POLITECNICO DI MILANO
DIPARTIMENTO DI SCIENZE E TECNOLOGIE AEROSPAZIALI
DOCTORAL PROGRAMME IN ROTARY WING AIRCRAFTS

VISUAL AIDED NAVIGATION AND MAPPING FOR AN AUTONOMOUS ROTORCRAFT

Doctoral Dissertation of:
Mehmet Suat Kay

Supervisor:
Prof. Carlo L. Bottasso

Tutor:
Prof. Lorenzo Trainelli

The Chair of the Doctoral Program:
Prof. Sergio Ricci

2013 – XXIV

Acknowledgements

In the name of God, the Most Gracious, the Most Merciful.

First of all, I would like to thank my advisor Prof. Bottasso for guiding me towards the interesting and challenging subject which made me possible to author this thesis.

I appreciate the help and friendly conversations of my teammates during my studies in Italy: Koray, Ehsan and Pietro. I specially thank my flatmate Fatih for his friendship, not only in University, but in the free time we shared. I also thank Selçuk Bayraktar for valuable discussions we had about stochastic estimation.

In particular, I thank my family for their precious support during my whole education life.

Last, but not least, I would like to thank my wife Hümeyra Betül for her patience and encouragement during the last and toughest period of my study.

Abstract

Autonomous rotorcraft in a confined environment that is preventing it to use a global localization system will end up with wrong estimates of velocity and position after a short time of dead reckoning initialization due to the inherent and external noise on the inertial sensors. Extracting corners from an image that is taken from an onboard camera and tracking them in subsequent images can be used as a velocity reference for state estimation algorithm. Scale ambiguity that is present in a monocular camera can be removed by using a fixed baseline stereo camera system. A measurement model using this setup with relative positional measurements is described in the first part of this thesis. The noise in the pixel measurements are back projected into three dimensional space and after calculation of the distance travelled during image acquisition interval, the weight of each feature point for the measurement update is characterized by the covariance matrix. This matrix obtained by algebraic combination of uncertainties in horizontal and vertical projection uncertainty of left and right cemeras. Extended Kalman Filtering with state equations for position in fixed frame,velocity in body frame and quaternion is used. Covariance representation for attitude representation is handled by a body fixed error approach described using Gibbs vector. Assertive simulation results are presented showing the improvement of velocity state vector. The second part discusses the mapping with known poses problem for a robotic vehicle. This is half of a more general problem called Simultaneous Localization and Mapping (SLAM). Occupancy grid mapping for vehicles moving on a two dimensional planar space is explained by many authors. An algorithm using this technique and utilizing dense disparity maps from a stereo camera is described for three dimensional mapping for a vehicle with six degrees of freedom. The algorithm generates random particles with probability distribution functions in accordance with the unertainty presented in the vehicle pose and vision sensors. The ratio of points collected in voxels to the total number of particles scattered for the measurement instant is used for a recursive Bayesian update. The validity of the approach is presented with simulation results.

Contents

Nomenclature

$\mathbf{a}$	Acceleration
$\mathcal{B}$	Body reference frame
$\mathcal{C}$	Camera reference frame
$\mathcal{E}$	Fixed reference frame
$\mathbf{E}$	Essential matrix
$\mathbf{e}_H$, $\mathbf{e}_V$, $\mathbf{e}_F$	Camera frame unit vectors
$\mathbf{e}_N$, $\mathbf{e}_E$, $\mathbf{e}_D$	Fixed frame unit vectors
$\mathbf{e}_X$, $\mathbf{e}_Y$, $\mathbf{e}_Z$	Body frame unit vectors
$\mathbf{F}$	Fundamental matrix
$\mathbf{F}$	System dynamics matrix
f	Focal length
$\mathbf{G}$	State-noise coupling matrix
$\mathbf{g}$	Gravitational acceleration vector
$\mathbf{H}$	Measurement sensitivity matrix
$\mathbf{I}$	Identity matrix
$\mathbf{K}$	Kalman gain matrix
m	Occupancy grid
m^i	Occupancy grid cell
m_k	Occupancy grid at instant k
$\mathbf{K}$	Camera calibration matrix

$\mathbf{P}$	Point in three dimensional space
$\mathbf{P}$	State covariance matrix
$\mathbf{p}$	Position
$\mathbf{p}$	Projection of a point on two dimensional image plane
$P(x)$	Probability of x
$p(x)$	Probability density function of x
$\bar{q}$	Quaternion
$\mathbf{Q}$	Process noise covariance matrix
$\mathbf{R}$	Measurement noise covariance matrix
$\mathbf{R}_{\mathcal{A}}^{\mathcal{B}}$	Rotation matrix from frame $\mathcal{A}$ to frame $\mathcal{B}$
$\mathbf{v}$	Velocity
$\mathbf{x}$	State vector
x_k	Pose at instant k
$\mathbf{z}$	Measurement vector
z_k	Measurement at instant k
z_k	Measurement at instant k
$\boldsymbol{\theta}$	Rotation vector
$\boldsymbol{\rho}$	Gibbs vector
$\boldsymbol{\Sigma}$	Covariance matrix
$\boldsymbol{\omega}$	Angular velocity

CHAPTER 1

Introduction

High flying unmanned air vehicles are free from static obstacles that can prevent them navigating safely during their missions. Meanwhile, the standard approach for estimating the vehicle state is fusing inertial sensor data with GPS signal. Although there are different techniques for fusion of these two, an UAV design for high flying missions always starts with this template for state estimation.

Rotary wing unmanned vehicles offer an alternative possible mission range with their vertical flight capabilities for scattered environments. These include search and rescue, law enforcement, examination and intervention in hazardous zones. These missions requires a complete or partial flight path which is proximate to obstacles. As a result, a safe flight becomes a more complicated problem for these vehicles with two reasons.

Firstly, the vehicle must identify a safe space to fly during its course. This can be possible with a map supplied to the navigation planner of the system before the flight. In that case, the vehicle knows where it is flyable and creates its path to be followed with an optimization of flight parameters.

Afterward, the localization problem exists. A flight path within a scattered environment means at least a degradation of the global positioning signal or a complete loss of it. In this case, the integration of inertial sensors accumulates the errors and a drift occurs from the actual path. Especially with cheaper strap-down sensors used in small UAVs, this error grows quickly, and the position data become useless. As a result, even the system has a very precise map of the surroundings, a safe navigation is not possible.

The aim of this thesis is to define and validate a framework for generating three dimensional occupancy grids which have just enough resolution to aid path planning and exploration tasks of a robotic rotorcraft. The rotorcraft is assumed to depend on its inertial sensors and vision information from an on-board stereo camera setup for state

estimation and map construction.

The problem to be addressed in this thesis can be described as follows: Using strapdown inertial measurements and an on-board stereo camera with a fixed baseline and position with respect to the airframe, localize the aircraft accurately with sufficient metric precision locally on the correct branch of a topological map description with a crude distance calculation to the nodes. Meanwhile construct three dimensional metric grids supplying unoccupied space information relative to the nodes which can be used for local path planning and obstacle avoidance.

The framework to be constructed is based on the following assumptions

- The air vehicle is an autonomous rotary wing robotic aircraft with onboard avionics capable of processing and storing adequent information in real time.
- The aircraft moves in a quasistatic manner, such that acceleration measurements can be used to estimate roll and pitch angles. Also the aircraft does not move in an agile manner, thus allowing use of image information for motion estimation. This smooth behaviour also dictates a limited roll and pitch angle during all phases of the operation.

The aim of the study is to design the system such that

- The mapping and localization algorithm runs on a limited sensor capability and computational sources.
- The aircraft can make a robust localization within a given map
- The aircraft can build a sufficiently dense three dimensional occupancy grid with stereo camera measurements

1.1 Contributions

This thesis deals with two important aspects in the navigation problem of an autonomous vehicle: state estimation and mapping. In particular case, a rotorcraft equipped with inertial measurement unit and a stereo camera setup, vision can be utilized in order to fulfill these two functions. The main contributions of the thesis actualized as stated in the following paragraphs:

- A detailed analysis of the state estimation algorithm which is using visual features as velocity measurements based on Extended Kalman Filtering technique is presented. Problems occurring due to the unobservable position and velocity states are explained and decoupling of the state vector is proposed as a solution. Uncertainty characteristics of velocity measurements gathered from visual features that are defined as ratio of two normally distributed random variables are approximated again as normally distributed using Rao's procedure.
- A three dimensional occupancy grid mapping algorithm using disparity maps as measurements are developed. The algorithm is based on random sampling of particles into the three dimensional space and the density of particles per voxel is used to calculate the probability of the occupation of the voxels according to the instant measurement. Instant probabilities are applied to the occupancy grid map in the regular recursive occupancy grid mapping algorithm.

CHAPTER 2

System Description

The STAR (Simulation and testbed for Autonomous Rotorcraft) project in the Department of Aerospace of Politecnico di Milano aims for a system with full autonomy of operation within a scattered environment like urban canyons or indoors by using relatively inexpensive hardware setup. The evaluation of the system can be traced in the PhD theses of Savini and Leonello [29, 37]. Savini's work describes the experimental test-bed with on board software and following the initial piloting test, simulation environment plant model, navigation, control algorithms and sensor fusion are explained. A non-linear model predictive controller proposition for the helicopter flight concludes this study. Leonello's work includes a mechanical improvement of the connection between the on-board computer and the helicopter fuselage in order to reduce vibrations transferred to the sensors. A vision based sensor fusion algorithm for state estimation is explained and simulation results that are done using synthetic points as features are followed by results using features extracted from images obtained from Gazebo robot simulator program. The study concludes with a path planning algorithm regarding the helicopter flight dynamics and flight envelope protection. Increase in the sensor weight required the base helicopter model changed. In this chapter, the current status of the experimental model will be explained with details of the sensors that are the related to the study that is done in this thesis.

2.1 Experimental Test-Bed

The test-bed consists of a model helicopter with a classical single main an tail rotor configuration, an on-board computer equipped with sensors and a ground control station. Since the last system description made by Leonello, some updates have been done both on the helicopter and on-board computer.

The model helicopter used is a Vario Benzin Trainer with a maximum take off weight around 11 kg [1]. It has a modified landing gear in order to provide room for the on-board computer box.

Onboard computer includes a PC104 industry standard computer using a CPU of 1.6 GHz clock rate. The architecture of the CPU is equivalent to a Pentium III. A Diamond-MM-16-AT data acquisition card is used for analog sensors. These sensors include a three dimensional accelerometer (Xbow - CLX10LP3) and three one axis gyroscopes (Silicon Sensing - CRS05) which are used in the homemade inertial measurement unit, a three dimensional magnetometer (CXM113) and the sonar sensor (Senscomp 6500). A Thales Navigation-A12 receiver supplies GPS position and velocity solution to the computer with an active antenna over serial communication port. Peripherals on the PCI bus are an Ethernet card, an IEEE 1394 interface and an RS232 expansion card. A MultimediaCard is used as a storage medium.

External communication with the ground computer is provided by a router which has a WiFi antenna. QGroundControl runs on the ground control station which is a laptop computer equipped with WiFi adapter [3].

Two sets of stereo camera used interchangeably for visual sensing. First one is a Bumblebee XB3 which is a three sensor multi-baseline camera systems supplying a single image per trigger including all sensor data at once. Its widest baseline is 24 centimeters. The other one is a home built system which includes two Unibrain Fire-i board pro VGA cameras with a baseline equalized to the widest XB3 baseline [5]. Both cameras communicate with the on-board computer over IEEE 1394 interface. Synchronization between two cameras in the later setup is realized using a broadcast software trigger over the IEEE1394 interface.

A redundant inertial measurement unit was added to the system in order to be able to have a comparison between the home-made unit and the commercial one and XSend MTi was chosen for this purpose [6]. It can supply calibrated acceleration, angular rate and magnetic field measurements or as an Attitude Heading Reference System it gives attitude information using its internal Extended Kalman Filtering algorithm. In the following sections, output from this IMU was utilized within the test results, so its specifications can be seen in Table2.1 and 2.2.

Table 2.1: *XSens Mti Technical Specifications as an AHRS*

	Specification
Static accuracy (roll/pitch)	<0.5 deg
Static accuracy (heading)	<1 deg
Dynamic accuracy	2 deg RMS
Angular resolution	0.05 deg

Table 2.2: *XSens Mti Technical Specifications for inertial sensors*

	Specification
Acceleration range	$\pm 50\,\mathrm{m/s^2}$
Acceleration measurement noise	$\pm 0.002\,\mathrm{m/s^2}/\sqrt{\mathrm{Hz}}$
Angular rate range	$\pm 300\,\mathrm{deg/s}$
Gyroscope noise	$\pm 0.05\,\mathrm{deg/s}/\sqrt{\mathrm{Hz}}$

Figure 2.1: *STAR Experimental testbed*

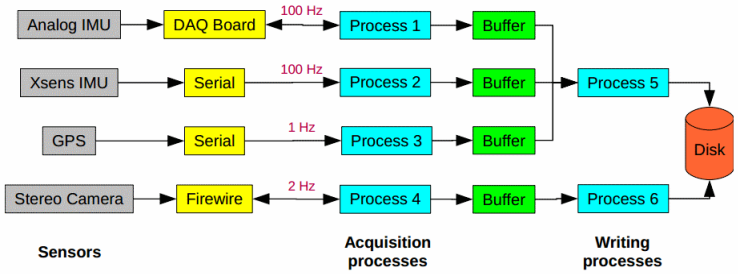

Figure 2.2: *Multi thread on-board sotware data flow diagram*

The onboard computer mounted on the helicopter can be seen in Figure 2.1.

2.2 Data Acquisition

Tests results obtained in this thesis depend on off-line processing of recorded data from on-board sensors and images taken from the cameras. In order to gather this data, a multi-threading software was coded whose data flow diagram is shown in Figure 2.2. The onboard computer has a Debian distribution Linux operating system. The kernel of the operating system was patched with Real Time Application Interface (RTAI) in order to have deterministic time intervals for periodic tasks [4].

During data acquisition, some problems were encountered related to data flow from computer random access memory (RAM) to disk and from camera to RAM. The first problem is related to the writing speed of the MMC card, and will be called as write congestion and the second is related to interrupts caused by the load in the PCI bus and will be called as image tearing. A brief description will be made for both them in the following paragraphs

2.2.1 Disk write congestion

The first hardware constraint in the implementation is that, when images are desired to be saved on the MMC card on the computer, the write cache allocated by the operating system was choked because of the data acquisition speed was exceeding disk writing speed. Linux, which is the operating system used during the study, uses a disk cache in order to be able to deal with fast write requests by keeping the data in an allocated space in the random access memory. When the cache is about to become filled or after a time interval has passed since the last write request, the data is written onto the physical disk. During this operation, write operations are rejected and data send for writing can be lost.

This operational bottleneck prevented acquisition of images with full camera frame rate available, because when the flush operation starts, images was lost and offline processing cannot be done for these segments. Linux has kernel parameters related to the size of the disk write cache and percentage value for starting to flush the data to the disk. Nevertheless, changes in these parameters did not end in the desired solution because of the hardware limitation. At the end, the disk was mounted not in asynchronous mode, instead in synchronous mode. This mode prevents the operating system using a cache as an intermediate step and write operations are directed to the disk immediately. With this configurations 4 images of 640 x 480 pixel resolution were saved to the disk per second. For a stereo pair, this means a 2 Hz image acquisition rate, which is used as a base for the tests.

2.2.2 Image tearing

Another difficulty encountered during the tests is the image tearing problem. The symptom is observed as a termination of one image acquisition prematurely and the remaining part of the image is filled with the next image taken. It is seen as a horizontal line separating two non-complete subsequent images as seen in Figure 2.3.

This problem is described in the technical note of the camera manufacturer [35] . This note relates the problem to an intense interrupt generation on the shared PCI bus. 1394 card on this bus is affected by this intensity and the image transfer over the 1394 cable is terminated before all the pixels in the camera frame is transferred to the PC.

Result quality of algorithms using image processing techniques depends on the consistency of scene projection. The problem described spoils the projection both in space and time. Feature points obtained from this image will evidently result in a faulty traction and disparity maps generated will have a wrong occupancy sense. With this manner, torn images must be identified and rejected before feeding it to the algorithm. It is also important to minimize this problem as much as possible.

Image tearing is easily recognizable by human eye with a distinct separation. But for computer vision, it needs another processing step in order to detect the occurrence of the problem. But with a limited computational power and constrained time, it is not wise to go on this way. One observation made during the tests is that, when this problem occurs, image transfer is not completed in the usual expected time. As this happens, the current image cannot be acquired and it is evident that the next image will be corrupt one. By using this diagnostic, faulty images are safely removed from the image stream.

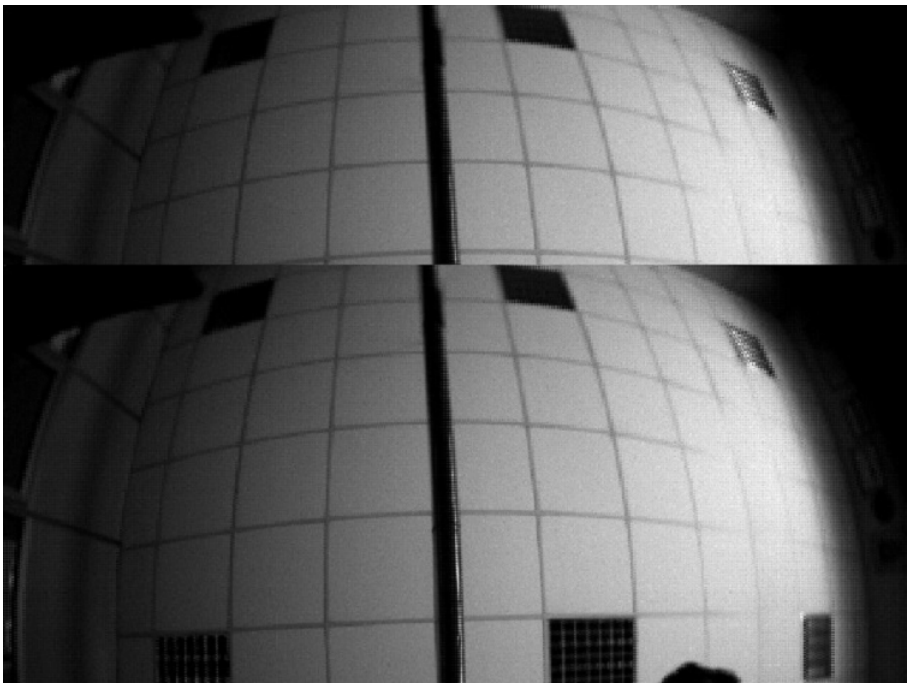

Figure 2.3: *Image tearing problem*

As told before, the problem depends on the shared bus configuration on the hardware and our experience directed us to the communication hardware as the source of the problem. When the amount of data being transferred to the ground is large, Ethernet card holds the bus busy and its interrupts prevents the 1394 card working healthy. Reducing the amount of data that is periodically transferred over the wireless communication made the returned torn image rate a scarce one.

2.3 Computing Performance

In order to prove that the image processing algorithms can run on the on-board computer with a reasonable rate, a code including a series of operations are executed. Results obtained for image acquisition, remapping for camera distortion removal, feature tracking, dense disparity mapping, motion estimate with RANSAC and image write operations can be seen on Figure 2.4 for a series of images with 320x240 pixels resolution. Average cycle time was obtained around 350 milliseconds for this test. The peak in the middle of the graph is the result of disk write congestion problem mentioned earlier.

2.4 Image Rectification

Camera lenses are a product of camera design for a sharper image with a high amount of light allowed inside the aperture. Unless a well optimized rectilinear camera lens is used, images taken from a camera will have a distortion, warping straight lines with a curvature. Especially cameras with a basic lens setup or with a wide field of view, the distortion becomes more evident. In order to be able to use a pinhole camera model

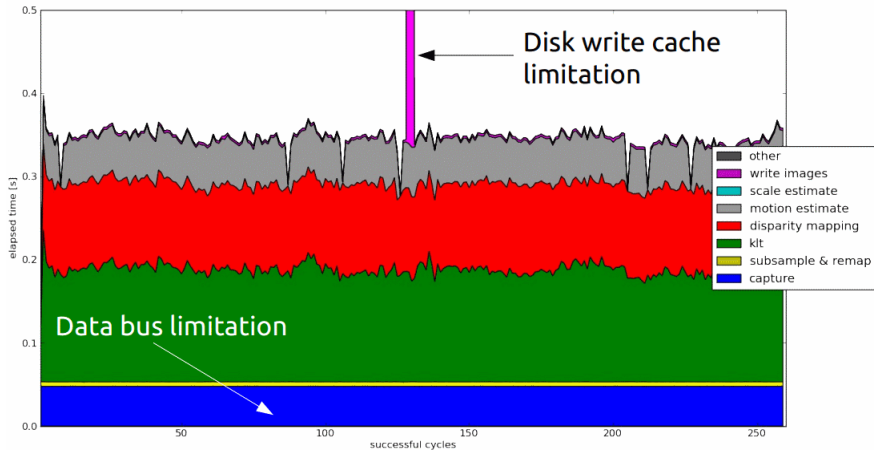

Figure 2.4: *Time elapsed for a series of image operations on the on-board computer*

for scene reconstruction or motion estimation, this distortion must be removed beforehand. The radial distortion model explained in [41] is used and parameters for our own setup was determined using Camera Calibration Toolbox for Matlab [10] . In order to minimize the computational cost relating to image rectification, a look up table was generated for rectification operation instead of using the calibration formula. Remap function from OpenCV library with linear interpolation was used for the final rectification [11].

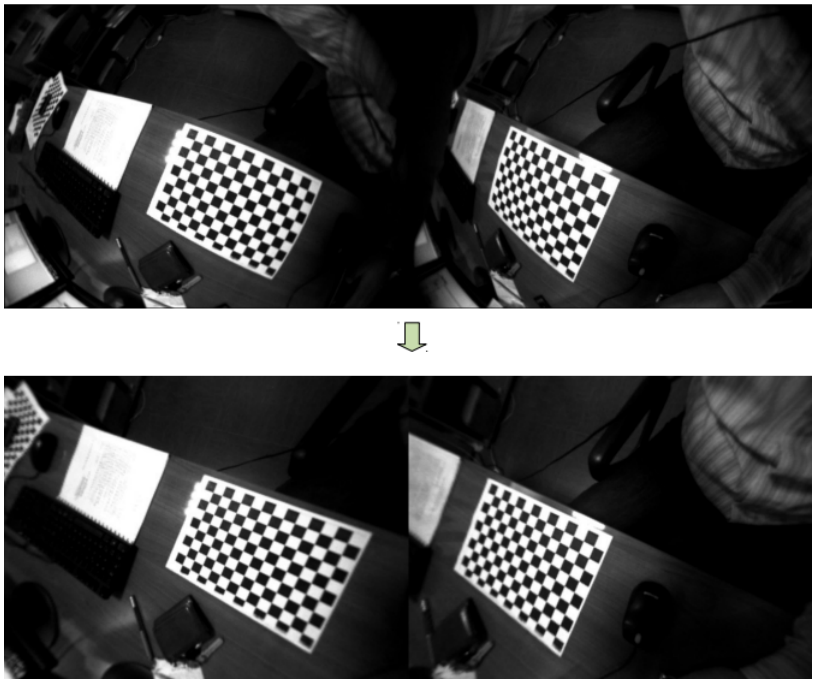

Figure 2.5: *Image rectification and camera calibration*

CHAPTER 3

EKF for Visual Odometry Using Stereo Vision

In order to fulfill a spatial map that can be used for guidance and obstacle avoidance, a robotic vehicle needs to know its pose within a certain degree of uncertainty. A 6 degrees of freedom pose consists of the position in $\mathbb{R}^3$ and orientation in $SO\left(3\right)$. An autonomous rotorcraft includes an inertial measurement unit, which outputs acceleration and angular rates, and its measurements can be integrated to calculate velocity and orientation. A second integration on velocity results in position.

Low cost MEMS inertial sensors that are used in mini robotic air vehicles are noisy and commercial grade accelerometers and gyroscopes have bias stability greater than 50 milli g and 1 deg/s respectively [7]. As a result, noise in their measurement is observed as a random walk in their integrations and an unbounded error accumulates in state variables .

For outdoor applications, where sky is visible enough, measurements from satellite based navigation systems like Global Positioning System (GPS) and Global Navigation Satellite System (GLONASS) can be fused to the inertial measurements in order to bound these errors using Extended Kalman Filtering (EKF). With this technique, it is also possible to estimate time varying inertial sensor characteristics like bias and scale factor. The integration of these two systems can be realized either using a position and velocity solution from GPS or directly taking pseudo-ranges to the satellites [38]. Simply using navigation satellite codes gives a positioning accuracy of less than 15 meters, which can be reduced to sub-meter accuracy using Differential GPS (DGPS) and even sub-decimeter range with Real Time Kinematic (RTK) systems [40].

In cluttered environments, urban canyons or indoors, performance of the satellite based systems degrades as the number of visible satellites diminish in contrary to increasing obstacles to be avoided. Several sensors can be used to detect surrounding objects relative to a robotic vehicle including sonar, radar and lidar rangers. Sophisti-

cated algorithms utilizing these sensors are demonstrated to achieve simultaneous localization and mapping (SLAM) solutions. While sonar and radar sensors offer range information of the nearest object within its measurement cone, lidars supply a point-wise sensing. Arrangement of multiple sensors in a regular pattern allows a wider perception. Scanning lidar systems allows a very high resolution of reconstruction of the environment being examined.

Optical cameras, contain a rich amount of information about objects in its field of view. Several techniques are proposed to be used for navigation, guidance and control of autonomous vehicles using vision. The practice spans almost all kind of robotic applications including autonomous airborne, underwater and terrain vehicles. Points, lines, planes and patterns in the image frames are used as features to be tracked. Position and orientation change rates can be detected from subsequent frames and previously visited places can be recognized from identified features by using advanced algorithms.

Because of the two-dimensional projection of three-dimensional space, scale ambiguity problem prohibits direct sensing of metric distances. This ambiguity can be overcome by using a pair of camera with known baseline and relative orientation. A common configuration of two cameras with identical orientation and coincident plane perpendicular to the image plane is called a stereo camera system. Dense disparity maps acquired by processing images taken at the same instant from a stereo set provides depth information to a degree. The uncertainty arises because of optical and electrical noise and discrete pixelization of digital camera sensors.

In this section, Extended Kalman Filtering technique will be used to estimate velocity of a vehicle with 6 degrees of freedom in body frame using measurements acquired from inertial sensors and a stereo camera system. This system is assumed to have a fixed baseline and parallel focal lines and projection planes with a pinhole camera model.

3.1 Frames of Reference

In this study, three frames of reference is used. A Cartesian fixed frame $\mathcal{E}$, which has origin at an arbitrary point on Earth and defined by orthogonal unit vectors $\mathbf{e}_N$ pointing to true North, $\mathbf{e}_E$ pointing to East and $\mathbf{e}_D$ pointing down parallel to the normal of Earth surface at origin defined by WGS84 ellipsoid [13]. This frame of reference is widely used in air vehicle state estimation for flight routes that are small enough for neglecting the non-planar surface and variable gravitational field of the Earth.

Body frame $\mathcal{B}$ has its origin on the center of mass of the vehicle. Unit vector$\mathbf{e}_X$ directs toward the point where the nose of the air vehicle points to. $\mathbf{e}_Y$ is aligned with starboard direction and $\mathbf{e}_Z$ conforms to the right hand rule and points down with respect to the vehicle. This is the common definition in aviation.

Camera frame $\mathcal{C}$ is defined in accordance with the definition in the computer vision literature. $\mathbf{e}_H$ and $\mathbf{e}_V$ defines the horizontal and vertical components in image frame and $\mathbf{e}_F$ extends from the focus of the camera to the depth of the scene. The image plane is defined to be in the field of view of the camera. All frames of reference mentioned are shown in Figure3.1.

For the system being analyzed, $\mathcal{C}$ is fixed with respect to the $\mathcal{B}$. Transformations between $\mathcal{E}$, $\mathcal{B}$ and $\mathcal{C}$ are done using direction cosine matrices. They can also be called

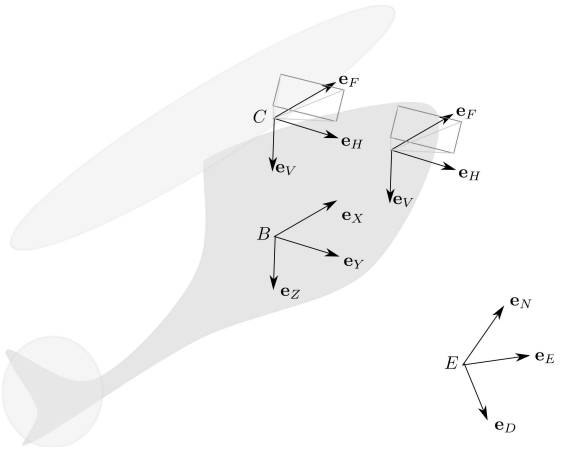

Figure 3.1: *Frames of reference*

as rotation matrices. For instance, pre-multiplying a vector defined in $\mathcal{E}$ with $\mathbf{R}_{\mathcal{E}}^{\mathcal{B}}$results in the same vector with its definition in $\mathcal{B}$. The inverse transformation is realized using pre-multiplication with $\mathbf{R}_{\mathcal{B}}^{\mathcal{E}}$ which is the transpose of $\mathbf{R}_{\mathcal{E}}^{\mathcal{B}}$. As stated before $\mathbf{R}_{\mathcal{C}}^{\mathcal{B}} = \left(\mathbf{R}_{\mathcal{B}}^{\mathcal{C}}\right)^T$ = constant is assumed in this thesis. Whenever a stereo set is used, both left and right cameras are oriented in $\mathcal{C}$. The origin of the $\mathcal{C}$ will be the left camera focus.

3.2 Rotation Formalism

According to Euler's theorem, a rotation can be identified using just three parameters. Euler angles and Rodrigues parameters are three parameter definitions, but some configurations yield singular cases for them. Rotation matrices, another form of rotation representation, are free from this defect, but 9 parameter definition constrained with orthonormal column and row requirements makes it a less preferred formalism.

Unit quaternions, on the other hand, are 4 parameter definitions without singularity problem. The unit magnitude constraint can be kept in numerical computations easier than a rotation matrix representation. Also consecutive rotations are calculated using quaternion algebra with less computational cost compared to rotation matrices. Because of these advantages, unit quaternions are generally the preferred rotation representation in state estimation problems.

Using Euler-Rodrigues parameters, a rotation can be described by a single rotation around an axis. If this axis is defined as a unit vector$\mathbf{e}_\theta$ and the amount of rotation is θ , then the rotation vector $\boldsymbol{\theta}$ is

$$\boldsymbol{\theta} = \theta \mathbf{e}_\theta. \tag{3.1}$$

Another similar form is defined as

$$\boldsymbol{\rho} = \tan\left(\frac{\theta}{2}\right)\mathbf{e}_\theta \tag{3.2}$$

where$\boldsymbol{\rho}$ is Gibbs vector. This form has some advantages while being used in Extended Kalman Filtering and it will be explained in following subsections.

Unit quaternion representation has a simple arithmetic for rotational transformations and does not suffer from discontinuity problems. While there are four components in a

quaternion, the unit magnitude constraint dictates the number of indepentent variables to three. The algorithms using this representation must be implemented considering this constraint. . Unit quaternion representation $\bar{q}$ can be related to the rotation vector by (3.3).

$$\bar{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\left(\theta/2\right) \\ q_1 \sin\left(\theta/2\right) \\ q_2 \sin\left(\theta/2\right) \\ q_3 \sin\left(\theta/2\right) \end{bmatrix} \tag{3.3}$$

If the result of n successive rotations are to be calculated, then quaternion multiplication defined in the Appendix A can be applied as

$$\bar{q} = \bar{q}_{[1]} \otimes \bar{q}_{[2]} \otimes ... \otimes \bar{q}_{[n]} \tag{3.4}$$

If $\bar{q}$ represents the orientation of the vehicle in fixed frame, then rotation matrix transforming vectors from $\mathcal{E}$ to $\mathcal{B}$ can be calculated using:

$$\mathbf{R}_{\mathcal{E}}^{\mathcal{B}} = \left(\mathbf{R}_{\mathcal{B}}^{\mathcal{E}}\right)^T = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2\left(q_1 q_2 + q_0 q_3\right) & 2\left(q_1 q_3 - q_0 q_2\right) \\ 2\left(q_1 q_2 - q_0 q_3\right) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2\left(q_2 q_3 + q_0 q_1\right) \\ 2\left(q_1 q_3 + q_0 q_2\right) & 2\left(q_2 q_3 - q_0 q_1\right) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix} \tag{3.5}$$

If rotation matrices are to be used for calculating n successive rotations, then later rotations must pre-multiply its predecessors:

$$\mathbf{R} = \mathbf{R}_{[n]} ... \mathbf{R}_{[2]} \mathbf{R}_{[1]} \tag{3.6}$$

3.3 Inertial Navigation

Inertial navigation solely depends on the accelerometer and gyroscope measurements. Earlier inertial measurement units (IMU) used a gimbal platform to hold accelerometers aligned with an inertial frame, and corrections to this alignment were made according to the angular rates sensed. Low cost MEMS systems of today are strap-down systems, in which inertial sensors are fixed on the body frame. Because the accelerometers move with the body frame, one must transform the acceleration vector to the fixed frame.

As the vehicle will sit on the ground or will be held in the air by the aerodynamic lift force, the gravity field will not be sensed and the reaction force from the ground or the lift force will be sensed as an acceleration in opposite direction of the gravity field. In order to calculate exact acceleration with respect to the fixed frame $\mathcal{E}$, gravitational acceleration must be added to the measurements. If the acceleration sensed by a strap-down accelerometer is denoted by $\tilde{\mathbf{a}}^{\mathcal{B}}$, the acceleration in fixed frame will be

$$\mathbf{a}^{\mathcal{E}} = \mathbf{R}_{\mathcal{B}}^{\mathcal{E}} \tilde{\mathbf{a}}^{\mathcal{B}} + \mathbf{g}^{\mathcal{E}}, \tag{3.7}$$

or it can be expressed in body frame as

$$\mathbf{a}^{\mathcal{B}} = \tilde{\mathbf{a}}^{\mathcal{B}} + \mathbf{R}_{\mathcal{E}}^{\mathcal{B}} \mathbf{g}^{\mathcal{E}}, \tag{3.8}$$

For sole inertial navigation, the change of orientation over time can be described using quaternions as

$$\frac{d}{dt}\bar{q}\left(t\right) = \frac{1}{2}\bar{q}\left(t\right) \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix}. \tag{3.9}$$

Here $\boldsymbol{\omega}$ represents the angular rate of the body frame. For small time increments τ, $\boldsymbol{\omega}$ can be assumed to be constant over sampling time and exact solution to 3.9 is given as [28]

$$\bar{q}\left(t+\tau\right) = \left[\cos\left(\frac{|\boldsymbol{\omega}\tau|}{2}\right)\mathbf{I}_{4\times 4} + \frac{\tau}{|\boldsymbol{\omega}\tau|}\sin\left(\frac{|\boldsymbol{\omega}\tau|}{2}\right)\Omega\left(\boldsymbol{\omega}\right)\right]\bar{q}\left(t\right). \tag{3.10}$$

where $\mathbf{I}$ is the identity matrix and $\Omega(\boldsymbol{\omega})$is defined as

$$\Omega\left(\boldsymbol{\omega}\right) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix}. \tag{3.11}$$

A first order Euler integration can also be used instead of 3.10 like

$$\bar{q}\left(t+\tau\right) = \bar{q}\left(t\right) + \tau\left[\frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}\tau \end{bmatrix} \otimes \bar{q}\left(t\right)\right]. \tag{3.12}$$

In both cases, renormalization of quaternion is required in order to prevent numerical error accumulation on the quaternion state.

The error in the orientation estimation will emerge because of the noise in the angular rate measurements. As a result $\mathbf{R}_{\mathcal{B}}^{\mathcal{E}}$ will not reflect the real orientation of the body. Considering the magnitude of gravitational acceleration being around $9.81^{\mathrm{m}}/_{s^2}$, it is seen that the orientation error will project part of this acceleration onto wrong axes. In order to prevent this, external references like gravity and magnetic field, can be used to correct orientation estimates [23, 33]. Even in the case of a perfect orientation estimation, accelerometer noise will be integrated twice and will affect both velocity and position estimates.

Velocity in body frame is more important than the velocity in fixed frame, because motion of a robotic vehicle is generally driven by the velocity command input. A closed loop control system needs the velocity output in body frame. For aerial vehicles, air velocity is more important than the absolute velocity of the vehicle. But in case of a rotary wing vehicle in a confined environment, absolute velocity becomes as important because of a crash possibility to an obstacle. Rate of change of the velocity in body frame can be expressed as

$$\dot{\mathbf{v}}^{\mathcal{B}} = \tilde{\mathbf{a}}^{\mathcal{B}} + \mathbf{R}_{\mathcal{E}}^{\mathcal{B}}\mathbf{g}^{\mathcal{E}} - \boldsymbol{\omega} \times \mathbf{v}^{\mathcal{B}}. \tag{3.13}$$

Rate of change of the position in fixed frame can be simply obtained by transforming the velocity vector from body frame to the fixed frame:

$$\dot{\mathbf{p}}^{\mathcal{E}} = \mathbf{R}_{\mathcal{B}}^{\mathcal{E}}\mathbf{v}^{\mathcal{B}}. \tag{3.14}$$

3.4 Visual Odometry

Egomotion of a vehicle can be judged according to the motion of features in the environment projected onto an image plane fixed to the body frame. This process is called visual odometry. Techniques using single cameras, omnidirectional cameras and multi-camera setups have been shown to be feasible.

Motion estimation depends finding position of the camera between two time steps. This can be realized by identifying and matching correspondent feature points between the images gathered at these time instants. For the general case, the points projected onto camera frames from two different views as $\mathbf{p}'$ and $\mathbf{p}$ onto a pinhole camera will satisfy the equation

$$\mathbf{p}'\mathbf{F}\mathbf{p} = 0 \tag{3.15}$$

Here $\mathbf{F}$ is fundamental matrix with size 3×3. $\mathbf{p}'$ and $\mathbf{p}$ are described in homogenous coordinates [24]. This equation does not depend on the intrinsic camera calibration parameters and can be used with different camera parameters for the former and later point values. If a single camera is used with fixed calibration parameters during time, essential matrix $\mathbf{E}$ can be used instead of $\mathbf{F}$. Relation between $\mathbf{F}$and $\mathbf{E}$ is given as

$$\mathbf{F} = \mathbf{K}'\mathbf{E}\mathbf{K} \tag{3.16}$$

where camera calibration matrix is defined as

$$\mathbf{K} = \begin{bmatrix} f_x & \alpha & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.17}$$

Camera calibration matrix includes focal lengths f_x , f_y, skewness α and projection plane center c_x,c_y. For a digital camera with square pixels with no skewness and projection plane center at origin will have its calibration matrix as

$$\mathbf{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.18}$$

In order to extract fundamental matrix from a given set of matched feature points at least 7 points are needed. But in practical applications mismatches in the acquired feature point set can lead a wrong solution or result in no solution at all. Outliers in the feature matches must be discarded before using these points in feature point extraction. Most used technique for filtering out outliers is Random Sample Consensus abbreviated as RANSAC [18]. One of the RANSAC derivatives that is using match quality values named Progressive Sample Consensus can be found in Chum's work [14]. Outlier detection phase can be followed by a guided matching for rearranging the match set and increasing correct matches.

For a vehicle with a motion estimation prior to image matching, guided matching can be initiated without using feature points. For fixed camera calibration, fundamental

matrix can be calculated using the displacement and rotation with respect to the older frame using the relation

$$\mathbf{F} = \mathbf{K}^{-T} \left[\mathbf{R}_{\mathcal{C}(t)}^{\mathcal{C}(t+\tau)} \mathbf{c}^{\mathcal{C}(t)} \right] \mathbf{R}_{\mathcal{C}(\mathbf{t})}^{\mathcal{C}(\mathbf{t}+\tau)} \mathbf{K}^{-1} \tag{3.19}$$

where $\mathbf{c}$ is the distance from the camera center in the former time step to the later one.

Fundamental matrix can be decomposed in order to get rotation and displacement values between camera centers in subsequent time frames with scale ambiguity. This ambiguity can be resolved either by using the feature point positions in three dimensional space as unknowns [27] or using a fixed baseline stereo camera setup [26].

3.5 Extended Kalman Filtering (EKF)

Extended Kalman Filtering (EKF) technique is a very common used form of state estimation technique for navigation and control problems. After the introduction of linear Kalman Filter by Rudolf Emil Kalman in 1960, Stanley F. Schmidt applied this to the nonlinear problem of trajectory estimation in Apollo project [22]. Since then it became the preferred method for estimation problems in aerospace beside many other applications. The most common form of EKF used in aerospace applications is INS/GPS integration [38].

EKF is also used in Simultaneous Localization and Mapping (SLAM) problems. In EKF-SLAM, the state vector of the robotic system is augmented with the feature positions and they are estimated concurrently with the pose of the vehicle.

For the purpose of this thesis, EKF is used to fuse information gathered from inertial sensors, with the known reference vector of gravity and visual odometry information. The state vector solely includes the pose of the vehicle with respect to the starting pose of the vehicle and feature positions are not included in the state vector. Features are just tracked in consecutive image frames and an average velocity is calculated regarding to the visual flow of these points. This technique was used by Leonello and it was shown that the visual information helped to improve accuracy of state variables and a better estimation in case of

3.5.1 System Description

The state vector of the system is defined by combining position and velocity vectors with the quaternion. As it can be seen, the vector consists of only the state of the body frame and does not contain any calibration parameters like accelerometer and gyro biases. Position will be defined in fixed frame $\mathcal{E}$, whereas velocity will be in the body frame $\mathcal{B}$. The state vector can be written in mathematical form as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}^{\mathcal{E}} \\ \hline \mathbf{v}^{\mathcal{B}} \\ \hline \bar{q} \end{bmatrix}. \tag{3.20}$$

Equations 3.9, 3.13 and 3.14 with white noise terms for accelerometer and angular

rate sensors can be put together to obtain the system of equations as:

$$\begin{bmatrix} \dot{\mathbf{p}}^{\mathcal{E}} \\ \hline \dot{\mathbf{v}}^{\mathcal{B}} \\ \hline \dot{\bar{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{\mathcal{E}}_{\mathcal{B}}\mathbf{v}^{\mathcal{B}} \\ \hline \tilde{\mathbf{a}}^{\mathcal{B}} - \boldsymbol{\nu_a} + \mathbf{R}^{\mathcal{B}}_{\mathcal{E}}\mathbf{g}^{\mathcal{E}} - (\tilde{\boldsymbol{\omega}} - \boldsymbol{\nu}_\omega) \times \mathbf{v}^{\mathcal{B}}. \\ \hline \frac{1}{2}\bar{q}\left(t\right) \otimes \begin{bmatrix} 0 \\ \tilde{\boldsymbol{\omega}} - \boldsymbol{\nu}_\omega \end{bmatrix} \end{bmatrix} \tag{3.21}$$

The system is nonlinear due to the dependence between velocity and quaternion variables with $\mathbf{R}^{\mathcal{B}}_{\mathcal{E}}$ term. Time change rate of error variables representing the difference between true state variables and estimated ones can be defined as:

$$\begin{bmatrix} \delta\dot{\mathbf{p}}^{\mathcal{E}} \\ \hline \delta\dot{\mathbf{v}}^{\mathcal{B}} \\ \hline \delta\dot{\bar{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{\mathcal{E}}_{\mathcal{B}}\delta\mathbf{v}^{\mathcal{B}} \\ \hline \tilde{\mathbf{a}}^{\mathcal{B}} - \boldsymbol{\nu_a} + \mathbf{R}^{\mathcal{B}}_{\mathcal{E}}\mathbf{g}^{\mathcal{E}} - (\tilde{\boldsymbol{\omega}} - \boldsymbol{\nu}_\omega) \times \delta\mathbf{v}^{\mathcal{B}}. \\ \hline \frac{1}{2}\delta\bar{q}\left(t\right) \otimes \begin{bmatrix} 0 \\ \tilde{\boldsymbol{\omega}} - \boldsymbol{\nu}_\omega \end{bmatrix} \end{bmatrix} \tag{3.22}$$

Using four component quaternion representation in EKF with state vector given as (3.22) causes numerical singularity problems in the gain calculation step. This is due to the fact that quaternion norm is unity and magnitude change in time is equal to zero. This problem is stated in [28] and three different methods to handle this problem is presented. All three technique relies on reducing the covariance matrix dimensions to three which related to the orientation. These approaches can be briefly defined as

1. A prescribed method to reduce covariance matrix dimensions from four to three,
2. Using only three of quaternion components, and using unit norm constraint to get the fourth one,
3. Body fixed covariance representation.

These methods are further discussed in [31], and the first two methods are concluded to have theoretical and performance problems. The third option is the preferred solution to represent orientation estimation uncertainty for the filtering process. Quaternion estimation error $\delta\bar{q}$ can be defined such that it corrects the estimated value to the real one by quaternion multiplication.

$$\bar{q} = \hat{\bar{q}} \otimes \delta\bar{q} \tag{3.23}$$

The error variable for the quaternion is defined as twice the Gibbs vector $\mathbf{a}_g = 2\boldsymbol{\rho}$. The error quaternion can be calculated from

$$\delta\bar{q}\left(\mathbf{a}_g\right) = \left(4 + \left\|\mathbf{a}_g\right\|^2\right)^{-1/2} \begin{bmatrix} 2 \\ \mathbf{a}_g \end{bmatrix} \tag{3.24}$$

The error magnitude in quaternion estimation is assumed to be small, so instead of (3.24), the following can be used:

$$\delta\bar{q}\left(\mathbf{a}_g\right) = \begin{bmatrix} 1 - \frac{\left\|\mathbf{a}_g\right\|}{2} \\ \frac{\mathbf{a}_g}{2} \end{bmatrix}. \tag{3.25}$$

Time propagation of the $\mathbf{a}_g$ parameter is given as

$$\dot{\mathbf{a}}_g = -\left[\tilde{\boldsymbol{\omega}}\right]_\times \mathbf{a}_g \tag{3.26}$$

where $\left[\boldsymbol{\omega}\right]_\times$ is defined as the cross product matrix

$$\left[\boldsymbol{\omega}\right]_\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{3.27}$$

The system with error variables can be rewritten replacing $\mathbf{a}_g$ with $\delta\bar{q}$ in (3.22)

$$\begin{bmatrix} \delta\dot{\mathbf{p}}^{\mathcal{E}} \\ \hline \delta\dot{\mathbf{v}}^{\mathcal{B}} \\ \hline \mathbf{a}_g \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{\mathcal{E}}_{\mathcal{B}}\delta\mathbf{v}^{\mathcal{B}} \\ \hline \tilde{\mathbf{a}}^{\mathcal{B}} - \boldsymbol{\nu_a} + \mathbf{R}^{\mathcal{B}}_{\mathcal{E}}\mathbf{g}^{\mathcal{E}} - (\tilde{\boldsymbol{\omega}} - \boldsymbol{\nu}_\omega) \times \delta\mathbf{v}^{\mathcal{B}}. \\ \hline -\left[\tilde{\boldsymbol{\omega}}\right]_\times \end{bmatrix} \tag{3.28}$$

3.5.2 System Linearization

The system given in (3.28) can be linearized for time instant t as

$$\frac{d}{dt}\Delta\mathbf{x} = \mathbf{F}\Delta\mathbf{x} + \mathbf{G}\mathbf{w} \tag{3.29}$$

where $\mathbf{F}\left(t\right)$ is given as

$$\mathbf{F}\left(t\right) = \left[\begin{array}{c|c|c} \mathbf{0}_{3\times3} & \mathbf{F}_{12} & \mathbf{F}_{13} \\ \hline \mathbf{0}_{3\times3} & -\left[\tilde{\boldsymbol{\omega}}\right]_\times & \mathbf{F}_{23} \\ \hline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\left[\tilde{\boldsymbol{\omega}}\right]_\times \end{array}\right] \tag{3.30}$$

and $\mathbf{G}\left(t\right)$ can be written in the following form:

$$\mathbf{G}\left(t\right) = \left[\begin{array}{c|c} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \hline -\mathbf{I}_{3\times3} & -\left[\tilde{\boldsymbol{\omega}}\right]_\times \\ \hline \mathbf{0}_{3\times3} & -\mathbf{I}_{3\times3} \end{array}\right]. \tag{3.31}$$

Here $\mathbf{F}_{12}$, $\mathbf{F}_{13}$, and $\mathbf{F}_{23}$ are Jacobien matrices with respect to the $\delta\mathbf{v}^{\mathcal{B}}$ and $\mathbf{a}_g$ variables,

$$\mathbf{F}_{12} = \left.\frac{\partial\left[\mathbf{R}^{\mathcal{E}}_{\mathcal{B}}\delta\mathbf{v}^{\mathcal{B}}\right]}{\partial\delta\mathbf{v}^{\mathcal{B}}}\right|_{\mathbf{a}_g=\mathbf{0}} \tag{3.32}$$

$$\mathbf{F}_{13} = \left.\frac{\partial\left[\mathbf{R}^{\mathcal{E}}_{\mathcal{B}}\delta\mathbf{v}^{\mathcal{B}}\right]}{\partial\mathbf{a}_g}\right|_{\mathbf{a}_g=\mathbf{0}} \tag{3.33}$$

$$\mathbf{F}_{23} = \left.\frac{\partial\left[\mathbf{R}^{\mathcal{B}}_{\mathcal{E}}\mathbf{g}^{\mathcal{E}}\right]}{\partial\mathbf{a}_g}\right|_{\mathbf{a}_g=\mathbf{0}} \tag{3.34}$$

The process noise covariance matrix is defined by the placing the accelerometer and gyro variances in the matrix diagonal as

$$\mathbf{Q} = \text{diag}\left(\begin{bmatrix} \boldsymbol{\sigma}_a^2 & \boldsymbol{\sigma}_\omega^2 \end{bmatrix}^T\right). \tag{3.35}$$

3.5.3 Covariance Propagation

Change of state covariance matrix $\mathbf{P}$ is given as a matrix Riccati equation in the following form:

$$\frac{d}{dt}\mathbf{P} = \mathbf{PF} + \mathbf{FP}^T + \mathbf{GQG}^T. \tag{3.36}$$

For small time steps, rate of change of the covariance matrix is assumed to be small and first order Euler integration is applied for the propagation of the covariance matrix as

$$\mathbf{P}_k = \mathbf{P}_{k-1} + \tau\left(\mathbf{P}_{k-1}\mathbf{F} + \mathbf{FP}_{k-1}^T + \mathbf{GQ}_{k-1}\mathbf{G}^T\right). \tag{3.37}$$

3.5.4 Visual Measurements

Two subsequent frames of a stereo camera setup containing points with known correspondences supplies the linear displacement and angular rotation measurement within the time interval of these two snapshots. As the motion is sensed not in a global frame but relatively within two time instants, this information can be translated to an average velocity measurement. For a stereo camera setup with xy planes overlapping and with a baseline of b, a point projected onto the left camera frame at time t, $\begin{bmatrix} x_L(t) & y_L(t) \end{bmatrix}^T$, and its correspondence on the right camera $\begin{bmatrix} x_R(t) & y_R(t) \end{bmatrix}^T$ can be combined as a three parameter measurement $\begin{bmatrix} x_L(t) & y_L(t) & d(t) \end{bmatrix}^T$ where $d = x_L - x_R$ representing disparity value. If measurement noise and pixelization is omitted, the position of the point in left camera frame can be gathered as:

$$Z^{\mathcal{C}} = \frac{fb}{d} \tag{3.38}$$

$$X^{\mathcal{C}} = \frac{x_L b}{d} \tag{3.39}$$

$$Y^{\mathcal{C}} = \frac{y_L b}{d}. \tag{3.40}$$

The position of the point in camera frame is then

$$\mathbf{x}^{\mathcal{C}} = \begin{bmatrix} X^{\mathcal{C}} \\ Y^{\mathcal{C}} \\ Z^{\mathcal{C}} \end{bmatrix}. \tag{3.41}$$

(3.41) can be transformed to the fixed frame by

$$\mathbf{x}^{\mathcal{E}} = \mathbf{R}_{\mathcal{C}}^{\mathcal{E}}\mathbf{x}^{\mathcal{C}} + \mathbf{c}^{\mathcal{E}}. \tag{3.42}$$

For two subsequent frames, the following equality shows the relation of the point coordinates in the $\mathcal{C}$ at two different time instants

$$\mathbf{R}^{\mathcal{E}}_{\mathcal{C}(t)}\mathbf{x}^{\mathcal{C}(t)} + \mathbf{c}^{\mathcal{E}}(t) = \mathbf{R}^{\mathcal{E}}_{\mathcal{C}(t+\tau)}\mathbf{x}^{\mathcal{C}(t+\tau)} + \mathbf{c}^{\mathcal{E}}(t+\tau) \tag{3.43}$$

yielding

$$\mathbf{c}^{\mathcal{E}}(t+\tau) - \mathbf{c}^{\mathcal{E}}(t) = \mathbf{R}^{\mathcal{E}}_{\mathcal{C}(t)}\mathbf{x}^{\mathcal{C}(t)} - \mathbf{R}^{\mathcal{E}}_{\mathcal{C}(t+\tau)}\mathbf{x}^{\mathcal{C}(t+\tau)} \tag{3.44}$$

In equation (3.44), the displacement of the camera is defined using a feature point projection on both frames which is observed, and orientation of the camera between two time steps. Since $\mathbf{R}^{\mathcal{E}}_{\mathcal{C}}$ represents the orientation of the camera with respect to the fixed frame, estimated state variable $\bar{q}(t)$ at time instants t and $(t+\tau)$ can be used to calculate this matrix. The measurement obtained between two frames can be found as

$$\Delta\mathbf{p}^{\mathcal{C}(t)} = \mathbf{x}^{\mathcal{C}(t)} - \mathbf{R}^{\mathcal{C}(t)}_{\mathcal{E}}\mathbf{R}^{\mathcal{E}}_{\mathcal{C}(t+\tau)}\mathbf{x}^{\mathcal{C}(t+\tau)}. \tag{3.45}$$

The projection error model of the feature points onto the left and right camera frames are assumed to be equal in both horizontal and vertical axes of the image plane normally distributed with a variance of σ_p^2. Since the sum of two random variables with normal distribution result in a normal distribution with the mean and variance equal to the sum of the mean and variance of its components, the disparity value d will have a variance of $\sigma_d^2 = 2\sigma_p^2$.

Covariance matrix of the measurement acquired from a single feature point can be obtained using the random variable divisions in (3.41). For $X^{\mathcal{C}}$, $Y^{\mathcal{C}}$ and $Z^{\mathcal{C}}$can be used as the expectation of the ratio of two normally distributed random variables. Using Rao's procedure, the variances of (3.39), (3.40) and (3.38) can be obtained by the following formula [20],

$$\mathrm{Var}\left(\frac{X}{Y}\right) = \frac{\sigma_x^2\mu_x^2 + \sigma_y^2\mu_y^2}{\mu_y^4} \tag{3.46}$$

3.5.5 Measurement Update and Reset

Measurement vector for n feature points and the current orientation is defined as

$$\tilde{\mathbf{z}} = \begin{bmatrix} \tilde{\mathbf{z}}_{fp_1} \\ \tilde{\mathbf{z}}_{fp_2} \\ \vdots \\ \tilde{\mathbf{z}}_{fp_n} \\ \mathbf{0}_{3\times1} \end{bmatrix} \tag{3.47}$$

Measurement sensitivity matrix is defined then as

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots \\ \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix} \tag{3.48}$$

Measurement covariance matrix is calculated using (3.46) for position measurements and a diagonal tiny matrix for the last term for rotation reset. This ensures that the

body velocity estimation is realized between acquired image measurements by keeping the uncertainty in the orientation small. This is not valid for the global orientation estimate, but this is applied in order to reset the orientation variable locally for the last image taken before the current image being processed. Kalman gain is calculated using

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left[\mathbf{R}_k + \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T \right]^{-1} \tag{3.49}$$

and the correction will be

$$\Delta \mathbf{x} = \mathbf{K} \left(\tilde{\mathbf{z}} - \hat{\mathbf{z}} \right). \tag{3.50}$$

Quaternion update is applied using the twice of the Gibbs vector found at the end of the error state vector calculation as

$$\hat{\bar{q}}_{unnorm} (+) = \hat{\bar{q}} (-) \otimes \begin{bmatrix} 2 \\ \mathbf{a}_g \end{bmatrix} \tag{3.51}$$

and a renormalization required like

$$\hat{\bar{q}} (+) = \frac{\hat{\bar{q}}_{unnorm} (+)}{\left\| \hat{\bar{q}}_{unnorm} (+) \right\|}. \tag{3.52}$$

Position and velocity corrections are applied linearly as

$$\hat{\mathbf{p}}^{\mathcal{E}} (+) = \hat{\mathbf{p}}^{\mathcal{E}} (-) + \Delta \mathbf{p}^{\mathcal{E}} \tag{3.53}$$

$$\hat{\mathbf{v}}^{\mathcal{B}} (+) = \hat{\mathbf{v}}^{\mathcal{B}} (-) + \Delta \mathbf{v}^{\mathcal{B}} \tag{3.54}$$

Covariance matrix update is realized using Joseph form in order to avoid numerical roundoff errors that can spoil the symmetric properties of the covariance matrix:

$$\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_k^- [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k]^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \tag{3.55}$$

3.6 Results

3.6.1 Simulation with synthetic points

A simulation was performed in order to see the performance of this approach for velocity updates in body frame. A path with a closed loop pattern was traversed with a target body velocity of $2\,{}^{m}/_{s}$ by the model defined for STAR robotic helicopter. The trajectory was calculated by the path planning algorithm described in Leonello's thesis [29]. The path travelled is shown in Figure (3.2) with a sample instant for left camera field of view pyramid showing tracked points in the last frame. Variance values for the accelerometers are taken as $0.001{}^{m^2}/_{s^4}$ and for the gyros as $0.3 \times 10^{-4} {}^{rad^2}/_{s^2}$. These values were obtained from the stagnant measurement taken from XSens-MTi attitude heading reference system device. Timestep for the inertial measurements and integration is 10 milliseconds.

The dead reckoning results at (3.3), (3.4) and (3.5) shows the clear divergence in velocity values calculated in the body frame. This is directly reflected to the position

calculation and a position error more than 1000 meters is obtained. Because there is no measurement except the inertial sensors in ths configuration, there is no feedback to the system in order to correct the state variables according to the reference. Euler angle calculations does not show a big error, mostly due to the low noise and single integration step.

Proposed EKF solution is applied with points to be projected onto a 640×480 image with a variance of 0.1 pixels. The discretization due to the pixel structure of a digital camera is also applied. Images are assumed to be taken at 500 milliseconds intervals. Stereo baseline is taken as 0.24 meters and field of view for the cameras are taken as 66 degrees. These are the values obtained from the Bumblebee XB3 stereo camera system which is used in the STAR helicopter. 20 and 100 points are tracked in two simulations.

Both 20 and 100 feature points tracking EKF solution gives a very good corrections on the body velocity vector of the vehicle. These results can be shown from Figure (3.6) to (3.11). Albeit the simulation is a relatively clean configuration with respect to a real world application, graphs shown proves the concept to be feasible. One important point to take into account is the requirement of a relatively accurate rotation estimation between image frames. These simulations relied gyro data between two time instants. Inthe case of a noisy environment, a fundamental matrix obtained using RANSAC can be used for this purpose.In this case another step must be added between the state propagation and measurement corrections in order to get a reliable rotation estimation.

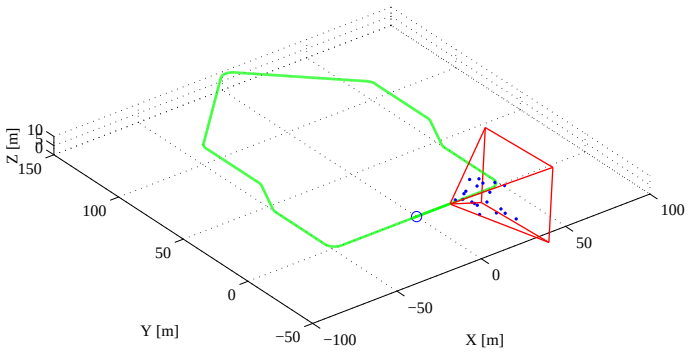

Figure 3.2: *Simulation trajectory*

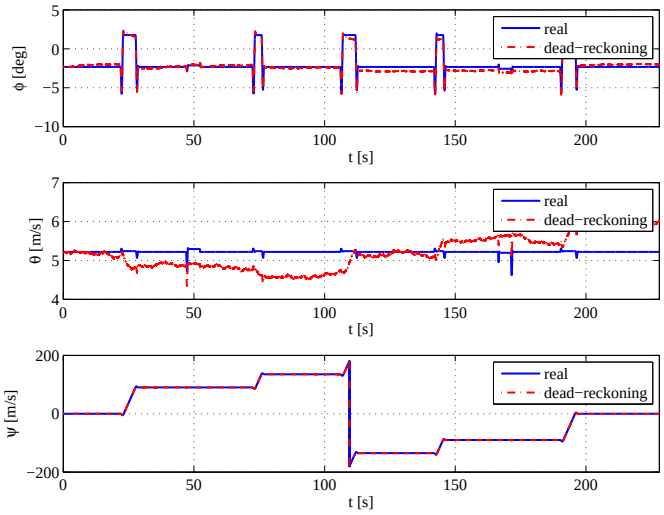

Figure 3.3: *Euler angles at the end of dead-reckoning solution*

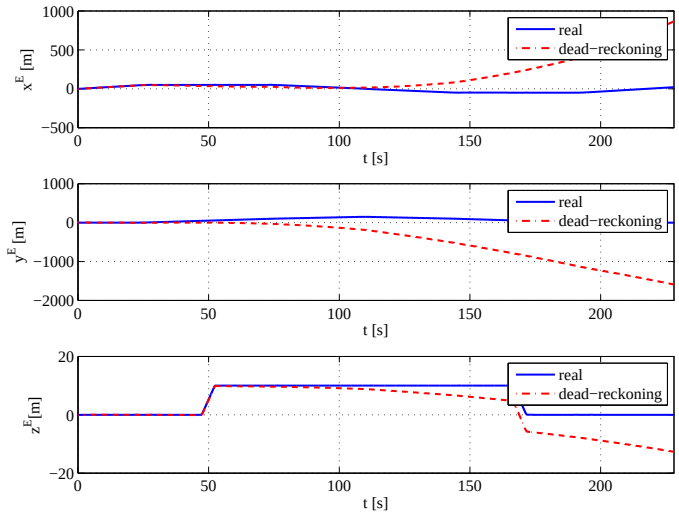

Figure 3.4: *Position in the fixed frame at the end of dead-reckoning solution*

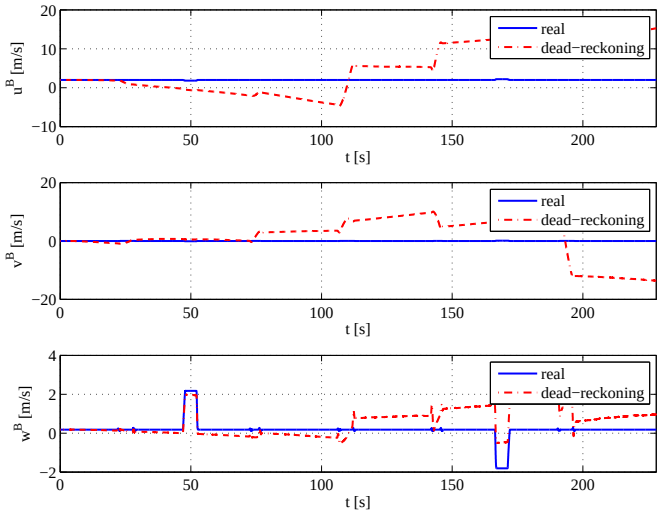

Figure 3.5: *Velocity in the body frame at the end of dead reckoning solution*

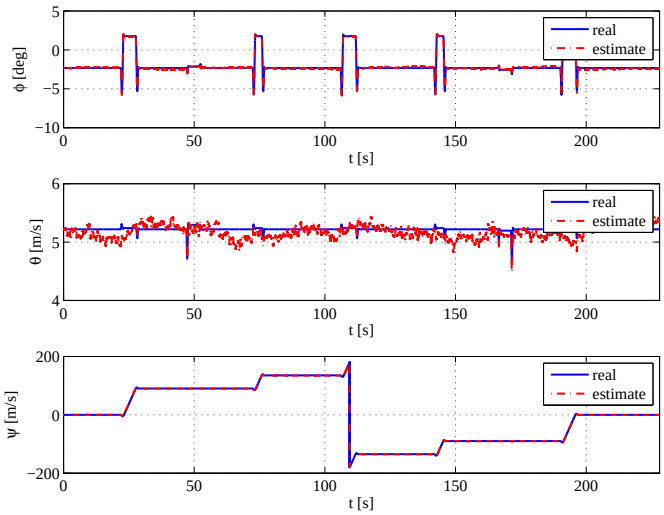

Figure 3.6: *Euler angles at the end of proposed EKF solution with 20 feature points tracked*

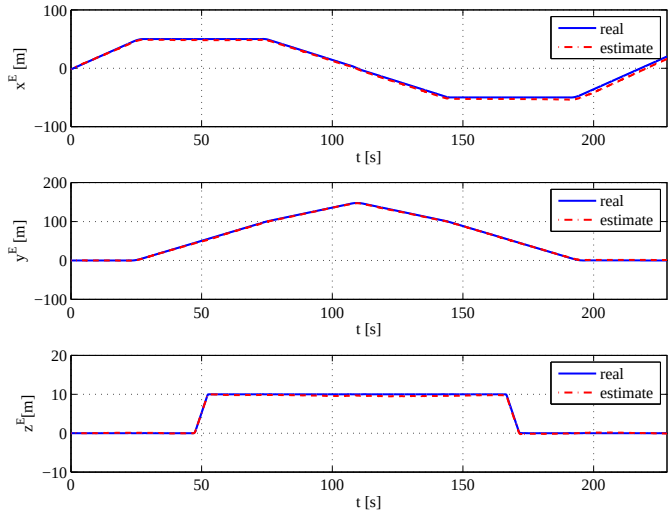

Figure 3.7: *Position in fixed frame at the end of proposed EKF solution with 20 feature points tracked*

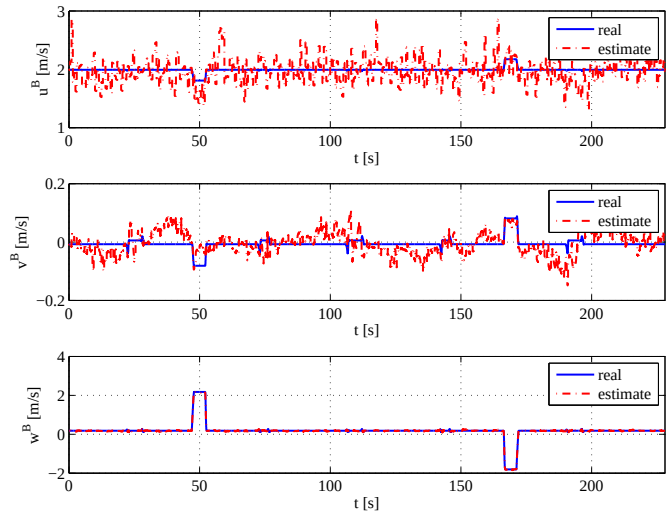

Figure 3.8: *Velocity in body frame at the end of proposed EKF solution with 20 feature points tracked*

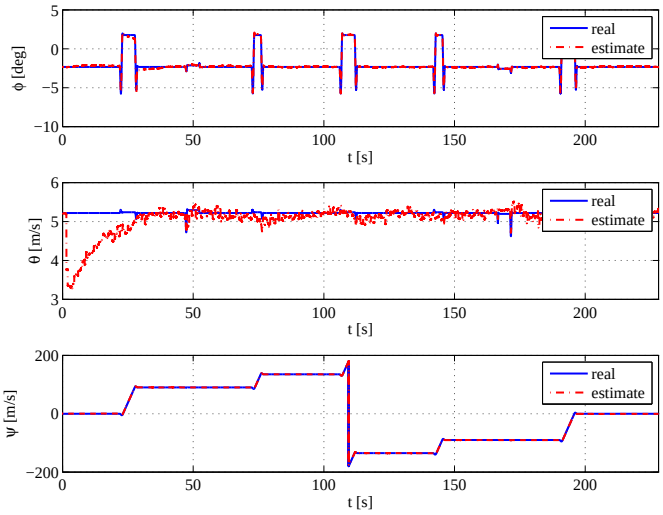

Figure 3.9: *Euler angles at the end of proposed EKF solution with 100 feature points tracked*

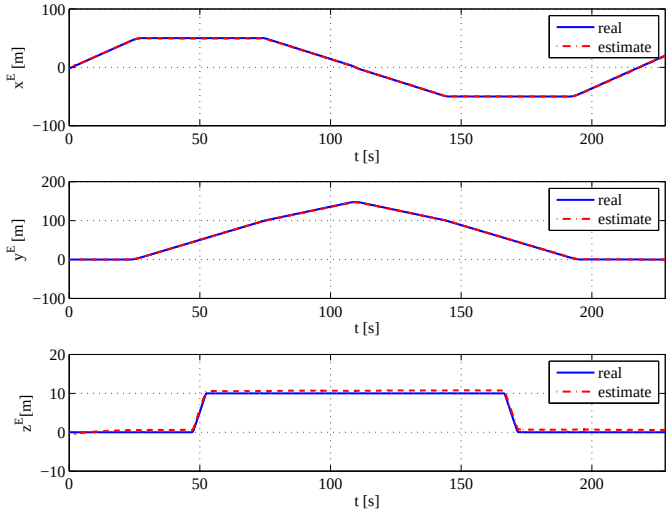

Figure 3.10: *Position in fixed frame at the end of proposed EKF solution with 100 feature points tracked*

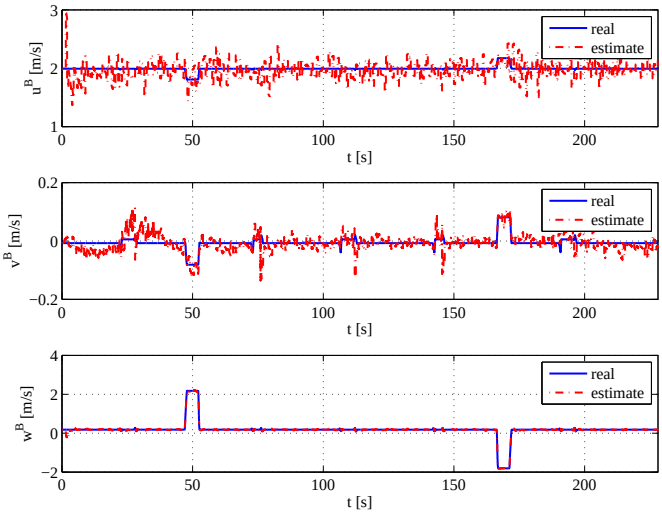

Figure 3.11: *Velocity in body frame at the end of proposed EKF solution with 100 feature points tracked*

3.6.2 Experimental test

The experimental test is conducted by carrying the on-board computer of the STAR3 vehicle by hand while it is operational and recording inertial measurements, GPS data and camera images. After real time acquisition of the data, it is processed off-line in order to see the effectiveness of the algorithm proposed in this chapter.

Data from XSens inertial unit was preferred because it also supplies orientation data using its magnetic sensors as well and processing the data with its internal Kalman filter. Acquisition was made in 100 Hz rate. At the beginning of the test, the box was set static, so the bias values of the gyroscopes were calculated using the average of angular velocity measurements. The average of accelerometer readings were used as initial roll and pitch of the system.

GPS measurements were made in 1 Hz, which is a limitation of the sensor itself. Its results were used for velocity calculation. Position measurements were not reliable in terms of precision, as a drift was observed in the altitude variable around 30 meters which was around 1 meters above the ground during the test.

For feature extraction and matching BRISK descriptor was preferred because of its lower computational cost [2, 30]. Outliers that exist in the feature point match set was removed using the gyroscope measurements between time instants of image acquisition. After the orientation change is calculated using angular velocities, displacement estimation of the left camera center is calculated using all the feature point matches and their disparity values. Median of the direction of estimates is taken as reference and elements that are not close enough to the median value are discarded from the match set. This step can be seen from Figure 3.12.

Figure 3.13 shows the divergence of the velocity variables as a result of sole inertial navigation. Non-realistic values are quickly reached and it is evident that velocity state estimation with dead reckoning for this system can not promise any good results. Although, results seen in Figure 3.14 shows a much better solution with velocity values close to the GPS velocity solution. The evident discrepancy between the visual aided solution and GPS solution is thought to be as a result of the less frequent measurements

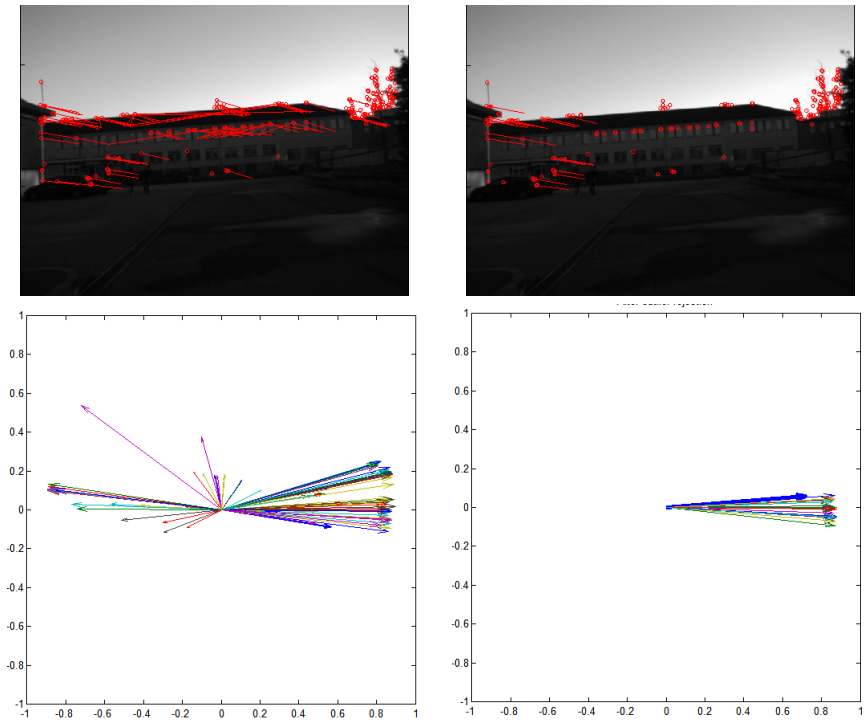

Figure 3.12: *Outlier removal using displacement vector orientation*

taken from the camera. Figure 3.15 shows the number of points tracked for each update step.

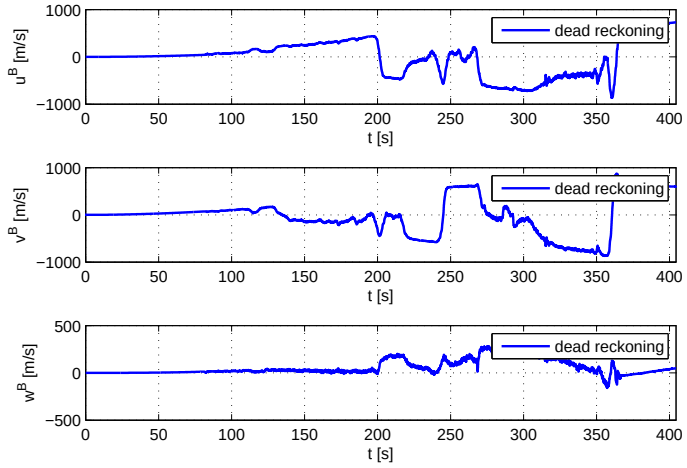

Figure 3.13: *Velocity in the body frame at the end of dead reckoning solution*

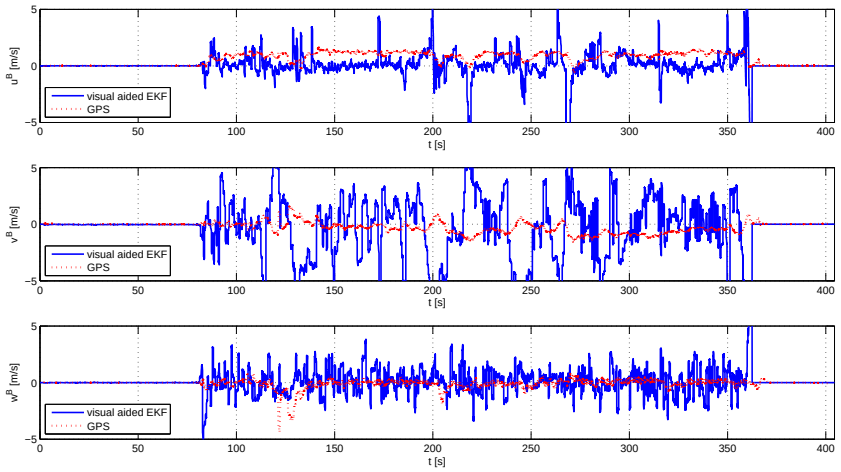

Figure 3.14: *Velocity in body frame at the end of proposed EKF solution*

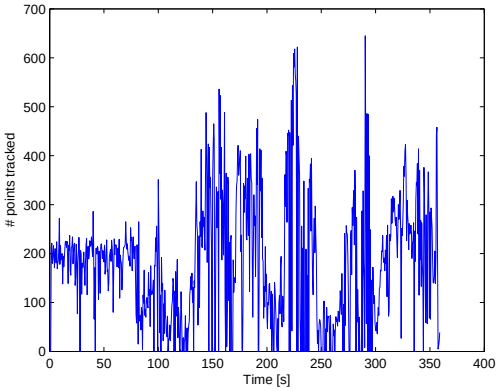

Figure 3.15: *Number of points tracked during the test*

CHAPTER 4

Three Dimensional Occupancy Grid Mapping

Using occupancy grids with probability of occupation to represent surrounding of a robot was first offered by Elfes and the technique became widespread [17]. The idea is to assume the map construction as a Markov process and the measurement updates are applied recursively. In his work Elfes used sonar sensor and stereo camera system for creating maps. Many authors described systems and results using other sensors as well. These sensors include lidar scanners, radar sensors and a variety of camera setups. The recursive update methodology is shown in Figure4.1.

In most cases, the robot is assumed to be moving in an unprepared environment for absolute pose measurement, so mapping becomes a part of a more complicated problem called Simultaneous Localization and Mapping (SLAM). Beside using other references for localization, uncompleted occupancy grids can also be used as a reference and the pose of the vehicle can be extracted from the instant measurements collected.

If the pose of a vehicle is exactly known and uncertainty only exists in measurements, this problem is called "mapping with known poses". While this is relatively easier one compared to the SLAM problem, there are some points that need to be considered while implementing a map update algorithm. These include:

- Sensor limitations like range, resolution, ambient factors,
- Space discretization and incorporating several measurements within discrete space elements,
- Fusing multiple sensor data and resolution of inconsistencies between measurements with different characteristics.

This section starts with a the problem statement and a brief introduction to occupancy grid mapping. Most of the work previously done using occupancy grids are formed

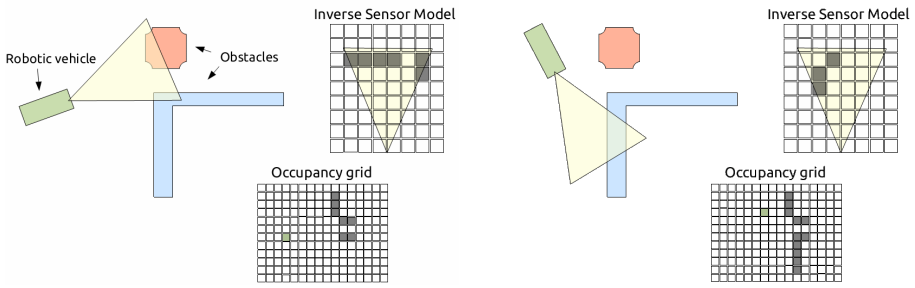

Figure 4.1: *The main idea of occupancy grid mapping, recursive update*

in two dimensional planar problems and some researchers focused on using the same concept for the three dimensional case. A literature survey about the three dimensional work is presented following the basics. Afterward, a novel algorithm is presented for creating three dimensional occupancy grid mapping with Cartesian form. The algorithm takes into account the uncertainty that exists in the pose estimation result as a six dimensional vector with Gaussian distribution. The algorithm explained is combined with a stereo camera sensor model and it is explained consecutively.

Probability calculation and updates are realized using particles generated according to the statistics of the uncertainty that is present in pose estimation and measurement. Computational aspects are discussed and subsequently results from simulation and experiment are presented.

4.1 Problem Statement

The environment around a robotic vehicle can be divided into two subsets that represent occupied and empty space. In order to solve a path planning problem in this environment, empty space can be used as the solution domain and the boundary between occupied and free space can be defined as the constraint for the calculation of optimal paths from a source to a destination. This representation is continuous and the solution is sought in an infinite set.

A grid representation of the spatial environment around the vehicle allows a metric measurement which can be accommodated in navigational optimization problems. With this regard, the problem can be divided into cells m^i , which form a grid with a fixed origin in a reference frame other than the vehicle itself. The grid elements can be in any non-overlapping shape form covering all the space represented, but mostly Cartesian or radial structured cells are preferred .The map m is the assembly of all the cells together.

$$m = \bigcup_i m^i \tag{4.1}$$

For a known environment, m^i can be either occupied or free. By assigning 1 to occupied cells and 0 to empty cells, a finite binary map m is constructed. If the map is to be generated, these values are not known a priori and updated according to the measurements taken. If the obstacles around the vehicle are sensed using range measurements from on-board sensors, these measurements will be corrupted by pose estimation and

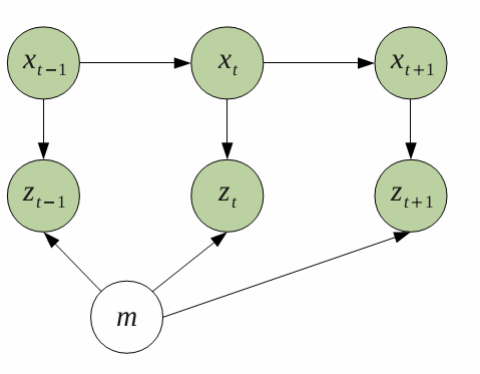

Figure 4.2: *Conditional dependency of map, measurement and pose*

measurement errors. Thus it is not possible to simply assign binary occupancy values to the individual cells.

The uncertainty encountered enforces using a probabilistic map instead of a binary representation. In this manner, one can not talk about the occupancy of a cell, instead of occupancy probability of it. The whole map is then the sum of the occupancy probability of individual cells.

$$P(m) = \sum_i P(m^i) \tag{4.2}$$

Measurements taken from range sensors are conditionally dependent on the pose and the environment. If x_k is the pose of the vehicle at instant k, then

$$P\left(z_k \mid x_k, m_k\right) \tag{4.3}$$

relates the measurement z_k to the pose of the vehicle and the map that is sought.

Here the measurement is represented as a probability instead of an equivalence of the measurement with a function of state like $z_k = f\left(x_k\right)$, simply because every sensor system contains some amount of error within their obtained measurements. If one wants to extract the map information from a measurement, this will yield an inverse sensor probability such as:

$$P\left(m_k \mid x_k, z_k\right) \tag{4.4}$$

The mapping problem with known poses and measurement data is defined as if x is the pose of the vehicle, and z is the measurement, find the posterior probability of the map m can be represented as:

$$P\left(m_k \mid x_{1\ldots k}, z_{1\ldots k}\right) \tag{4.5}$$

Here m_k represents the map at the end of the k^{th} measurement and $x_{1\ldots k}$ and $z_{1\ldots k}$ represents all the pose and measurement data from the first to the k^{th}measurement. Here a sequential update of the map is assumed and the map can contain no data at the beginning of the construction, or there can be an initial map. Figure 4.2 shows the relation between map, poses and measurements.

4.2 Three Dimensional Mapping with Occupancy Grids

In many applications, a planar robotic vehicle is used and the world for the robotic vehicle is a two dimensional planar surface. In this case, the pose state of the robotic vehicle is composed of just three variables that are two translational and one heading. An extension to this algorithm is using elevation maps that assign an elevation value for every grid element. Occupancy in this case is defined by the elevation of the cell. This technique is implemented for robotic air vehicles and used with path optimization. While elevation maps seem just an extension to the two dimensional problem, utilization of three dimensional sensor models and 6 degrees of freedom (DOF) of the vehicle are required to be integrated into the algorithm.

Three dimensional occupancy grids have also found application for both land, marine and air vehicles. For land vehicles that do not have to move on a planar surface, 6 DOF motion model must be used. Beside this, a two dimensional map representation will be insufficient to describe the environment. It is evident that marine and air vehicles that moves close to the objects should have the environment information in a three dimensional form and avoid obstacles while planning their route to a defined point.

Because of the more degrees of freedom involved, the size and complexity of the problems grows while extending the two dimensional occupancy grid concept to the three dimensional form. Applying an inverse sensor model for calculating occupancy requires using a proper rotation formalism and more computation. If a fine map is required, memory resources must be considered for the excess amount of information to be held.

A two dimensional occupancy grid generally uses two position and a heading variable for pose. The map represents the environment as a two dimensional planar surface. Assignment of height values to cells is used for terrain mapping by some researchers. In that case, 6 degrees-of-freedom of the robotic vehicle must be considered in order to apply the sensor model measurements to the map. A full implementation of occupancy grid concept to the three dimensional case requires dividing the spatial environment into non overlapping subsets like cubic voxels as done in the general case. The increment in both pose variables and grid elements brings additional burden for computational cost and required memory. Following are some of the work related to three dimensional occupancy grid mapping with different spatial representation, sensor models and update methods.

4.3 Previous Work

An example of using elevations over a two dimensional occupancy grid is implemented by Marks et al [32]. The occupancy map is produced as part of the SLAM problem, which uses a Rao-Blackwellized particle filter. Statistics of point positions observed from stereo camera that are binned into a rectangular with the same alignment of occupancy grid are used to update the estimate of the height in each grid element in this study.

Ryde and Hu used a multi resolution voxel list to represent a moderately large office environmnet in centimeter resolution [36]. They used the number of observations for voxels observed as occupied in a list to determine their occupancy status. A list approach instead of a complete occupancy grid supplied a resolution with less resources.

Another approach described by Dryanovski et al is using a two dimensional grid as a base and define interleaved rectangular volumes in maps representing the occupied and free cells and called the method “multi-level occupancy grid” [16]. They used laser scans to construct maps and use entry and exit points of laser scannings to generate volumes over the two dimensional map. Triebel et al used a similar concept but called the name [39]

Braillon et al fused stereo disparity mapping with optical flow using occupancy grid framework to detect obstacles in front of the sensor system [12]. The sensor model used for the stereo camera assigns a decreasing probability of occupancy with height above ground for measured disparity. The measurement is projected onto a two dimensional occupancy grid with the maximum probability in the three dimensional measurement above the grid elements.

A diverse setup using two omnidirectional camera in a stereo setup is explained in Correra and Okamoto’s work [15]. Feature points extracted from two images are correlated with each other and disparity values of sparse features are used to update the two dimensional occupancy grid. Their sensor model is empirically adjusted bimodal normal distribution and in the vicinity of the camera occupied cells are updated with the normal distribution and other cells are updated with a small probability value. Farther cells are updated with a half probability that represents the low observability. Instant measurement is applied to the map with Bayes rule.

Gahouzani et al described an algorithm that is calculating the probability of each voxel in a three dimensional occupancy grid using an disparity maps acquired from a stereo camera [21]. The algorithm uses the distance between the voxel center, all pixel triangulation results and triangulation errors to define an observation function for each voxel. This instant observation depends on an emprical constant. State of a voxel is updated according to the age of the observation and state of the neighboring voxels in order to prevent hanging single occupied pixels. The three dimensional map is then projected on a two dimensional one in order to lower computational cost for mapping large areas.

Hu and Mordohai used two seperate three dimensional maps representing the belief of occupied and free cells seperately [25]. Their algorithm projects the boundaries of voxels onto the image plane as a rectangular window. Each voxel was allowed to update either the occupied or free belief map. The decision was made with comparison of the calculated disparity value of the cell of the voxel and measured disparity values in the window. The conflict arising from existence of disparity values both smaller or larger than the voxel center disparity value was resolved with a threshold value for the ratio of these controversial voters.

Another similar approach to Hu and Mordohai was explained by Pirker et al [34]. The algorithm they presented projects each voxels front and back plane according to camera and constructs a pyramid. The log odd value for a grid element is determined according to the inverse model generated by the pixels witin the pyramid. the The implementation used graphical processor unit (GPU) to parallelize and fasten the computation.

4.4 Spatial Resolution and Sensor Model

Spatial resolution of a map is adjusted according to the application considered. If the vehicle position can be computed very precisely, then smaller grid elements are helpful for a path planning algorithm in order to find the optimal path to be traversed for a certain target. If the size of the vehicle is large compared to the characteristic grid element length, very high resolution makes less sense unless one is more interested in the map of the environment instead of the navigation of the vehicle itself.

Another topic that needs to be considered while generating these grids are the resolution of the onboard range sensor that is used to update the map. While sensors with wide field-of-view cone like sonars are more suitable for detecting free space, point measurement sensors like laser range arrays are good for finer representations.

Range measurements are essential for mapping. For a robotic system in an unprepared environment, range measurements can be done either by sending a signal from the robot itself and receiving it, or signals originated from another source can be processed [19]. Sonar, radar or lidar are examples of the first type of range measurement techniques. Stereo imaging is an instant of the latter. Monocular imaging has scale ambiguity , so unless assisted by another measurement or correlated with motion information, can not be used for absolute range sensing alone.

Sonar sensors are useful when one needs to be sure of an unoccupied space in the measurement cone of the sensor. As it can be seen from Figure 4.3, a Sonar senses the fastest return of its own dissipated sound burst which is interpreted as the nearest reflecting surface. This information is geometrically crude, not picturing the surrounding well, but a good measure of empty space. This rough information can also result in an incoherent map.

Radar sensors are similar to sonar sensors, but instead of sound waves, they use radio waves. They possesses the same geometric limitations but outperforms the former in terms of update rates.

Finer range sensing can be done using a lidar sensor. These sensors emit lights and use arrival time of the reflected signal as the measurements. Generally laser is used as the light source. Almost point ranging is available for a single laser beam. In robotic vehicle systems, a laser beam is oriented along a scanning pattern during its operation in order to acquire ranges within a field of view. An accurate representation of the environment can be obtained by this method. One difficulty to be encountered while using this type of sensor is the amount of data acquired. Processing this data can be cumbersome for systems with limited computational resources. The ranging pattern of a laser sensor is shown in Figure 4.4.

With fine resolution of a three dimensional occupancy grid, voxels will be effected mostly with a single pixel of the digital image plane. In that case, it is easy to apply the sensor model of a stereo camera by calculating the probability of occupancy of a voxel with the corresponding pixel and after traversing all pixels, the instantaneous measurement is applied to the map with Bayes rule. Although it is a straightforward method, mapping bigger volumes become problematic as the computational burden and memory requirements compels available limits. Because of this limitations, methods proposed are applied into smaller volumes, limited to single instants of measurement or aided by a two dimensional map with bigger coverage. Techniques showing complete

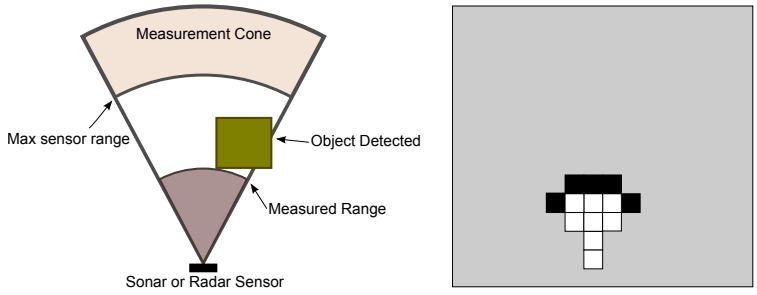

Figure 4.3: *a) An object detected in the measurement cone of a sonar or radar sensor. b) Instant measurement update in the occupancy grid*

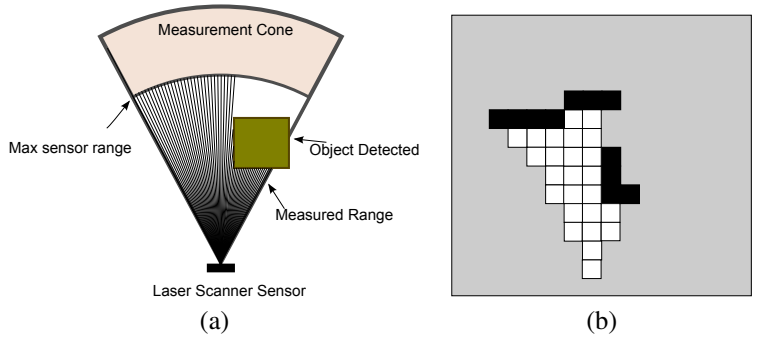

Figure 4.4: *a) An object detected in the measurement cone of a laser scanner sensor. b) Instant measurement update in the occupancy grid*

three dimensional results for larger volumes generally required offline data processing or computers with highly parallelized subsystems or specialized hardware.

A useful three dimensional mapping technique proposed in this thesis introduces a more coarse discretization of the space with larger voxels. This helps reducing computing resource and memory requirements and renders it possible of a real time application. The problem encountered with a coarse grid structure is that, a sufficiently accurate calculation of probabilities is essential for the voxels affected by many pixels in the disparity image. A numerical solution for the probability in a voxel that is effected by many pixels, measured at a pose with uncertainty require calculation of intersections of many rectangular pyramids with voxels and the prevalent probabilities within these intersections. In order to avoid computational complexity that can help identifying the occupancy probabilities with an over refined correctness, an algorithm using randomly generated particles representing a uniform probability of occupancy in its residing cell is used with a much more simpler implementation.

4.5 Stereo Camera Inverse Sensor Model

Parallel focal line stereo camera systems offer the advantage of epipolar lines being paralel. This allows one to use this information to seek for correspondences between two image data. In this case, all feature matching will be done through scanlines.

The projection of a point onto the camera plane can be expressed with the multivariate Gaussian probability distribution of:

$$p(\mathbf{p} \mid \mathbf{P}) = (2\pi\mathbf{\Sigma})^{-1/2} \exp\left(-\frac{1}{2}(f(\mathbf{P}) - \mathbf{p})^T \mathbf{\Sigma}^{-1} (f(\mathbf{P}) - \mathbf{p})\right) \tag{4.6}$$

Here $\mathbf{\Sigma}$ is the covariance matrix, and $f(\mathbf{P})$ is the ideal projection of the point $\mathbf{P}$ onto the camera plane. This measurement is corrupted by the process noise and measured as $\mathbf{p}$. The components of the three dimensional position vector of the point are $\mathbf{P} = \begin{bmatrix} X & Y & Z \end{bmatrix}^{\mathrm{T}}$, and their two dimensional projection is $\mathbf{p} = \begin{bmatrix} x & y \end{bmatrix}^{\mathrm{T}}$. The covariance matrix can be written as follows:

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}. \tag{4.7}$$

In order to find the probability distribution of a point position $\mathbf{P}$ with a given measurement $\mathbf{p}$, the distribution $p(\mathbf{P} \mid \mathbf{p})$ can be found using the Bayes rule:

$$p(\mathbf{P} \mid \mathbf{p}) = \frac{p(\mathbf{p} \mid \mathbf{P})\, p(\mathbf{P})}{p(\mathbf{p})} = \frac{p(\mathbf{p} \mid \mathbf{P})\, p(\mathbf{P})}{\int p(\mathbf{p} \mid \mathbf{P}')\, p(\mathbf{P}')\, d\mathbf{P}'} \tag{4.8}$$

Assuming *a priori* distribution $p(\mathbf{P})$ is uniform within a domain D, $\int p(\mathbf{p} \mid \mathbf{P}')\, p(\mathbf{P}')\, d\mathbf{P}'$will yield to $p(\mathbf{P}) \int p(\mathbf{p} \mid \mathbf{P}')\, d\mathbf{P}' = p(\mathbf{P})$ resulting *a posteriori* distribution given in 4.8 as

$$p(\mathbf{P} \mid \mathbf{p}) = p(\mathbf{p} \mid \mathbf{P}) \tag{4.9}$$

Because of the discrete structure of the camera pixel array, a feature defined on a pixel does not correspond to a single point in the three dimensional space, instead its correspondence space covers a rectangular pyramid, extending from the focus of the camera to the infinity passing through the rectangle defined by the pixel boundaries. The domain D can be defined as this rectangular pyramid, so the equality shown in (4.9) will be valid within it.

It can also be seen from (4.9) that this inverse probability distribution does not give any information about the depth of the point $\mathbf{P}$. This is due to the line to point mapping property of the projective geometry as explained before.

A two view measurement will narrow down this space to the intersection of two rectangular pyramids. Moreover, this intersection will be formed as a convex hexahedron with quadrilateral faces for a parallel focal line camera system.

$$\mathbf{\Sigma} = \mathrm{diag}\left(\begin{bmatrix} \sigma_{xl}^2 & \sigma_{yl}^2 & \sigma_{xr}^2 & \sigma_{yr}^2 \end{bmatrix}\right) \tag{4.10}$$

$$\mathbf{\Sigma} = \mathrm{diag}\left(\begin{bmatrix} \sigma_x^2 & \sigma_y^2 & \sigma_x^2 & \sigma_y^2 \end{bmatrix}\right) \tag{4.11}$$

Considering a parallel focal line camera system, $\mathbf{p}$ can be defined with three components, representing the coordinates of the measurements along the baseline with two parameters, and a single one perpendicular to the baseline. A perfect measurement on a pixel will define a probability distribution such that

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right) \tag{4.12}$$

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{x_p - w_p/2}^{x_p + w_p/2} \exp\left(-\frac{1}{2}\frac{(x - x')^2}{\sigma^2}\right) dx' \tag{4.13}$$

$$p(x) = \frac{1}{2}\left\{\text{erf}\left(\frac{2x_p - 2x + w_p}{2\sqrt{2}\sigma}\right) - \text{erf}\left(\frac{2x_p - 2x - w_p}{2\sqrt{2}\sigma}\right)\right\} \tag{4.14}$$

Where error function erf (x) is defined as

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp\left(-x'^2\right) dx' \tag{4.15}$$

In order to draw samples from this probability distribution, one can use its cumulative distribution. The cumulative distribution function $P(x)$ is defined as

$$P(x) = \int_{-\infty}^{\infty} p(x)\, dx \tag{4.16}$$

This function is nondecreasing the and approaches to zero and one while x approaches infinities, the inverse function is

$$P^{-1}(y) = \inf\left(x;\, F(x) \geq y\right),\ 0 < y < 1 \tag{4.17}$$

If a uniform sampling is done such as $0 < y < 1$ then $P^{-1}(y)$ will map this distribution to the desired distribution with cumulative distribution function $P(y)$. For numerical computation, a table with uniform intervals between zero and one will suffice to generate this samples using the inverse function.

4.5.1 Particle Generation with Pose and Measurement Uncertainty

Particle generation for representing pose uncertainty must depend on the pose error covariance matrix which is deduced from the covariance matrix of the state estimator. Pose error vector is composed of the position and orientation error of the vehicle. The orientation representation defines the size of the pose error vector and thus the dimensions of the pose covariance matrix. For a vector description of orientation error state, the pose covariance matrix will be composed of 6×6 elements.

If all of the pose error variables were uncorrelated, pose covariance matrix would be diagonal. A number of particles whose statistics are approximately equal to this covariance matrix could have been generated by simply generating normally distributed random values for every pose variable with the corresponding diagonal covariance matrix element. However, time propagation in the state estimation will create a correlation between the orientation and position states.

In order to generate particles conforming to this statistical distribution, pose covariance matrix can be diagonalized and a random particle vector with the same size of the pose error vector can be transformed accordingly.

Let Σ be the state covariance matrix of the system including position and orientation error uncertainties. By definition Σ is symmetric positive-semidefinite matrix and any principal submatrix of a positive-semidefinite matrix is again a positive-semidefinite matrix [8]. If rows and columns unrelated to the position and orientation error states are deleted, pose uncertainty matrix Σ_{pose} remains.

Principal component analysis of the pose covariance matrix can be carried out by eigenvalue decomposition. This decomposition is denoted by

$$\mathbf{\Sigma}_{pose} = \mathbf{Q}\mathbf{D}_{pose}\mathbf{Q}^T \tag{4.18}$$

where $\mathbf{D}_{diag}$ represents the diagonal matrix with variance values calculated as the eigenvalues of Σ_{pose} and $\mathbf{Q}$ is an orthogonal matrix composed of the eigenvectors. 4.18 is actually a singular value decomposition,

A random pose generation following the eigenvalue decomposition can be realized as

$$\Delta\mathbf{x}_{pose} = \mathbf{Q}\check{\mathbf{r}} \tag{4.19}$$

where $\check{\mathbf{r}}$ is the generated random vector of with the same size of the pose error vector. For current study $\Delta\mathbf{x}_{pose} = \begin{bmatrix} \Delta\check{\mathbf{p}}^T & \check{\mathbf{a}}_g^T \end{bmatrix}^T$, and the generated random position $\check{\mathbf{p}}$and orientation $\check{\check{q}}$are written as

$$\check{\mathbf{p}} = \hat{\mathbf{p}} + \Delta\check{\mathbf{p}} \tag{4.20}$$

$$\check{\check{q}}_{unnorm} = \hat{\check{q}} \otimes \begin{bmatrix} 2 \\ \check{\mathbf{a}}_g \end{bmatrix} \tag{4.21}$$

$$\check{\check{q}} = \frac{\check{\check{q}}_{unnorm}}{\|\check{\check{q}}_{unnorm}\|}. \tag{4.22}$$

After generating the random pose of the vehicle, measurement uncertainty that is present in the dense disparity mapping is disposed. It is assumed that the projection errors on left and right camera are uncorrelated. Also the projection error in horizontal and vertical components are also uncorrelated. As a result, a principal component analysis is not required for these variables. If a random back-projection is to be generated around a measurement in the stereo camera model, $\check{\mathbf{R}}_{\mathcal{C}}^{\mathcal{E}}$ is calculated from $\check{\check{q}}$ and the generated particle with all pose and measurement uncertainty is found as:

$$\check{\mathbf{p}}_{fp} = \check{\mathbf{p}} + \check{\mathbf{R}}_{\mathcal{C}}^{\mathcal{E}} \frac{fb}{\check{d}} \begin{bmatrix} \check{x}_L \\ \check{y}_L \\ 1 \end{bmatrix} \tag{4.23}$$

The uncertainty pattern that is being modeled by equation 4.23 is shown in 4.5 by parts and all combined.

4.5.2 Map Update

A measurement taken from the stereo camera will generate a number of particles according to the specific probability distribution as explained in the previous section. If n particles are generated from a pixel, and these particles are contained in voxels v_i, then the probability of occupancy for these voxels are defined according to the ratio of the number of points accumulated in a voxel n_i to total number of particles generated which can be expressed in as

$$P\left(v_i = 1\right) = \frac{n_i}{n}. \tag{4.24}$$

In practical application, one must consider the number of voxels to be checked after each particle generation for a pixel which is done for several times while processing

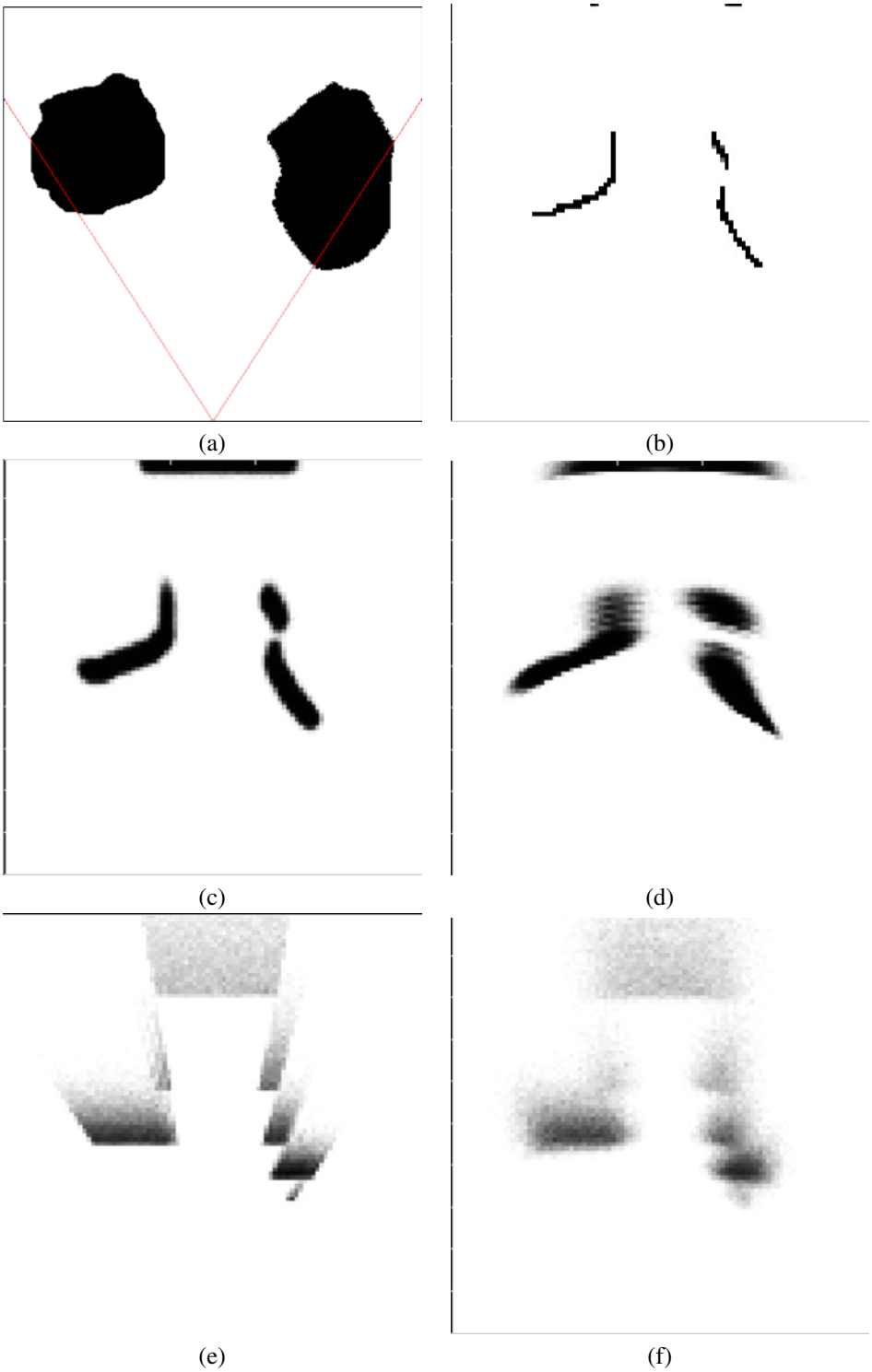

Figure 4.5: *Effect of uncertainties in different parameters for the sensor a) Real scene b)Groundtruth range sensing c) Range sensing with position uncertainty d) Range sensing with orientation uncertainty e) Pixelization uncertainty f) All uncertainties combined*

just one image. Care must be taken in order to make the update step in an efficient way. For this reason, voxel update is not done just after processing a pixel. While the particles are being generated, the voxels which they are delivered are marked by the currently processed pixel identifier. If the last pixel identifier of the voxel is not equal to the current pixel identifier, it is understood that the number of particles in that voxel are remains from the previous pixel and immediately the occupancy probability of this voxel is calculated with respect to the last pixels distribution.

The probability of occupancy of a voxel v_i according to the pixels p_1 to p_n can be calculated using

$$P(v_i = 1|p_1, p_2, \ldots, p_k) = P(v_i = 1|p_1) \cdot P(v_i = 1|p_2) \cdot \ldots \cdot P(v_i = 1|p_n) \quad (4.25)$$

under the assumption of independence of the pixel probabilities.

After every pixel is processed in the image a last check of all voxels in the field of view is required in order to clear all remnant particles and reflect them in the calculation. This is a possibility not just for the last pixel processed, but several particles can be left unaccounted because no other pixel generated at least one particle that falls on that voxel afterwards. Subsequently, a Bayesian update is applied with the instantaneous map to the general map. This is done in log-odds form.

4.5.3 Computational Aspects

Computational cost of the mapping algorithm presented depends on two factors. These are the number of particles generated at each measurement and number of voxels that are inside the field of view of the stereo camera system and allowed maximum update range. So ,the number of voxels in the map is not critical, but the size of the voxels are.

Voxel dimensions can be changed according to the resolution desired. Smaller voxels will better represent the environment, but as long as the shape of the surrounding is not important, cubic voxels with edge lengths equal to the maximum dimensions of the air vehicle can be used for the safe navigation of the air vehicle. Slightly bigger voxels will increase the safety factor, but if the voxels are too big and the environment is too cluttered including passages smaller than an edge of a voxel, the map obtained will not be a useful reference for path planning and navigation.

If two parts of the algorithm, particle generation and map updating are examined separately, it can be seen that both can be accelerated using parallel computing. If it is possible to use a computing unit with either a multiple core central processing unit (CPU) or a graphical processing unit (GPU), the time required to process the map can be lowered significantly.

It is important to analyze how random number generation can effect the accuracy of the resulting occupancy grid. There are many algorithms that can be implemented and their performance are measured according to their sequence repeating period, distribution of the numbers generated and computational cost. Randomly generated numbers can be produced online with different seeds for different variables subject to uncertainty, or a pre-produced random number table can be used, reducing the time spent to get a random number from the algorithm sequence cost to just random memory access with an incremental index.

4.5.4 Results

The algorithm explained was coded in C language and tested with stereo image sets acquired from images from a simulation . The prepared code is wrapped in an S-Function which can be used in Simulink. The block is accompanied with several other blocks that are used for pose virtual simulation environment, image processing algorithms and can be used easily with a state estimator. The modular block structure of Simulink enables the user to easily use the algorithm for both simulation and real sensor data by just changing the inputs supplied to the block.

The simulation environment is prepared using OGRE graphics engine. A village scene that is prepared by Leonello [29] is constructed with geometric meshes and rendering elements . Position and orientation of the vehicle is obtained from the same work.

The algorithm depicted in [9] results in a adequate dense disparity map for the mapping application in this work. A slightly modified version of this algorithm is implemented in legacy module of OpenCV computer vision library [11] and used for acquiring dense disparity maps for feeding the mapping function as an input.

The problem of mapping with known poses was tried in the simulation. Since the exact pose of the simulator vehicle is know, particle generation can be done with very small pose covariance matrix. Results obtained from this simulation enables comparing the results with the groundtruth and shows an insight of how well the sensor model behaves in multiple passes from the same space. Increasing the covariance matrix of the pose gives an idea of what can happen when one makes a mapping update in a high uncertain situation for the pose.

First example is a two dimensional simulation. Camera moves along the lines shown in Figure4.6. Black areas are obstacles sensd by a stereo-camera-like sensor. The map generated at the end of this simulation can be seen in Figure 4.7 where blue cells are highly probable for being free. Red cells are places where obstacles are sensed and inverse sensor model increased the probability of occupancy of them.

The three dimensional simulation environment and the path of the vehicle which is carrying the camera is shown in Figure4.8. The groundtruth of this scene can be seen in Figure 4.9. Incremental updates being applied to the three dimensional map is shown through Figure 4.10 to Figure 4.12. Comparison of the groundtruth and generated map can be seen in Figure 4.13. The map results show cells that have a probability of occupancy over a given threshold. Other cells under this threshold are aither empty or not sensed yet. For path planning purposes, the occupancy probability values of these cells can be used for weighing the cost function while seeking for the optimal paths. The recursive structure of the algorithm makes navigation towards unexplored cells safer unless the camera is not directed towards to the direction of flight.

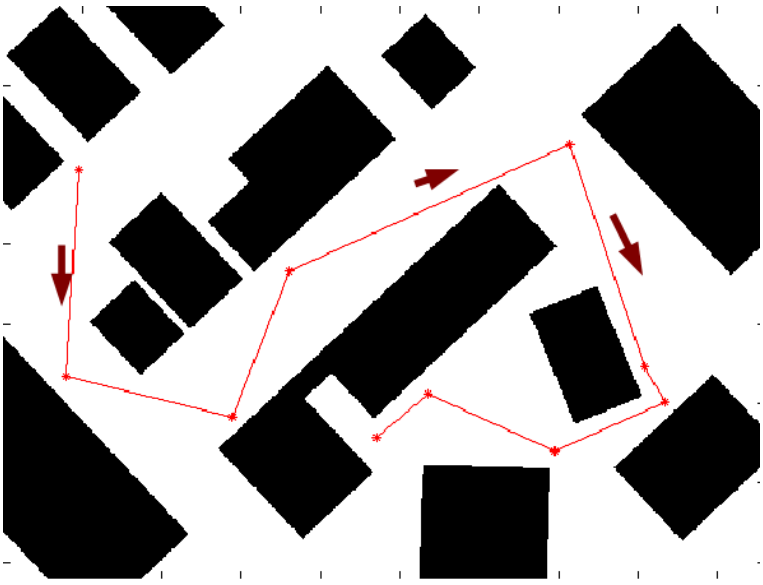

Figure 4.6: *Two dimensional simulation with stereo-camera-like sensor model*

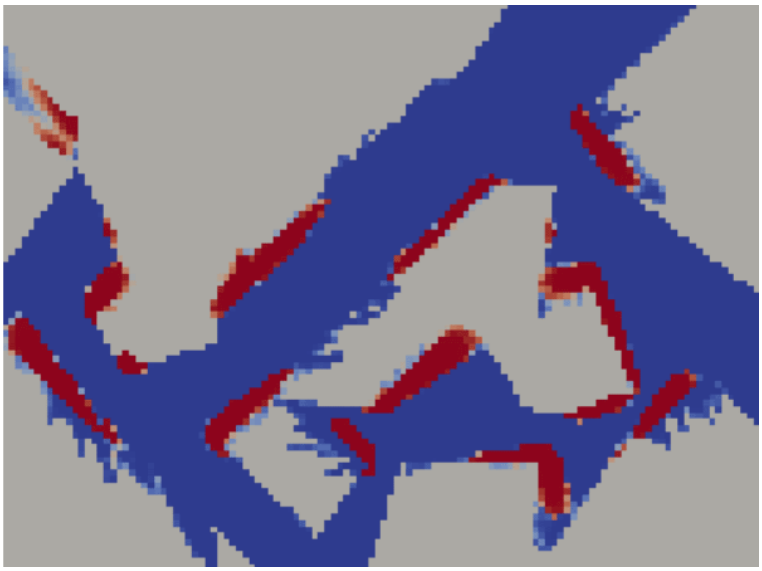

Figure 4.7: *Result of the two dimensional simulation. Blue cells are with high probability of being free, while red cells are marked to be occupied*

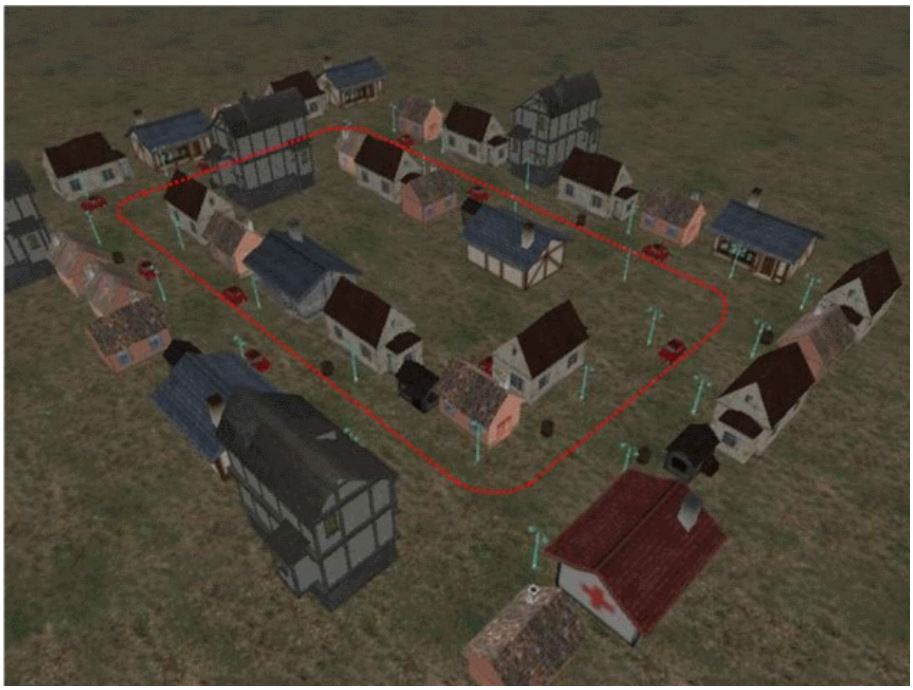

Figure 4.8: *Simulation scene*

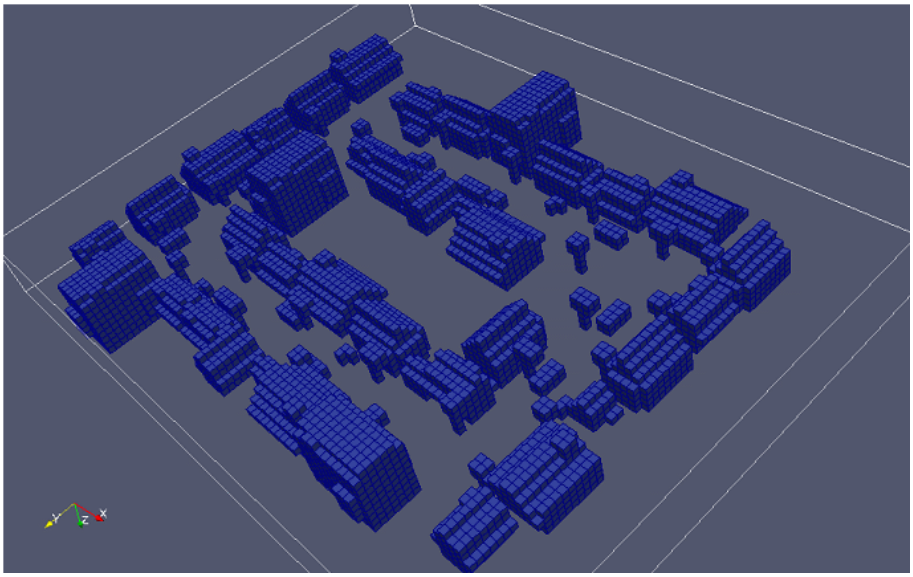

Figure 4.9: *Groundtruth for the simulation scene*

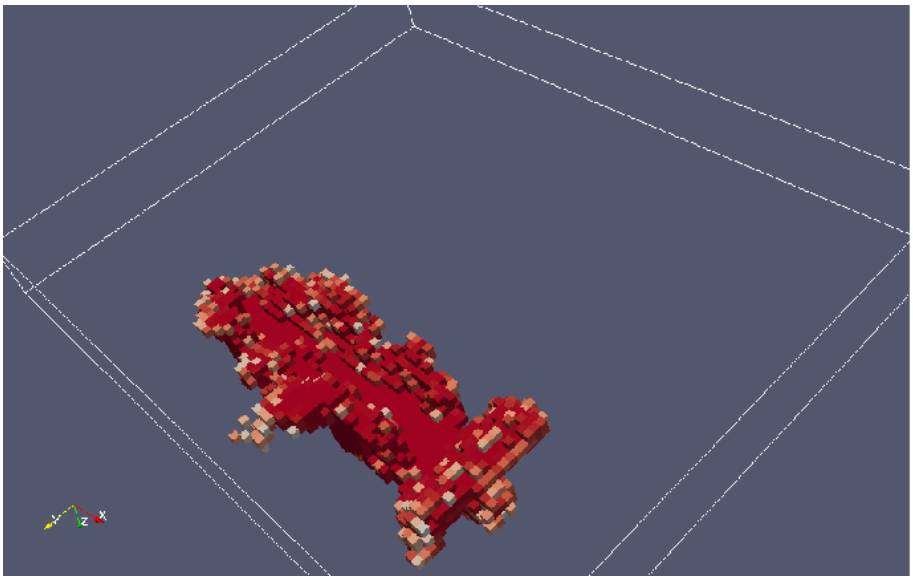

Figure 4.10: *Incremental mapping - 1*

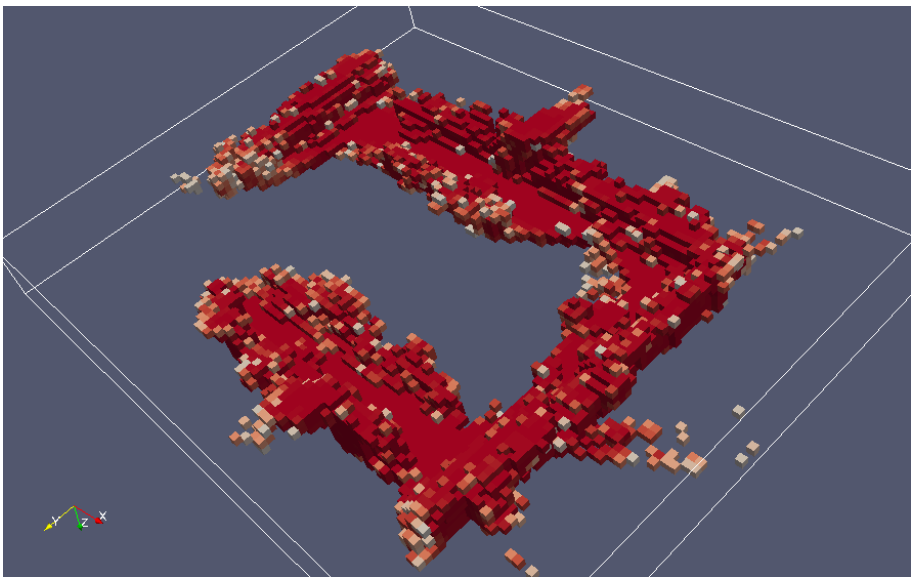

Figure 4.11: *Incremental mapping - 2*

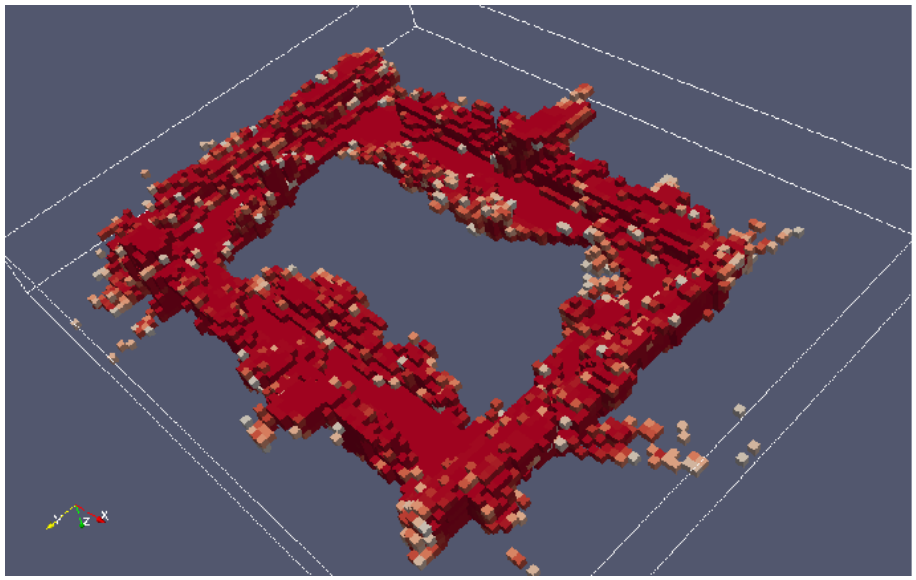

Figure 4.12: *Completed map*

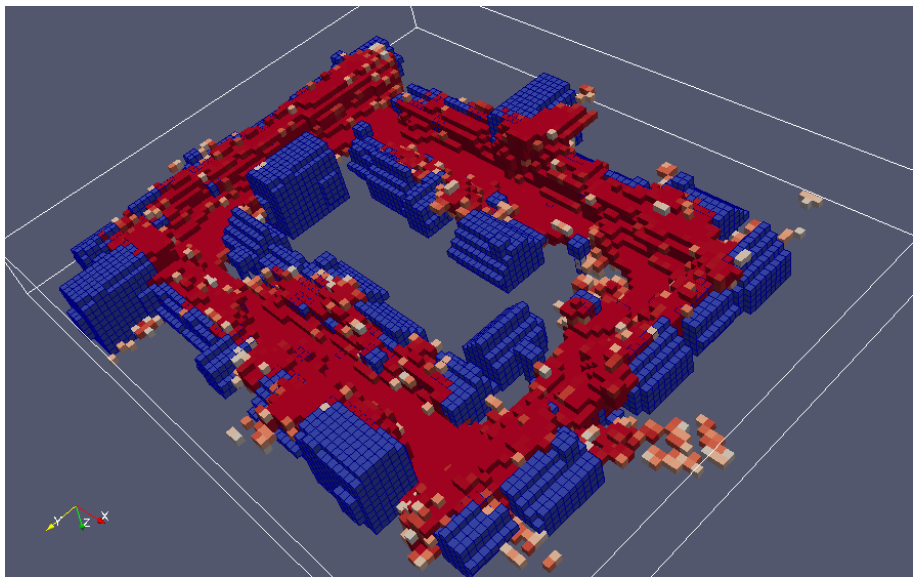

Figure 4.13: *Completed map and groundtruth*

CHAPTER 5

Conclusion and Future Prospects

State estimation and situational awareness are two key issues in order to control an autonomous rotorcraft. Controlling the rotorcraft with the desired command and avoiding possible obstacles around it requires sufficiently accurate algorithms providing state data and spatial constraints to be obeyed. In a higher control layer, spatial domain to be navigated and localization of the vehicle inside the given map must be supplied to the vehicle in order to have a solution for its path planning and possibly optimizing it. In most cases, this data is not readily and accurately available to the vehicle and it must construct a representation of the surrounding for the purpose mentioned.

This thesis focused on two navigational problems. Firstly, estimating body velocity of the vehicle in an environment lacking global localization aids like satellite based solutions. With this manner, visual features tracked by a stereo camera was incorporated with inertial measurement unit. After that, a three dimensional mapping algorithm was presented using dense disparity maps obtained from the on-board stereo camera. A widely used technique for two dimensional planar mapping problem called occupancy grid mapping was used with an extension to the third dimension. An efficient algorithm was presented that relies on the statistics of randomly generated particles with probability distribution function defined by the stereo camera sensor model.

Study related to visual aided state estimation relied on Extended Kalman Filtering technique which has a formulation containing vehicle position in global frame, vehicle velocity in body frame and orientation with quaternion formalism. Time integration of frequent inertial sensor data were updated by visual features with much less frequency. Results obtained from simulation data affirmed this approach and the technique later tested by visual simulation and real test data. The test was realized by relatively low cost and proportionally noisy sensors. This application required dealing with image processing algorithms for feature point extraction, matching and dense disparity map-

ping. Alreadily available libraries were implemented for image tasks and outlier rejection was done using the inertial sensor data between time frames instead of using a RANSAC based technique which is a common way of estimation of motion between two different views.

Certainly, velocity correction in body frame helps to reduce the positioning error in the global frame. Nevertheless a drift will always occur due to the unobservable position variable. Without a reference positioning signal, this drift can be measured only by recognition of previously visited points. By referencing that point, the drift in the current estimate and on the previous path traced can be removed. Implementation of a possible solution was not included in this study.

The mapping algorithm presented used dense disparity maps obtained from the same images taken for the feature tracking system. Statistics of accumulated particles which were generated randomly defines the total measurement of the occupancy for that time instant. Bayesian update was applied for subsequent measurements according to the occupancy grid concept. Three dimensional solutions are not very common in the literature due to the computational burden required for the great number of cells to be updated for each instant. In order to overcome this difficulty, voxel cells were held bigger as the size of the vehicle. The algorithm was also designed to be efficient in order to realize computations for each voxel in the field of view with minimum pass.

The techniques adverted were shown to be efficient in simulations. In order to prove its efficiency also in the real world, batch processing on collected data was done with these algorithms. Some drawbacks encountered during implementation which were told in the second chapter. Adequacy of the system available was questioned and it was shown that the system can perform required operations with small size images in infrequent intervals. Data transfer problems and their resolution within current limitations were shown.

5.1 Future Outlook

It is clear that computational power is the bottleneck of the current outdated system. Realizing described techniques on the onboard system will require a better computing hardware. There are a number of paths that can be traced which can be enlisted as follows:

- Using decentralized processing for each task instead of using a single processor,
- Parallel processing of the tasks that are suitable. There are variety of algorithms ported to be run on Graphical Processing Units (GPU). Also, the mapping algorithm described is sitable for parallelization,
- Implementing purpose-spesific hardware like Field Programmable Gate Arrays (FPGA). These systems allow designing a computation unit with highly parallel computation with high rates of input output.

For visual aided state estimation, higher frame rates will allow a better solution for the body velocity variables. Also, an increase in the number of points tracked will allow a more precise velocity measurement. For mapping algorithm, smaller voxels make an improvement towards a better representation of the environment. The aim in this thesis was stated like to have an algorithms sufficient enough for pat planning and obstacle

avoidance. Improvement in the computational performance will allow a very good reconstruction of the objects around the vehicle and can be used for other applications like real time photogrammetry.

Although the drift in the body velocity variable can be corrected using visual features, the position drift is still a problem to be solved. Place recognition will be a necessity for position correction without an external reference signal. For this purpose, handling of a database constructed from previously visited placed will be required. The database can be constructed according to the features like corners, lines, shapes and they must be labeled with a single descriptor for each. Another alternative is using semantic image segmentation for identification of objects.

Place recognition can also be realized by using the maps generated during the navigation. In that case mapping will be a part of Simultaneous Localization and Mapping problem. With that purpose, the maps can be further processed in order to get topological data and this can be utilized with feature based maps to get positional infomration inside the map.

APPENDIX $\mathcal{A}$

Appendix

A.1 Quaternion Multiplication

Quaternions are composed of a real and three imaginary numbers

$$\bar{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}. \tag{A.1}$$

The basis elements in (A.1) have multiplication results as follows:

$$\begin{array}{lll} \mathbf{i}\otimes\mathbf{i} = -1 & \mathbf{j}\otimes\mathbf{i} = \mathbf{k} & \mathbf{k}\otimes\mathbf{i} = \mathbf{j} \\ \mathbf{i}\otimes\mathbf{j} = \mathbf{k} & \mathbf{j}\otimes\mathbf{j} = -1 & \mathbf{k}\otimes\mathbf{j} = -\mathbf{i} \\ \mathbf{i}\otimes\mathbf{k} = -\mathbf{j} & \mathbf{j}\otimes\mathbf{k} = \mathbf{i} & \mathbf{k}\otimes\mathbf{k} = -1 \end{array} \tag{A.2}$$

Considering (A.1) and (A.2), quaternion multiplication can be carried out by either matrix products

$$\bar{a}\otimes\bar{b} = \begin{bmatrix} a_0 & -a_1 & -a_2 & -a_3 \\ a_1 & a_0 & -a_3 & a_2 \\ a_2 & a_3 & a_0 & -a_1 \\ a_3 & -a_2 & a_1 & a_0 \end{bmatrix} \bar{b} \tag{A.3}$$

$$\bar{a}\otimes\bar{b} = \begin{bmatrix} b_0 & -b_1 & -b_2 & -b_3 \\ b_1 & b_0 & b_3 & -b_2 \\ b_2 & -b_3 & b_0 & b_1 \\ b_3 & b_2 & -b_1 & b_0 \end{bmatrix} \bar{a} \tag{A.4}$$

The inverse of a quaternion $\bar{q}^{-1}$ is the one which is premultiplied or post multiplied by $\bar{q}$ results in the identity quaternion $\bar{q}_I = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$.

$$\bar{q}^{-1} \otimes \bar{q} = \bar{q}_I \tag{A.5}$$

$$\bar{q} \otimes \bar{q}^{-1} = \bar{q}_I \tag{A.6}$$

Conjugate of a quaternion is defined as

$$\bar{q}^* = q_0 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k} \tag{A.7}$$

and the inverse of a quaternion can be calculated as follows

$$\bar{q}^{-1} = \frac{\bar{q}^*}{\|\bar{q}\|}. \tag{A.8}$$

List of Figures

List of Tables

Bibliography

[1] 3337 Benzin Trainer Complete Set. http://www.vario-helicopter.net/products/pod-and-boom/petrol-pod-boom/benzin-trainer/petrol-trainer.html, 17.02.2013.

[2] BRISK: Binary Robust Invariant Scalable Keypoints. http://www.asl.ethz.ch/people/lestefan/personal/BRISK, 04.03.2013.

[3] QGroundControl. http://qgroundcontrol.org/, 17.02.2013.

[4] RTAI - Real Time Application Interface for Linux by DIAPM. https://www.rtai.org, 17.02.2013.

[5] Unibrain Fire-i board VGA Camera. http://www.unibrain.com, 17.02.2013.

[6] XSens - Mti. http://www.xsens.com/en/general/mti, 17.02.2013.

[7] N. Barbour and K. Flueckiger. Understanding Commonly Encountered Inertial Instrument Specifications. Technical report, Draper Laboratory, Cambridge, Massachusets, 2010.

[8] A. Berman and N.S. Monderer. *Completely Positive Matrices*. World Scientific, New Jersey, 2003.

[9] Stan Birchfield and Carlo Tomasi. Depth Discontinuities by Pixel-to-Pixel Stereo. *Int. J. Comput. Vision*, 35(3):269–293, 1999.

[10] J.Y. Bouguet. Camera Calibration Toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/, 3.3.2013.

[11] G Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[12] C. Braillon, C. Pradalier, K. Usher, J. L. Crowley, and C. Laugier. Occupancy Grids from Stereo and Optical Flow Data. In Oussama Khatib, Vijay Kumar, and Daniela Rus, editors, *Experimental Robotics*, volume 39 of *Springer Tracts in Advanced Robotics*, pages 367–376. Springer Berlin Heidelberg, 2008.

[13] G. Cai, B. M. Chen, and T. H. Lee. *Unmanned Rotorcraft Systems*. Springer-Verlag London Limited, London, 2011.

[14] O Chum and J Matas. Matching with PROSAC - progressive sample consensus. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 220 – 226 vol. 1, 2005.

[15] F. R. Correa and J. Okamoto. Omnidirectional stereovision system for occupancy grid. In *Advanced Robotics, 2005. ICAR '05. Proceedings., 12th International Conference on*, pages 628–634, 2005.

[16] I. Dryanovski, W. Morris, and J. Xiao. Multi-volume occupancy grids: An efficient probabilistic 3D mapping model for micro aerial vehicles. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1553–1559, 2010.

[17] A Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[18] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[19] R. B. Fisher and K. Konolige. *Range Sensors*. Springer Handbook of Robotics. 2008.

[20] F. Frishman. On The Arithmetic Means and Variances of Products and Ratios of Random VAriables. Technical report, Army Research Office, Durham, North Carolina, 1971.

[21] H. Ghazouani, M. Tagina, and R. Zapata. Robot Navigation Map Building Using Stereo Vision Based 3D Occupancy Grid. *Journal of Artificial Intelligence: Theory and Application*, 1(3):63–72, 2010.

[22] M.S. Grewal and A.P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. John Wiley and Sons, New Jersey, third edition, 2008.

[23] J. K. Hall, N. B. Knoebel, and T. W. McLain. Quaternion attitude estimation for miniature air vehicles using a multiplicative extended Kalman filter. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 1230–1237, May 2008.

[24] R.˜I. Hartley and A Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[25] X. Hu and P. Mordohai. Robust Probabilistic Occupancy Grid Estimation from Positive and Negative Distance Fields. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 539–546, 2012.

[26] B Kitt, A Geiger, and H Lategahn. Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 486–492, 2010.

[27] K Konolige and M Agrawal. FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping. *Robotics, IEEE Transactions on*, 24(5):1066–1077, 2008.

[28] E. J. Lefferts, F. L. Markley, and M. D. Shuster. Kalman Filtering for Spacecraft Attitude Estimation. *Journal of Guidance, Control, and Dynamics*, 5(5):417–429, September 1982.

[29] D. Leonello. *Navigation Techniques for Autonomous Rotorcraft*. Phd thesis, Politecnico di Milano, 2009.

[30] S Leutenegger, M Chli, and R Y Siegwart. BRISK: Binary Robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555, 2011.

[31] F. L. Markley. Attitude Error Representations for Kalman Filtering. *Journal of Guidance, Control, and Dynamics*, 26(2):311–317, 2003.

[32] T. K. Marks, A. Howard, M. Bajracharya, G. W. Cottrell, and L. Matthies. Gamma-SLAM: Using stereo vision and variance grid maps for SLAM in unstructured environments. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3717–3724, May 2008.

[33] R. Munguia and A. Grau. Attitude and Heading System based on EKF total state configuration. In *Industrial Electronics (ISIE), 2011 IEEE International Symposium on*, pages 2147–2152, 2011.

[34] K. Pirker, M. Ruther, H. Bischof, and G. Schweighofer. Fast and accurate environment modeling using three-dimensional occupancy grids. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1134–1140, 2011.

[35] Pointgrey Research. Image Tearing Causes and Solutions, 2011.

[36] J. Ryde and H. Hu. 3D mapping with multi-resolution occupied voxel lists. *Autonomous Robots*, 28(2):169–185, 2010.

[37] B. Savini. *Development of a Test-bed for Simulation and Control fo an Unmanned Rotorcraft*. Phd thesis, Politecnico di Milano, 2007.

[38] G. T. Schmidt and R. E. Phillips. INS/GPS Integration Architectures, 2011.

[39] R. Triebel, P. Pfaff, and W. Burgard. Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2276–2282, 2006.

[40] Trimble. *GPS - The First Global Navigation Satellite System*. Trimble Navigation Ltd., Sunnyvale, California, 2007.

[41] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666 –673 vol.1, 1999.