segmentation using multiple models. Thesis No. 405, 1995. ISBN 91-7871-627-6.

H. Fortell: Algebraic approaches to normal forms and zero dynamics. Thesis No. 407, 1995. ISBN 91-7871-629-2.

A. Helmersson: Methods for robust gain scheduling. Thesis No. 406, 1995. ISBN 91-7871-628-4.

P. Lindskog: Methods, algorithms and tools for system identification based on prior knowledge. Thesis No. 436, 1996. ISBN 91-7871-424-8.

J. Gunnarsson: Symbolic methods and tools for discrete event dynamic systems. Thesis No. 477, 1997. ISBN 91-7871-917-8.

M. Jirstrand: Constructive methods for inequality constraints in control. Thesis No. 527, 1998. ISBN 91-7219-187-2.

U. Forssell: Closed-loop identification: Methods, theory, and applications. Thesis No. 566, 1999. ISBN 91-7219-432-4.

A. Stenman: Model on demand: Algorithms, analysis and applications. Thesis No. 571, 1999. ISBN 91-7219-450-2.

N. Bergman: Recursive Bayesian estimation: Navigation and tracking applications. Thesis No. 579, 1999. ISBN 91-7219-473-1.

K. Edström: Switched bond graphs: Simulation and analysis. Thesis No. 586, 1999. ISBN 91-7219-493-6.

M. Larsson: Behavioral and structural model based approaches to discrete diagnosis. Thesis No. 608, 1999. ISBN 91-7219-615-5.

F. Gunnarsson: Power control in cellular radio systems: Analysis, design and estimation. Thesis No. 623, 2000. ISBN 91-7219-689-0.

V. Einarsson: Model checking methods for mode switching systems. Thesis No. 652, 2000. ISBN 91-7219-836-2.

M. Norrlöf: Iterative learning control: Analysis, design, and experiments. Thesis No. 653, 2000. ISBN 91-7219-837-0.

F. Tjärnström: Variance expressions and model reduction in system identification. Thesis No. 730, 2002. ISBN 91-7373-253-2.

J. Löfberg: Minimax approaches to robust model predictive control. Thesis No. 812, 2003. ISBN 91-7373-622-8.

J. Roll: Local and piecewise affine approaches to system identification. Thesis No. 802, 2003. ISBN 91-7373-608-2.

J. Elbornsson: Analysis, estimation and compensation of mismatch effects in A/D converters. Thesis No. 811, 2003. ISBN 91-7373-621-X.

O. Härkegård: Backstepping and control allocation with applications to flight control. Thesis No. 820, 2003. ISBN 91-7373-647-3.

R. Wallin: Optimization algorithms for system analysis and identification. Thesis No. 919, 2004. ISBN 91-85297-19-4.

D. Lindgren: Projection methods for classification and identification. Thesis No. 915, 2005. ISBN 91-85297-06-2.

R. Karlsson: Particle Filtering for Positioning and Tracking Applications. Thesis No. 924, 2005. ISBN 91-85297-34-8.

J. Jansson: Collision Avoidance Theory with Applications to Automotive Collision Mitigation. Thesis No. 950, 2005. ISBN 91-85299-45-6.

E. Geijer Lundin: Uplink Load in CDMA Cellular Radio Systems. Thesis No. 977, 2005. ISBN 91-85457-49-3.

M. Enqvist: Linear Models of Nonlinear Systems. Thesis No. 985, 2005. ISBN 91-85457-64-7.

T. B. Schön: Estimation of Nonlinear Dynamic Systems — Theory and Applications. Thesis No. 998, 2006. ISBN 91-85497-03-7.

I. Lind: Regressor and Structure Selection — Uses of ANOVA in System Identification. Thesis No. 1012, 2006. ISBN 91-85523-98-4.

J. Gillberg: Frequency Domain Identification of Continuous-Time Systems Reconstruction and Robustness. Thesis No. 1031, 2006. ISBN 91-85523-34-8.

M. Gerdin: Identification and Estimation for Models Described by Differential-Algebraic Equations. Thesis No. 1046, 2006. ISBN 91-85643-87-4.

C. Grönwall: Ground Object Recognition using Laser Radar Data – Geometric Fitting, Performance Analysis, and Applications. Thesis No. 1055, 2006. ISBN 91-85643-53-X.

A. Eidehall: Tracking and threat assessment for automotive collision avoidance. Thesis No. 1066, 2007. ISBN 91-85643-10-6.

F. Eng: Non-Uniform Sampling in Statistical Signal Processing. Thesis No. 1082, 2007. ISBN 978-91-85715-49-7.

E. Wernholt: Multivariable Frequency-Domain Identification of Industrial Robots. Thesis No. 1138, 2007. ISBN 978-91-85895-72-4.

D. Axehill: Integer Quadratic Programming for Control and Communication. Thesis No. 1158, 2008. ISBN 978-91-85523-03-0.

G. Hendeby: Performance and Implementation Aspects of Nonlinear Filtering. Thesis No. 1161, 2008. ISBN 978-91-7393-979-9.

J. Sjöberg: Optimal Control and Model Reduction of Nonlinear DAE Models. Thesis No. 1166, 2008. ISBN 978-91-7393-964-5.

D. Törnqvist: Estimation and Detection with Applications to Navigation. Thesis No. 1216, 2008. ISBN 978-91-7393-785-6.

P-J. Nordlund: Efficient Estimation and Detection Methods for Airborne Applications. Thesis No. 1231, 2008. ISBN 978-91-7393-720-7.

H. Tidefelt: Differential-algebraic equations and matrix-valued singular perturbation. Thesis No. 1292, 2009. ISBN 978-91-7393-479-4.

H. Ohlsson: Regularization for Sparseness and Smoothness — Applications in System Identification and Signal Processing. Thesis No. 1351, 2010. ISBN 978-91-7393-287-5.

S. Moberg: Modeling and Control of Flexible Manipulators. Thesis No. 1349, 2010. ISBN 978-91-7393-289-9.

J. Wallén: Estimation-based iterative learning control. Thesis No. 1358, 2011. ISBN 978-91-7393-255-4.

J. Hol: Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS. Thesis No. 1368, 2011. ISBN 978-91-7393-197-7.

D. Ankelhed: On the Design of Low Order H-infinity Controllers. Thesis No. 1371, 2011. ISBN 978-91-7393-157-1.

C. Lundquist: Sensor Fusion for Automotive Applications. Thesis No. 1409, 2011. ISBN 978-91-7393-023-9.

P. Skoglar: Tracking and Planning for Surveillance Applications. Thesis No. 1432, 2012. ISBN 978-91-7519-941-2.

K. Granström: Extended target tracking using PHD filters. Thesis No. 1476, 2012. ISBN 978-91-7519-796-8.

C. Lyzell: Structural Reformulations in System Identification. Thesis No. 1475, 2012. ISBN 978-91-7519-800-2.

J. Callmer: Autonomous Localization in Unknown Environments. Thesis No. 1520, 2013. ISBN 978-91-7519-620-6.

D. Petersson: A Nonlinear Optimization Approach to H2-Optimal Modeling and Control. Thesis No. 1528, 2013. ISBN 978-91-7519-567-4.