

# **STEREO VISION-BASED VISUAL ODOMETRY FOR PLANETARY EXPLORATION**

by

Kieran Kneisel

B.Eng, Ryerson University, 2009

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the Program of

Aerospace Engineering

Toronto, Ontario, Canada, 2011

©Kieran Kneisel 2011

## **Author's Declaration**

I hereby declare that I am the sole author of this thesis or dissertation.

I authorize Ryerson University to lend this thesis or dissertation to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis or dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

## **Abstract**

### STEREO VISION-BASED VISUAL ODOMETRY FOR PLANETARY EXPLORATION

by

Kieran Kneisel

B.Eng, Ryerson University, 2009

A thesis presented to Ryerson University in partial fulfillment of the requirements for the degree of Master of Applied Science in the Program of Aerospace Engineering

Ryerson University  
Toronto, Ontario, Canada, 2011

The ability to localize an unmanned vehicle is an essential requirement for extraterrestrial robotic exploration missions. The goal of this thesis is to develop a visual odometry algorithm capable of operating in real-time and in natural unstructured environments. Accuracy, repeatability and computational cost were the primary considerations during the development of the algorithm. The resulting visual odometry algorithm can operate in real-time and provides the foundations for further development. More commonly used approaches for localization include the use of inertial measurement units (IMU) or wheel odometry, which are prone to drift and slippage respectively, making them unreliable for long duration missions. Visual odometry also experiences error accumulation, however, it offers the possibility of mitigating this problem through techniques such as loop closing and bundle adjustment. The performance of the Iterative Closest Point (ICP) algorithm in conjunction within the visual odometer was also evaluated and shown to have improved overall localization performance.





## **Acknowledgements**

I would like to first acknowledge my academic supervisor, Dr. G. Okouneva, as well as my industry supervisors, Dr. F. Aghili and S. Gemme from the Canadian Space Agency. Without their guidance, criticism and continuous support this project could not have been carried out.

I would also like to thank our research partners, the Canadian Space Agency and the Neptec Research Group. I would especially like to thank Sébastien Gemme from the CSA for contributing so much of his time and expertise which proved invaluable in making the project a success. I would also like to thank Erick Dupuis from the CSA and Chad English from Neptec for making my presence at the CSA possible.

Next I would like to thank the members the CSA Robotics Group for welcoming me into the department and always making themselves available to me.

Finally I would like to thank those people who are closest to me that have never failed to offer me with their support and encouragement. Especially my parents, who inspired me to reach this point in my career path and to whom I owe so much else. To my sister and friends, for reminding me of the value of laughter and keeping me sane when the stress began to get to me. Finally to Reda El Amraoui and Sana Medelci, for welcoming me into their home and making me feel like family during my time in Montreal.



Dedicated to my family



## Table of Contents

Abstract.....	iii
Acknowledgements.....	v
Table of Contents.....	ix
List of Tables .....	xi
List of Figures .....	xii
List of Appendices .....	xiv
Nomenclature .....	xv
Abbreviations .....	xviii
1 Introduction and Previous Work.....	1
1.1 Motivation .....	1
1.2 Previous Work .....	2
1.3 Scope of this Thesis .....	5
1.4 Thesis Organization .....	6
2 Theoretical Foundations of the Work.....	9
2.1 Summary of Camera Mathematics .....	9
2.1.1 Epipolar Geometry .....	9
2.1.2 Fundamental Matrix.....	10
2.2 Feature/Interest Point Detection.....	12
2.2.1 Scale Invariant Feature Transform (SIFT) .....	13
2.2.2 Speeded Up Robust Features (SURF) .....	14
2.2.3 GPU-SURF .....	16
2.3 Data Correspondence.....	17
2.3.1 Descriptor Matching Schemes.....	17
2.3.2 Outlier Rejection/Inlier Acceptance .....	17
2.4 Pose Estimation .....	18
2.4.1 Triangulation.....	18
2.4.2 Horn Method .....	19
2.4.3 Iterative Closest Point Algorithm .....	20

2.5 Coordinate Systems.....	23
3 Preliminary Studies for Visual-Odometer Development .....	27
3.1 Preliminary Studies.....	27
3.1.1 Maximum Tracking Range .....	27
3.1.2 Impact of Maximum Tracking Range in an Unstructured Environment .....	30
3.2 Data Collection .....	32
4 Visual Odometer Design .....	35
4.1 Triangulation & Alignment .....	37
4.2 Feature Association & Outlier Rejection .....	38
4.3 Selection of a Feature Detector .....	41
5 Experimental Characterization .....	43
5.1 Experimental Setup .....	43
5.2 Experimental Results.....	44
5.2.1 Characterization of the Visual Odometer.....	44
5.2.2 Impact of ICP.....	50
5.2.3 Computational Cost .....	55
5.3 Functionality Verification .....	56
6 Conclusions .....	61
6.1 Summary of Specific Results .....	62
6.2 Future Work .....	63
Appendices.....	65
Appendix A: Algorithm Characterization Figures.....	65
Appendix B: Camera Specifications.....	71
References .....	73

## List of Tables

Table 1 - Average algorithm processing time for different feature detectors .....	41
Table 2 - Average error experienced during multiple traverses.....	46
Table 3 - Average errors shown in order of the length of the associated traverse .....	47
Table 4 - Characterization case 1 average error .....	51
Table 5 - Characterization case 2 average error .....	54
Table 6 - Average computational cost .....	55
Table 7 - Transformation applied to the synthetic data .....	58
Table 8 - Average error in localization estimates for synthetic data.....	59

## List of Figures

Figure 1 - Mars Emulation Terrain at the Canadian Space Agency .....	2
Figure 2 - Bumblebee® stereo camera system .....	6
Figure 3 - Epipolar geometry [31] .....	9
Figure 4 - Projection of point x onto the image planes [31] .....	11
Figure 5 - Visual representation of the SIFT descriptor assignment [42] .....	14
Figure 6 - Left: Approximation of the Gaussian second order partial derivative in the y-direction; Right: Corresponding box-filter approximation [7] .....	15
Figure 7 - Pose estimation using the ICP algorithm .....	23
Figure 8 - Body-centered frame relative to the stereo camera .....	24
Figure 9 - CSA Mars Emulation Terrain with world coordinate frame .....	24
Figure 10 - Maximum Tracking Range Experiment Setup: A – Bumblebee® camera system; B – Default tracking pattern .....	27
Figure 11 - Constant Radius Patterns Results, Radius of Circles is 0.01 m .....	28
Figure 12 - Constant Distance Pattern Results, Distance Between Circles is 0.06 m .....	29
Figure 13 - Scaled Pattern Results .....	29
Figure 14 – Sample of terrain used for the visual odometry algorithm development .....	30
Figure 15 - Sample left camera frame with descriptors .....	31
Figure 16 - Matched descriptors overlaid on the associated left and right camera frames .....	31
Figure 17 – Aerial view of the Mars Emulation Terrain at the Canadian Space Agency ..	32
Figure 18 - Instrumented cart used for data collection .....	33
Figure 19 – Visual odometer structure .....	36
Figure 20 - Time required to match for a set of descriptors using varying search areas ..	39
Figure 21 - Number of inliers detected for varying search area sizes .....	40
Figure 22 – The MRT equipped with a stereo camera and GPS frame .....	43
Figure 23 - Various environmental conditions experienced during the characterization of the visual odometer .....	45
Figure 24 - Comparison of rover traverses 2 and 8 .....	47



Figure 25 - Displacement error vs. the length of the traverse .....	48
Figure 26 - Yaw error vs. the length of the traverse .....	48
Figure 27 - Graphical output of offline processing .....	49
Figure 28 - Characterization case 1 rover trajectory .....	50
Figure 29 - Characterization case 1 terrain texture .....	50
Figure 30 - Characterization case 1 rover displacement .....	51
Figure 31 - Characterization case 1 rover yaw angle .....	52
Figure 32 - Characterization case 2 shown on a CAD model of the MET .....	53
Figure 33 - Characterization case 2 terrain texture .....	53
Figure 34 - Characterization case 2 rover displacement .....	54
Figure 35 - Characterization case 2 rover yaw angle .....	55
Figure 36 - Outlier rejection results during varying weather conditions .....	57
Figure 37 - Synthetic data used to compare the modes of the visual odometry algorithm .....	58
Figure 38 - Traverse 3 displacement error.....	65
Figure 39 - Traverse 3 yaw angle error .....	66
Figure 40 - Traverse 4 displacement error.....	66
Figure 41 - Traverse 4 yaw angle error .....	67
Figure 42 - Traverse 5 displacement error.....	67
Figure 43 – Traverse 6 displacement error .....	68
Figure 44 - Traverse 7 displacement error.....	68
Figure 45 - Traverse 8 displacement error.....	69
Figure 46 - Traverse 8 yaw angle error .....	69
Figure 47 - Bumblebee® camera specifications [65] .....	71
Figure 48 - Bumblebee® camera dimensional drawings [65] .....	72

**List of Appendices**

Appendices..... 65

    Appendix A: Algorithm Characterization Figures..... 65

    Appendix B: Camera Specifications..... 71

## Nomenclature

$\Delta$	Euclidean distance between descriptors
$\varepsilon$	Error threshold for the ICP cost function
$\varepsilon_{Rr}$	Error threshold for the ICP relative magnitude of rotation matrices
$\varepsilon_{Tr}$	Error threshold for the ICP relative magnitude of translation matrices
$\varepsilon_{Ra}$	Error threshold for the ICP absolute magnitude of rotation matrices
$\varepsilon_{Ta}$	Error threshold for the ICP absolute magnitude of translation matrices
$\theta$	Rover pitch
$\tau$	Epipolar plane
$\sigma$	Standard deviation, scale
$\phi$	Rover roll
$\psi$	Rover yaw
$A_{i,j}$	Image pixel
$b$	Baseline
$\mathbf{C}$	Camera center
$\mathbf{D}$	Points representing the Data set for ICP
$D_{xx}$	Approximation of the second order derivative of the Gaussian function
$d_i$	Descriptor value for dimension $i$
$d_x$	Haar wavelet response in the x direction
$d_y$	Haar wavelet response in the y direction
$\mathbf{e}$	Epipole
$F$	Fundamental matrix
$f_{ij}$	Entry of the fundamental matrix corresponding to position $(i,j)$
$f$	Camera focal length

$g$	Gaussian function
$H$	Homography mapping all points $\mathbf{x}$ to their corresponding points $\mathbf{x}'$
$H$	Hessian matrix
$I$	Image
$i$	Point Index
$K$	Camera calibration matrix
$k$	Number of trials required by RANSAC
$L_{ij}$	Convolution of the Gaussian second order derivative
$l$	Epipolar line
$M$	Points representing the Model set for ICP
$M_{i,j}$	Image gradient magnitude
$n$	Number of data points randomly selected by RANSAC
$P$	Camera projection matrix
$Q$	Rotation matrix to transform coordinates from the rover frame into the world frame
$R$	Rotation matrix
$R_{i,j}$	SIFT keypoint orientation
$r_l$	Current coordinate frame vector
$r_r$	Previous coordinate frame vector
$r'$	Point coordinate expressed relative to the centroid of the data set
$r_{l,c}$	Coordinates of the data set centroid expressed in the left frame
$s$	Scale factor
$T$	Translation matrix
$w$	Probability that a given feature correspondence is an inlier

$\mathbf{X}$	Point in 3D space
$\mathbf{X}_r$	Point in 3D space, rover body-centered frame
$\mathbf{X}_w$	Point in 3D space, world frame
$\mathbf{x}$	Image point
$X$	Cartesian x-axis coordinate
$x$	Pixel x- axis coordinate for image point $\mathbf{x}$
$x_c$	Pixel x-axis coordinate of the camera focal point in the image
$Y$	Cartesian y-axis coordinate
$y$	Pixel y- axis coordinate for image point $\mathbf{x}$
$y_c$	Pixel y-axis coordinate of the camera focal point in the image
$Z$	Cartesian z-axis coordinate
$z$	Probability that a random selection of $n$ feature correspondences consists entirely of inliers

## Abbreviations

API	Application Programming Interface
CCD	Charge Coupled Device
CSA	Canadian Space Agency
GPU	Graphics Processing Unit
ICP	Iterative Closest Point Algorithm
IDC	Iterative Dual Correspondence
IMRP	Iterative Matching-Range-Point
IMU	Inertial Measurement Unit
LIDAR	Light Detection and Ranging
MET	Mars Emulation Terrain
MRT	Mars Rover Testbed
PSM	Polar Scan Matching
RANSAC	Random Sampling Consensus
RTK	Real Time Kinematic
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SURF	Speeded-Up Robust Features
VME	Visual Motion Estimation
VO	Visual Odometry

# **1 Introduction and Previous Work**

## **1.1 Motivation**

Localization is an essential capability for a robotic vehicle on an extraterrestrial exploration mission. The time delay for communication from Earth to other bodies makes it impractical to have a human component in the loop to performing this task. Additionally, the lack of Earth-based assets, such as the Global Positioning System (GPS), means that a vehicle cannot rely on them for localization. The localization problem is currently is being addressed through the development of visual odometry algorithms.

Visual odometry, or visual motion estimation, is a method of performing localization. This can be performed using one or more cameras and can be done using any combination of feature detection, feature matching, outlier rejection and pose estimation techniques. In the case of stereo-based visual odometry, the task is to determine motion from the images captured by the stereo camera system.

Localization algorithms have most commonly been developed using wheel odometry or inertial measurement units (IMU). Wheel odometry is unreliable due to the wheel slippage on terrain where traction is poor and IMUs become unreliable as they drift over time. These problems cause error to accumulate in the sensor readings causing increasingly large errors in localization estimates.

Light detection and ranging (LIDAR) equipment has been used to perform localization, however they have a high mass, volume and power consumption, compared to a stereo camera system. It is extremely important to minimize these factors when designing space missions and hardware for them. Finally, LIDAR has the additional drawback of requiring the vehicle to stop movement while it scans the terrain, whereas a camera can collect imagery in real-time.

Stereo camera systems have also been used to perform obstacle avoidance and slip detection when wheel odometry was used as the primary localization method. The results of these algorithms demonstrated the potential of visual-odometry and also

showed that the error accumulation in wheel odometry could be greatly reduced through the use of this system.

Current visual odometry algorithms, while gradually improving in accuracy, have either post-processed data or required that their vehicles move slowly. A major focus of this thesis is the development and testing of a visual odometry algorithm capable of operating in real-time while providing accurate localization estimates. Another goal was to assess the utility of incorporating the Iterative Closest Point (ICP) algorithm into the visual odometer to improve localization estimates. All experimental characterization of the completed algorithm was performed using an actual rover in the Canadian Space Agency's Mars Emulation Terrain (MET) shown in Figure 1.



**Figure 1 - Mars Emulation Terrain at the Canadian Space Agency**

## **1.2 Previous Work**

Visual odometry is a branch of computer vision that is essential to the advancement of autonomous robotics. It estimates a rigid transformation between data points or descriptors obtained from camera frames. Ego-motion estimation, or estimation of a vehicle's motion, through visual odometry has applications for all manner of vehicles.



Autonomous navigation of rovers designed for planetary exploration is the application for which the visual odometry algorithm presented here is intended. Other applications include navigation of vehicles such as cars ([14] and [75]), rescue vehicles [19] and even robotic museum tour guides [6]. Visual odometry can also be used as a secondary system to complement localization estimates obtained through other means such as wheel odometry ([12],[33] and [47]). A variety of camera configurations exist for visual odometry including stereo ([15],[20],[35],[48],[58],[59], [67] and [69]), monocular omnidirectional ([13] and [75]) and multi-ocular omnidirectional [66] configurations.

Feature detection and description, the process by which data is obtained from images, is essential to visual odometry. Building on the work of Canny [11] and Moravec [55], Harris and Stephens first developed an algorithm capable of locating corners within images, an approach which gained wide acceptance within the image processing research community [30]. Much of the work in visual odometry has been either directly or indirectly based on Harris and Stevens. One such example is the algorithm developed by Lowe [40], called scale invariant feature transform (SIFT). SIFT is currently the most widely used feature descriptor in use within the visual odometry community.

While SIFT feature descriptors have been proven to be robust in a variety of environmental conditions ([3]), the high cost in terms of computation is a drawback for real-time systems. Further development has resulted in a large number of variations and alternatives including Speeded Up Robust Features (SURF) [7], maximally stable extremal regions, MSER [49], PCA-SIFT [37] and gradient location and orientation histogram, GLOH [50]. SURF in particular is gaining wide acceptance as it provides comparable accuracy to SIFT with less intensive computation requirements. This is due to SURF's use of a 64 dimension descriptor [7] as opposed to the 128 dimension descriptor used by SIFT [40][44]. Finally, SURF also offers 32 dimension and 128 dimension descriptor versions as well as upright SURF, or U-SURF, which forgoes rotation invariance for increased speed [7]. Work has also been performed by Valgren

and Lilienthal on determining the robustness of SIFT and SURF to the environmental changes caused by changing seasons [78].

Feature association, or feature matching, is inherently linked with feature detection and description. Unless an association of descriptors can be achieved between frames, there is not enough available information to perform pose estimation. Methods of performing feature matching are largely derivatives of the Nearest-Neighbour matching technique [40] which matches descriptors based on the lowest Euclidean distance between them ([35], [42], [61] and [66]). Variations of the Nearest-Neighbour approach include Nearest-Neighbour Ratio ([6], [51] and [75]), Nearest-Neighbour Threshold ([58] and [59]) and Preferred Mate matching [61]. An alternative method has been proposed by Zhang et al. to match using an iterative heuristic greedy rounding process [81] and another has been described by Zhang et al. using epipolar geometry [82]. In both cases the alternative approaches were found to perform similarly or better than typical nearest neighbour approaches.

Erroneous matches are inevitable when matching large data sets against one another. In the majority of work performed in visual odometry, outlier rejection is performed independently of feature association and is almost synonymous with using the Random Sample Consensus (RANSAC) to estimate the epipolar geometry of the images and eliminate the matches that do not fit the model [24]. The work of Zhang et al. estimates and utilizes the epipolar geometry as part of the feature association step instead of performing it independently after matching is completed [82].

Pose estimation can be performed a variety of ways. Optical flow is a viable method for performing pose estimation and has been used by Corke et al. [13]. During development of the algorithm presented here, the Horn method [34] was chosen to determine the initial pose estimation for the ICP algorithm. A number of other approaches have been used as well including sparse bundle adjustment as used by Sünderhauf et al. ([22] and [74]) and the Hough transform approach used by Se et al. [71].

The ICP algorithm, developed by Besl and McKay [9], is an integral part of the visual odometry algorithm presented here. Alternative approaches to the ICP algorithm have been proposed. The Iterative Dual Correspondence (IDC) algorithm, a hybrid of the original ICP algorithm and the new Iterative Matching-Range-Point (IMRP) algorithm was proposed by Lu and Milios [45]. As well, Diosi and Kleeman developed the Polar Scan Matching (PSM) algorithm [16] and have demonstrated its value for simultaneous localization and mapping (SLAM) algorithms [17].

Past research into visual odometry has yielded promising results. All of the reported data indicates that visual odometry is reliable for translational motion. The stereo-based visual odometry algorithm reported by Helmick et al. consistently achieved an overall displacement error of less than 2.5 % [32]. Similarly, Olson et al. were able to achieve a displacement error of approximately 1.2 % with their stereo-based visual odometry algorithm [63]. Finally, the work of Nistér et al. [61] demonstrated a visual odometry algorithm with a 1-2 % error in displacement and up to 5° in rotation. Rotation is a particularly difficult aspect of motion for a visual odometer to estimate, as features pass out of the field of view faster than when undergoing translational motion.

Alternative approaches to vision-based pose estimation have included radar [18], sonar ([21],[76] and [80]), and LIDAR based systems ([20] and [45]), as well as inertial systems [5].

### **1.3 Scope of this Thesis**

The primary contribution of this thesis is the design of a visual odometry algorithm that is capable of operating in real-time to localize a vehicle in an unstructured environment, without the use of external infrastructure, such as artificial markers. This addresses the problem of determining position and orientation information for a vehicle while seeking to reduce the limitations visual odometry can impose in terms of the vehicle's speed.

The algorithm also provides the basis for conducting future research by offering a framework on which to build. As well, this thesis provides an evaluation of the

capability of ICP to improve pose estimates obtained from stereo imagery. Characterization of the algorithm as a whole and of the ICP algorithm is done by processing the imagery from the stereo camera system shown in Figure 2.

Due to the use of ICP by the visual odometer, it is necessary to generate 3D Cartesian coordinates for each descriptor. Consequently the work presented here lends itself to fusion with a mapping algorithm, however, there are certain capabilities of such an algorithm that are outside of the scope of this thesis. The algorithm has been tested in simple traverses in both laboratory and relevant field environments. Capabilities such as the loop closing problem and kidnapped robot scenario are beyond the scope of this work.



**Figure 2 - Bumblebee® stereo camera system**

## **1.4 Thesis Organization**

Chapter I discusses the motivation behind the project, previous work performed by other researchers and a brief description of the scope of the thesis.

Chapter II contains a discussion of the definitions and theoretical knowledge required for this work.

Chapter III contains a description of the preliminary experiments performed to determine the capabilities of the hardware with which development would be conducted as well as the data collection process.

Chapter IV describes the different components of the visual odometry algorithm. The decision making processes used in selecting the components are also discussed.

Chapter V outlines the characterization experiments conducted to test the finalized algorithm as well as presenting the testing methodology used in Canadian Space Agency's Mars Emulation Terrain (MET) on the CSA Mars Rover Testbed (MRT).

Chapter VI outlines the specific conclusions of the research and makes recommendations for future work.



## 2 Theoretical Foundations of the Work

A significant number of algorithms and concepts are required in order to construct a functional visual odometer. The following section contains necessary theoretical foundations for the project.

### 2.1 Summary of Camera Mathematics

#### 2.1.1 Epipolar Geometry

The epipolar geometry is the projective geometry between two views, and is independent of the scene viewed by the cameras. The primary applications of the epipolar geometry are the outlier rejection technique, RANSAC, and the triangulation of the matched stereo pairs. The only relevant parameters are the internal camera parameters, being the focal length and the coordinates of the principle point, and the relative pose between cameras.

Suppose a point,  $X$ , in 3D space is imaged in two views, let  $x$  denote the image point in the first image and  $x'$  in the second. The camera centers for each view are denoted as  $C$  and  $C'$  respectively. It can be seen in Figure 3 that rays back-projected from the camera centers through the image points will intersect at  $X$ . Thus,  $X, x, x', C$  and  $C'$  are coplanar. This plane is referred to as the epipolar plane [31].

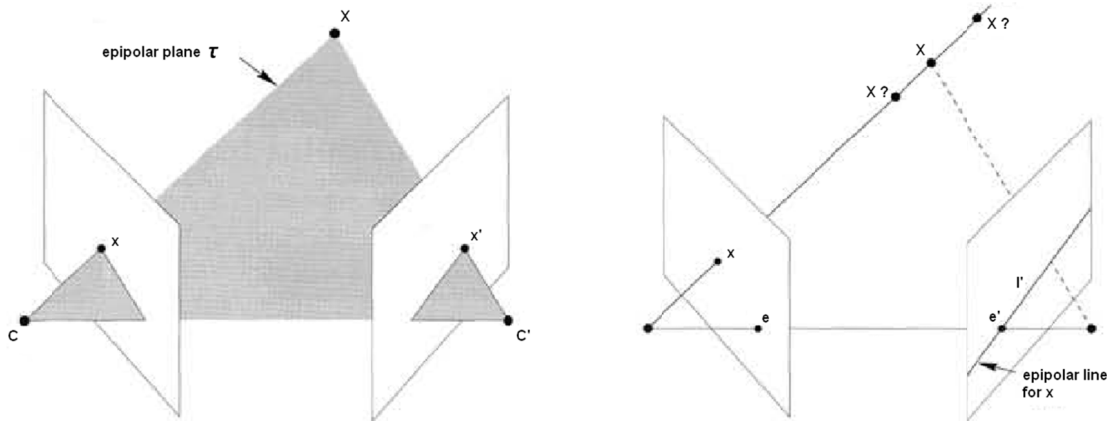


Figure 3 - Epipolar geometry [31]

Also shown in Figure 3 is the epipolar line corresponding to  $\mathbf{x}'$ , denoted  $l'$ , as well as the epipoles of each view, denoted  $\mathbf{e}$  and  $\mathbf{e}'$  respectively. The line connecting camera centers  $\mathbf{C}$  and  $\mathbf{C}'$  is referred to as the baseline.

The epipole of an image is located at the intersection of the baseline with the image plane. This point corresponds to the location of the camera center of one view in the other view. An epipolar line,  $l$ , is a line in the image plane that intersects the image epipole,  $\mathbf{e}$ , and the image point,  $\mathbf{x}$ . Each epipolar line intersects the image epipole. Additionally, each epipolar plane contains the baseline and intersects the image planes along the epipolar lines [31].

These relations are of particular importance to visual odometry and scene reconstruction. More specifically, this directly impacts feature correspondence between images. Knowledge of epipolar geometry can be used to speed up feature matching by limiting the search to the correct epipolar line. This also has implications for outlier rejection as well, in that correct matches should be located along the corresponding epipolar lines. Feature correspondence and outlier detection is discussed further in Section 2.4.

### **2.1.2 Fundamental Matrix**

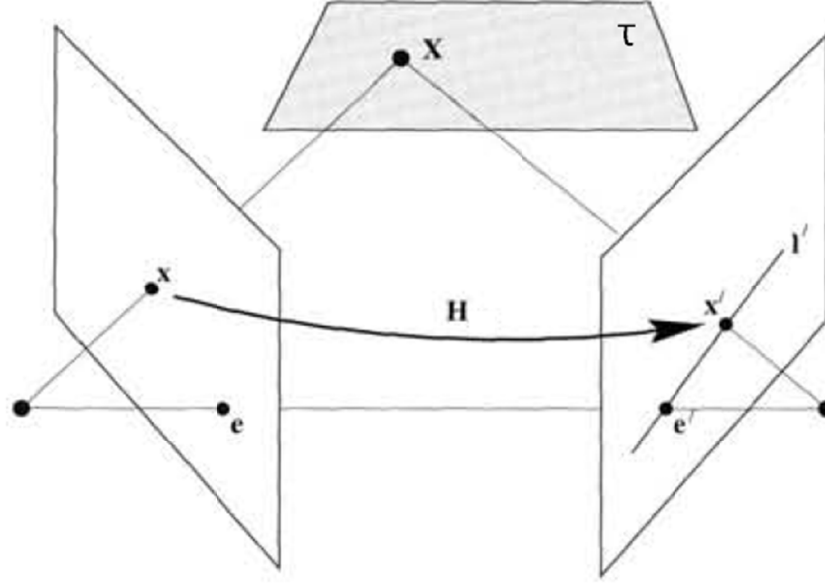
The fundamental matrix,  $F$ , is a 3x3 homogeneous matrix of rank 2 that encapsulates the geometry of a stereo camera system. It is a mapping from one point in one image to the corresponding epipolar line in another image.

The following is a geometric derivation as described by Hartley and Zisserman [31]. An algebraic derivation in terms of the camera projection matrices,  $P$  and  $P'$ , can be found in G. Xu and Z. Zhang's text *Epipolar Geometry in Stereo, Motion and Object Recognition* [81].

There are two steps when deriving the fundamental matrix geometrically. The first is to map  $\mathbf{x}$  from one image to  $\mathbf{x}'$  in the other image through the plane  $\tau$ . The plane  $\tau$  does not intersect either camera center. The ray through the first camera and the image



point  $\mathbf{x}$  intersects the plane  $\tau$  at  $\mathbf{X}$ . The world point  $\mathbf{X}$  is then projected onto the other image at  $\mathbf{x}'$ . As the point  $\mathbf{X}$  lies on the ray connecting  $\mathbf{x}$  with the corresponding camera center, the projected point  $\mathbf{x}'$  must lie on the epipolar line  $\mathbf{l}'$ , which is the image of the ray in the other image. This is illustrated in Figure 4.



**Figure 4 - Projection of point  $\mathbf{x}$  onto the image planes [31]**

The set of all points  $\mathbf{x}_i$  and the corresponding point  $\mathbf{x}'_i$  are projectively equivalent to  $\mathbf{X}_i$  and are therefore projectively equivalent to each other. Hence, there is a transformation,  $\mathbf{H}$ , that maps  $\mathbf{x}_i$  to  $\mathbf{x}'_i$ , shown by Figure 4.

The second step in deriving the fundamental matrix geometrically is to determine the epipolar line from the point projected onto the other image. Knowing that the epipolar line,  $\mathbf{l}'$ , passes through  $\mathbf{x}'$  and  $\mathbf{e}'$ , we can write the equation of the epipolar line as

$$\mathbf{l}' = [\mathbf{e}']_x \mathbf{x}' \quad (1)$$

where  $[\mathbf{e}']_x$  is the skew symmetric matrix

$$[\mathbf{e}']_x = \begin{bmatrix} 0 & -e_z & e_y \\ e_z & 0 & -e_x \\ -e_y & e_x & 0 \end{bmatrix} \quad (2)$$

Since  $\mathbf{x}'$  can be expressed as  $\mathbf{x}' = H \cdot \mathbf{x}$ , we can write that

$$\mathbf{l}' = [e']_x \cdot H \cdot \mathbf{x} = F \cdot \mathbf{x} \quad (3)$$

From (3) it can be seen that the fundamental matrix is a mapping of  $\mathbf{x}$  to  $\mathbf{l}'$ , or of  $\mathbf{x}'$  to  $\mathbf{l}$ , as expressed by (4a) and (4b) respectively.

$$\mathbf{l}' = F \cdot \mathbf{x} \quad (4a)$$

$$\mathbf{l} = F^T \cdot \mathbf{x}' \quad (4b)$$

where,

$$\mathbf{l} \cdot \mathbf{x} = 0 \quad (5a)$$

$$\mathbf{x}'^T \cdot \mathbf{l}' = 0 \quad (5b)$$

It can be easily shown that if a point,  $\mathbf{X}$ , in 3D space is imaged in two views denoted by  $\mathbf{x}$  and  $\mathbf{x}'$ , then the image points must satisfy the correspondence condition.

$$\mathbf{x}'^T \cdot F \cdot \mathbf{x} = 0 \quad (6)$$

This relation also has significant implications for visual odometry. It is the basis of outlier rejection, namely RANSAC, applied to the feature matches obtained as part of the visual odometry algorithm and is further described in Section 2.4.2.

## 2.2 Feature/Interest Point Detection

Feature detection is an integral part of a visual odometry algorithm, and can be broken down into two main steps. First, interest points must be detected in the images from unique areas of each image. Typically, such areas are characterized by corners, T-junctions and blobs. The most important property for feature detection is repeatability in detection of these areas. If an algorithm cannot detect the same interest points in the same scene, but viewed under different conditions, then correct matches are unlikely. Second, the area surrounding each interest point must be described by a vector which must be both unique and robust to noise and detection errors [7].

### 2.2.1 Scale Invariant Feature Transform (SIFT)

The first feature detection method evaluated for use in the visual odometer is Scale Invariant Feature Transform (SIFT) developed by Lowe [40]. The algorithm consists of both feature detection and feature description. For detection, a Gaussian pyramid is constructed. A Gaussian pyramid is created by repeatedly blurring and scaling down a series of images. Doing this multiple times results in a collection of successively smaller images. Local maxima and minima, referred to as keypoints or interest locations, are identified in the pyramid using multiple difference-of-Gaussian images. Because the 2D Gaussian function is separable, Lowe computes each convolution by applying the 1D Gaussian function, given by (7), in the horizontal, followed by the vertical direction:

$$g(x, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (7)$$

Where  $x$  is the image coordinate, and  $\sigma$  is the standard deviation of the Gaussian distribution, which is taken to be  $\sqrt{2}$ . The image gradient magnitude,  $M_{p,q}$ , and orientation,  $R_{p,q}$ , at each pixel are determined:

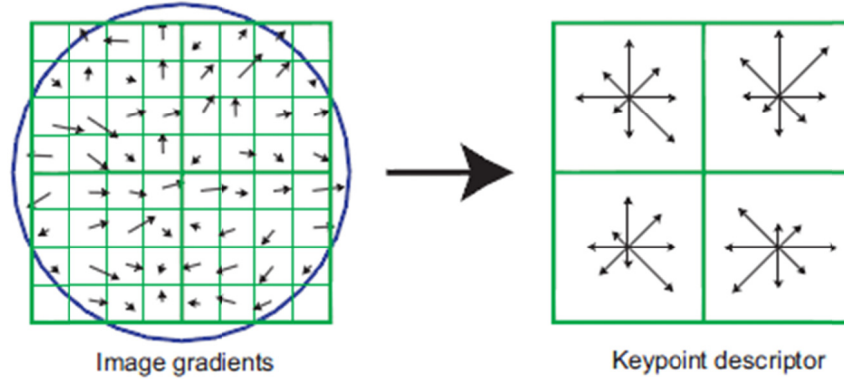
$$M_{i,j} = \sqrt{(A_{p,q} - A_{p+1,q})^2 + (A_{p,q} - A_{p,q+1})^2} \quad (8)$$

$$R_{p,q} = \tan^{-1}(A_{p,q} - A_{p+1,q}, A_{p,q+1} - A_{p,q}) \quad (9)$$

where  $A_{p,q}$  is the pixel associated with the pixel coordinate  $(p,q)$ . Each key is assigned an orientation, determined by a peak in a histogram of local image gradient, to ensure rotation invariance. The histogram is created using a Gaussian-weighted window with a standard deviation which is three times higher than that used for smoothing when creating the Gaussian pyramid.

Finally, for each keypoint, the pixel sampling from the pyramid level at which this keypoint was detected is used to assign it a descriptor. SIFT uses an orientation histogram formed from the gradient orientations, for each region of a 4x4 grid assigned to each keypoint. With eight orientation planes being used for each of the regions of

the 4x4 grid assigned to the keypoint, the resulting descriptor has 128 dimensions. Orientation planes are defined by Lowe as representations of the local image region using multiple images representing each of a number of orientations. Finally, the descriptor is normalized to unit length to improve invariance to illumination changes.



**Figure 5 - Visual representation of the SIFT descriptor assignment [42]**

The assignment of the descriptor based on the orientation planes is illustrated in Figure 5 where the arrows show the gradient magnitude and orientation at each region of the 4x4 grid. Figure 5 shows a 2x2 descriptor array computed from an 8x8 set of samples [42].

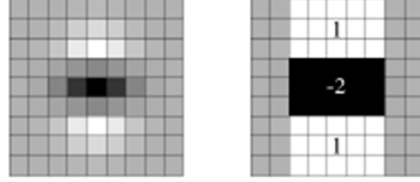
### **2.2.2 Speeded Up Robust Features (SURF)**

In developing Speeded Up Robust Features, SURF, Bay et al. [7] were able to reduce the number of dimensions in the descriptor vector from 128 to 64, without sacrificing much in terms of accuracy. The resulting algorithm is an attractive alternative to SIFT due to its significantly lower computational cost and comparable accuracy [7]. The SURF algorithm can be broken down into two operations, detection and description.

The SURF detector is based on the Hessian matrix, using its determinant to select both location and scale of interest points. Given a point,  $\mathbf{x}$ , in an image with coordinates  $[x, y]$ , the Hessian,  $H(\mathbf{x}, \sigma)$ , is defined as

$$H(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix} \quad (10)$$

where  $\sigma$  is the scale and  $L_{xx}$  is the convolution of the Gaussian second order derivative  $\frac{\delta}{\delta x^2} g(x, \sigma)$  with the image  $I$  in the point  $\mathbf{x}$  and similarly for  $L_{xy}$  and  $L_{yy}$  [7]. SURF approximates the Gaussian second order derivatives using box filters, illustrated in Figure 6.



**Figure 6 - Left: Approximation of the Gaussian second order partial derivative in the y-direction; Right: Corresponding box-filter approximation [7]**

The second order derivatives are evaluated using integral images. Taking the determinant of the Hessian matrix yields:

$$\det(H) = D_{xx}D_{yy} - (0.9D_{xy})^2 \quad (11)$$

where  $D_{xx}$ ,  $D_{yy}$  and  $D_{xy}$  denote the approximations of the second order derivatives. Filters of increasing size are applied to the images in order to generate an image pyramid. Finally, a non-maximum suppression in a 3x3x3 neighbourhood is applied in order to localize interest points [7]. Non-maximum suppression means that interest points occur where the gradient magnitude assumes a local maximum.

Once the location and scale of features have been determined, feature description is applied to generate a descriptor for each interest point. The process begins by assigning an orientation to each interest point by calculating the Haar wavelet responses in the x and y directions. This is within a circle of radius equal to six times the scale of the interest point.

The next step is to construct a square region surrounding the interest point, oriented using the previously assigned orientation. The region is subdivided into a 4x4 area, with each sub-region being assigned a feature vector. This vector is given by:

$$\mathbf{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \quad (12)$$

where  $d_x$  and  $d_y$  denote the Haar wavelet responses in the horizontal and vertical directions respectively. By assigning each of the 16 sub-regions a vector of four dimensions, a descriptor containing 64 dimensions is created for the interest point [7].

This approach allows for a significantly reduced computation time. More detailed information can be found in [7].

### **2.2.3 GPU-SURF**

GPU-SURF is an implementation of the SURF algorithm using the NVIDIA CUDA API [26]. CUDA, compute unified device architecture, is NVIDIA's parallel computing architecture which enables increases in computing performance by utilizing the GPU for processing [62]. While the theory and equations used in developing GPU-SURF are as described for SURF in Section 2.3.2, the performance deviates from the SURF algorithm of Bay et al. [7]. This is a result of the closed source nature of the original SURF distribution. Despite this inability to access SURF source code, the GPU-SURF implementation achieves comparable performance while doing so at a much lower computational cost [26].

There are some limitations identified by Furgale and Tong [26] that result from using the GPU instead of the CPU for processing. The first such issue is that the GPU hardware has single precision floats which results in slightly different computational results. The reported keypoint scale is larger than what SURF produces for a blob of comparable size. Also, the Gaussian weighting used by the GPU-SURF descriptor appears to use a different value from those generated by the original SURF distribution [7].

Because of the unique nature of GPU-SURF, it is treated as a separate from the original SURF algorithm when assessing its value for the visual odometer.

## 2.3 Data Correspondence

### 2.3.1 Descriptor Matching Schemes

Feature correspondences are formed using both the Nearest-Neighbour matching approach, proposed by Lowe [42], and the Preferred-Mate matching scheme [61]. The Preferred-Mate matching approach is a variant of the Nearest-Neighbour matching scheme which determines matches based on the Euclidean distance between descriptors. For descriptors of  $m$  dimensions,

$$\Delta = \sqrt{(d_1 - d'_1)^2 + (d_2 - d'_2)^2 + \dots + (d_m - d'_m)^2} \quad (13)$$

where  $\Delta$  is the Euclidean distance between descriptors,  $d$  is the descriptor for the feature in the left image,  $d'$  is the descriptor in the second image and the subscript  $i$  in  $d_i$ , and  $d'_i$ , is the dimension. Unlike matching using the Nearest-Neighbour approach, which searches for matches in one direction, that is, from one image to the second, the Preferred Mate approach searches in both directions. Only those feature pairs, in which the two descriptors have found each other to be the most likely match, are accepted as a valid [61].

### 2.3.2 Outlier Rejection/Inlier Acceptance

All point correspondences are subjected to the Random Sample Consensus (RANSAC) algorithm, the basis of which is the computation of the fundamental matrix from a set of nine randomly selected point correspondences. As previously mentioned, the fundamental matrix is defined by the correspondence condition, equation (6).

Given nine valid point correspondences, this condition can be used to compute the entries of  $F$ . The coefficients of this equation are functions of the known coordinates for two points, specifically the equation for the points  $(x, y, 1)$  and  $(x', y', 1)$  is

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0 \quad (14)$$

where  $f_{ij}$  are the coefficients of the matrix  $F$ . Generally, from a set of  $n$  point correspondences we obtain a homogeneous set of linear equations

$$Af = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} f = 0 \quad (15)$$

Using nine point correspondences a least-squares solution is obtained for this system to determine the entries of  $F$ . If only eight point correspondences are available then the solution will be unique up to scale [31].

Using the fundamental matrix,  $F$ , the correspondence condition, equation (6) can be applied to the data set to determine which point correspondences are inliers. Inliers are those points that satisfy the correspondence conditions, while outlier are those that do not. The RANSAC algorithm repeats the process of estimating the fundamental matrix until it has determined, within the specified confidence interval, that no larger set of inliers exists.

The algorithm bases the decision to stop selecting new random sets of data to compute  $F$  based on the expected number of trials that are required to select a data set that is error free. This is determined by the following equation:

$$k = \frac{\log(1-z)}{\log(1-w^n)} \quad (16)$$

where  $k$  is the required number of trials,  $z$  is the probability that at least one of the randomly selected data sets consists entirely of valid matches,  $w$  is the probability that given point is an inlier and  $n$  is the number of points in the data set [24]. Because it is not possible to determine all of the parameters used by RANSAC beforehand, it is important to properly tune the algorithm.

## 2.4 Pose Estimation

### 2.4.1 Triangulation

After the outliers have been removed from the data set using RANSAC, triangulation is performed to determine a Cartesian coordinate for each feature pair. This is performed using the Point Grey API which is a part of the software package accompanying the



Bumblebee<sup>®</sup> camera system. The software triangulates each pixel coordinate pair according to the equations below.

$$Z = \frac{fb}{x_2 - x_1} \quad (17)$$

$$X = \frac{x_c - x_1}{b} Z \quad (18)$$

$$Y = \frac{y_c - y_1}{b} Z \quad (19)$$

where  $f$  is the focal length of the left camera,  $b$  is the baseline of the stereo camera system,  $(x_c, y_c)$  are the pixel coordinates of the focal point in the left image,  $(x_1, y_1)$  are the pixel coordinates of the feature in the image taken by the left camera and  $(X, Y, Z)$  are the Cartesian coordinates associated with the feature. Values for the camera parameters can be found in Appendix B.

#### 2.4.2 Horn Method

Presented by Horn [34], this method is a closed form solution to the least squares problem of absolute orientation of two Cartesian coordinate frames, referred to as the current and previous frames. This algorithm is advantageous for visual odometry as a pose estimate can be determined in a single step without any iteration. In order to describe the process of determining the rotation and translation, let there be  $n$  points which are defined in each coordinate frame as  $\{\mathbf{r}_{l,i}\}$  and  $\{\mathbf{r}_{r,i}\}$  respectively. The translation is determined such that

$$\mathbf{r}_r = s\mathbf{R}(\mathbf{r}_l) + \mathbf{T} \quad (20)$$

where  $s$  is a scale factor,  $\mathbf{T}$  is the 3x1 translation matrix,  $\mathbf{R}$  is the 3x3 rotation matrix and  $\mathbf{r}_r$  and  $\mathbf{r}_l$  are the coordinates of the point expressed in three dimensions using the coordinate frames whose origins coincide with the current and previous camera frames respectively. This equation is solved using a least-squares approach to minimize the residual error given by

$$\sum_{i=1}^n \|\mathbf{r}_{r,i} - s\mathbf{R}(\mathbf{r}_{l,i}) - \mathbf{T}\|^2 \quad (21)$$

The Horn method requires that the rotation matrix is determined first. The determination of the rotation matrix is performed using quaternions. Minimization of (20) is equivalent to maximization of the function below

$$\sum_{i=1}^n \mathbf{r}'_{l,i} \cdot \mathbf{R}(\mathbf{r}'_{l,i}) \quad (22)$$

where the primes denote the position vectors expressed in a new coordinate frame relative to the centroid of the data set. The new coordinates are determined by

$$\mathbf{r}'_{l,i} = \mathbf{r}_{l,i} - \mathbf{r}_{l,c} \quad (23a)$$

$$\mathbf{r}'_{r,i} = \mathbf{r}_{r,i} - \mathbf{r}_{r,c} \quad (23b)$$

Where  $\mathbf{r}'_{l,i}$  and  $\mathbf{r}'_{r,i}$  denote the coordinates of the data relative to the centroid of the data set  $\mathbf{r}_{l,c}$  and  $\mathbf{r}_{r,c}$  denote the coordinates of the data set centroid in the current and previous frames respectively. For each coordinate pair nine possible products are determined as  $x'_l x'_r, x'_l y'_r, \dots, z'_l z'_r$  and the sum is taken to obtain  $S_{xx}, S_{xy}, \dots, S_{zz}$  where

$$S_{xx} = \sum_{i=1}^n x'_{l,i} x'_{r,i} \quad (24a)$$

$$S_{xy} = \sum_{i=1}^n x'_{l,i} y'_{r,i} \quad (24b)$$

Using this information the symmetric matrix  $N$  is determined according to the following.

$$N = \begin{bmatrix} S_{xx} + S_{yy} + S_{zz} & S_{yz} - S_{zy} & S_{zx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & S_{xx} - S_{yy} - S_{zz} & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} - S_{xz} & S_{xy} + S_{yx} & -S_{xx} + S_{yy} - S_{zz} & S_{yz} + S_{zy} \\ S_{xy} - S_{yx} & S_{zx} + S_{xz} & S_{yz} + S_{zy} & -S_{xx} - S_{yy} + S_{zz} \end{bmatrix} \quad (25)$$

The eigenvalues of  $N$  are then determined and the corresponding eigenvector for the most positive root is obtained. Finally, as stated by Horn, the quaternion representing the rotation is a unit vector in the same direction.

### 2.4.3 Iterative Closest Point Algorithm

The Horn method provides a non-iterative closed-form solution to the problem of determining the orientation between two coordinate frames. The Iterative Closest Point (ICP) algorithm proposed by Besl and McKay [9] iteratively refines the Horn method's closed-form solution.

The goal of the ICP algorithm is the iterative minimization of the cost function, presented in (26), by determining rotation and translation matrices that minimizes the least squared distance between the points of the “Reference Data” and “Collected Data” sets. The Reference Data the set of data to the newly acquired data, called Collected Data, is aligned. The cost function is given by

$$\min_{\mathbf{R}, \mathbf{T}} \sum_h \|M_h - (\mathbf{R}D_h + \mathbf{T})\|^2 \quad (26)$$

For the visual odometer presented herein both data sets are point clouds, the Reference Data consists of data points triangulated previous image frames and the Collected Data consists of the coordinates triangulated from the current stereo frame correspondence. The ICP algorithm operates using five steps and may be stopped by meeting any one of four stopping criteria. The process used by the algorithm is:

1. The cumulative transformation parameters for rotation,  $\mathbf{R}$ , and translation,  $\mathbf{T}$ , are initialized to the initial pose estimate.
2. For each point in the Collected Data set, select the closest point in the Reference Data based on the Euclidean distance.
3. Updated values for  $\mathbf{R}$  and  $\mathbf{T}$  are determined by minimizing the cost function.
4. The transformation determined in the previous step is then applied to all points in the Collected Data set.
5. If a stopping criterion has been satisfied the algorithm terminate, otherwise the process is repeated from step two.

The first two stopping criteria are based on the incremental change in the transformation matrices. The first stopping criterion is described by

$$\frac{|\mathbf{R}_h|}{|\mathbf{R}_{h-1}|} < \varepsilon_{Rr} \quad (27a)$$

$$\frac{|\mathbf{T}_h|}{|\mathbf{T}_{h-1}|} < \varepsilon_{Tr} \quad (27b)$$

where  $h$  is the iteration number,  $\mathbf{R}_h$  and  $\mathbf{R}_{h-1}$  are the rotation matrices determined for the current and previous iterations respectively,  $\varepsilon_{Rr}$  is the threshold specified for the relative magnitudes for the rotation matrices. Similarly,  $\mathbf{T}_h$  and  $\mathbf{T}_{h-1}$  are the translation matrices determined for the current and previous iterations respectively,  $\varepsilon_{Tr}$  is the threshold specified for the relative magnitudes for the translation matrices. The algorithm stops if the conditions given by equations (27a) and (27b) are satisfied.

The second stopping criterion is given by

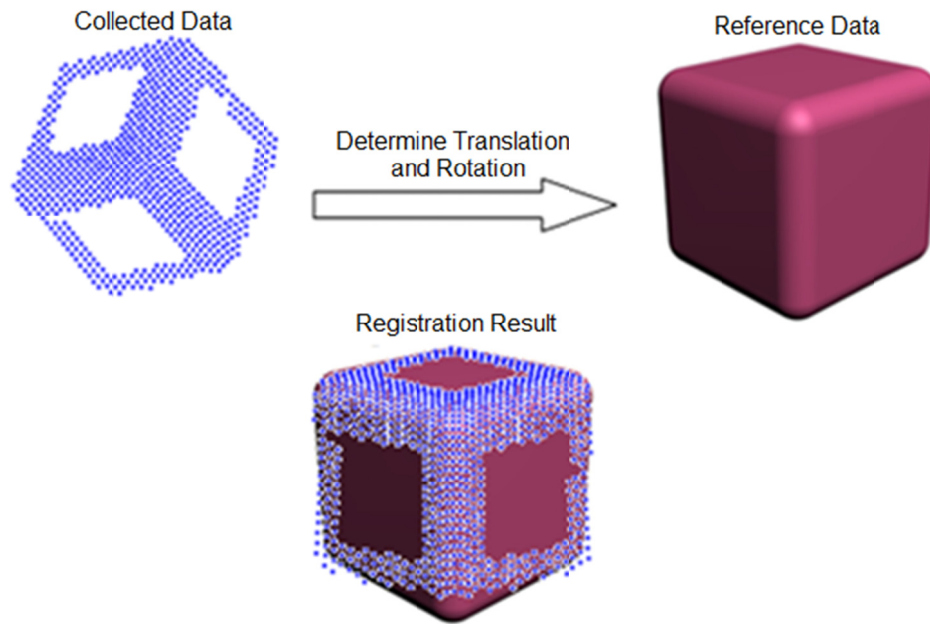
$$|\mathbf{R}_h| < \varepsilon_{Ra} \quad (28a)$$

$$|\mathbf{T}_h| < \varepsilon_{Ta} \quad (28b)$$

where  $\varepsilon_{Ra}$  is the threshold specified for the absolute magnitude of the rotation matrix and  $\varepsilon_{Ta}$  is the corresponding term for the translation matrix. Again both the conditions of both (28a) and (28b) must be satisfied for this stopping criterion to be met. The third stopping criterion is based on equation (26):

$$\min_{\mathbf{R}, \mathbf{T}} \sum_h \|\mathbf{M}_h - (\mathbf{R}\mathbf{D}_h + \mathbf{T})\|^2 < \varepsilon \quad (29)$$

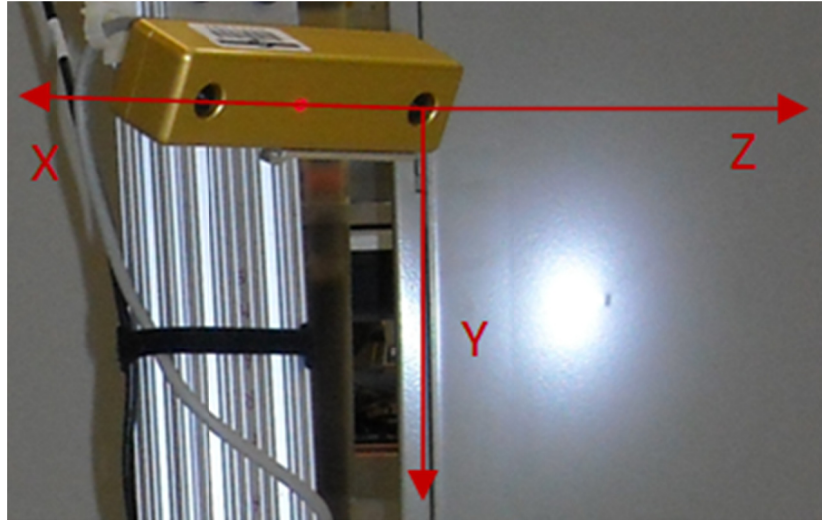
where  $\varepsilon$  is the error threshold. Finally, the algorithm can be stopped if a specified number of iterations are reached. The process of aligning the Collected Data set with the Reference Data is illustrated by Figure 7.



**Figure 7 - Pose estimation using the ICP algorithm**

## **2.5 Coordinate Systems**

The coordinate systems used by the visual odometer and for experimental characterization are presented in this section together with the equation used to convert between them. Two coordinate systems are used. The first, a body-centered coordinate frame is used by the visual odometer to represent the position of each point of interest in three dimensional space. The origin of the body-centered frame coincides with the left camera on the Bumblebee® camera system and is oriented such that the y-axis is normal to and positive towards the ground. The x- and z-axes are oriented along and normal to the front face of the rover respectively as shown in Figure 8.



**Figure 8 - Body-centered frame relative to the stereo camera**

The world frame is that used by the GPS system to track the rover's movements during data collection. The origin of this frame is coincident with the north-west corner of the CSA's Mars Emulation Terrain and is oriented such that the x- and y-axes are positive in the south and east directions respectively with the positive z-axis coinciding with the zenith direction. The world coordinate frame is shown in Figure 9.



**Figure 9 - CSA Mars Emulation Terrain with world coordinate frame**

The ground truth data describing the rover's position and orientation are defined in the world frame, as they are determined by the GPS. While the GPS only provides position

estimates, orientation is obtained by comparing the relative position of three individual antennas mounted on the vehicle.

The GPS readings are transformed from the world frame to the body-centered frame for the purposes of comparing the visual odometry estimates to ground truth data. The equation to rotate from the body-centered frame to the world frame is:

$$\mathbf{X}_r = Q \cdot \mathbf{X}_w \quad (26)$$

where  $Q$  is the rotation matrix,  $\mathbf{X}_r$  is the coordinate in the body-centered frame and  $\mathbf{X}_w$  is the coordinate in the world frame. The rotation matrix,  $Q$ , is an Euler angle yaw-pitch-roll sequence:

$$Q = \begin{bmatrix} \cos \theta \cos \psi & \sin \left( \varphi - \frac{\pi}{2} \right) \sin \theta \cos \psi - \cos \varphi \sin \psi & \cos \left( \varphi - \frac{\pi}{2} \right) \sin \theta \cos \psi + \sin \left( \varphi - \frac{\pi}{2} \right) \sin \psi \\ \cos \theta \sin \psi & \sin \left( \varphi - \frac{\pi}{2} \right) \sin \theta \sin \psi + \cos \varphi \cos \psi & \cos \left( \varphi - \frac{\pi}{2} \right) \sin \theta \sin \psi - \sin \left( \varphi - \frac{\pi}{2} \right) \cos \psi \\ -\sin \theta & \sin(\varphi - \pi) \cos \theta & \cos \varphi \cos \theta \end{bmatrix} \quad (27)$$

where  $\psi$  is the yaw angle,  $\theta$  is the pitch angle and  $\varphi$  is the roll angle.





### 3 Preliminary Studies for Visual-Odometer Development

#### 3.1 Preliminary Studies

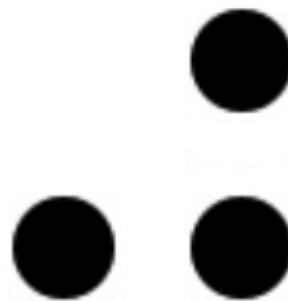
Preliminary work was conducted to determine the capabilities of the camera as well as the most advantageous conditions under which to operate. These studies focused primarily on determining the optimal range at which the camera was able to operate in an effort to determine how to best mount the camera on the rover.

##### 3.1.1 Maximum Tracking Range

The first preliminary experiment was conducted in the lab and consisted of a simple tripod setup, as shown in Figure 10A. The purpose of the experiment was to determine the limitations of the hardware using the software provided with the camera API to track a known pattern, such as the one shown in Figure 10B. A pattern was suspended in the field of view of the camera and was slowly translated and rotated away from the camera until tracking failed. The camera's estimation of the pattern's pose was observed and stored in real time using the camera software. Once the software stopped updating the pose estimates, the individual experiment was halted and data recorded.



A

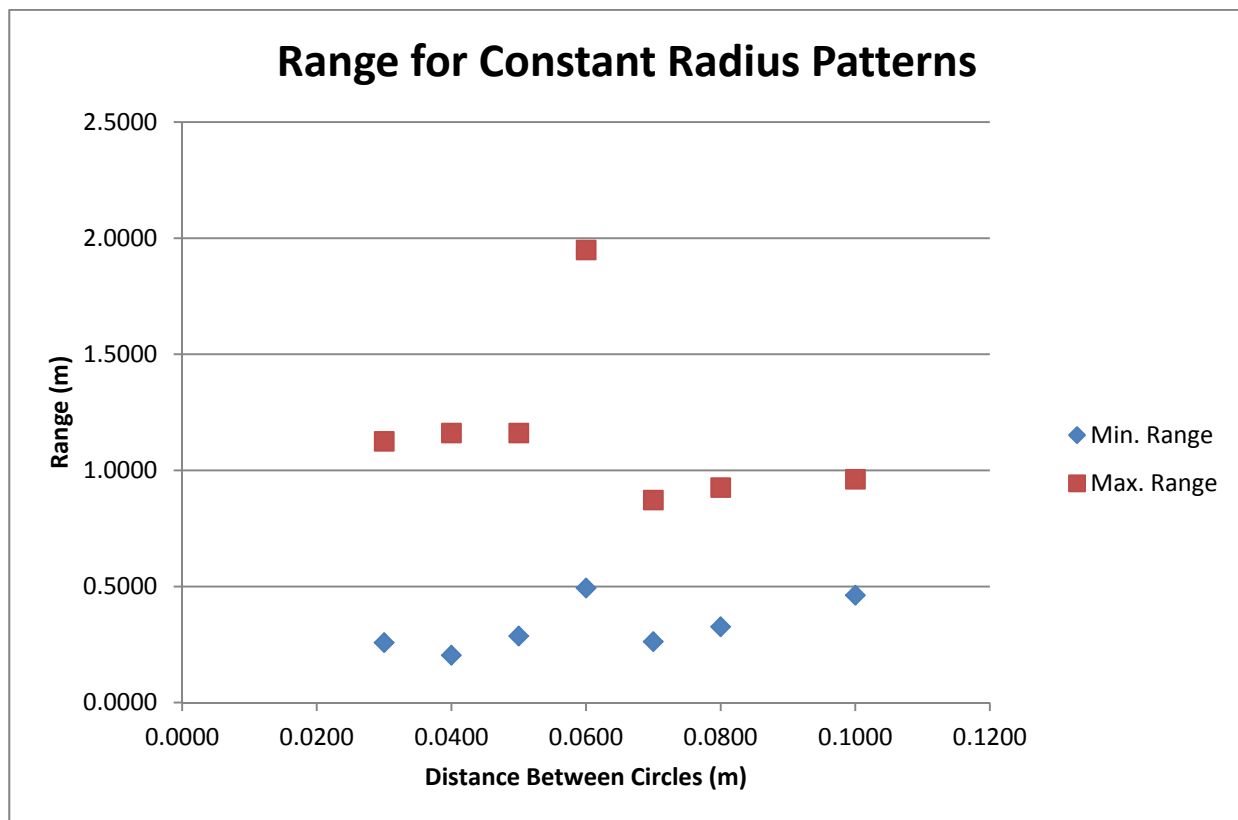


B

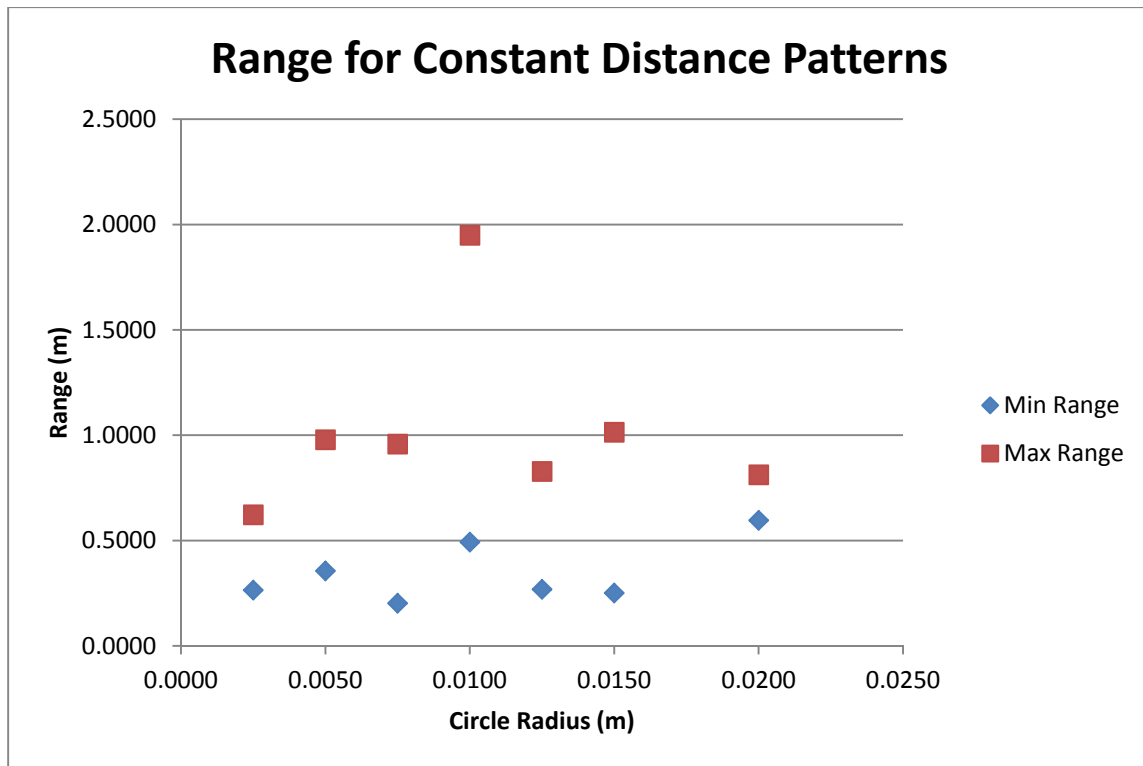
**Figure 10 - Maximum Tracking Range Experiment Setup: A – Bumblebee® camera system; B – Default tracking pattern**

Multiple trials were conducted in a controlled environment for a variety of patterns in order to obtain a measure of the average capability of the camera beyond the specifications provided by the manufacturer, see Figure 47 and Figure 48. The trials varied based on the patterns used for tracking. The default pattern used, shown in Figure 10B, was then used as a template for new patterns that were created by varying the distance between and the size of pattern features.

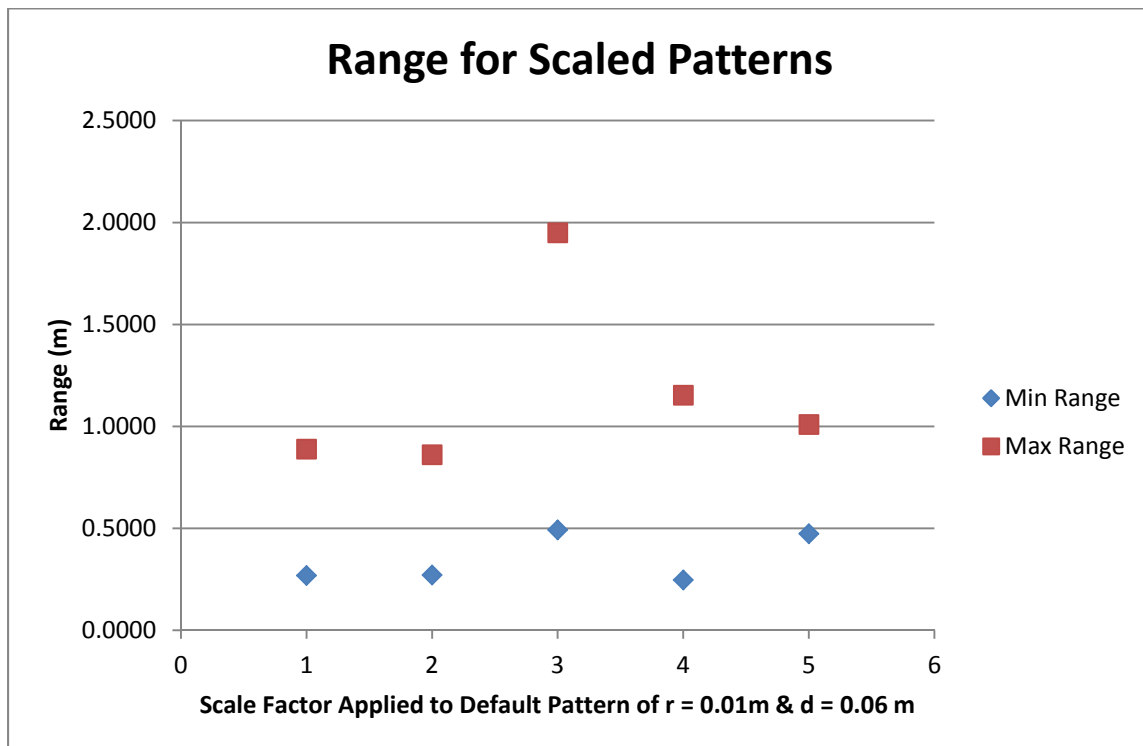
The results of this experiment are given below. Figure 11 illustrates the test results for those patterns which used identically sized circles set at varying distances. Figure 12 shows the results for tests using patterns in which the circle size was varied, but their positions were constant. Finally, Figure 13 shows the results for patterns which were scaled versions of the default pattern found in Figure 10B.



**Figure 11 - Constant Radius Patterns Results, Radius of Circles is 0.01 m**



**Figure 12 - Constant Distance Pattern Results, Distance Between Circles is 0.06 m**



**Figure 13 - Scaled Pattern Results**

The results of the tracking experiment indicated that the camera was able to consistently track the patterns between 0.25 *m* and 1.00 *m*. Some tracking data was able to be collected beyond 1.00 *m*. The default pattern, which is associated with the center peak of each figure, was consistently tracked up to 2.50 *m*. It is assumed that a similar experiment was performed by the manufacturer in order to provide the best pattern to the customer, which resulted in the default pattern performing significantly better than the others. Pose estimates from beyond 2.5 *m* were obtained more sporadically, and therefore were considered to be outside the camera's optimal range. Results from these tests are shown in Figure 11, Figure 12 and Figure 13.

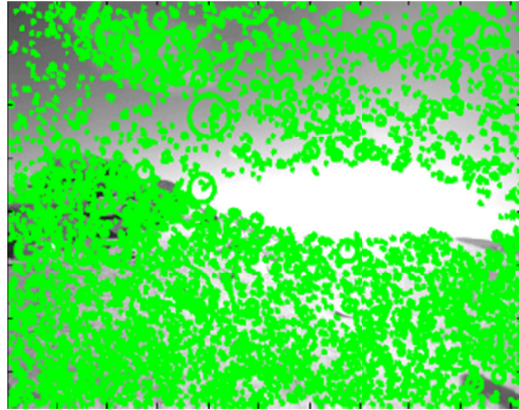
### ***3.1.2 Impact of Maximum Tracking Range in an Unstructured Environment***

A second experiment was conducted off-line in MATLAB using data from the MET to determine the impact of the camera's maximum tracking range on the results of the visual odometer. The primary consideration of this test was to determine impact of signal noise, which was expected to have a greater impact on descriptors corresponding to distant features. The frames were obtained using the stereo camera mounted on a stationary tripod. One such frame is shown in Figure 14.



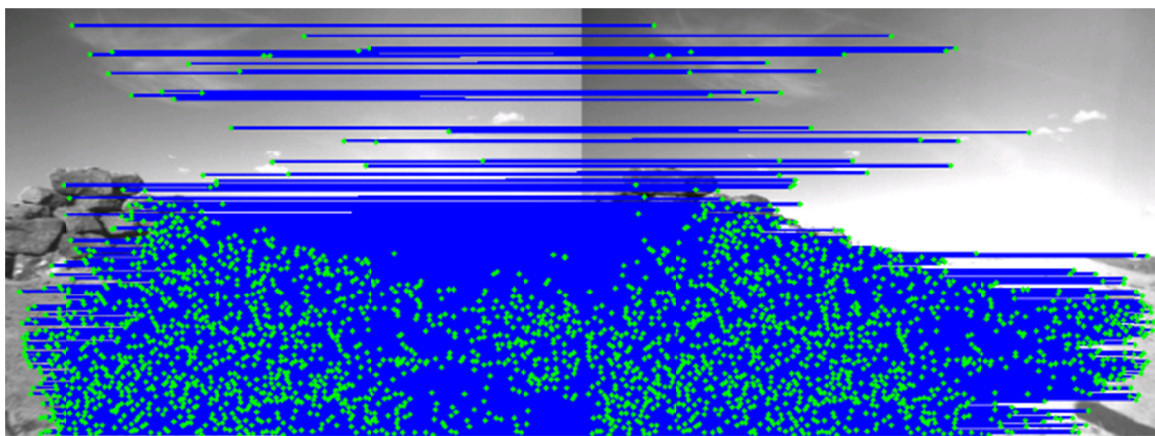
**Figure 14 – Sample of terrain used for the visual odometry algorithm development**

Performing feature detection on the sample image demonstrated that descriptors are detected in all areas of the image. This is illustrated in Figure 15 where a green marker is located to the position of each detected feature.



**Figure 15 - Sample left camera frame with descriptors**

By performing feature matching on the sample frame and plotting the results on top of a composite image, it was determined which areas of the image produced the most successfully matched data points.



**Figure 16 - Matched descriptors overlaid on the associated left and right camera frames**

Figure 16 is a composite image created using a stereo image pair showing a dot to mark each descriptor which was successfully matched to the correct feature in the corresponding image. A line is also shown for each descriptor in the left frame connecting it with its counterpart in the right frame. It is evident from Figure 16 that the vast majority of successfully matched descriptors were detected on the terrain as opposed to the horizon or the sky.

This test was performed in an off-line manner, and results were evaluated qualitatively with images similar to Figure 16. The results confirmed that distant points in the images were much more susceptible to noise and consequently fewer successful matches are obtained from these points. As a result, the decision was made to mount the camera at an angle such that it was  $30^\circ$  below the horizontal on the rover's pitch axis. This orientation limited the field of view of the camera such that the majority of the terrain was 0.5 – 3.0 m from the camera.

### 3.2 Data Collection

A variety of data was collected for use in development of the visual odometer. This was done to ensure the robustness of the resulting system. Three separate traverses of the MET were performed to collect preliminary data, and are shown on an aerial view of a CAD model of the MET in Figure 17.



**Figure 17 – Aerial view of the Mars Emulation Terrain at the Canadian Space Agency**



Collection of the data was performed using the instrumented cart as shown in Figure 18, while the experiment itself was performed offline. The red arrows indicate the paths followed by the cart during data collection and are marked with their corresponding number. Also, visible in Figure 18 are the Bumblebee<sup>®</sup> camera system and the GPS setup used.



**Figure 18 - Instrumented cart used for data collection**

Of the three traverses, shown in Figure 17, paths 1 and 2 were taken with the camera positioned horizontally relative to the rover wheel base. This orientation did not limit the viewing range of the camera. The result of this was that the sky and structures were visible in the frame, which allowed clouds to introduce interest points that did not often match successfully while the buildings created glare.

The imagery from path 3, also shown in Figure 17, was collected with the camera tilted 30° below the horizontal on the pitch axis. This orientation limited the visible range of the camera preventing the inclusion of the sky and distant objects in the frames.



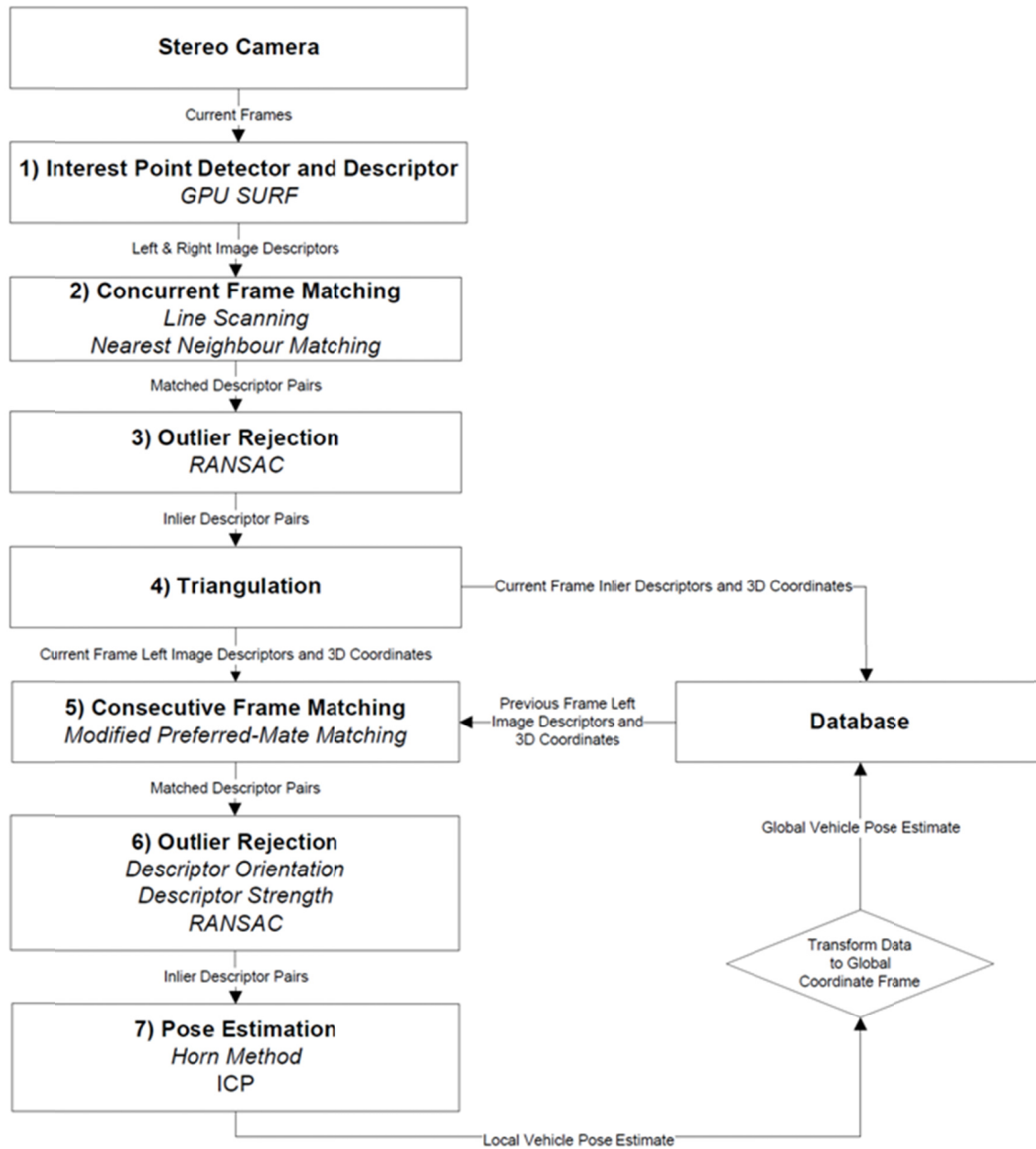


## 4 Visual Odometer Design

The processes of a visual odometer can be broken down into seven individual steps.

- 1) The first step is pre-processing, feature detection and feature description. This step encompasses all operations performed on the raw image data, such as rectification, as well as the application of a feature detector.
- 2) The second step is feature correspondence for concurrent frames. In this case matching is performed using the Nearest Neighbour matching scheme with line scanning applied to reduce the computation time.
- 3) The third step is outlier rejection for the concurrent frame point correspondences. This is performed using RANSAC.
- 4) Triangulation based on the correspondences achieved from the current frames is the fourth step of the algorithm.
- 5) The fifth step is to perform feature correspondence between the features of the current and previous left frames.
- 6) Outlier rejection is performed using RANSAC as well as strength and orientation information for the consecutive frames match set.
- 7) The seventh and final step of the algorithm is the pose estimation. The Horn method is used to determine an initial alignment of the data and the ICP algorithm refines this alignment.

The flow of data through the components of the visual odometer is illustrated in Figure 19.



**Figure 19 – Visual odometer structure**

In practice, it was not possible to select methods to perform the individual steps in the order they are used. It was also not to select them individually. This was due to the goal of developing a real-time implementation and evaluating the potential for the ICP algorithm to improve pose estimates. For example, the size of the feature descriptor used significantly impacted the computational cost of not only the feature detection and description phase, but also the feature correspondence phase. To reflect this, the visual

odometer components are presented in the order that they were selected as opposed to the order they are used in the algorithm.

#### **4.1 Triangulation & Alignment**

Triangulation is the process of recovering the 3D coordinate of a point from its projection onto 2D images. More specifically the triangulation component of the algorithm uses the point correspondences obtained from the feature association

The Horn method was selected as the method of determining the initial pose estimate because of its accuracy as well as its use of a closed form solution as opposed to an iterative method. This allowed an estimate to be obtained while minimizing the time required for processing.

ICP was selected to refine the pose estimates produced by the Horn method due to the success of previous work implementing it into pose estimation in other fields. To evaluate the effectiveness of ICP's inclusion in visual odometer, two modes were created within the visual odometer:

Mode 1: the algorithm uses only the Horn method to estimate the transformation, pose, between two consecutive frames associated with the rover motion.

Mode 2: at each frame, the algorithm refines the pose using ICP with the Horn method's pose estimate as the initial guess.

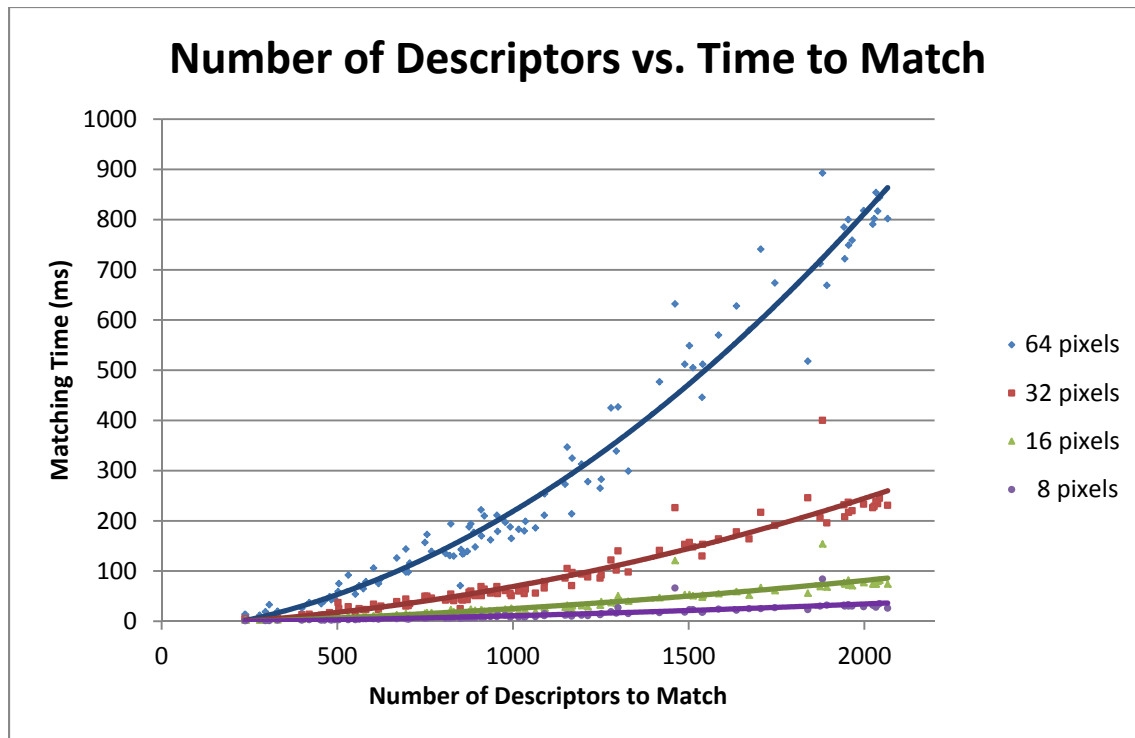
It is commonly known that the computational cost of ICP increases significantly as the matched data sets increase in size. For a system using large data sets seeking to run in real-time, the computational cost of ICP would have been prohibitively large. Due to the comparatively small data sets generated from the stereo camera, when compared to a laser scanner, the computational cost of ICP for pose estimation was found to be acceptable in this application. This is discussed further in Section 5.2.

## 4.2 Feature Association & Outlier Rejection

Different approaches to determine feature correspondence were selected for use in matching concurrent and consecutive frame descriptors. A technique called line scanning is used for concurrent frame matching to find candidate matches while the matching itself is performed using the Nearest-Neighbour approach. Line scanning is when the search for a matching descriptor in the corresponding image is restricted to the corresponding epipolar line and the area immediately adjacent to it. This technique was implemented to reduce the number of candidate matches and by extension the computation time. This was possible due to the geometry of the stereo camera system. Because the individual cameras are contained in one unit they are always mounted parallel to each other and at the same height. This means that corresponding descriptors occur in the same y-coordinate range of the captured images. To take advantage of this, the search for matching descriptors is limited to a  $\pm 2$  pixel range in the y-direction when matching descriptors from concurrent frames.

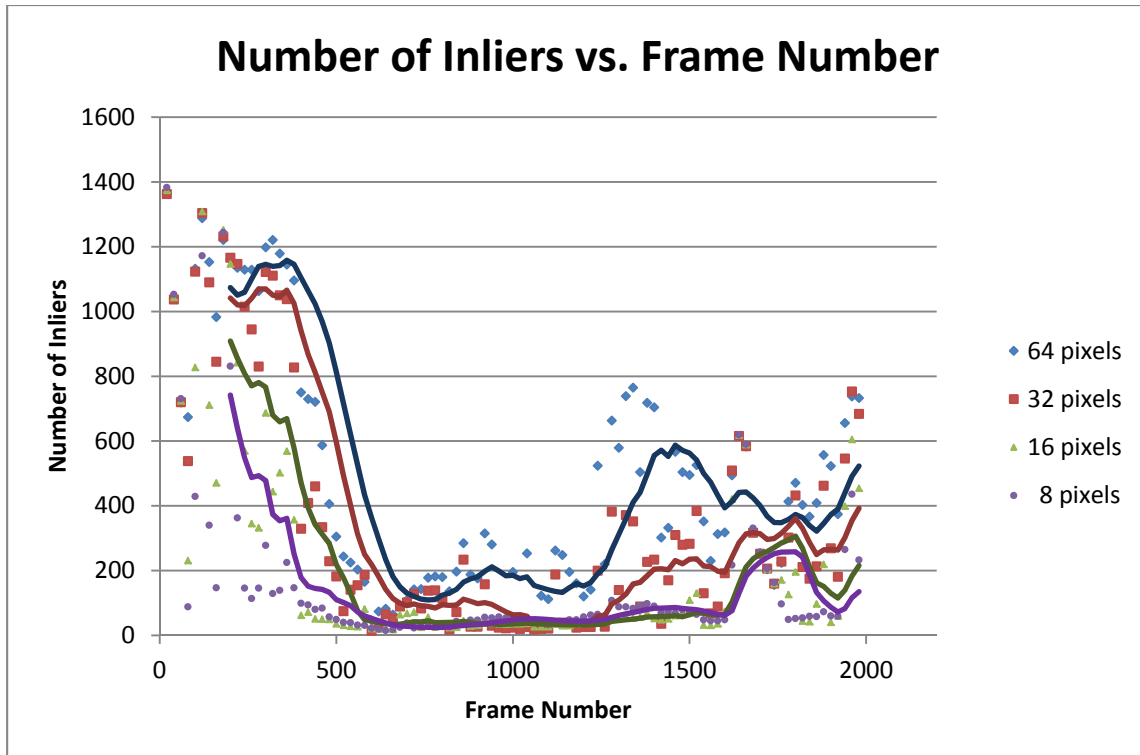
Matching for consecutive frames was performed using a modified version of the Preferred-Mate matching method instead of the simpler Nearest-Neighbour method. The reason for this was that this method performs better in rotation than the Nearest-Neighbour approach. In order to compensate for the additional computational cost associated with the Preferred-Mate matching technique, the search area for candidate matches was restricted.

By restricting the area searched for candidate matches, a significant savings in computational cost was achieved. It can be seen from Figure 20 that by reducing the search area around the descriptor from 64x64 pixels to 32x32 pixels, the time for computation dropped significantly. Reducing the area size to 16x16 pixels or 8x8 pixels resulted in further reductions in computation time, though the savings decrease exponentially. Overall it was found that a search area of 32x32 provided the best balance between computation time and the number of descriptors successfully matched.



**Figure 20 - Time required to match for a set of descriptors using varying search areas**

The number of successful matches obtained for varying scan area sizes, across a series of 2000 images, is illustrated in Figure 21. It is important to note that from frames 500 to 1000, significantly fewer inliers were detected for all scan area sizes. This is a reflection of the algorithm obtaining fewer descriptors from these images than the others. Therefore it is important to only consider the number of inliers of each trial relative to the others. It is evident that even reducing the search area from 64x64 pixels to 32x32 pixels has a measurable impact on the number of inliers detected.



**Figure 21 - Number of inliers detected for varying search area sizes**

For the final implementation of the visual odometer, a scan area of 32x32 pixels is used for consecutive frame matching. This decision came as a result of the need to balance accuracy with computation time. The reduction in time required for this type of matching eliminated a significant obstacle to creating a visual odometer that operates in real-time. As well, it should be noted that the decline in the number of inliers for all search area sizes around frame 550 is a result of the terrain yielding fewer descriptors. This reduction in the number of inliers can be addressed by having the vehicle traverse more slowly across the terrain in order to reduce the disparity between consecutive images and consequently reduce the likelihood of poor pose estimates.

The dominant method used for outlier rejection in visual odometry research is RANSAC as it is a robust approach that has been shown to remove outliers from data sets in which the number of outliers, or erroneous matches, is as high as 60%. Based on the past performance of the RANSAC algorithm and the lack of proven alternatives it was selected for use in the visual odometer as the primary outlier rejection method.

For consecutive frame descriptor correspondence, candidate matches are evaluated based a comparison of descriptor orientations and strengths, as defined by Lowe [43]. Any pairs where the difference in orientation or strength exceeds a threshold were removed prior to evaluating the set of descriptor pairs using RANSAC. This combination of methods was found to require somewhat less computation time than running RANSAC alone. This was not the case when matching descriptors from concurrent frames. RANSAC was the only outlier rejection algorithm used for these matches.

### 4.3 Selection of a Feature Detector

Three candidate feature detectors were identified from the literature review performed at the onset of the project to be the most robust. These feature detectors are SIFT, SURF and GPU-SURF. SIFT has been used very successfully by the research community since its development, and SURF, though only recently developed, has also proven to be both accurate and reliable in the short time it has been available. Finally, GPU-SURF is a new approach that has not yet been widely implemented, but was selected as a candidate for its potential to provide comparable results to the other algorithms at a reduced computational cost.

Given that the accuracy of the three candidate algorithms has been reported by related research to be comparable, the performance indicator used in the selection process was the processing time. Each candidate was applied to the same series of frames to evaluate their performance, the average results are given by Table 1. The computer used for the evaluation is equipped with a Pentium 4HT processor.

**Table 1 - Average algorithm processing time for different feature detectors**

Descriptor Type	Processing Time [ms]	Operating Frequency [Hz]
SIFT	7517.74	0.1330187
SURF	2447.66	0.408553476
GPU-SURF	559.73	1.78657567

It can be seen from Table 1 that the computational cost of GPU-SURF is an average of 4.37 times lower than SURF and 13.43 times lower than SIFT. Consequently, in order to obtain a comparable level of accuracy, a vehicle utilizing the SIFT or SURF algorithms would be required to travel much slower than a vehicle utilizing GPU-SURF respectively. While irrelevant to an algorithm intended for use as a post-processing error correction tool, the significantly lower operating frequency makes the computational costs of SIFT and SURF prohibitively large for use in real-time. Consequently, GPU-SURF was selected as the feature detector and descriptor for the visual odometer.



## 5 Experimental Characterization

### 5.1 Experimental Setup

Characterization of the algorithm was performed off-line using additional data collected from the Canadian Space Agency's Mars Emulation Terrain. Instead of reusing the instrumented cart shown in Figure 18, the stereo camera system was mounted on the CSA's MRT. The reason for this was the intention to validate the algorithm under the motion of an actual rover as opposed to a cart controlled by a person.

The MRT is shown in Figure 22 with the stereo camera mounted at the top at a downward angle of  $30^\circ$  and a GPS frame consisting of three real time kinematic, RTK, GPS receivers. The RTK GPS receivers provided the ground truth data that was the basis of the algorithm characterization, but were unused by the visual odometer itself. The ground truth data is accurate to within  $0.04\text{ m}$ .

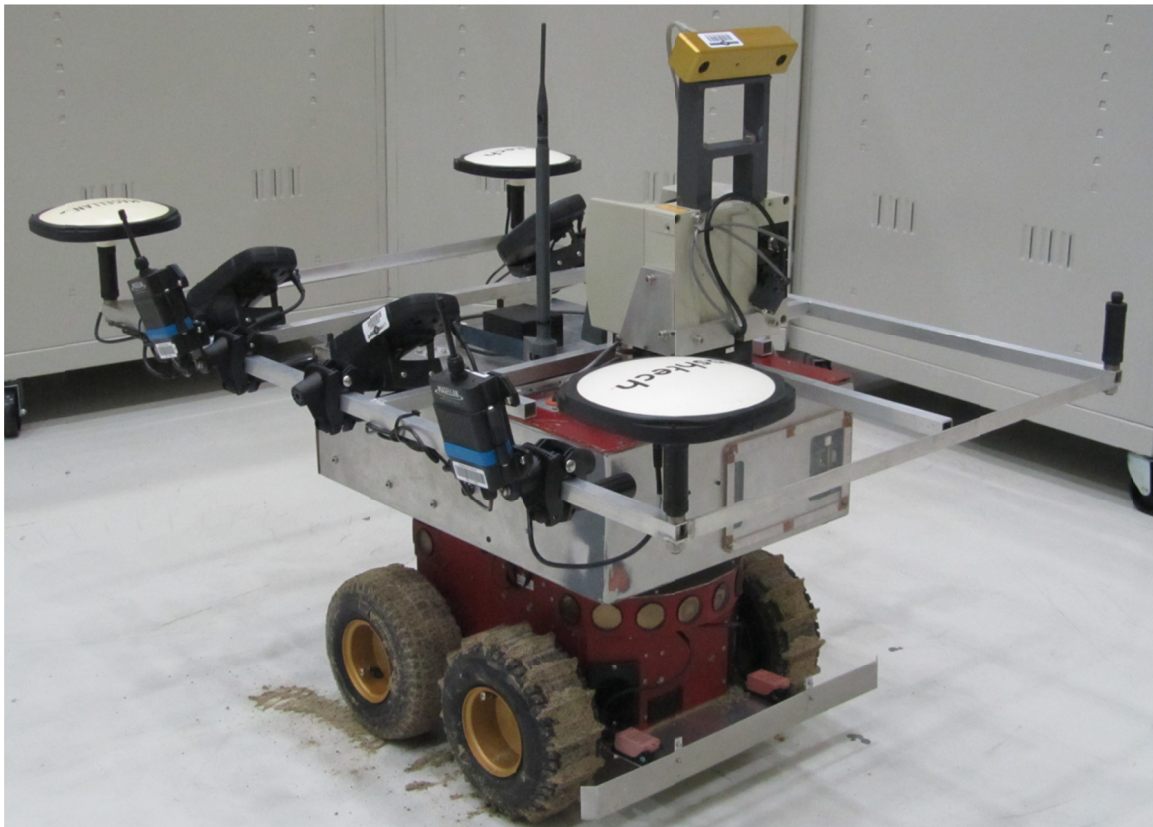


Figure 22 – The MRT equipped with a stereo camera and GPS frame

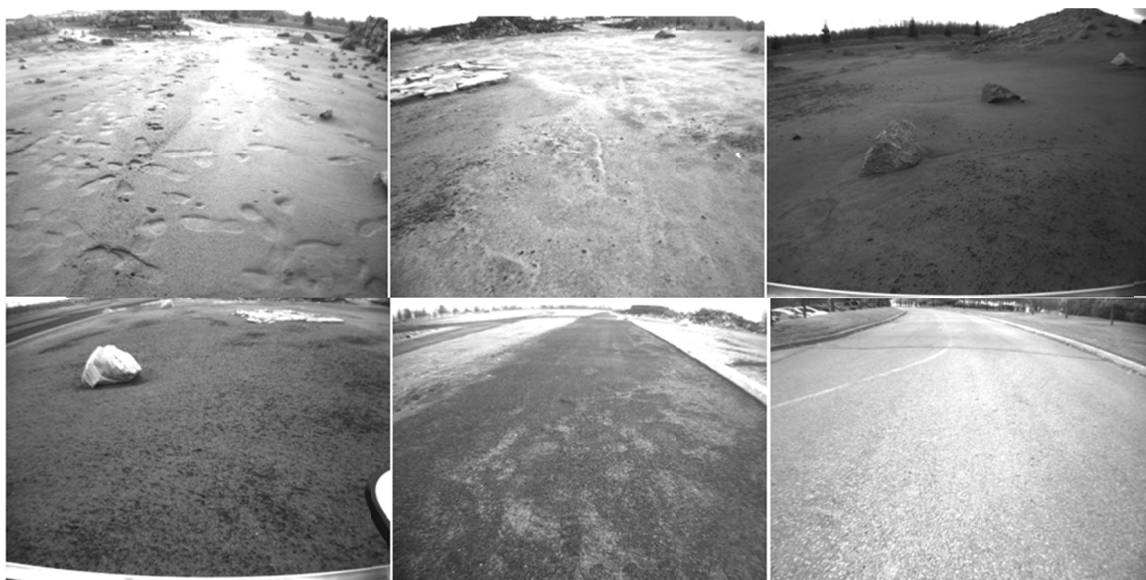
The rover itself is a modified Pioneer 2AT from Mobile Robots Inc. The computing processing unit (CPU) of the rover consists of a CF-30 Toughbook™ and the operating system of the rover is a Linux distribution, namely Ubuntu 8.0.4. The rover features a series of custom electromechanical interfaces to allow for additional hardware to be added easily.

## **5.2 Experimental Results**

A number of traverses were performed in and around the MRT, at a speed of 0.10 *m/s* to generate data with which to characterize the visual odometer and to evaluate the ICP algorithm as a part of the visual odometer. The results of off-line processing of the data using the visual odometry algorithm in both Modes of operation are discussed below. The data for each trial was collected and processed off-line by the visual odometer. The error metrics used for evaluation during the test cases are the average percent error in the overall rover displacement and the average error in yaw angle estimation. Data on rotation for roll and pitch are not compared because the terrain did not cause the vehicle to pitch or roll noticeably. This would result in inflated error estimates.

### ***5.2.1 Characterization of the Visual Odometer***

In order to characterize the visual odometer under varying conditions, traverses were performed in different types of terrain and under various lighting conditions. A composite image of the test cases is shown in Figure 23.



**Figure 23 - Various environmental conditions experienced during the characterization of the visual odometer**

As can be seen from Figure 23, the terrain over which localization was performed included wet and dry sand, gravel and pavement. Data was not collected from extremely rocky terrain due to the ground clearance limitations of the rover used. Lighting conditions varied as well, as data was collected during both clear and overcast weather conditions. The results of the traverses are given by Table 2.

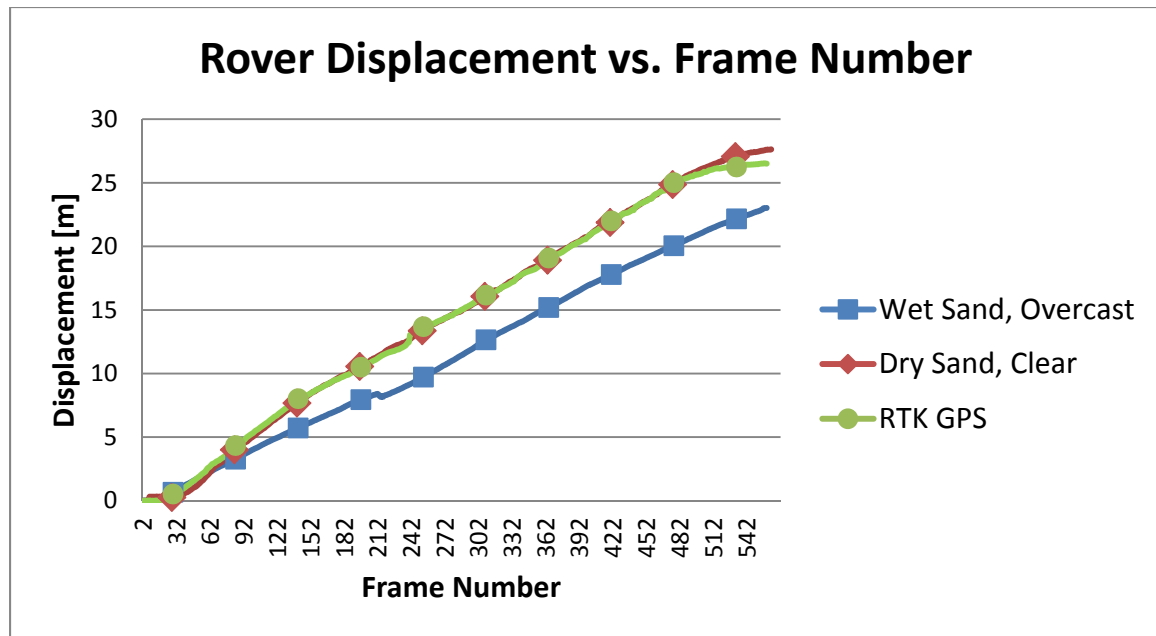
It should be noted that ground truth data was only available for displacement during traverses 5, 6 and 7. The reason for this is the fact that the data for these cases was collected using the instrumented cart instead of the rover, which was only equipped with a one RTK GPS antenna. The other five traverses were performed using the rover equipped with three RTK GPS antennas.

**Table 2 - Average error experienced during multiple traverses**

Traverse	Displacement [%]	Yaw Angle [°]	Terrain	Displacement [m]
1	5.085%	4.572	gravel, clear	14.075
2	20.480%	6.504	wet sand, overcast	26.503
3	20.249%	13.008	wet sand, overcast	32.993
4	43.212%	47.147	wet sand, clear	90.830
5	14.588%	N/A	wet sand, clear	42.717
6	89.600%	N/A	wet sand, clear	145.534
7	67.031%	N/A	pavement, overcast	109.048
8	6.670%	31.395	dry sand, clear	25.264

It can be seen that both the displacement error and the yaw error varied significantly between the different traverses. Comparing traverses 4, 5, 6 and 8, we can see that the texture change between wet and dry sand had a significant impact on localization error, with lower error being seen during dry conditions. This is likely due to the decreased texture of the environment seen when the terrain becomes saturated. Taking note of the weather conditions, it is evident that the best results were predominantly obtained from data collected on days having clear weather.

The impact of the weather conditions is most evident when comparing traverses 2 and 8. The paths followed for these data sets are nearly identical, having been started from the same initial position and following the same trajectory. The change in environmental conditions is the only significant difference between the two traverses yet the localization error is significantly higher during traverse 2. Figure 24 illustrates this, showing only one of the ground truth estimates to avoid cluttering the figure.



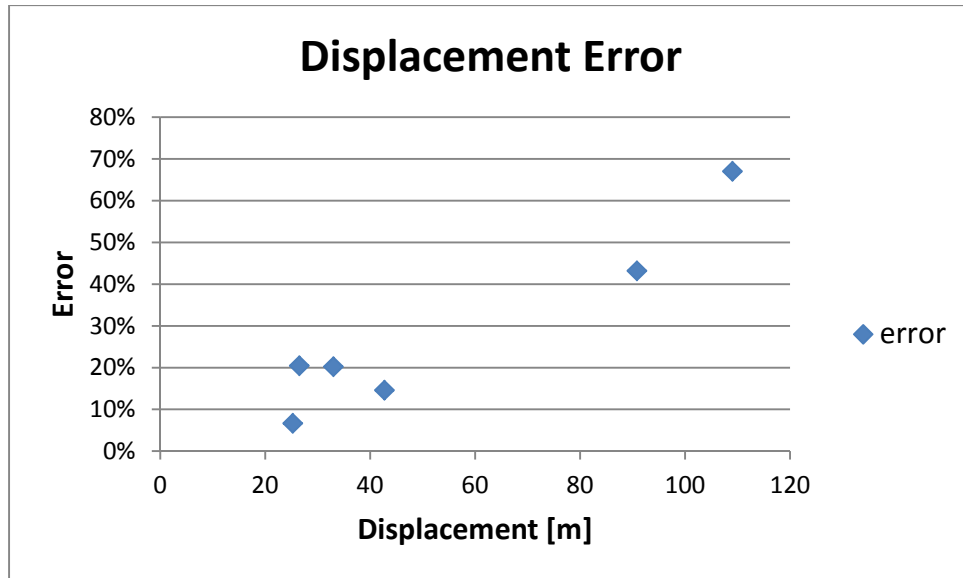
**Figure 24 - Comparison of rover traverses 2 and 8**

Another factor that should be addressed for its potential to have an impact on the localization accuracy, is the duration of the traverses. Table 3 gives the data shown before, in Table 2, in order of increasing traverse duration based on the number of frames collected.

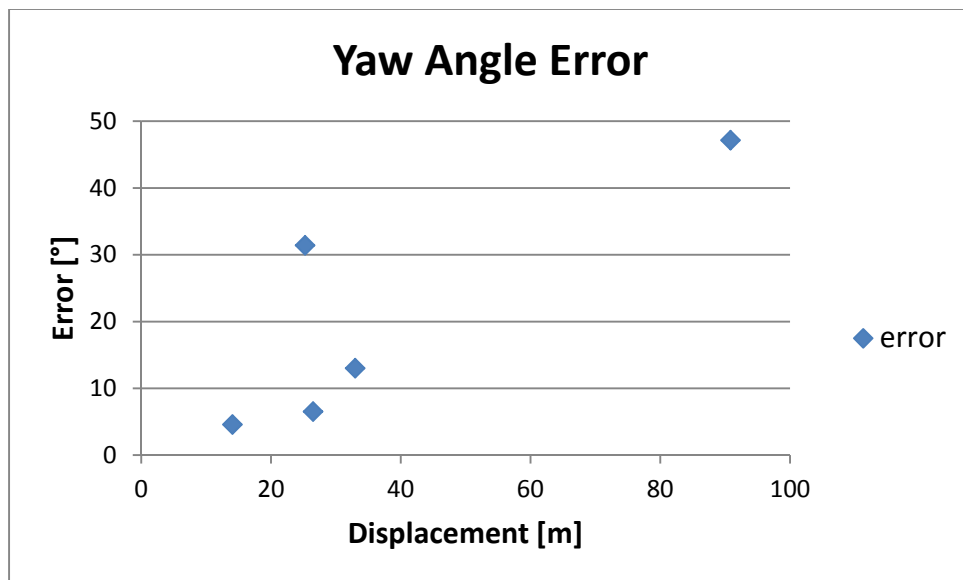
**Table 3 - Average errors shown in order of the length of the associated traverse**

Displacement [m]	Displacement [%]	Yaw Angle [°]
14.075	5.085%	4.572
25.264	6.670%	31.395
26.503	20.480%	6.504
32.993	20.249%	13.008
42.717	14.588%	N/A
90.830	43.212%	47.147
109.048	67.031%	N/A
145.534	89.600%	N/A

Looking at Table 3, there appears to be an overall trend to increase the average translation error as the traverses become longer. Similarly the average yaw error also appears to increase for the longer traverses. Figure 25 and Figure 26 illustrate this.



**Figure 25 - Displacement error vs. the length of the traverse**

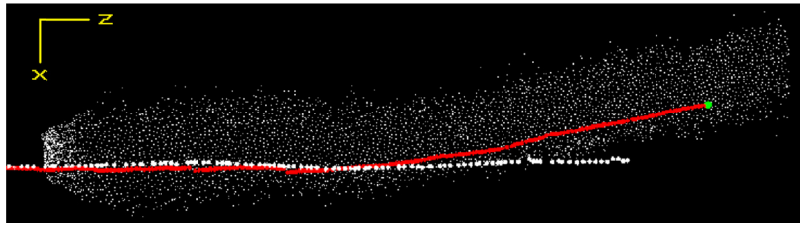


**Figure 26 - Yaw error vs. the length of the traverse**

Individual figures illustrating the visual odometer performance compared to the ground truth data can be found in Appendix A.

The yaw error has an important impact on the error in the final position. Even in those cases where the visual odometer displacement error was relatively low, the final

position can still significantly deviate from the ground truth estimate. This is illustrated below in Figure 27.



**Figure 27 - Graphical output of offline processing**

Figure 27 shows the graphical output of the off-line visual odometer testing where the green sphere represents the rover and the red line is the path estimated by the visual odometer. The white line is the actual path followed, constructed from the ground truth data, and the point cloud is the triangulated data used by the visual odometer for the ICP algorithm. The impact of erroneous yaw angle estimations is apparent where the estimated rover path deviates from the ground truth data. This is evident after approximately 40 % of the traverse is completed when the algorithm estimates a sharp turn. After this point no significantly high errors are found in the yaw estimates, however the path continues to deviate due to the new heading.

It can be seen from the results presented here that the presented visual odometer was not able to achieve the results published in other research. The traverses on which the algorithm performed best achieved approximately 5.0 % and 6.5 % error in terms of displacement and 4.5° and 6.5° error in terms of yaw. Data published by other researchers put their displacement and yaw rotation errors at approximately 2.0% and 5° respectively.

A more definitive statement can be made about the impact of ICP on the accuracy of the algorithm. The first two traverses were also processed without the use of the ICP algorithm in order to form a basis on which to assess the impact of ICP on localization accuracy. The results of this are discussed below.

### **5.2.2 Impact of ICP**

*Characterization Case 1.* As seen in Figure 28 and Figure 29, the first characterization case was a traverse over a flat terrain characterized by gravel. In addition, the terrain was dry and the weather was clear on the day of the experiment.

This scenario was evaluated for the challenge it provided to the camera, and would provide to a laser scanner based system, which would have been unable to localize properly under such uniform environmental conditions.



**Figure 28 - Characterization case 1 rover trajectory**



**Figure 29 - Characterization case 1 terrain texture**

Table 4 gives a summary of the average errors for the algorithm applied to the data in both modes of operation. The visual odometer algorithm running ICP (Mode 2) had an average displacement estimate error of 5.085%, while the associated error for the algorithm omitting ICP is 19.535%. The average yaw error using ICP is also measurably



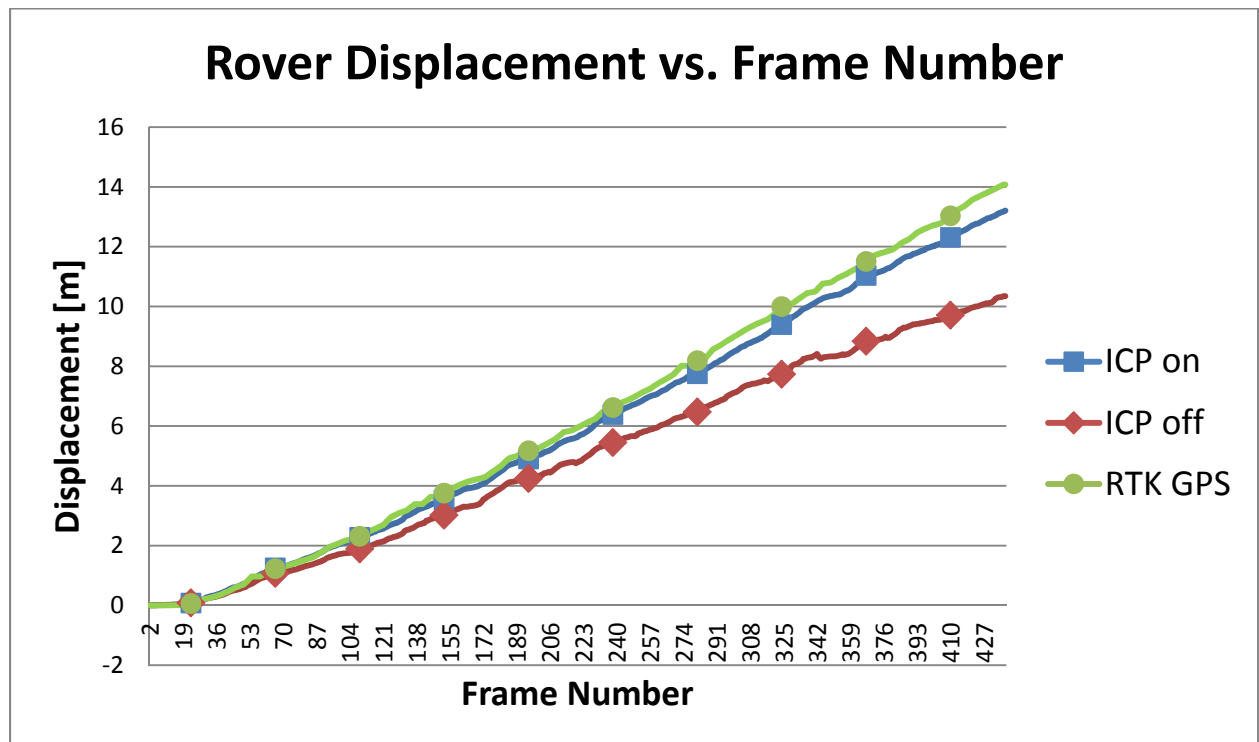
less than the average error obtained without using ICP, at 4.572° and 8.355° respectively.

The results of the first case appear to indicate that the ICP algorithm significantly improves both the displacement and yaw errors. A more complex traverse is analyzed for case 2 to determine if the same trend is visible under different conditions.

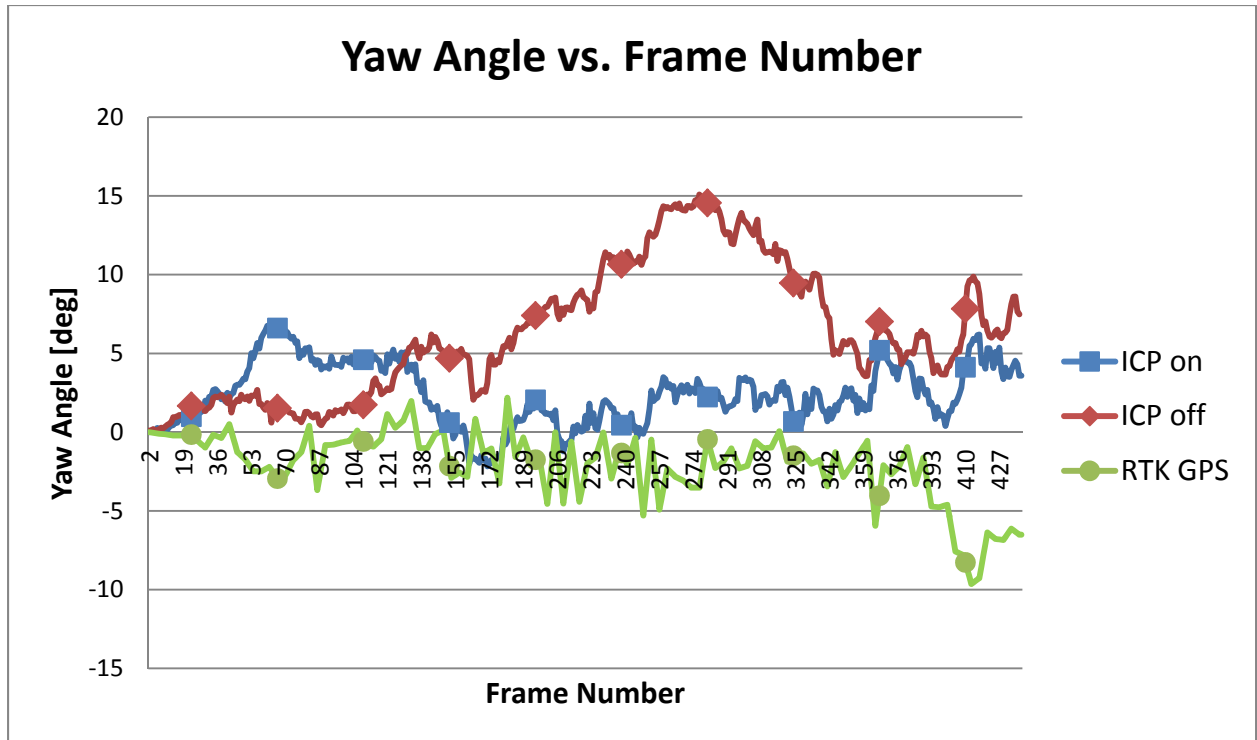
**Table 4 - Characterization case 1 average error**

Mode	Pose Estimate	Yaw Error
ICP on	5.085%	4.572°
ICP off	19.535%	8.355°

The experimental results obtained for the rover displacement and yaw angle during the first case are given by Figure 30 and Figure 31 respectively.



**Figure 30 - Characterization case 1 rover displacement**

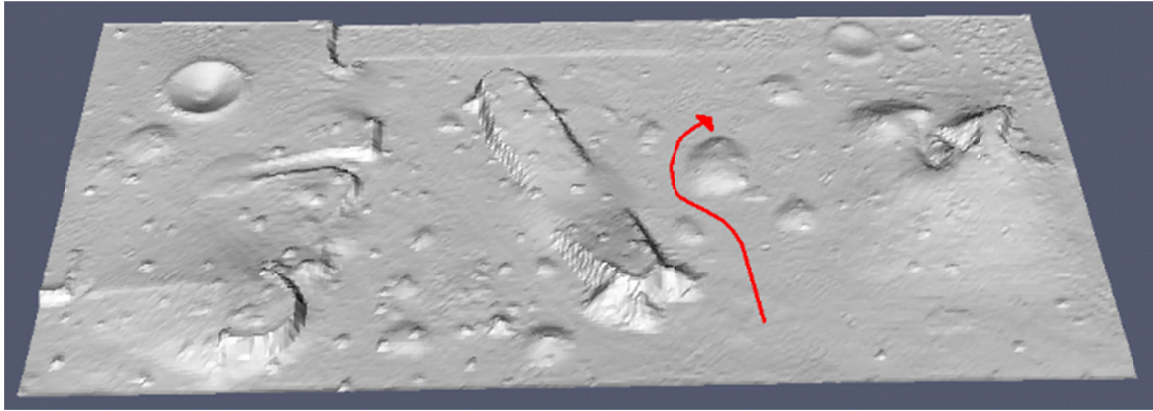


**Figure 31 - Characterization case 1 rover yaw angle**

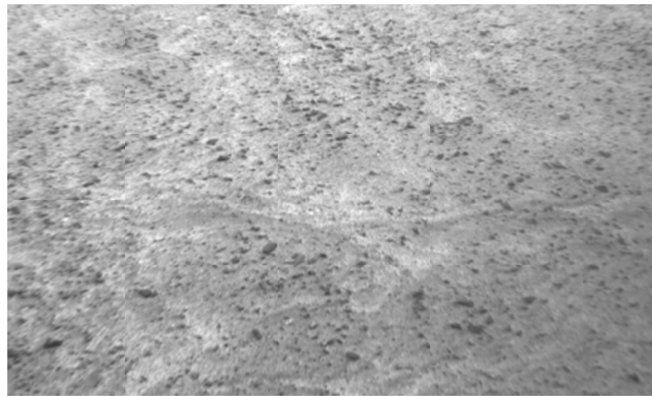
A deviation of up to  $15^\circ$  from the ground truth GPS reading is seen in Figure 31 for the algorithm when used without ICP. Though it is known that the yaw angle is particularly difficult for a visual odometry algorithm to estimate, this deviation is higher than anticipated. This likely indicates that a larger number of incorrect matches were characterized as inliers during the middle portion of the traverse, which adversely affected the estimate obtained from the Horn method. This is significant because the Horn method will only be able to obtain the optimal solution to the alignment problem if the data is composed entirely of correct matches. The results, specifically the reduced error in terms of displacement and the deviation in the yaw angle midway through the traverse, demonstrate the benefit obtained from introducing the ICP algorithm.

*Characterization Case 2.* As seen in Figure 32 and Figure 33, the second case is a curved traverse through terrain characterized by sand and a variety of small to large rocks. This scenario was selected as it is a very similar environment to that found during planetary exploration missions to Mars. Additionally, the shape of the path was chosen to provide

a higher level of difficulty than the previous case. Rotational motion causes features to pass out of the camera's field of view more quickly than during translational motion, resulting in fewer inliers, making rotational motion more difficult to correctly estimate. Finally, the experiment was performed while the sand was wet and the sky was overcast, the opposite of the conditions in the previous case.



**Figure 32 - Characterization case 2 shown on a CAD model of the MET**



**Figure 33 - Characterization case 2 terrain texture**

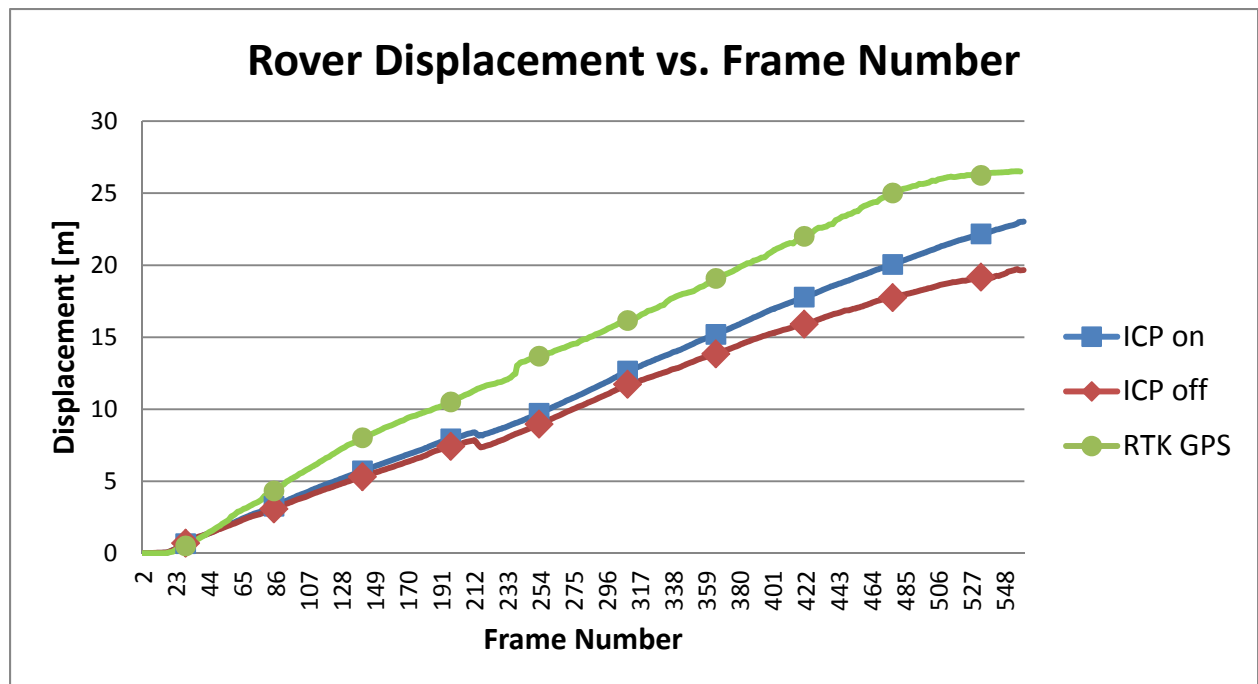
Table 5 summarizes the average errors in the results of the visual odometer pose estimates. The average displacement errors for the ICP on and ICP off cases are 21.611% and 29.158% respectively while the associated average yaw errors are 6.419° and 13.725° respectively. It is again evident that the algorithm performed significantly better when utilizing ICP. This result matches that seen in the first case despite the significantly different environmental conditions.

**Table 5 - Characterization case 2 average error**

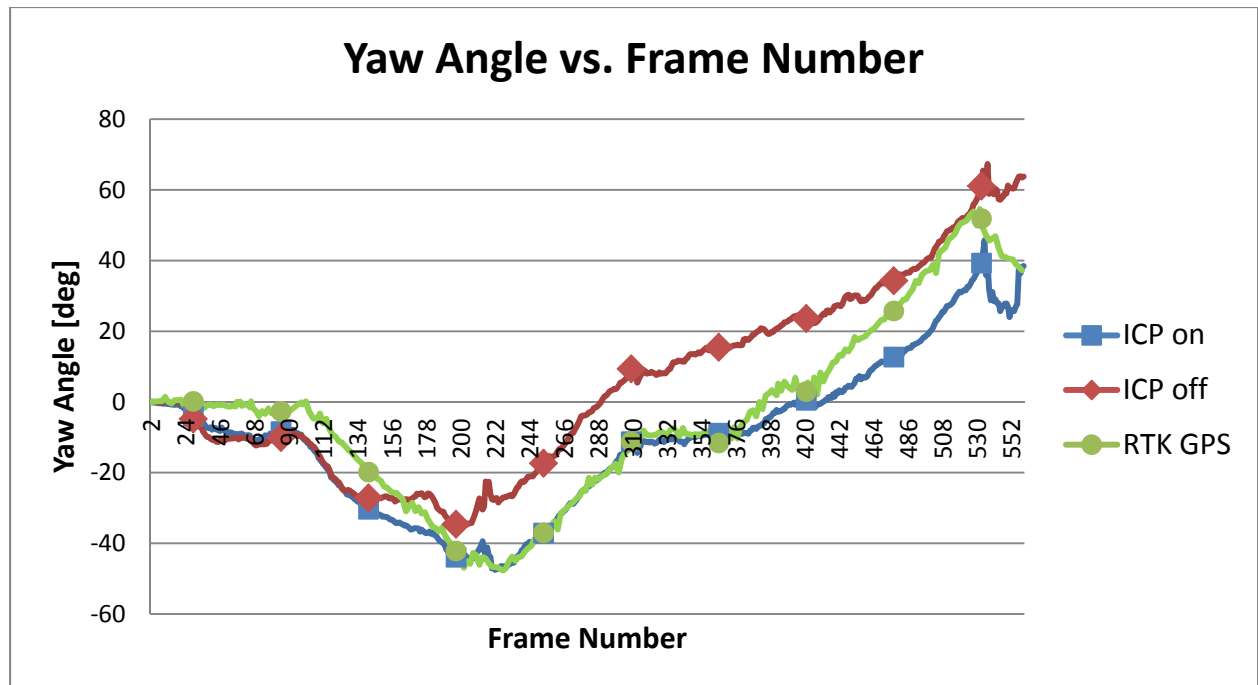
Mode	Pose Estimate	Yaw Error
ICP on	21.611%	6.419°
ICP off	29.158%	13.725°

The results confirm what was seen in the first case. The presence of the ICP algorithm improved the pose estimate obtained by the Horn method, which was influenced by the presence of erroneous matches in the inlier data set.

Figure 34 and Figure 35 show the estimated rover displacements and yaw angles for both visual odometer modes compared to the ground truth data. The experimental results obtained from the second experiment showed a decrease in localization error when using ICP. While no significant deviations, such as those seen in Figure 31 are evident, the deviation of the algorithm's estimates from the ground truth data is less severe when using ICP.



**Figure 34 - Characterization case 2 rover displacement**



**Figure 35 - Characterization case 2 rover yaw angle**

By comparing the results of the second case to the first, it can be seen that the ICP algorithm had a positive impact on algorithm accuracy when estimating both the displacement and the yaw angle of the rover. This trend was seen despite varying environmental conditions and terrain types.

### 5.2.3 Computational Cost

Computational cost was measured in term of the time required to perform the individual operations of the visual odometer. The average cost is summarized in Table 6.

**Table 6 - Average computational cost**

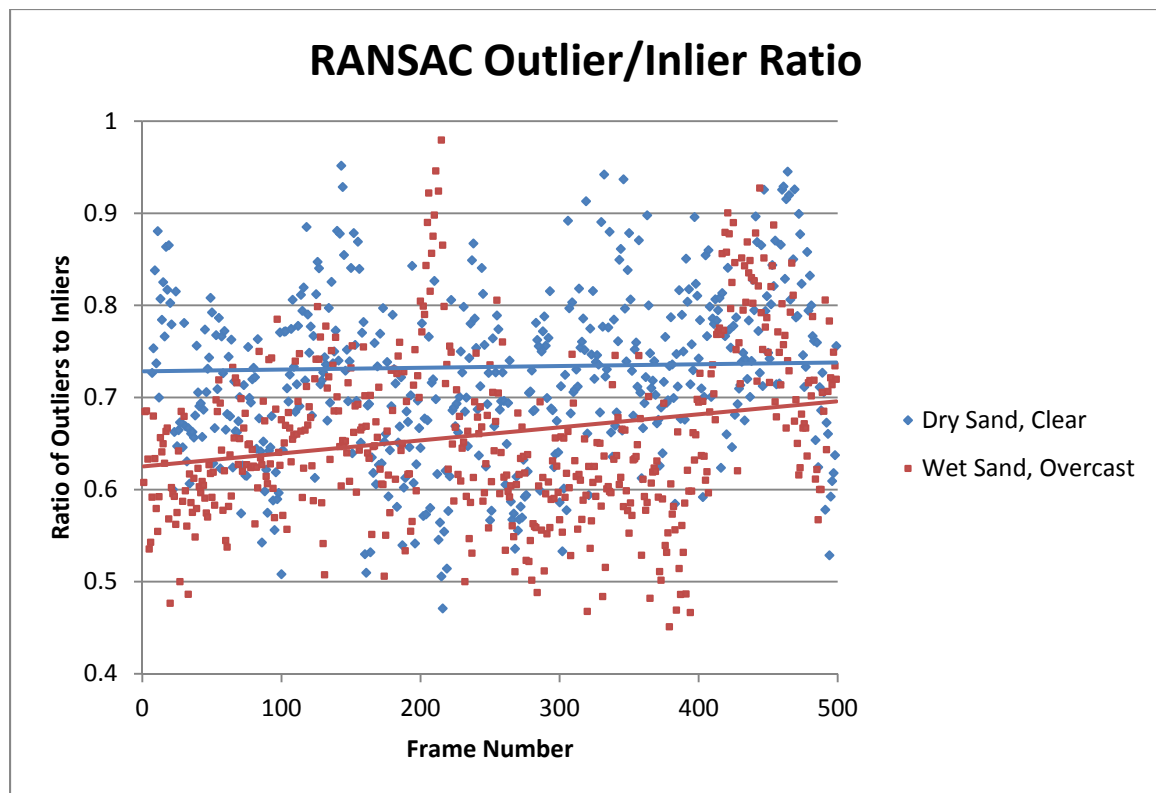
Process	Time (ms)
GPU-SURF	433.31
Feature Matching & RANSAC	96.7
Horn Method & ICP	29.71
Total	559.73

It is evident that feature detection remains the largest contributor to computational cost, though there is room for further improvement in the feature matching and outlier rejection time as well. As well, it is important to note that while the ICP algorithm is typically computationally intensive, its impact in this context is sufficiently small in comparison to the other algorithm operations that it can be used in this application. This is attributed to the smaller data set obtained by a stereo camera then from a laser scanner, the sensor most often used in conjunction with ICP.

### **5.3 Functionality Verification**

Some additional tests were performed to verify that the outlier rejection and pose estimation components of the algorithm are functioning correctly in light of the high error seen in the localization results from those days which had poor weather conditions.

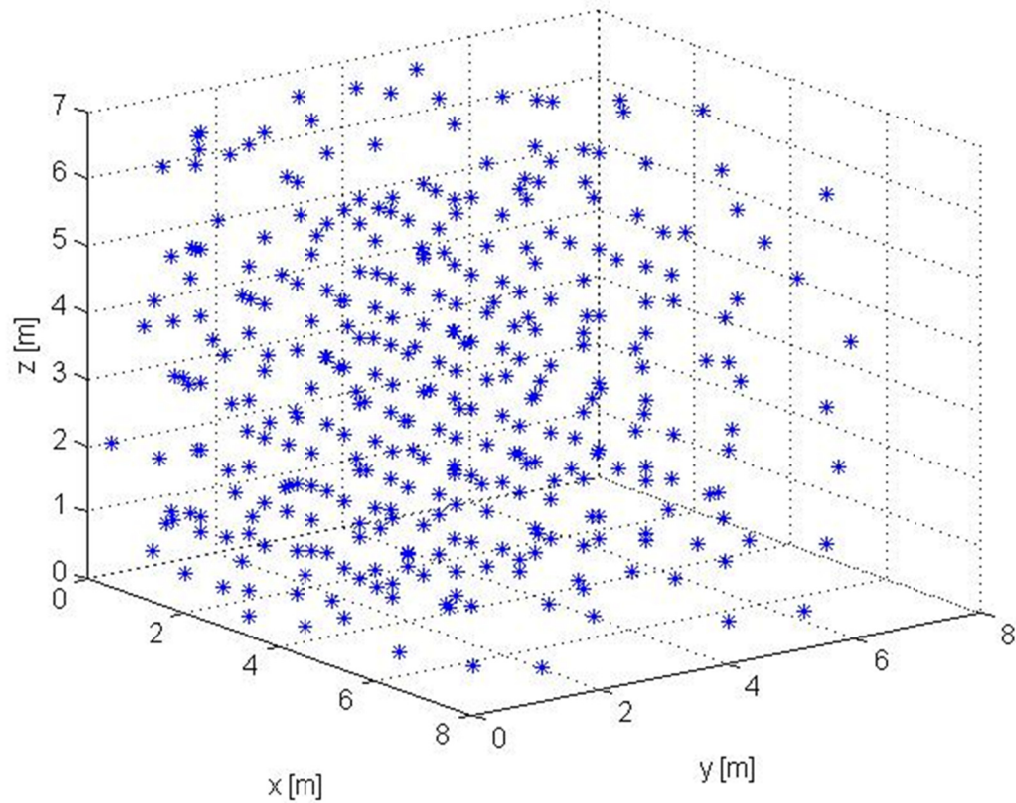
*Functionality Test 1.* The first test was to determine if the outlier rejection algorithm, RANSAC, was performing properly, the results of which are illustrated by Figure 36.



**Figure 36 - Outlier rejection results during varying weather conditions**

It can be seen from Figure 36 that the ratio of outliers to inliers is consistently higher during the traverse of fair weather conditions. By detecting a lower percentage of outliers during the traverse under poor weather conditions, more outliers are erroneously categorized as inliers and are used to perform motion estimation. This is a source of error which contributes to the higher error seen under poor weather conditions. It is possible that by adjusting the parameters used by RANSAC, the performance during poor weather could be improved somewhat.

*Functionality Test 2.* A comparison of the localization components of the different modes of the visual odometer was also performed on synthetic data to evaluate the algorithm's response to a 100% inlier data set. A point cloud in the shape of a cube with rough face was generated using MATLAB to form the reference data, and a known transformation, given in Table 8, was applied to generate the collected data. The synthetic data can be seen in Figure 37.



**Figure 37 - Synthetic data used to compare the modes of the visual odometry algorithm**

**Table 7 - Transformation applied to the synthetic data**

Transformation	Magnitude
X [m]	0.06
Y [m]	0.01
Z [m]	0.36
Yaw [°]	5

The results showed that Mode 1, using the Horn method without ICP refinement performed, better than Mode 2, which used the ICP algorithm. The results are given below.



**Table 8 - Average error in localization estimates for synthetic data**

Error	Mode 1	Mode 2
Displacement	0.0002%	0.7069%
Yaw	0.0003°	0.3230°

The results indicate that both modes of the algorithm are capable of accurately performing localization of the data set shown in Figure 37. The mode 2 results are slightly different from Mode 1 results because of noise injected into the simulated data when performing ICP.



## 6 Conclusions

Accurate localization estimation is a critical task for autonomous vehicles including those used for exploration of other planets. The majority of systems currently in practical use perform localization using either IMU's or wheel odometry. Such approaches become increasingly inaccurate over time due the drift experienced by IMU's and the slippage that is unaccounted for in wheel odometry estimates.

An alternative means of performing localization is visual odometry. It also experiences error accumulation but this can be mitigated by the visual odometry algorithm itself through the use of bundle adjustment and loop closing techniques.

The primary contribution of this thesis is the development of a visual odometry algorithm capable of operating in real time and in the unstructured environments of other planets. This work also provides a framework for use in future research. Imagery is collected using a stereo-camera system which is used to obtain localization estimates. The visual odometry algorithm can function in two modes.

Mode 1: the algorithm uses only the Horn method to estimate the transformation, pose, between two consecutive frames associated with the rover motion.

Mode 2: at each frame, the algorithm refines the pose using ICP with the Horn method's pose estimate as the initial guess.

Evaluation of ICP in the context of the visual odometry algorithm found that it had a minimal impact on the overall computational cost of the visual odometer. The incorporation of ICP resulted in a significant reduction in the error of the yaw angle estimates, a parameter typically difficult to estimate accurately using visual odometry. The impact of ICP on the translation error estimates was less conclusive and should be studied further using larger traverses.

In conclusion, running the algorithm in Mode 2 is found to be beneficial for rotation accuracy, with minimal additional computational cost.

It is recommended that sensor fusion be explored in future work as a means of further reducing pose estimation error.

Data used for the development of the visual odometer was collected in from the Canadian Space Agency's Mars Emulation Terrain (MET). Performance evaluation of the completed algorithm was also founded on data collected from the MET, ensuring that the algorithm could operate in a relevant field environment.

### **6.1 Summary of Specific Results**

Given the problem of localizing a vehicle in an unknown and unstructured environment, this thesis works to develop a visual odometry algorithm that is capable of accurately localizing in real time. It was found that:

1. The SIFT and SURF algorithms significantly limited to allowable speed at which a vehicle could travel if localizing in real time.
2. The GPU-SURF algorithm was able to partially alleviate the limitations in vehicle speed by allowing the algorithm to operate at a higher frequency.
3. Even a single inaccurate estimation of the vehicle yaw angle resulted in significant deviation of the estimated final position from the true final position.
4. The integration of the ICP algorithm into the visual odometer substantially reduced the error in vehicle yaw angle estimates, a parameter that is typically difficult to estimate using visual odometry.
5. While ICP is computationally intensive on large data sets, the computation time was sufficiently small in this application to be viable for use in a real time algorithm.
6. The algorithm demonstrated a sensitivity to the lighting conditions, a known problem in visual odometry research
7. The algorithm which was developed is capable of operating in real-time, however the accuracy achieved is not comparable to other implementations.

## 6.2 Future Work

Additional work is recommended to reduce the pose estimation error and improve robustness, as the present version is not yet suitable for practical application. It is recommended that sensor fusion, loop closing and bundle adjustment be explored as potential means of achieving this goal. As well, obstacle avoidance, the kidnapped robot scenario and self-calibration are areas of research with the potential to further the visual odometer presented here.

Sensor fusion offers another means by which the estimation error could be reduced and should be considered for future work. It is possible that error could be reduced through the integration of a second, independently determined, initial pose estimate. While IMU's or laser scanners could be applied, these options incur an additional cost in terms of volume, mass or power consumption. An alternative sensor, which should be considered, is the right camera of the stereo camera system. The consecutive frames from the left camera are the basis for the visual odometer's initial pose estimate. The consecutive frames from the right camera, currently unused, could produce a second, independently determined, initial pose estimate. By comparing the two initial pose estimates, this approach has the potential to improve the localization estimation error at a very low cost.

A focus of current research is the loop closing problem. Loop closing occurs when a vehicle conducting localization and mapping operations returns to a previously visited location. In this event, the ideal scenario would be that the newest data corresponds exactly with previously collected data. In reality this is unlikely due to error in the previous estimates or differing perspectives on the environment. If the algorithm is able to close the loop and achieve correspondence between new and old data, then the error in the previous localization estimates can be reduced via bundle adjustment. Future integration of bundle adjustment is highly recommended.

Bundle adjustment is the refinement of the 3D coordinates and relative motion estimates obtained from imagery of a scene taken from multiple viewpoints.

Performing bundle adjustment has been shown to increase pose estimation accuracy and should be considered for future implementations of the algorithm.

The visual odometry algorithm would also benefit from work on the kidnapped robot scenario. The kidnapped robot scenario is when an event occurs that prevents the vehicle from successfully localizing. The problem is that when the vehicle is able to resume normal operation, it must attempt to localize relative to previously collected data even though it may have continued to traverse the terrain during the period where localization had failed. A major component of this problem, as with loop closing, is that the vehicle must be capable of recognizing previously viewed terrain from different perspectives if it becomes turned around.

As well, the visual odometer could be advanced by integrating it with an obstacle avoidance algorithm. The algorithm in this thesis required that the rover be driven by an operator. A major step in removing the need for human assistance in the rover's operations is to incorporate obstacle avoidance capabilities.

Work could also be done to incorporate self-calibration into the visual odometer. This would result in a more robust algorithm that would be readily transferable to vehicles utilizing different stereo vision systems. As well, the algorithm described here could be used for doing studies on slip detection. By comparing visual odometry to wheel odometry, the algorithm could be used to better characterize conditions under which slippage is most severe and to test methods of preventing it.

## Appendices

### Appendix A: Algorithm Characterization Figures

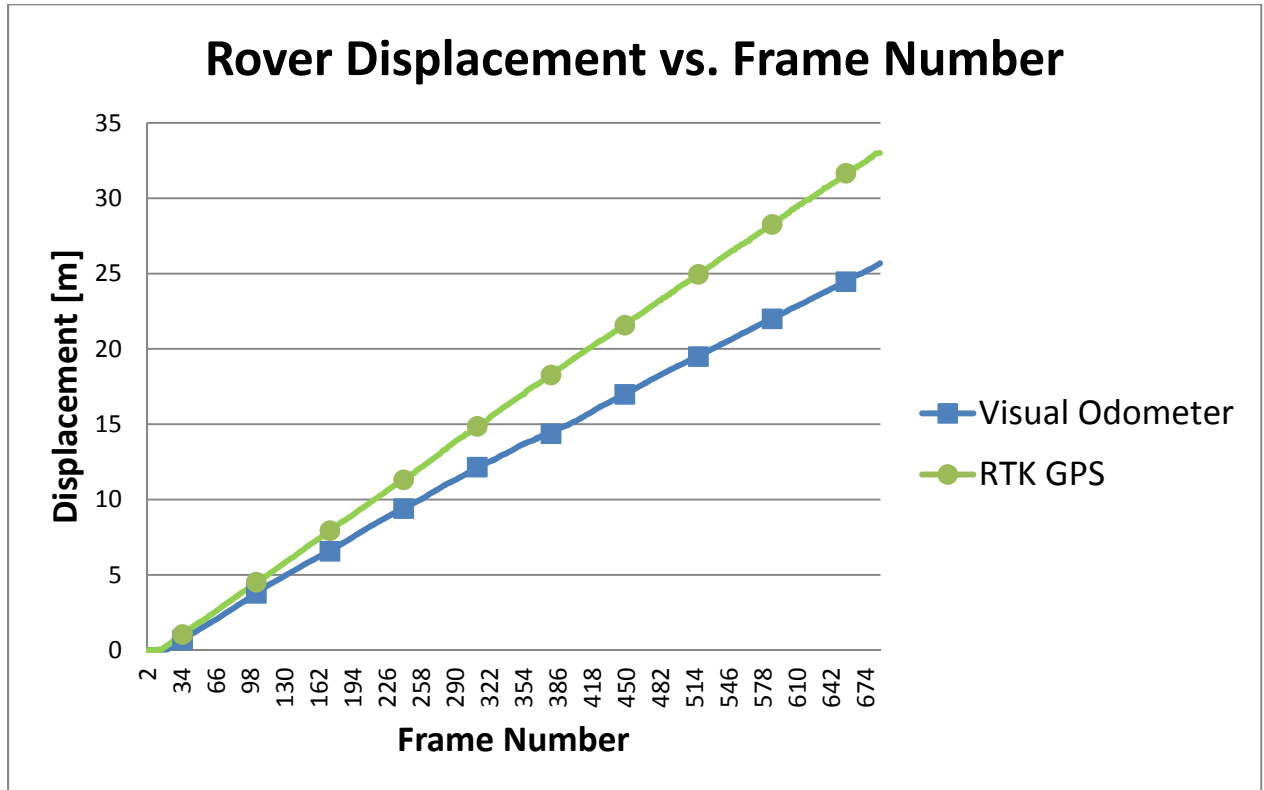


Figure 38 - Traverse 3 displacement error

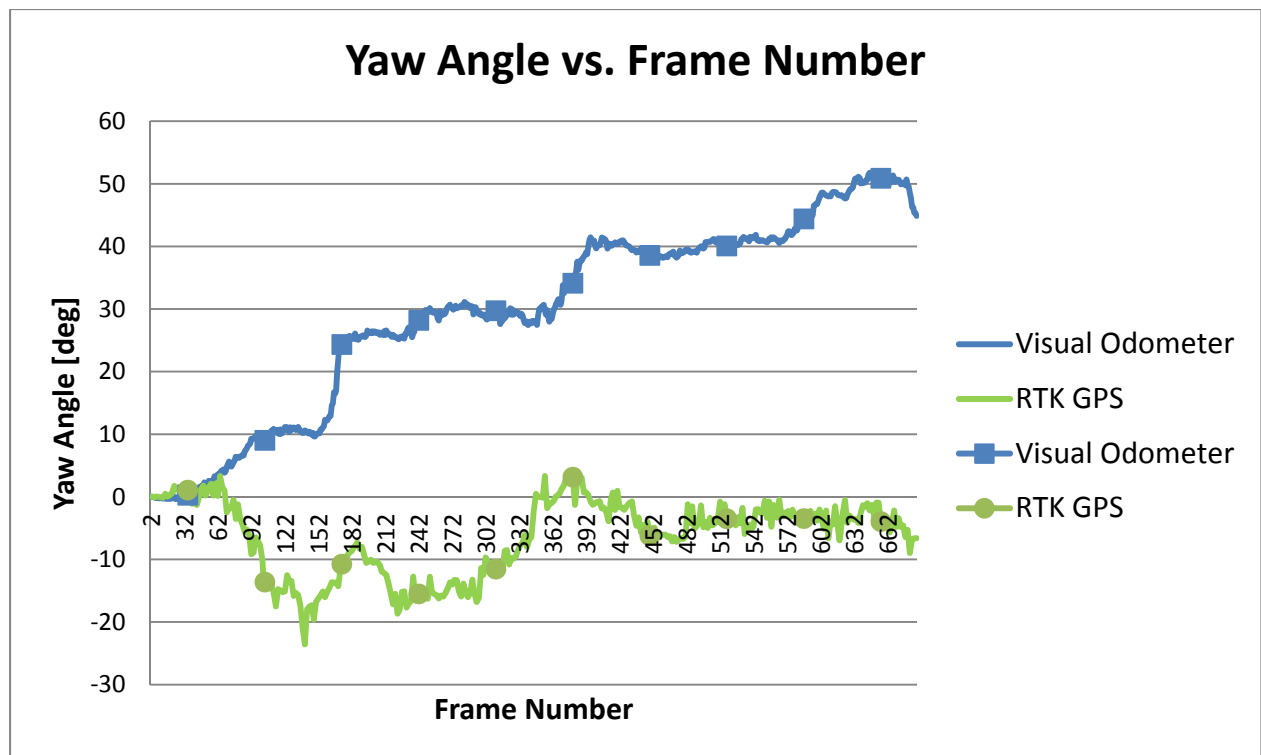


Figure 39 - Traverse 3 yaw angle error

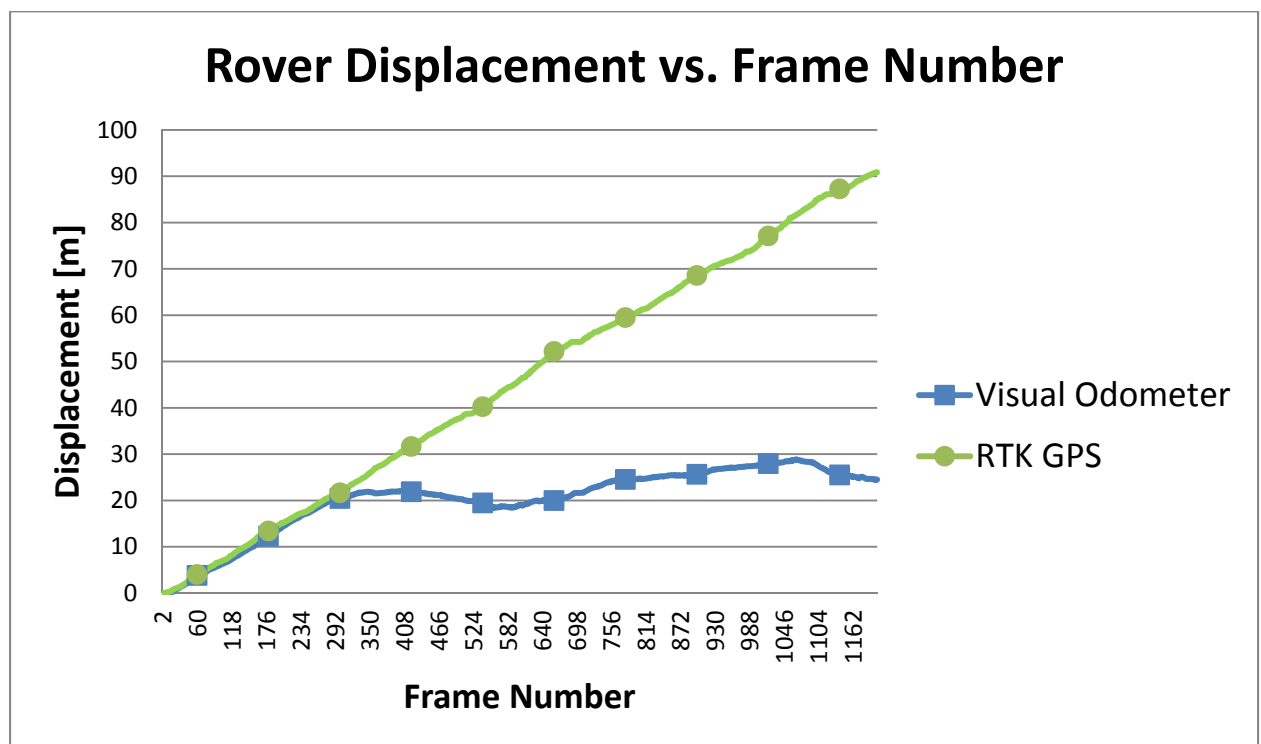


Figure 40 - Traverse 4 displacement error



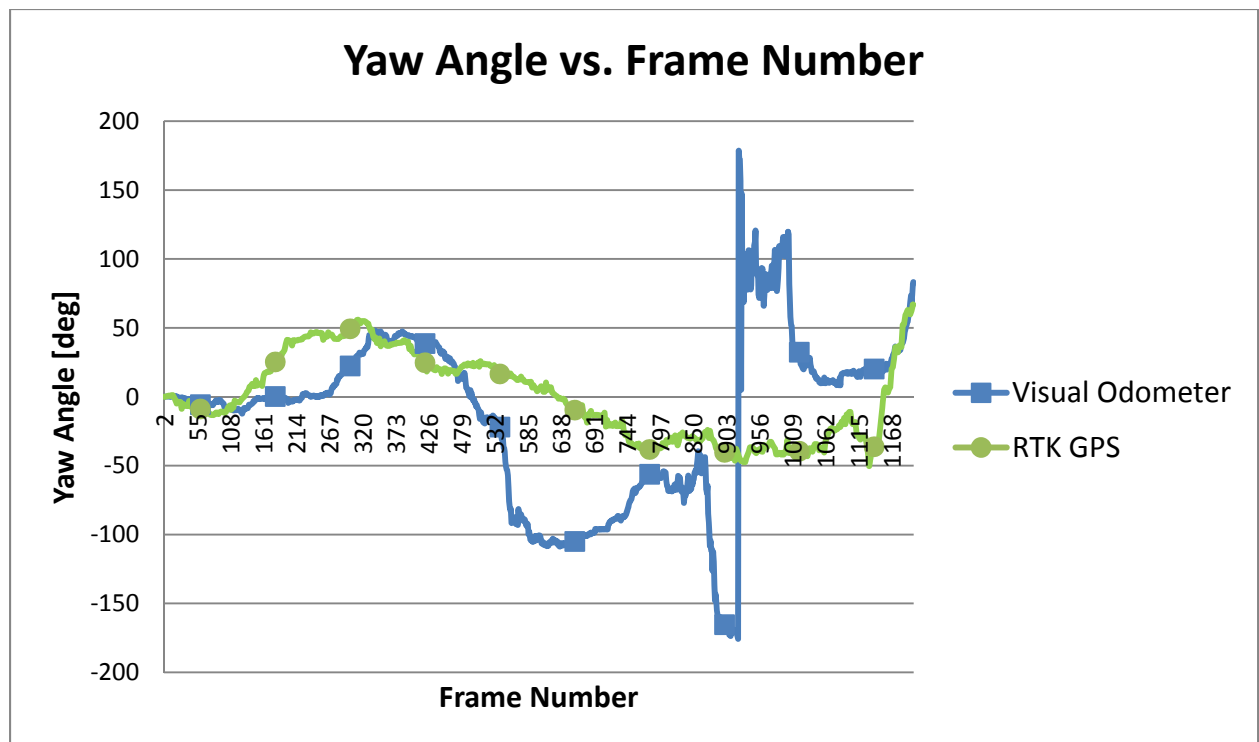


Figure 41 - Traverse 4 yaw angle error

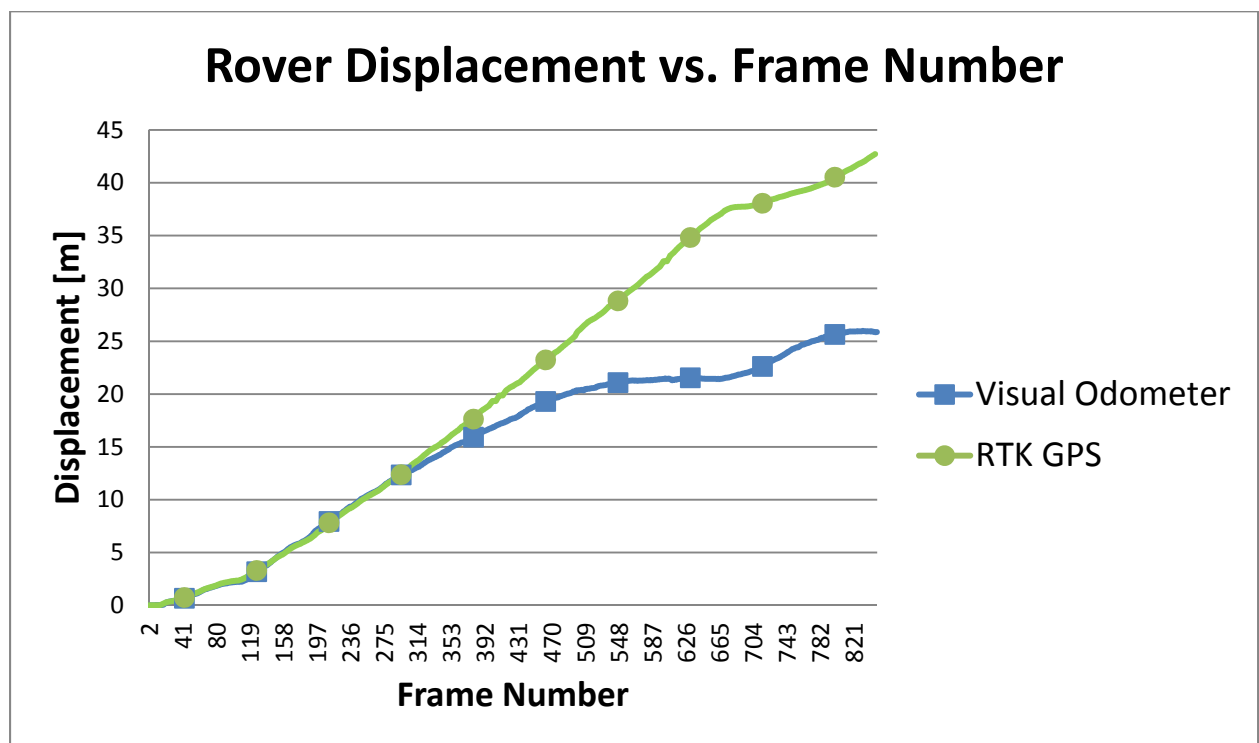


Figure 42 - Traverse 5 displacement error

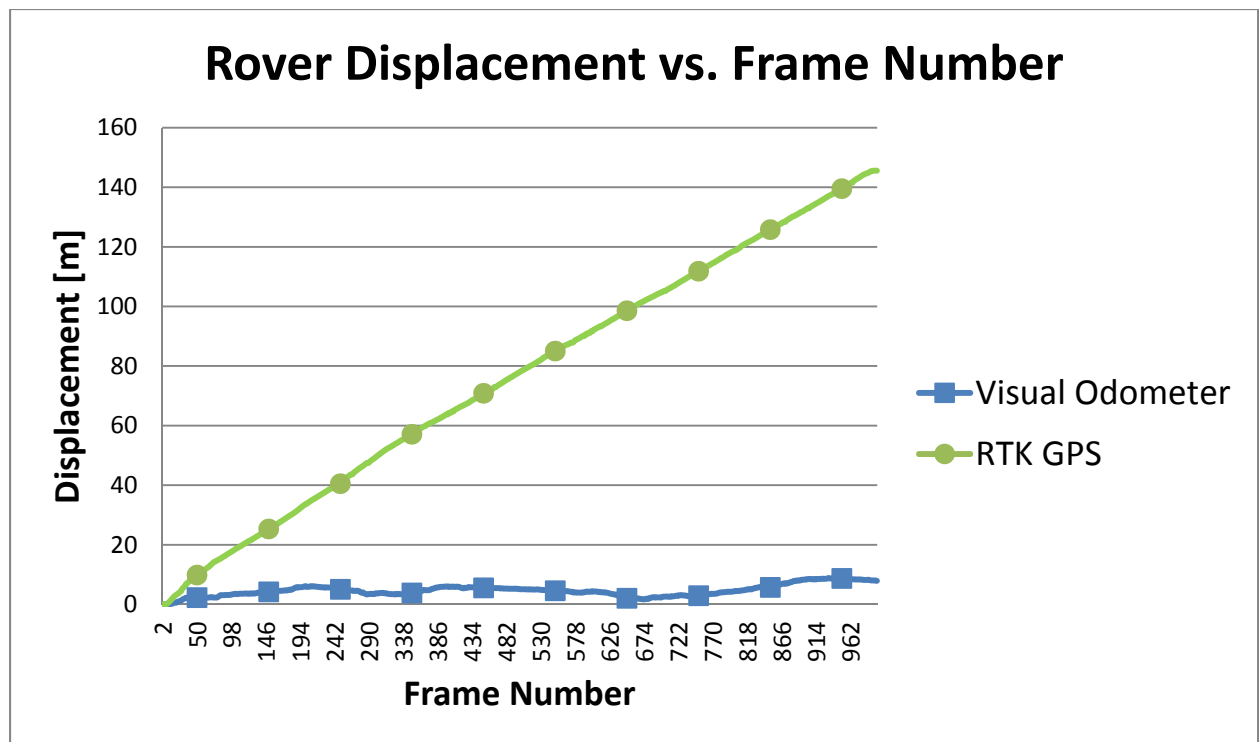


Figure 43 – Traverse 6 displacement error

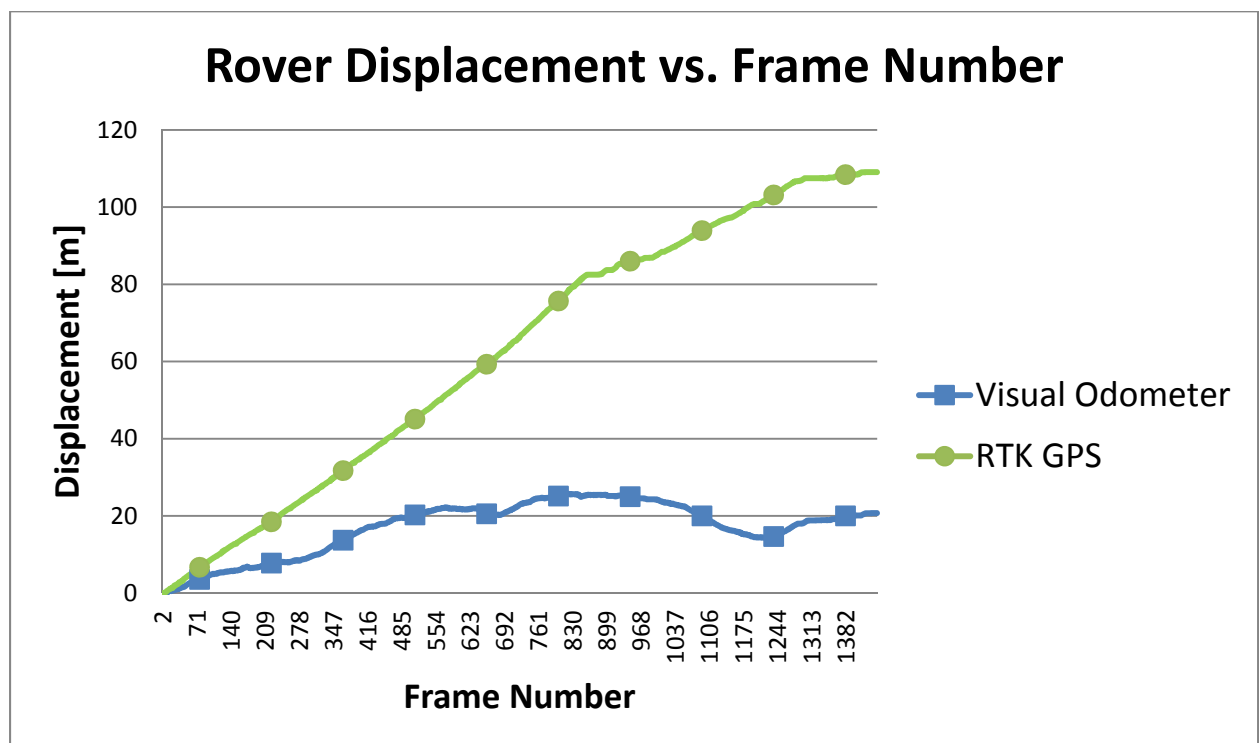


Figure 44 - Traverse 7 displacement error

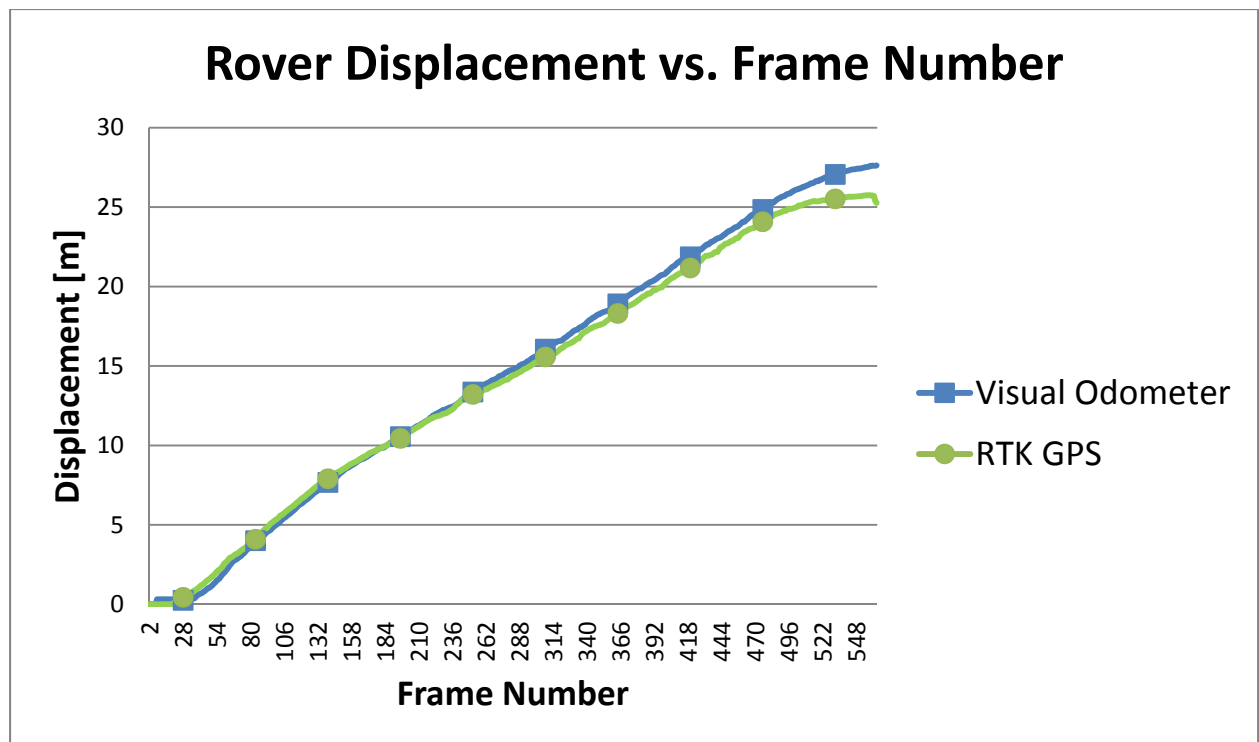


Figure 45 - Traverse 8 displacement error

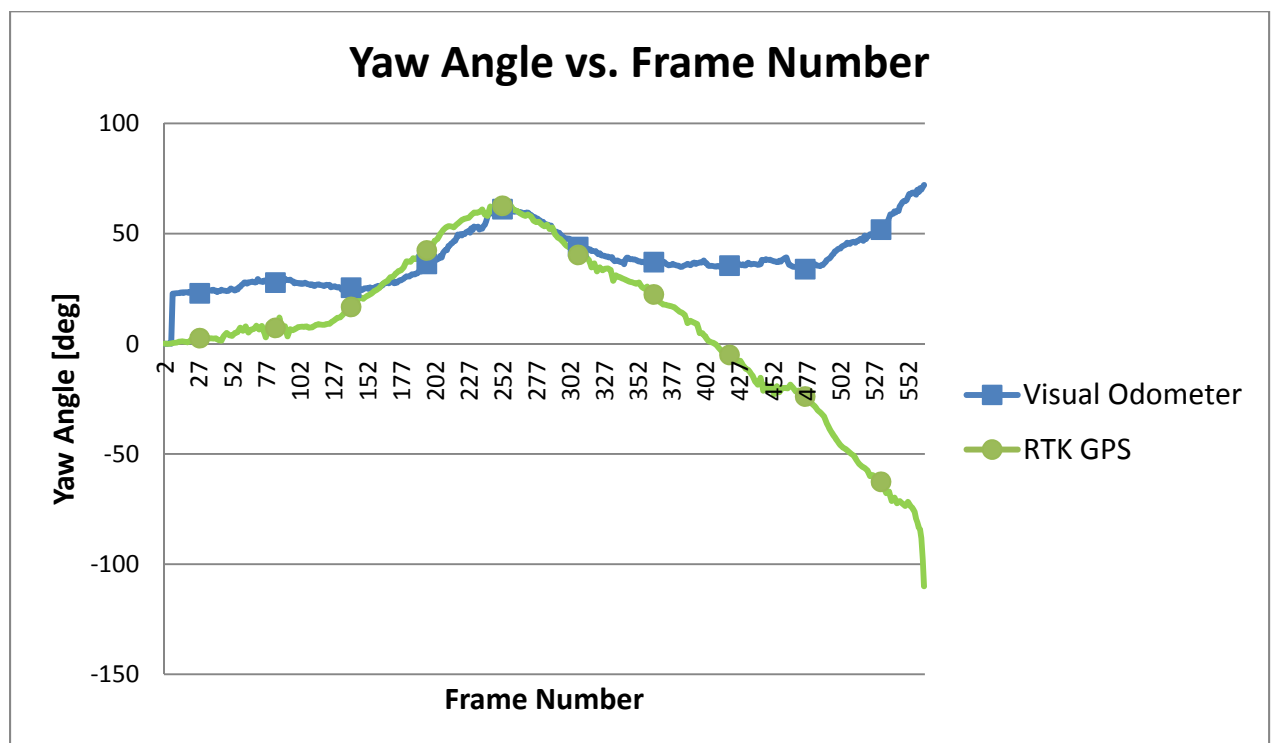


Figure 46 - Traverse 8 yaw angle error



## Appendix B: Camera Specifications

### Bumblebee® Specifications

Specification	BB2-03S2	BB2-08S1	BBX3
Image Sensor Type	Sony® 1/3" progressive scan CCD		
	ICX424 (648x488 max pixels) 7.4µm square pixels	ICX204 (1032x776 max pixels) 4.65µm square pixels	ICX445 (1280x960 max pixels) 3.75µm square pixels
Baseline	12 cm		12 cm and 24 cm
Focal Lengths	2.5mm with 97° HFOV (BB2 only) or 3.8mm with 66° HFOV or 6mm with 43° HFOV		
A/D Converter	12-bit analog-to-digital converter		
White Balance	Automatic / Manual (Color model)		Manual (Color model)
Frame Rates	48 FPS	20 FPS	16 FPS
Interfaces	6-pin IEEE-1394a for camera control and video data transmission 4 general-purpose digital input/output (GPIO) pins		2 x 9-pin IEEE-1394b for camera control and video data transmit 4 general-purpose digital input/output (GPIO) pins
Voltage Requirements	8-30V via IEEE-1394 interface or GPIO connector		
Power Consumption	2.5W at 12V		4W at 12V
Gain	Automatic/Manual		
Shutter	Automatic/Manual, 0.01ms to 66.63ms at 15 FPS		
Trigger Modes	DCAM v1.31 Trigger Modes 0, 1, 3, and 14		DCAM v1.31 Trigger Modes 0, 1, 3, and 14
Signal To Noise Ratio	60dB		54dB
Dimensions	157 x 36 x 47.4mm		277 x 37 x 41.8mm
Mass	342 grams		505 grams
Camera Specification	IIDC 1394-based Digital Camera Specification v1.31		
Lens mount	2 x M12 microlens mount		3 x M12 microlens mount
Emissions Compliance	Complies with CE rules and Part 15 Class A of FCC Rules		
Operating Temperature	Commercial grade electronics rated from 0° to 45°C		
Storage Temperature	-30° to 60°C		

**Figure 47 - Bumblebee® camera specifications [65]**

## Bumblebee2

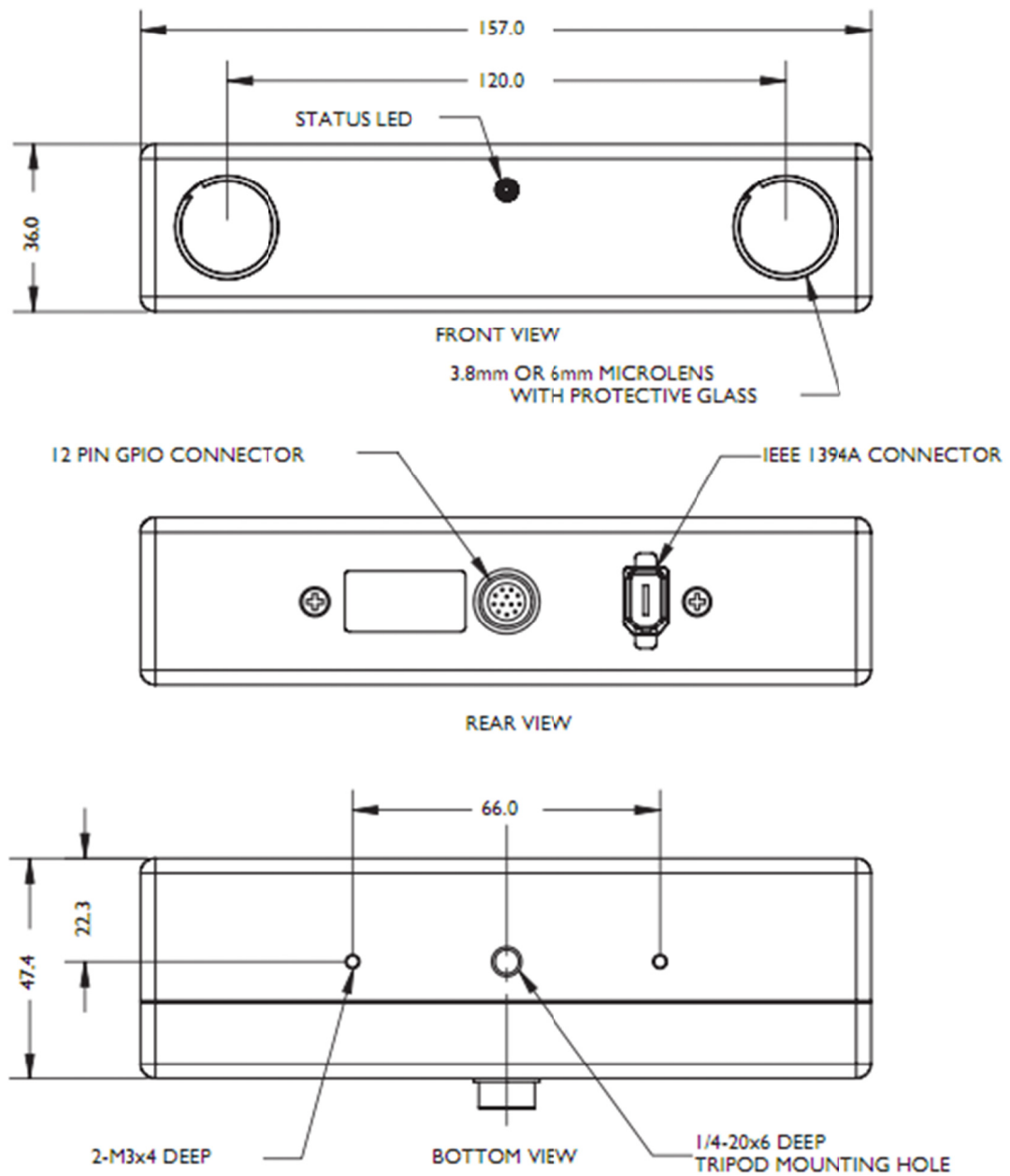


Figure 48 - Bumblebee® camera dimensional drawings [65]

## References

- [1] Active Media Robotics LLC, Pioneer 2/PeopleBot Operations Manual, Version 9, Peterborough, USA, 2001.
- [2] T. Bailey, "Mobile Robot Localisation and Mapping in Extensive Outdoor Environments" Thesis, University of Sydney, Sydney, Australia, Aug. 2002.
- [3] T. Barfoot, "Online Visual Motion Estimation using FastSLAM with SIFT Features" In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.
- [4] T. Barfoot, S. Se, and P. Jasiobedzki, "Vision-based Localization and Terrain Modelling for Planetary Rovers" In *Chapter in Intelligence for Space Robotics*, A. Howard and E. Tunstel, eds., TSI Press, Albuquerque, USA, 2006.
- [5] B. Barshan and H.F. Durrant-Whyte, "Inertial Navigation Systems for Mobile Robots" In *IEEE Trans. On Robotics and Automation*, vol. 11, no. 3, pp. 328-342, June 1995.
- [6] H. Bay, B. Fasel and L.V. Gool, "Interactive Museum Guide: Fast and Robust Recognition of Museum Objects" In *Proc. Of the First International Workshop on Mobile Vision*, May 2006.
- [7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features" In *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.
- [8] J. L. Bentley, "Multidimensional binary search trees used for associative searching" In *Communications of the ACM*, vol. 18, number 9, pp. 509-217, 1975.

- [9] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes" In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [10] V. Bevilacqua, G. Mastronardi, F. Menolascina and D. Nitti, "Stereo-Matching Techniques Optimisation Using Evolutionary Algorithms" In *Intelligent Computing*, pp. 612-621, 2006.
- [11] J.F. Canny, "Finding Edges and Lines in Images" MIT Artificial Intelligence Laboratory, report: 720, pp. 1-149, May, 1983.
- [12] Y. Cheng, M. Maimone and L. Matthies, "Visual Odometry on the Mars Exploration Rovers" In *IEEE Robotics & Automation Magazine*, vol. 13, issue 2, pp. 54-62, 2006.
- [13] P. Corke, D. Strelow, and S. Singh, "Omnidirectional visual odometry for a planetary rover" In *Proc. IROS 2004: IEEE/RSJ Inter. Conf. Intelligent Robots and Systems*, Japan, 2004.
- [14] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet, "A generic framework for topological navigation of urban vehicle" In *Proc. ICRA09 Workshop on Safe Navigation in Open and Dynamic Environments Application to Autonomous Vehicles*, Kobe, Japan, May, 2009.
- [15] A. Cumani and A. Guiducci, "Fast Stereo-based Visual Odometry for Rover Navigation" In *WSEAS Transactions on Circuits and Systems*, vol. 7, issue 7, July, 2008.
- [16] A. Diosi and L. Kleeman, "Laser Scan Matching in Polar Coordinates with Application to SLAM" In *Proc. of IEEE International Conference on Intelligent Robots and Systems*, pp. 3317-3322, Aug. 2005.



- [17] A. Diosi and L. Kleeman, "Scan Matching in Polar Coordinates with Application to SLAM" Monash University, report: MECSE-29-2005, pp. 1-53, July, 2005.
- [18] M. G. Dissanayake, P. Newman, S. Clark, and H. F. Durrant-Whyte, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem" In *IEEE Transactions on Robotics and Automation*, vol 17, issue 3, pp. 229–241, 2001.
- [19] C. Dornhege and A Kleiner, "Visual Odometry for Tracked Vehicles" In *Proc. of IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR 2006)*, Gaithersburg, USA, 2006.
- [20] E. Dupuis, J.C. Piedboeuf, and E. Martin, "Canadian Activities in Intelligent Robotic Systems: An Overview", 2007.
- [21] A. Elfes, "A Sonar-Based Mapping and Navigation System" In *Proc. of IEEE International Conference on Robotics and Automation*, Feb. 1986.
- [22] C. Engels, H. Stewénus and D. Nistér "Bundle Adjustment Rules" In *Photogrammetric Computer Vision (PCV'06)*, 2006.
- [23] A. Farjadpour, D. Roundy, A. Rodriguez, M. Ibanescu, P. Bermel, J.D. Joannopoulos and S.G. Johnson "Improving accuracy by subpixel smoothing in the finite-difference time domain" In *Optics Letters*, vol. 31, no. 20, pp. 2972-2974, Oct. 2006.
- [24] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography" In *Proc. Image Understanding Workshop*, pp. 71-88, Apr. 1980.
- [25] P. Furgale, T. D. Barfoot, and N. Ghafoor "Rover-Based Surface and Subsurface Modeling for Planetary Exploration" In *Proc. Field and Service Robotics (FSR)*, Cambridge, USA, 14–16 July, 2009.

- [26] P. Furgale and C.H. Tong, “*Speeded Up Speeded Up Robust Features*”  
*ONLINE* [<http://asrl.utias.utoronto.ca/code/gpusurf/>], Updated: Apr. 30, 2010.
- [27] A. Fusiello, E. Trucco and A. Verri “A Compact Algorithm for Rectification of Stereo Pairs” In *Machine Vision and Applications*, vol. 12, pp. 16-22, 2000.
- [28] P. Goel, S.I. Roumeliotis and G.S. Sukhatme, “Robot Localization Using Relative and Absolute Position Estimates” In *Proc. of IEEE International Conference on Intelligent Robots and Systems*, Kyongju, South Korea, pp. 1134-1140, Oct. 1999.
- [29] J. Guivant and E. Nebot, “Optimization of the Simultaneous Localization and Map Building Algorithm for Real Time Implementation” In *IEEE Trans. Of Robotics and Automation*, May, 2001.
- [30] C. Harris and M. Stephens “A Combined Corner and Edge Detector” In *Fourth Alvey Vision Conference*, Manchester, UK, pp.147-151, 1988.
- [31] R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision” Cambridge: Cambridge University Press, 2003.
- [32] D. Helmick, Y. Cheng, S.I. Roumeliotis, D. Clouse, and L. Matthies, “*Path Following using Visual Odometry for a Mars Rover in High-Slip Environments*” In *IEEE Aerospace Conference*, Big Sky, USA, 2004.
- [33] D. Helmick, S.I. Roumeliotis, Y. Cheng, D. Clouse, M. Bajracharya, and L. Matthies, “Slip compensation for a Mars Rover” In *Proc. IEEE Inter. Conf. Intelligent Robots and Systems*, Edmonton, Canada, pp. 1419–1426, Aug. 2005.
- [34] B.K.P. Horn, “Closed-form solution of absolute orientation using unit quaternions” In *Jour. Of the Optical Society of America A*, vol. 4, pp. 629-642, 1987.

- [35] A. Howard, "Real-Time Stereo Visual Odometry for Autonomous Ground Vehicles" In *Proc. IROS 2008: Inter. Conf. Intelligent Robots and Systems*, Nice, France, Sept. 2008.
- [36] S.J. Julier and J.K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems" In *Proc. SPIE*, vol. 3068, pp. 182-193, 1997.
- [37] Y. Ke and R. Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors" In *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 511-517, Washington, USA, 2004.
- [38] B. Kuipers and Y. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations" In *Robotics and Autonomous Systems*, vol. 8, pp. 47-63, 1981.
- [39] J.J. Leonard and H.J.S. Feder, "A Computationally Efficient Method for Large-Scale Concurrent Mapping and Localization" In *Robotics Research: the Ninth International Symposium*, London, UK, 2000.
- [40] D. G. Lowe, "Object recognition from local scale-invariant features" In *Proc. Inter. Conf. Computer Vision*, pp. 1150–1157, Corfu, Greece, Sept. 1999.
- [41] D. Lowe, "Local Feature View Clustering for 3D Object Recognition" In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1682-1688, 2001.
- [42] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints" In *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [43] D. Lowe, "Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image" United States Patent 6,711,293, 2004.

- [44] D.G. Lowe, "Three-Dimensional Object Recognition from Single Two-Dimensional Images" In *Artificial Intelligence*, vol. 31, no. 3, pp. 355-395, Mar. 1987.
- [45] F. Lu and E. Miliot, "Robot pose estimation in unknown environments by matching 2d range scans" In *Journal of Intelligent and Robotic Systems*, 1998.
- [46] B.D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision" In *Proc. Of Imaging Understanding Workshop*, pp. 121-130, 1981.
- [47] M. Maimone, Y. Cheng, and L. Matthies, "Two years of Visual Odometry on the Mars Exploration Rovers" In *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.
- [48] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey and K. Konolige, "The Office Marathon: Robust Navigation in an Indoor Office Environment" In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010.
- [49] J. Matas, O. Chum, M. Urban and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions" In *British Machine Vision Conference*, Cardiff, Wales, pp. 384-393, 2002.
- [50] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors" *PAMI*, 2004.
- [51] A. Milella, B. Nardelli, D. Di Paola, and G. Ciciirelli, "Robust Feature Detection and Matching for Vehicle Localization in Uncharted Environments" At *IROS 2009 3rd Workshop: Planning, Perception and Navigation for Intelligent Vehicles – Inter. Conf. Intelligent Robots and Systems*, St. Louis, USA, Oct. 2009.
- [52] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem" In

- Proc. Of the National Conference on Artificial Intelligence*, no. 18, pp. 593-598, 2002.
- [53] M. Montemerlo and S. Thrun, "Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM" In *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1985-1991, 2003.
- [54] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, "FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges" In *Proc. of International Joint Conference on Artificial Intelligence*, vol. 18, pp. 1151-1156, 2003.
- [55] H. Moravec, "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover" Carnegie-Mellon University: Robotics Institute, report: CMU-RI-TR-3, Sept. 1980.
- [56] L.P. Morency, A. Rahimi, and T. Darrell, "Fast 3D Model Acquisition from Stereo Images" In *1<sup>st</sup> International Symposium on 3D Data Processing Visualization and Transmission (3DPVT 2002)*, pp. 19-21, 2002.
- [57] A.C. Murillo, J.J. Guerrero, and C. Sagüés, "SURF features for efficient robot localization with omnidirectional images" In *Proc. IEEE Inter. Conf. Robotics and Automation*, pp. 3901–3909, Rome, Italy, Apr. 2007.
- [58] D. Murray and C. Jennings, "Stereo vision based mapping and navigation for mobile robots" In *Proc. of IEEE International Conference on Robotics and Automation*, no. 2, pp. 1694-1699, 1997.
- [59] D. Murray and J. Little, "Using real-time stereo vision for mobile robot navigation" In *Proc. of the IEEE Workshop on Perception for Mobile Agents*, Santa Barbara, USA, June 1998.

- [60] P.M. Newman, "On the Structure and Solution of the Simultaneous Localisation and Map Building Problem" Doctoral thesis, University of Sydney, Sydney, Australia, Mar. 1999.
- [61] D. Nistér, O. Naroditsky, and J. Bergen, "Visual Odometry for Ground Vehicle Applications" In *Journal of Field Robotics*, vol. 23, no. 1, pp. 3– 20, 2006.
- [62] NVIDIA, "What is CUDA?", *ONLINE* [http://www.nvidia.com/object/what\\_is\\_cuda\\_new.html](http://www.nvidia.com/object/what_is_cuda_new.html). Updated: 2011.
- [63] C.F. Olson, L.H. Matthies, M. Schoppers and M.W. Maimone, "Rover navigation using stereo ego-motion" In *Robotics and Autonomous Systems*, vol. 43, no. 4, pp. 215-229, 2003.
- [64] Panasonic, Toughbook CF-30FTSAFxx Service Manual, Order No. CPD0711217AE, Matsushita Electric Industrial Co., Ltd., 2007.
- [65] Point Grey, "Bumblebee Stereo Vision Camera Systems", *ONLINE* [http://www.ptgrey.com/products/bumblebee2/bumblebee2\\_xb3\\_datasheet.pdf](http://www.ptgrey.com/products/bumblebee2/bumblebee2_xb3_datasheet.pdf) Updated: Feb. 2009.
- [66] D. Scaramuzza and R. Siegwart, "Appearance-Guided Monocular Omnidirectional Visual Odometry for Outdoor Ground Vehicles" In *IEEE Transactions on Robotics*, pp. 1–12, Oct. 2008.
- [67] S. Se, D. Lowe, and J. Little, "Local and Global Localization for Mobile Robots using Visual Landmarks" In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '01)*, Hawaii, USA, 2001.
- [68] S. Se, D. Lowe, and J. Little, "Vision-based Mobile Robot Localization and Mapping using Scale-Invariant Features" In *Proc. of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 2051-2058, 2001.

- [69] S. Se, D. Lowe, and J. Little, "Global Localization using Distinctive Visual Features" In *Proc. of International Conference on Intelligent Robots and Systems, IROS 2002*, Lausanne, Switzerland, pp. 226-231, 2002.
- [70] S. Se, D. Lowe, and J. Little, "Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks" In *International Journal of Robotics Research*, vol. 21, no. 8, pp. 735-738, 2002.
- [71] S. Se, D. Lowe, and J. Little, "Vision-based Global Localization and Mapping for Mobile Robots" In *IEEE Transactions on Robotics*, vol. 21, issue 3, pp. 364–375, June 2005.
- [72] S. Se, T. Barfoot and P. Jasiobedzki, "Visual Motion Estimation and Terrain Modeling for Planetary Rovers" In *Proc. of ISAIRAS 2005 Conference*, Munich, Germany, Sept. 2005.
- [73] R.C. Smith and P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty" In *The International Journal of Robotics Research*, vol. 5, no. 4, 1986.
- [74] N. Sünderhauf, K. Konolige, S. Lacroix, and P. Protzel, "Visual Odometry Using Sparse Bundle Adjustment on an Autonomous Outdoor Vehicle" P. Levi, M. Schanz, R. Lafrenz, and V. Avrutin, eds., In *Tagungsband Autonome Mobile Systeme 2005*, pp. 157–163, 2005.
- [75] J.P. Tardif, Y. Pavilidis, and K. Danniilidis, "Monocular Visual Odometry in Urban Environments Using an Omnidirectional Camera" In *Proc. IROS 2008: IEEE Inter. Conf. Intelligent Robots and Systems*, Nice, France, Sept. 2008.
- [76] J.D. Tardós and J. Neira, "Robust Mapping and Localization in Indoor Environments using Sonar Data" In *The International Journal of Robotics Research*, June 2002.

- [77] S. Thrun, "Bayesian Landmark Learning for Mobile Robot Localization" In *Machine Learning*, vol. 33, pp. 41-76, 1998.
- [78] C. Valgren and A. J. Lilienthal, "SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments" In *Robotics and Autonomous Systems*, vol. 58, issue 2, pp. 149–156, Feb. 2010.
- [79] L. Vincent and P. Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations" In *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 13 no. 6, pp. 583-598, June 1991.
- [80] S.B. Williams, G. Dissanayake and H. Durrant-Whyte, "An Efficient Approach to the Simultaneous Localisation and Mapping Problem", 2006.
- [81] G. Xu and Z. Zhang, "Epipolar Geometry in Stereo, Motion and Object Recognition" Norwell, USA: Kluwer Academic Publishers, 1996.
- [82] Z. Zhang, R. Deriche, O. Faugera and Q.T. Luong, "A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry" In *Artificial Intelligence*, vol. 78, pp. 87-119, 1995.
- [83] S. Zhang, L. Xie and M. Adams, "An Efficient Data Association Approach to Simultaneous Localization and Map Building" In *The International Journal of Robotics Research*, vol. 24, no. 1, pp. 49-60, Jan. 2005.