Estimating Epipolar Geometry With The Use of a Camera Mounted Orientation Sensor

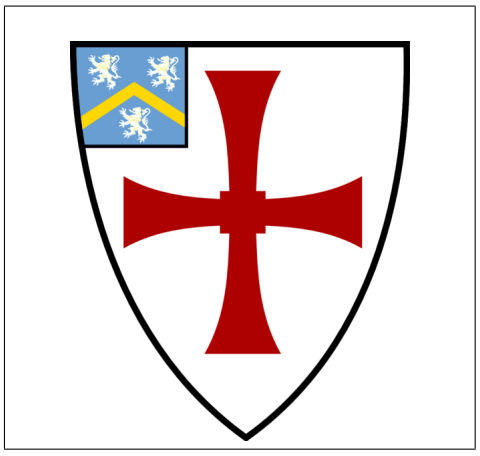
BARBER, ALASTAIR,EDWARD

MSc. Thesis

Estimating Epipolar Geometry With The Use of a Camera Mounted Orientation Sensor

Alastair Barber BSc. (Hons) Dunelm

2012



University of Durham

School of Engineering and Computer Sciences

Abstract

Context: Image processing and computer vision are rapidly becoming more and more commonplace, and the amount of information about a scene, such as 3D geometry, that can be obtained from an image, or multiple images of the scene is steadily increasing due to increasing resolutions and availability of imaging sensors, and an active research community. In parallel, advances in hardware design and manufacturing are allowing for devices such as gyroscopes, accelerometers and magnetometers and GPS receivers to be included alongside imaging devices at a consumer level.

Aims: This work aims to investigate the use of orientation sensors in the field of computer vision as sources of data to aid with image processing and the determination of a scene's geometry, in particular, the *epipolar geometry* of a pair of images - and devises a hybrid methodology from two sets of previous works in order to exploit the information available from orientation sensors alongside data gathered from image processing techniques.

Method: A readily available consumer-level orientation sensor was used alongside a digital camera to capture images of a set of scenes and record the orientation of the camera. The *fundamental matrix* of these pairs of images was calculated using a variety of techniques - both incorporating data from the orientation sensor and excluding its use.

Results: Some methodologies could not produce an acceptable result for the Fundamental Matrix on certain image pairs, however, a method described in the literature that used an orientation sensor always produced a result - however in cases where the hybrid or purely computer vision methods also produced a result - this was found to be the least accurate.

Conclusion: Results from this work show that the use of an orientation sensor to capture information alongside an imaging device can be used to improve both the accuracy and reliability of calculations of the scene's geometry - however noise from the orientation sensor can limit this accuracy and further research would be needed to determine the magnitude of this problem and methods of mitigation.

Declaration of Authorship

I, Alastair Edward Barber, declare that this thesis entitled "Estimating Epipolar Geometry With The Use of a Camera Mounted Orientation Sensor" and the work presented in it are my own. I confirm that no part of the material provided has previously been submitted by the author for a higher degree at Durham University or any other University. All the work presented here is the sole work of the author. The copyright of this thesis rests with the author. No quotation from it should be published without the author's prior written consent and information derived from it should be acknowledged

List of Figures

Contents

Acknowledgments

I would like to acknowledge and thank my academic supervisor, Dr. Ioannis Ivrissimtzis for his hard work, dedication, support and encouragement throughout this project. His enthusiasm for the subject has been an inspiration and I have learned a great deal over my time working with him.

I would also like to thank my family, and colleagues and friends at the School of Engineering and Computer Sciences and also at St. Chad's College for their support and encouragement throughout my time in Durham.

This work was funded by a one-year scholarship from the School of Engineering and Computer Sciences, Durham University.

1 Introduction

1.1 Background

Advances in digital photography have vastly increased the amount of data that can be captured by a camera. It is now common to find consumer level devices that offer sensor resolutions of upwards of 10 megapixels and at ever decreasing costs. Similarly, the increased upward trend in computer processing power has allowed for the data captured by digital cameras to be analysed quickly and cheaply - and as such image processing techniques that are able to infer a great deal of information from an image, or set of images, are commonplace across a large number of industries and applications. Naturally, this is currently an active and broad area of research. In particular, one area of considerable focus is the use of digital image processing in order to infer the geometry of a particular scene, the most important aspect of which is the ability to determine 3 dimensional depth from one or more 2 dimensional images. However, until recently - much of this work has been focussed purely on using image data captured from cameras in a process known as image registration, or captured by multiple cameras that have previously and precisely been calibrated to form a 'rig'. Due to the complexities and cost of the latter, it is not particularly suited to consumer applications or portable photography. The former is more widespread and makes use of a variety of image processing algorithms and statistical methods, however, is computationally expensive and is prone to error. It is however, now becoming common to find devices, in particular 'Smartphones' that contain a camera alongside multiple other sensors, such as a gyroscope, accelerometer and magnetometer in order to determine the orientation of the device, alongside a GPS receiver to determine the location of the device on a global scale. The focus of this work is on using data acquired from such sensors to enhance traditional methods of determining the scene geometry from a set of 2 images of a scene. In particular, we focus on methods for solving the *Fundamental*

Matrix of a particular set of images. This matrix, described in detail in following chapters, along with other information relating to the camera - is able to encode all the information necessary to re-construct a scene in 3 dimensions, although such a reconstruction is outside of the scope of this work.

1.2 Image Geometry and Registration

In order to achieve the aim of determining scene geometry, the vast majority of algorithms currently in use require an input of a set of matched points over a two or more images of the same scene. The process of determining matching points in a pair of images is know as Image Registration. This can be performed by hand, or, more commonly, using automated methods, typically depending on corner detection algorithms to identify potential objects in a scene, followed by statistical methods to determine possible matches between the candidate points. In itself, this field is subject to a great amount of research, a comprehensive survey of which is presented in [58], and a detailed description of a subset of methods for performing this process is given in later chapters.

1.3 MEMS Orientation Sensors

In parallel to the improvements in camera sensor technology, developments in the field of Micro Electro-Mechanical Systems (MEMS) have seen the inclusion of such sensors as Gyroscopes, Accelerometers and Magnetometers in many consumer level devices. Devices such as the Apple iPhone contain gyroscopes and accelerometers for the purposes of determining the orientation and motion of the device, alongside Global Positioning System receivers in order to determine the device's location relative to the earth's surface and combine these with high-resolution cameras. It is already common for images captured with such devices to be 'Geo-Tagged' with location and orientation (portrait or landscape) information in the form of meta-data encoded alongside the image [23]. Inertial measurement units comprised of Gyroscopes, Accelerometers and Magnetometers have also found applications in the field of computer vision and image processing when combined with a digital camera. Such examples include Image Stabilization, as described in [18], and also for use

in the correction of 'Rolling Shutter' [25] distortion from moving a mobile phone camera whilst filming a scene.

This work will be focussed on how it is possible to exploit the data provided by a combination of MEMS Gyroscopes, Accelerometers and Magnetometers (together forming and inertial measurement unit) in order to improve the process of registering a pair of images of a scene and determining the Fundamental Matrix, alongside traditional image registration techniques.

1.4 Project Summary

In summary, this work aims to compare three different methods for computing the Fundamental Matrix of a pair of corresponding images of a scene, taken at different times and with the camera having undergone an unconstrained rotation and translation of a value that is not known beforehand. The first method uses a combination of existing feature extraction and feature matching methods and uses an existing method of estimation of the Fundamental Matrix using RANSAC. The remaining methods will use a 3D Orientation Sensor mounted on the camera to provide data to be used during the calculation; The second method uses a method proposed in [41] by Okatani and Deguchi, whilst the final method is a novel method introduced in this work which is based upon the Okatani and Deguchi method and uses orientation data in the point matching stage, and RANSAC to estimate the fundamental matrix using all candidate point matches found by matching points using the data from the inertial measurement unit. The motivation behind this is to investigate whether this algorithm would be more robust than the original Harris-Normalised Cross Correlation-RANSAC based algorithm in identifying potential point correspondences, which according to Okatani and Deguchi it should be due to information obtained from the orientation sensor, but also aims to ensure that error introduced by the orientation sensor is not included in the final calculation for the Fundamental matrix, and hence achieve a higher accuracy than was obtained using the original Okatani and Deguchi method.

This work also aims to investigate the use of MEMs based orientation sensors within computer vision applications in general and hence aims to research the accuracy and error of such devices. This work aims to answer the research question of how such devices might be incorporated into both cameras, and image processing and

computer vision applications, as well as their effectiveness and current weaknesses and hardware limitations.

All of the comparisons are performed on a set of real images, and a discussion is included as to the suitability of these methods and suggestions for sources of error and further research are made. The results obtained suggest that using data from an inertial sensor provides a significantly more robust method for determining the Fundamental Matrix, and by using the Okatani and Deguchi method the amount of point correspondences needed to formulate a good result for the Fundamental Matrix is significantly reduced, as only two correct correspondences are needed. However, they also highlight the fact that, whilst a Fundamental Matrix can be computed in more circumstances - it is likely to be slightly less accurate, especially in the case where rotation information is used directly in the computation of the Fundamental Matrix. It is suggested that a probable cause for this is noise introduced by the inertial sensors themselves.

1.5 Thesis Structure

This thesis is divided into the following chapters:

- **Chapter 1:** Introduction
- **Chapter 2:** The Geometry of a Stereoscopic System. This chapter presents an overview of the theory, mathematical details and the literature surrounding the formation of a 2D image in a camera, and the relationship between two images of the same scene given a set of known point correspondences.
- **Chapter 3:** Orientation Sensors. In this chapter, an introduction to a the theory and literature behind the field of orientation sensors are presented, focussing on the workings of both mechanical and MEMs sensors. Attention is given to the sources and significance of error in these sensors and methodologies for reducing the impact of such error. The chapter concludes with a summary of the uses and limitations of such sensors an how they may best be used to augment the image registration process.
- **Chapter 4:** Related Work. This chapter explores the literature and past work focussing on the problem of image registration and determining scene

geometry, alongside previous works on the use of orientation sensors in this process.

- **Chapter 5:** Proposed Method. This chapter presents an overview of the methods to be implemented, alongside a new method that combines aspects of the previously presented methodologies.
- **Chapter 6:** Implementation. In this chapter, an overview of how the necessary hardware and software systems were developed for testing is presented.
- **Chapter 7:** Results. This chapter presents the results obtained from the tests of the implementation, and a methodology for evaluating their effectiveness. An evaluation of these results is also presented.
- **Chapter 8:** Conclusions and Further Work. In this chapter, conclusions are drawn as to the effectiveness of the methods shown in this work, as well as suggestions for further research areas in this field.

2 The Geometry of a Stereoscopic System

This chapter focuses on how two 2D images of a 3D scene can be related to one another and to the 3D scene. This is important as by determining a relationship between these images and the scene we are able to use the information gathered in order to recover the information of the dimension (depth) lost by imaging the scene in 2D. In this work, we assume that these images have been obtained by the same camera from different viewpoints in such a fashion as described in [58].

In order to determine this relationship, we must describe the geometric parameters of a camera and how 3D scene information is translated to that of a 2D image (presented in Section 2.1). We then describe how the geometry of two 2D images relate to that of the 3D scene (Section 2.2), and finally how the information needed to determine this geometry is gathered and a method of doing so (Section 4.2).

2.1 Camera Geometry

In this section, we describe how a 3D scene is captured by a camera capable of capturing an image in 2D dimensions. Throughout this work, we refer to the camera as a digital camera that forms a digital representation of an image by use of a CCD Sensor.

2.1.1 Perspective Camera Model

A camera creates a 2D representation of a 3D scene by way of imaging a 'projection' of the light emitted or reflected from objects in the scene. Described in this subsection is the method by which this is achieved, and the parameters that govern this representation.

The simplest of cameras can be described as a *pinhole camera.* Such a camera is described by [51] as having a *focal plane* F at a fixed distance f (the *focal length)* in front of an *image plane* I. A pinhole, referred to as the *optical centre* C, is made such that rays of light arriving at the camera from the scene form an inverted image of the scene on the image plane. Thus, for each point on the object, there is a corresponding point made on the image plane I. This manner of projection from 3D scene to the 2D image plane is referred to as *perspective projection.* This arrangement is illustrated in Fig. 2.1.

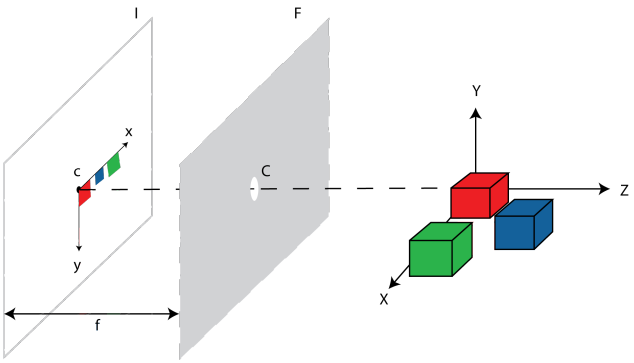


Figure 2.1: An illustration of a pinhole camera and the Perspective Projection of a 3D Scene to a 2D image plane.

Planes F and I are parallel to one another and the point at which a line passing through C, perpendicular to F (known as the *optical axis)* meets I is described as the *principal point* c. It then follows that the coordinate system of I (the *image coordinate system*), (x, y) is defined such that the origin lies at the point c. The 3D coordinate system of the scene being photographed can be referred to as (X, Y, Z), with an origin of C and where the X and Y axes are parallel (but in opposite directions - as the image is inverted) to those of the image coordinate system, and the Z axis coinciding with the camera's optical axis. The relationship between 2D image coordinates and 3D space coordinates can then be represented as:

$$\frac{x}{X} = \frac{y}{Y} = \frac{f}{Z} \tag{2.1}$$

Coordinate Systems In this work, and in particular this Chapter, numerous references are made to points in different coordinate systems. For clarity they are described here along with the corresponding notation:

World Coordinate System: This is the 3D coordinate system of objects in a scene. It is not affected by movement of the camera and the location of a point in the scene in this coordinate system is constant regardless of movement of the camera. The origin of this system is arbitrary and is defined by the extrinsic parameters of a camera (see Section 2.1.2).

Camera Coordinate System: This coordinate system is used to represent 3D points (X, Y, Z) in a scene as viewed from the camera. The origin of this system lies at C, the optical centre of the camera. In situations whereby two cameras are being used, a single point, M in space is represented by the coordinates M_l and M_r, where M_l is the location of the point in the left camera coordinate system, and M_r represents this same point but in the coordinate system of the right camera.

Image Coordinate System: The image coordinate system is the 2D coordinate system of points formed on the image plane I of a camera. The origin of this system lies at the principal point c.

Pixel Coordinate System: The pixel coordinate system is the coordinate system of a digital image captured by the camera. It is similar to the image coordinate system however is produced after application of the intrinsic parameters of the camera have been factored in. (Fig. 2.4). The origin of this coordinate system is usually at the upper left corner of the resulting image.

Whilst the image plane is physically behind the camera, it is standard practice in literature, and hence this work, to represent the projection of the image as a plane *in front* of the plane F, known as a *virtual image plane* as illustrated in Fig. 2.2.

This arrangement makes no difference to the geometry of the camera system, but allows for easier notation and representation of the scene and it's associated geometry.

2.1.2 Extrinsic Camera Parameters

The previous explanation of the perspective projection model assumes that the camera's axes are parallel to those of the scene being photographed, and the origin of the scene coordinates (referred to as the *world coordinates*) lies along the optical

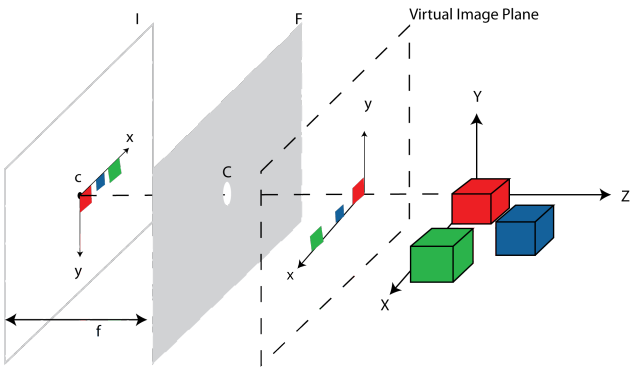


Figure 2.2: Pinhole Camera Model with the Virtual Image Plane

axis. This is acceptable should we be using only a single image of a scene. However for preforming an analysis on multiple images of the same scene that have been taken from different viewpoints, it is necessary to ensure that the scene's origin is constant across all images. In order to achieve this, it is necessary to know the *extrinsic* camera parameters for a set of two images.

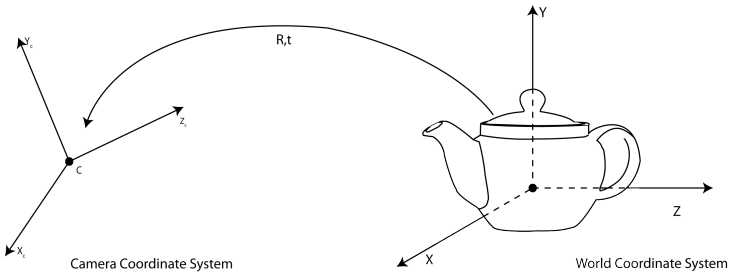


Figure 2.3: The Extrinsic Parameters of a Camera System. The origin of the Camera Coordinate System, C, is mapped to that of the World Coordinate System through Rotation R and Translation t.

These consist of the translation of the camera and the rotation of the relative to the origin of the world coordinate system, as illustrated in Fig. 2.3. As rotation and translation can be expressed simply in linear algebra, it is convenient to rewrite the relationship between 3D space and image coordinates (Equation 2.1) in the following

format, where $x = U/S$, $y = V/S$ and $S \neq 0$. (Equation 2.2).

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.2}$$

Thus, for a 3D point $M = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$ and it's 2D image coordinate $m = \begin{bmatrix} x & y \end{bmatrix}^T$, and

$$P = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

we can represent the relationship between the two as shown in Equation 2.3.

$$s\tilde{m} = P\tilde{M} \tag{2.3}$$

where $s = S \neq 0$, and $\tilde{m}$ and $\tilde{M}$ are m and M which have been homogenized. A homogenous coordinate is defined as for any arbitrary vector $x = \begin{bmatrix} a & b & c & ... \end{bmatrix}^T$, $\tilde{x}$ is the vector augmented with a 1 added to as the final element.

So far, we have been able to represent 3D coordinates in the camera system in relation to 2D image coordinates, however, as mentioned previously, it is more useful to relate 3D world coordinates to 2D image coordinates. This can be accomplished by first relating 3D world coordinates centered at the world origin, to 3D camera coordinates centered at the camera's origin (the optical centre C), then using equation 2.3 to determine the 2D point at which this 3D point lies. We can convert between the world origin O to camera origin C by performing a rotation R followed by a translation t, therefore:

$$M_{camera} = RM_{world} + t \tag{2.4}$$

R and t are thus referred to as the camera's *extrinsic properties.*

2.1.3 Intrinsic Camera Parameters

It is important to consider how various parameters inside the camera affect the mapping of world coordinates to image coordinates. There are several parameters affecting this mapping, the most significant being described in [51] as:

- The origin of the image plane is not known in advance, generally, it will not coincide with the intersection of the optical axis and the image plane. (Typically - the origin of the image plane is defined as the top left corner, whereas the optical axis will usually coincide with the centre of the image plane.)
- The units of the image coordinate axes are not necessarily equal, and are determined by the sampling rate of the devices.
- The two axis of the image plane may not form a right angle.

These parameters are determined by the internal physics of the camera. In a digital camera, the image plane consists of a *sensor* that in turn consists of many *pixels* that measure the colour and intensity of light rays landing on them. These values are then digitally encoded to represent the image being photographed. As mentioned, it is the shape of this sensor and how the images are encoded that will affect the intrinsic parameters, for example, pixels themselves are not square, or the horizontal and vertical axis of the sensor do not meet at right angles - this will affect how the image is formed relative to the camera and hence world coordinates of a scene. Furthermore, distortions introduced by the lens being used will also change how points are matched to the image (although these cannot be modeled and represented in the method described here).

These issues are illustrated in Fig. 2.4.

In this figure, the original image coordinate system (x, y) has its origin at the principal point c, and has the same units on both the x and y axis. The coordinate system (u, v) is the coordinate system by which pixels in the resulting image are addressed. Usually, the origin of these images is located at the upper left corner as opposed to the principal point. Typically, the x axis is assumed to run parallel to the u axis and the units along the u and v axis can be denoted as $k_u$and k_v respectively. As shown in Fig. 2.4 it the v and u axes may not be exactly perpendicular (and therefore the pixels may not be square), as such the angle between the two axis is denoted as θ. The location of the principal point, C, is denoted in v u coordinates as $\begin{bmatrix} u_0 & v_0 \end{bmatrix}^T$. Therefore, for a given point $m_0 = \begin{bmatrix} x & y \end{bmatrix}^T$ in the original x y coordinate system,

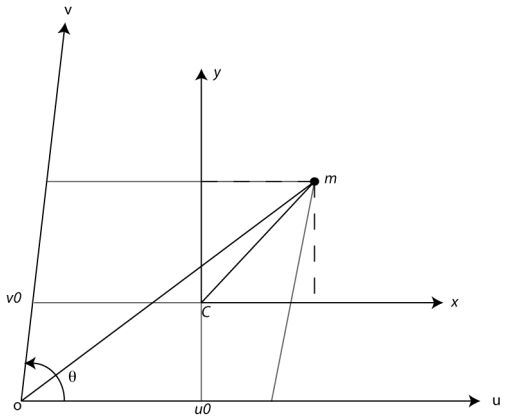


Figure 2.4: Camera Intrinsic Parameters

its location in the new u v coordinate system is written as $m_1 = \begin{bmatrix} u & v \end{bmatrix}^T$. The relationship between these two can then be written as:

$$\tilde{m_1} = H\tilde{m_0}$$

where

$$H = \begin{bmatrix} k_u & k_u \cot\theta & u_0 \\ 0 & k_v / \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

These five parameters $((k_v / \sin\theta), k_u, (k_u \cot\theta), u_0, v_0)$ are referred to as the intrinsic parameters of the camera, and remain the same regardless of camera position and movement relative to the scene being photographed. The process of obtaining these parameters is known as camera calibration - and there are many different methods of performing this calibration, such that it is subject to a significant amount of research. One established method of determining these intrinsic parameters for a wide range of given cameras is described in [54]. In this method, the intrinsic parameters are calculated using a set of photographs of a checkerboard with known properties (square dimensions and quantities) from different angles and translations. This process only needs to be done once for a single camera and lens. Furthermore, it is capable of correcting for other parameters such as lens distortion that are not

considered in this equation.

Equation 2.5 can then be included within equation 2.3 to produce $s\hat{m}_1 = HP\tilde{M}$, and so,

$$P_{new} = HP = \begin{bmatrix} \alpha_u & \alpha_u \cot\theta & u_0 & 0 \\ 0 & \alpha_v / \sin\theta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.6}$$

where $\alpha_u = fk_u$ and $\alpha_v = fk_v$ (where f is the focal length of the camera).

2.2 Epipolar Geometry

So far, we have described how a camera is able to map points from a 3D world coordinate system, to a 2D image coordinate system - taking into account how the camera lies in relation to the scene (the *extrinsic parameters*). In this section, we consider how this relates to a series of two images of the same scene, known as a *stereoscopic* image - where the images are taken by the same camera that has undergone a translation and rotation around a scene. The geometry of how these images relate to one another, and how the 3D coordinates of a scene are represented by two 2D images is known as the *epipolar geometry* of a scene. Specifically, we focus on the situation where this rotation and translation between images are unknown. Presented in this section is a method whereby if the position of objects, or points in one image can be matched with those of the other images, it is possible to deduce the rotation and translation between images, and hence scene geometry from this information.

2.2.1 Epipolar Geometry

The geometry of a stereo view of a scene is referred to as *epipolar geometry* and is illustrated in Fig. 2.5. Two projection centres C_l and C_r represent the location of two pinhole cameras (as described in Section 2.1.1), and I_l I_r their (virtual) image planes. The point m_l in I_l corresponds to the identical point (on the scene being imaged) m_r in I_r, i.e. m_l and m_r are both the 2D image points of the 3D point M in the scene. The points $e_l$and e_r are referred to as the *epipoles* of their

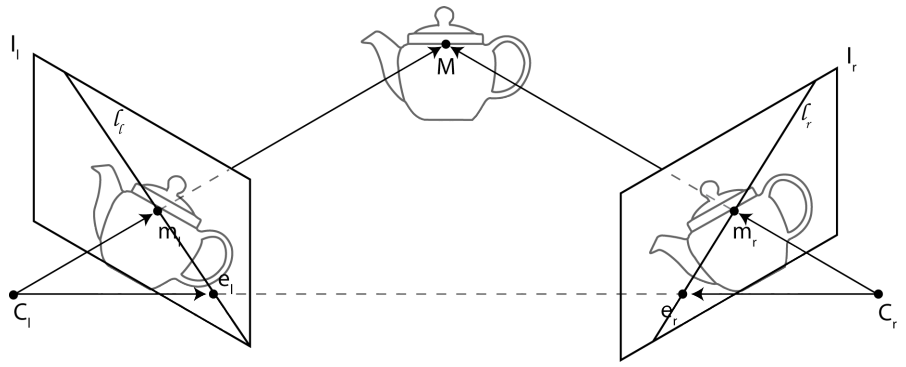


Figure 2.5: Epipolar Geometry

respective images, and can be described as the image of the projection centre (C) of one camera in the other. Epipoles are therefore intersections of the line C_lC_r (which can be referred to as the *baseline*) with the respective image planes. The *epipolar line* l_l (and vice-versa for l_r and e_r) is the line passing through the point m_l and e_l. Both epipolar lines in I_l and I_r lie on the intersection of the plane Π (the *epipolar plane*, defined by $m_lC_lC_r$) with the image planes. Thus, given a point m_l, the location of this point in the second image m_r is constrained to lie along the epipolar line l_r in this image. [49][51][31] This is significant, as if, say, a set of points have been identified on the left image, and corresponding epipolar lines have been calculated for the second image, it is possible to restrict the search for the location of the matching points in the right image to along the epipolar lines, thereby reducing this problem of locating matching points to a 1 dimensional search, and conversely, if point correspondences are known - enables the epipolar geometry to be estimated with greater ease.

2.2.2 The Essential Matrix

As it is now known that a scene point in one image will lie along the epipolar line of the other, it is necessary to consider how this relationship can be mapped. This is achieved by the use of the *essential matrix* originally described in [31]. The epipolar plane through a point M can be written as the coplanarity condition of the vectors M_l, T and $M_l - T$ (where M_l represents the same 3D point M in space, viewed from the left camera perspective, and T is the translation vector between cameras) as $(M_l - T)^T T \times M_l = 0$. If we factor in the relationship between M_l and M_r (the 3D point M as viewed from the right camera perspective) as being the rotation R

followed by the translation T, such that, $M_r = R(M_l - T)$, we can deduce that:

$$(R^T M_r)^T T \times M_l = 0 \tag{2.7}$$

An arbitrary vector product, such as $T \times M_l$ can be written as a multiplication by a rank-deficient matrix: $T \times M_l = SM_l$ where

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

Applying this to Equation 2.7 allows Equation 2.7 to be written as:

$$M_l^T E M_l = 0 \tag{2.8}$$

Where $E = RS$ is the *essential matrix*. This matrix establishes a link between the epipolar constraint and the extrinsic parameters of a stereo system, however equation 2.8 only provides a link between 2 known 3D points as viewed from either camera's perspective, by itself it does not allow for the relationship between 2D image point and epipolar line to be derived. The first step in achieving this is achieved by taking into account the perspective model of the camera as described in equation 2.2, and dividing by $Z_l Z_r$, equation 2.8 can be rewritten as:

$$m_l^T E m_r = 0 \tag{2.9}$$

Where m_l and m_r are 2D image location points of the 3D point M in space on the left and right image planes (I_l and I_r). It is then possible to express Em_r in equation 2.9 as a projective line (l_r) in the right image plane, I_r that passes through point m_r and the epipole e_r, giving the mapping between a point in the left image and its epipolar line in the right image:

$$l_r = E m_l \tag{2.10}$$

2.2.3 The Fundamental Matrix

So far, the relationship between points in one image to those in the other image has focussed on knowing the extrinsic parameters of the stereo system. Furthermore, the points described in the essential matrix are expressed in terms of the 2D Camera coordinates, as opposed to pixel coordinates from the resulting images. In this section, it is shown how pixel point matches from the resulting images can be used to determine the epipolar geometry of a scene.

Let H_l and H_r be the intrinsic parameters of the left and right cameras. Let $\bar{m}$ represent the pixel coordinates of a point m in the camera coordinate system. The relationship between m and $\bar{m}$ for both the left and right hand cameras is defined as $m_l = H_l^{-1}\bar{m}_l$ and $m_r = H_r^{-1}\bar{m}_r$. Substituting these equations for the point m into equation 2.9 results in:

$$\bar{m}_r^T F \bar{m}_l = 0 \tag{2.11}$$

where

$$F = H_r^{-T} E H_l^{-1} \tag{2.12}$$

F is known as the *fundamental matrix.* It is clear that there is a strong similarity between F and the *essential matrix* described previously. However, the main advantage afforded by the use of the fundamental matrix as opposed to the essential matrix is that all points are represented in pixel coordinates as opposed to camera coordinates. This means that F can be computed using image processing techniques performed on the resulting images without knowledge of the extrinsic parameters of the camera system. As follows, there are many different methods in which this can be achieved, and all depend upon searching for a matched set of points in both images in a process known as *image registration,* described in further depth in Section 4.2

3 Orientation Sensors

In this work we aim to evaluate how image processing, specifically determining epipolar geometry from two images, may be enhanced by the use of orientation sensors attached to a camera. An orientation sensor, also referred to as an Inertial Measurement Unit consists of several discrete components that measure different types of motion, and a method of combining the data received from these individual sensors in order to determine the orientation of the sensor. Presented in this chapter is an overview of the constituent parts of the orientation sensor, methodologies of combining this data, and the capabilities and limitations of an orientation sensor system.

3.1 Individual Sensors

In this section, an overview of the different constituent motion sensors that make up an orientation sensor is presented, and the movement they measure. Highlighted for each sensor are the characteristics that are likely to induce an error into the final reading and how this can be compensated for. Initially, we present an overview of how these sensors are manufactured.

3.1.1 The Manufacture of Sensors

Originally, the sensors presented in this chapter would have been mechanical systems consisting of moving parts. Whilst development of these systems had lead to significant improvements in terms of size, accuracy and reliability, it would not be feasible to use these mechanical based sensors in a consumer-level camera system. Fig. 3.1illustrates a typical mechanical orientation sensor from 1996.

Many consumer devices, (such as the Apple iPhone [3]) are available that contain gyroscopes, accelerometers and other sensors that can be used for detecting device

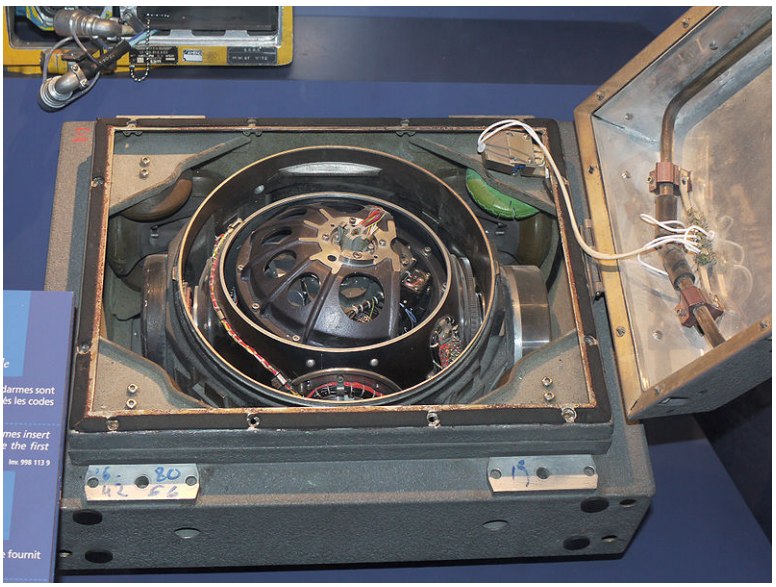


Figure 3.1: A Mechanical Inertial Measurement Unit - Image from wikipedia.org

movement and orientation. These devices rely on Micro-Electro-Mechanical Systems (MEMS) manufacturing techniques to produce the parts required that are small, accurate and have a low power consumption. Fig. 3.2 shows a modern day MEMS based inertial measurement unit.

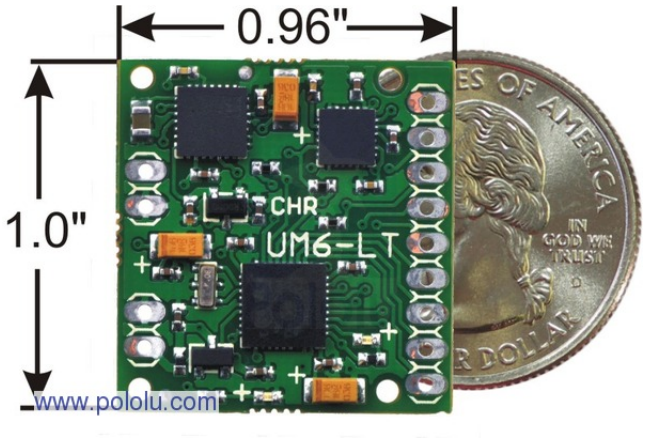


Figure 3.2: A MEMs Inertial Measurement Unit, shown with dimensions and coin for scale - Image from Pololu Robotics and Electronics Corporation

It is clear that the sensor presented in Fig. 3.2 is more suited to our application, as it allows for integration with the camera system, and does not affect the portability of a moving camera. MEMS devices have recently undergone a great deal of research [52], and their use is widespread. Most importantly, they allow for the manufacture

of parts that would previously have been based on large mechanical systems, to be produced on a small microchip size package and at a fraction of the cost. Presented below is a brief description of MEMs Technology:

3.1.1.1 MEMs Technology

MEMs is described in [32] as "a process technology used to create tiny integrated devices or systems that combine mechanical and electrical components". Whilst their manufacture is based on traditional Silicon Integrated Circuit production techniques, it also combines direct manipulation of the silicon substrates used to build the device. Typically, the production of MEMS structures can be achieved by constructing a series of patterned layers that are formed by a combination of both depositing new material, and selectively etching away existing material by the process of photolithography [2]. This process allows for devices and components of a size range of between just tens of microns to a few millimeters to be manufactured cheaply, reliably and quickly. [52][2][42]. It is important to note however, that whilst MEMS sensors may have the appearance of traditional silicon based Integrated Circuits, as they contain moving parts (albeit extremely small ones) - they are not, strictly speaking, 'solid state'. However, it is clear that whilst the principle of operation is very similar to the mechanical devices they have superseded, the moving parts are nothing like those shown in Fig. 3.1. For this reason, throughout this work, MEMS Inertial Measurement Sensors shall be treated as if they were discrete sold-state components.

The remainder of this section will describe the function, operation and basic design of the various component sensors in an inertial measurement unit.

3.1.2 Accelerometer

At it's simplest, an accelerometer is an instrument that measures acceleration along a single axis. Mechanically, it can be thought of as a proof mass suspended by springs, along with a displacement pickoff to provide a reading of the acceleration of the mass, as shown in Fig. 3.3:

There are several types of solid state accelerometers [42], one such is the surface acoustic wave (SAW) accelerometer. A SAW accelerometer consists of a cantilever beam holding a mass that resonates at a particular frequency. This mass is free to

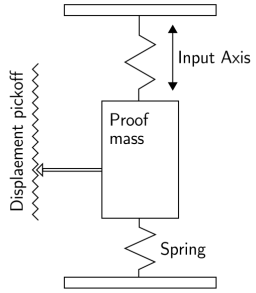


Figure 3.3: A Mechanical Accelerometer (Taken from [47] in [42])

move, whilst the other end of the beam is rigidly attached to the case. When an acceleration is applied to this case, the beam will bend and the frequency of the surface wave (traveling along the beam) will change proportionally to this strain. This frequency change is detected and the acceleration can then be determined. This arrangement is illustrated in Fig. 3.4.

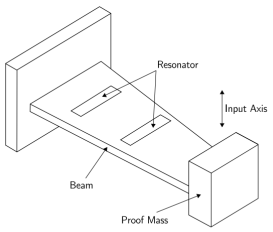


Figure 3.4: A solid-state, Surface Acoustic Wave Accelerometer, taken from [47] in [42]

Both of these methods of detecting acceleration can be manufactured using MEMS techniques. [42] Notes that there are two main classes of MEMS accelerometers that each operate based on the two previously described principles.

However, using an accelerometer to determine position produces a significant amount of error that must be counteracted. As an accelerometer measures acceleration, a single integration of the value over time should produce a result for velocity, and further integration of the velocity over time should enable to distance moved to be deduced. Any error in the original reading is therefore likely to have a significant effect on the final position result. The two main sources of error for a MEMs gyroscope are uncorrected bias and white-noise random-walk errors [42], however other errors such as inaccurate calibration and the effect of temperature changes on the device can also be a source for error.

Bias Error The offset of the output signal from the true value of acceleration in m/s^2 is known as the Bias Error. As mentioned, we shall be double-integrating the value of acceleration in order to determine position, thus, a constant error e causes the error in position to grow quadratically with time. Where t is the time of integration, the accumulated error due to the sensor's bias will be:

$$s(t) = e\frac{t^2}{2} \tag{3.1}$$

where s is the distance (in meters assuming acceleration is measured in m/s^2).

The Bias error is relatively simple to correct for - as the average long term output can be taken whilst the device is not undergoing any acceleration. However, the effect of gravity (which will cause a constant acceleration on the device) complicates this, and as such specialist equipment capable of moving the accelerometer through accelerations known relative to gravitational field is necessary. Many commercially available accelerometers undergo this calibration at the factory of manufacture due to this, however, in many applications, multiple accelerometers set apart orthogonally from one another are used to determine the acceleration due to gravity at any particular angle - and therefore provide a mechanism for estimating and filtering bias from the system (described in a further section).

Noise Based Random Walks Like any electro-mechanical system, the output of a MEMs accelerometer will contain noise. [42] defines this noise as a 'sequence of zero-mean uncorrelated random variables'. In the case of an MEMs sensor, 'each random variable is identically distributed and has a finite variance σ^2'. When integrated over a timespan $t = n \times \delta t$, this noise produces the result:

$$\int_0^t e(\tau)d\tau = \delta t \sum_{i=1}^{n} N_i \tag{3.2}$$

where $N_i$is the i^{th} random noise variable and n is the number of samples received from the device over the timespan and δt is the time between successive samples.

Thus, double integrating noise from the sensor will result in:

$$\int_0^t \int_0^t e(\tau)d\tau d\tau = \delta t \sum_{i=1}^{n} \delta t \sum_{j=1}^{n} N_j = \delta t^2 \sum_{i=1}^{n} (n-i+1)N_i \tag{3.3}$$

The expected error position is, according to [42],

$$E(\int_0^t \int_0^t e(\tau)d\tau d\tau) = \delta t^2 \sum_{i=1}^{n} (n-i+1)E(N_i) = 0 \tag{3.4}$$

with a variance of

$$Var(\int_0^t \int_0^t e(t)d\tau d\tau) = \delta t^4 \sum_{i=1}^{n} (n-i+1)^2 Var(N_i) = \frac{\delta t^4 n(n+1)(2n+1)}{6} Var(N) \approx \frac{1}{3}\delta t \times t^3 \sigma^2 \tag{3.5}$$

where approximation assumes that δt is small (i.e. the sampling frequency is large - which is a valid assumption for MEMs sensors). We can then deduce that this noise creates a second order random walk in position, with a mean of 0 and a standard deviation

$$\delta_s(t) \approx \sigma \times t^{\frac{3}{2}} \times \sqrt{\frac{\delta t}{3}} \tag{3.6}$$

which grows proportionally to $t^{\frac{3}{2}}$.

Manufacturers of accelerometers quote the power spectral density RMS noise (PSD) of an accelerometer in the unit $g/\sqrt{Hz}$.

We can convert this to $g/\sqrt{h}$ (g per $\sqrt{\text{hour}}$) with the formula:

$$g/\sqrt{h} = \frac{1}{60}\sqrt{PSD} \tag{3.7}$$

For example the 'Freescale Semiconductor MMA7361L' 3-Axis MEMS accelerometer quotes a noise level of $350\mu g/\sqrt{Hz}$. [16] This means that after one hour, the standard deviation of the error from the device will be $0.0058g$, which, if we assume $g = 9.8ms^{-2}$, we can calculate that this will be $0.057ms^{-2}$. As the noise from the sensor is random, the 'random walk' error introduced by it cannot be removed from the output. However, as this noise level is quoted by the manufacturer, it is possible to estimate the accuracy of the results produced by the sensor if the length of time it has been operating for is known.

3.1.3 Gyroscope

Traditional mechanical gyroscopes measure orientation by way of exploiting the conservation of angular momentum. Such gyroscopes would consist of a spinning wheel mounted on two frames, known as Gimbals, that allow it to rotate in all three axis, as shown in Fig. 3.5.

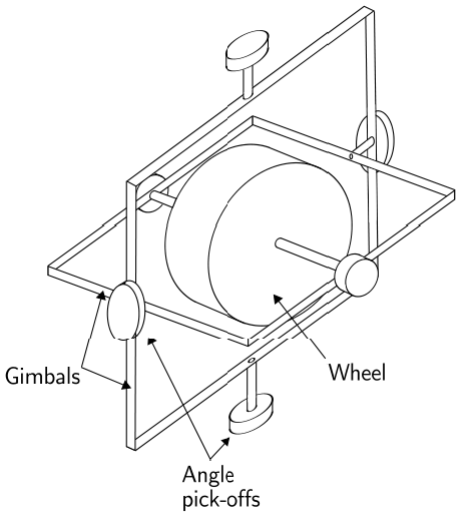


Figure 3.5: A Conventional Mechanical Gyroscope, taken from [47] in [42].

When the gyroscope is rotated around an axis, the wheel will remain at a constant global orientation, and the angles between adjacent gimbals will change. These angles can be detected and will represent the rotation the gyroscope has undergone from its initial starting position.

A MEMS based gyroscope relies on the Coriolis Effect in order to determine angular

velocity (or rate of change of angle) and are often referred to as 'rate gyros'. [42] The Coriolis effect states that in a frame of reference rotating at angular velocity ω with a mass m moving with velocity v experiences a force F described in Equation 3.8.

$$F_c = -2m(\omega \times v) \tag{3.8}$$

In a MEMS gyroscope, a mass is usually driven to vibrate along a drive axis. When the gyroscope is rotated, a secondary vibration is induced perpendicular to this drive axis, as is shown in Fig. 3.6.

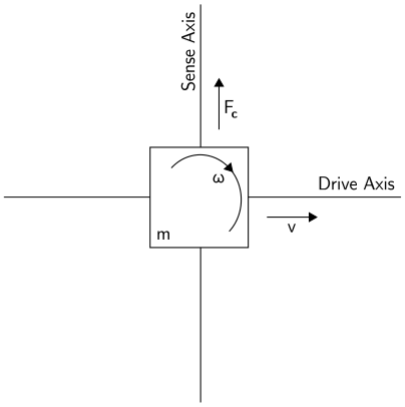


Figure 3.6: A Gyroscope Based on a Vibrating Mass

As with MEMs accelerometers, it is expected that the output of a gyroscope will be susceptible to some error. Indeed, the sources of noise are very similar, and are described below.

Bias Errors The bias of a Rate Gyro can be determined by measuring the average output of a gyroscope over a period of time whilst it is not undergoing any rotation, and is usually specified in degrees per hour ($^\circ h^{-1}$). Unlike an accelerometer, we only need to integrate the angular velocity measurement once in order to determine position. For this reason, a constant bias error e, when integrated, causes an angular error which grows linearly with time: $\theta(t) = et$, therefore, once the bias is known, it is trivial to remove its effects from the results.

Noise Based Random Walks This situation is almost identical to that of the MEMS accelerometer described previously, however, as before, only a single integration step is performed on the data from the rate gyroscope in order to determine the orientation of the sensor. Assuming, the noise detected follows from equation 3.2, the single integration of the mean noise value (E) can be written as:

$$E(\int_0^t e(\tau)d\tau) = \delta t \times n \times E(N) = 0 \tag{3.9}$$

with a variance of:

$$Var(\int_0^t e(\tau)d\tau) = \delta t^2 \times n \times Var(N) = \delta t \times t \times \sigma^2 \tag{3.10}$$

(where $Var(N_i) = Var(N) = \sigma^2$), the noise introduced into the integrated signal will be a zero mean, random walk error with standard deviation:

$$\sigma_\theta(t) = \sigma \times \sqrt{\delta t \times t} \tag{3.11}$$

that grows proportionally to the square root of time.

MEMS gyroscopes are not as accurate as their physical counterparts as MEMS accelerometers are to theirs and can suffer from significant noise levels. [52][42] However, the recent development of MEMS gyroscopes has lead to the production of new MEMS gyroscopes that produce a negligible amount of noise. [11]. In a similar fashion to accelerometers, manufactures will specify the expected error of the integrated signal of a gyroscope. This is commonly referred to as the 'Angle Random Walk' (ARW) measurement, where ARW= $\sigma_\theta(1)$, and units are specified in $^\circ/\sqrt{h}$. One such example is the popular ST Microelectronics L3G4200D that is used in many consumer products, such as the Apple iPhone [3] , which specifies a *Rate Noise Density* (RND) of $0.03/\sqrt{Hz}$[45].

3.1.4 Magnetometer

Compasses, capable of measuring the earth's magnetic field for navigational purposes have been used for many centuries. It is nowadays commonplace to find solid-state electromagnetic compasses in a wide variety of devices, such as smartphones and navigational aids. Typically, such a solid state integrated system will be based on *magnetoresistive* materials. These sensors consist of thin strips of Permalloy (an alloy of nickel and iron), whose electrical resistance changes with the changes in applied magnetic field [8]. Such arrangements have proven to be extremely sensitive - with some being capable of sensing below 0.1 miliguass (1 gauss is equivalent to 1×10^{-4} Tesla - the SI unit for magnetic flux density) [8]. This makes such sensors appropriate for detecting the earth's magnetic field, which varies from between 0.1 gauss to 1 gauss [29], with a typical value of 0.5 - 0.6 gauss [8], and has a component parallel to the earth's surface that always points to magnetic north. Thus, if a sensor is configured in order to determine the direction of magnetic north, the location of which remains constant regardless of the sensors orientation or global location[1], this could be useful in determining motion of an inertial measurement unit.

The resistance of a strip of permalloy decreases as the direction of magnetization (in this case, the earth's magnetic field) rotates away from the direction in which current flows through the strip. By measuring the electrical resistance in two strips mounted perpendicular to one another to form two axis (XY) on a horizontal plane, the heading of the device - i.e. degrees from North, can be measured using the equation

$$\text{Heading} = \arctan(y/x) \tag{3.12}$$

where y and x are the readings obtained from the two axis. To account for the tangent function being valid over 180°and not allowing the $y = 0$ division, the following equations can be used, as described in [8]:

[1] Constant here implies constant sufficient for the applications of this work. The earth's magnetic field does in fact reverse after a period of time, however this is an occurrence that has been found to happen only once every 400,000 years. [29]

$$
\begin{aligned}
\text{Heading}(x = 0, y < 0) &= 90.0 \\
\text{Heading}(x = 0, y > 0) &= 270.0 \\
\text{Heading}(x < 0) &= 180 - [\arctan(y/x)] \times \frac{180}{\pi} \\
\text{Heading}(x > 0, y < 0) &= -[\arctan(y/x)] \times \frac{180}{\pi} \\
\text{Heading}(x > 0, y > 0) &= 360 - [\arctan(y/x)] \times \frac{180}{\pi}
\end{aligned}
$$

This system assumes that the compass is operation on a horizontal plane. However, this is unlikely to be the case, and as such, many magnetometer ICs are packaged in a format capable of measuring the magnetic field across three axis. Furthermore, by itself, knowing the third angle of rotation by using an additional magnetoresistive detector is insufficient as in order to rotate the XY magnetic readings back to horizontal, the pitch and roll angles relative to horizontal must also be known. One such method of achieving this is by use of an additional sensor, such as those described in Section 3.1.2 and Section 3.1.3.

A significant source of error for a magnetometer is the nearby presence of ferrous metals and strong magnetic fields to the sensor. In certain applications, such as the detection of vehicles on a section of road or detection of ships and submarines, this is desirable, but not so for detecting small changes in the earth's magnetic field as a sensor moves. In the application described in this work, it is likely that an orientation sensor, possibly consisting of a magnetometer, will be attached to a camera containing ferrous metals, and actuators (such as a motor for shutter release) that consist of magnets that are capable of interfering with the magnetometer. Compensating for magnetic interference is a relatively simple process. Contained in [8] is a method for compensating for the magnetic interference effected on an electronic compass made of a set of magnetometers by a car, and presented here is a summary of the process:

3.1.4.1 Correcting for Distortion due to Nearby Ferrous Effects

This subsection presents a methodology described in [8]:

If no interference is received by the magnetometer, if the magnetometer is rotated around 360°, the readings shown in Fig. 3.7 should be expected:

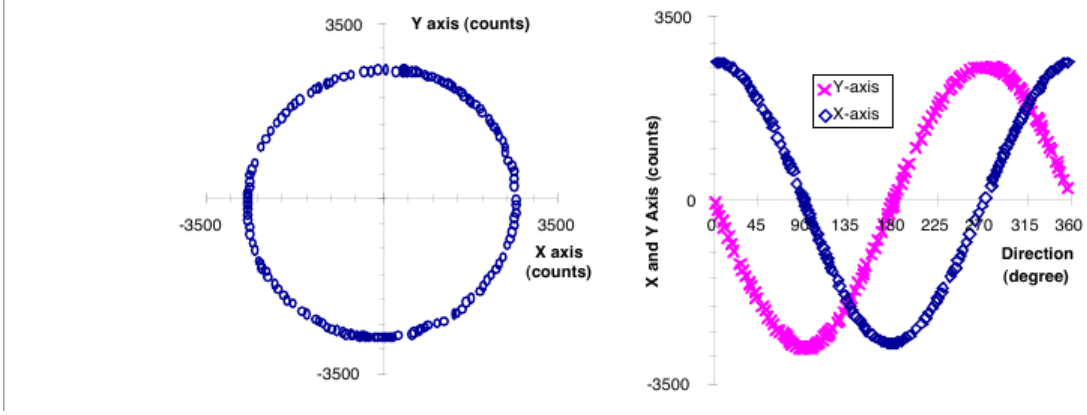


Figure 3.7: No Interference of Magnetometer Readings For 360°Rotation in Level Plane - Source [8]

As can be seen, the plot is circular with an origin of $(0, 0)$. However, should the magnetometer be mounted in a situation where is is affected by ferrous interference, such as in a car, the following pattern is produced. (Fig. 3.8)

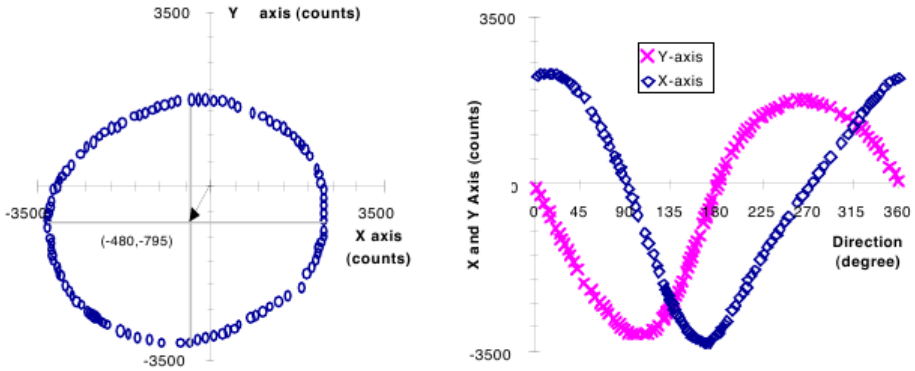


Figure 3.8: A Magnetometer Rotated Horizontally Around 360°Whilst Experiencing Interference from A Car Body and Engine - Source [8]

As is shown here, the reading plot becomes an ellipse, and has an origin of $(-480, 795)$, we can describe the interference of causing an offset $(X_{\text{off}}, Y_{\text{off}})$ and scale distortion $(X_{\text{sf}}, Y_{\text{sf}})$, which can be computed with the following simple algorithm, as described in [8]:

- Mount the compass on a camera (or in the case used in [8], a car) and move the camera in a circle on a horizontal plane.
- Find the maximum and minimum values of the X and Y magnetic readings.
- Use these four values to determine the scale factors determine the X and Y scale factors (X_{sf} Y_{sf}) and the zero offset values ($X_{\text{off}}Y_{\text{off}}$):
  - $X_{\text{sf}} = 1$ or $(Y_{\text{max}} - Y_{\text{min}})/(X_{\text{max}} - X_{\text{min}})$ whichever is greater
  - $Y_{\text{sf}} = 1$ or $(X_{\text{max}} - X_{\text{min}})/(Y_{\text{max}} - Y_{\text{min}})$ whichever is greater
  - $X_{\text{off}} = [(X_{\text{max}} - X_{\text{min}})/2 - X_{\text{max}}] \times X_{\text{sf}}$
  - $Y_{\text{off}} = [(Y_{\text{max}} - Y_{\text{min}})/2 - X_{\text{max}}] \times Y_{\text{sf}}$

This calibration procedure is an important step in preparing a magnetometer for use in its intended environment - and it is recommended by manufacturers that calibration is performed on every device after having been installed.

3.2 Data Processing

So far in this section, the basic operating principles and concepts of an accelerometer, gyroscope and magnetometer have been introduced. It is clear that by themselves, they are capable of measuring a variety of components of motion, however, in the application discussed in this work, it is necessary to determine the position of the device attached to them. It has been discussed that for this information to be determined from an accelerometer and gyroscope, it would be necessary to perform a double integration of the acceleration information received in order to determine position. We have also mentioned that there are significant sources of error to this methodology. In this section, an overview of a complete orientation sensing system is presented, along with a methodology in which to minimize the effect of accumulated error on the system.

3.2.1 Inertial Measurement Units

An inertial measurement unit (IMU) is a device that is capable of tracking the position and orientation of an object relative to a known starting position. They will typically contain three orthogonal rate gyroscopes, and three orthogonal accelerometers. Almost all IMUs fall into two distinct categories [42]. Those based on a 'stable

platform' design and those based on a 'strap down' configuration. Stable platform gyroscopes mount all of the necessary sensors on a platform that is in turn mounted on a set of gimbals, that allow for an object rigidly attached to the IMU to rotate around 3 axis, but the IMU itself remains at a constant orientation, much like the mechanical gyroscope described in Fig. 3.5. Angle pickoffs record the device's orientation, and the accelerometer data is integrated twice to determine position relative to original position. In a strapdown system, all sensors are mounted rigidly to a single platform. In order to position and orientation to be determined, the signals from the rate gyroscopes are integrated to determine the rotation around three axis, and then this rotation information is used in order to rotate the readings from the accelerometers onto a global axis, prior to double integration.

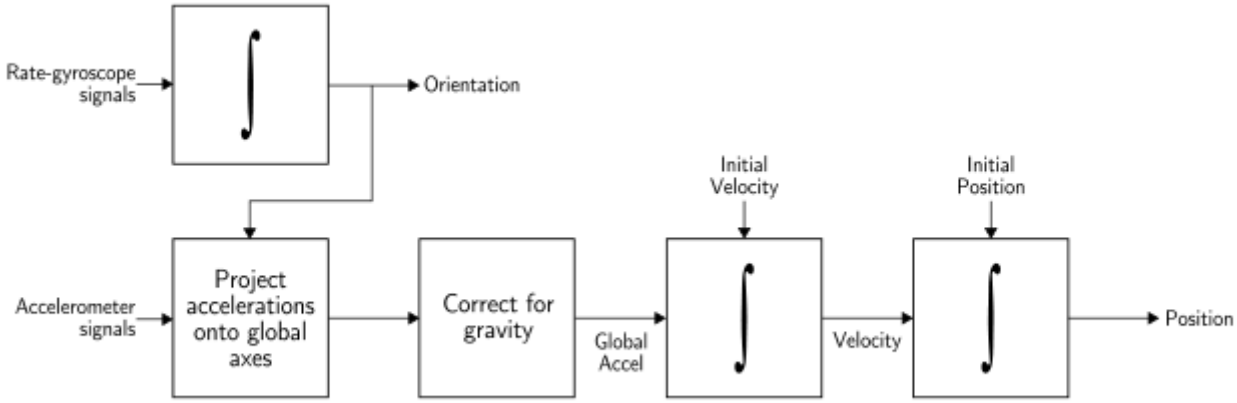


Figure 3.9: A Block-Diagram of A Strapdown IMU Algorithm, source [42]

Strapdown systems are typically smaller and have fewer moving parts than stable-platform systems, and although they require more complex algorithms to implement, as the cost of computation has decreased, they have become the dominant type of IMU. [42]. For this reason, this work will utilise a commercially available strapdown based IMU.

3.2.2 Correcting For Error

As discussed previously, double integrating any error produced by the orientation sensor will produce an overall error in the reading that will be significant, and will grow exponentially over time. In order to try to reduce the effect of noise from the

sensors on final readings, a common approach is to implement a system of filtering out noise from the sensors. There are several different methods, each ranging in complexity, an overview of which is presented here.

3.2.2.1 Low Pass Filter

In an analogue system, a low pass filter as a device that allows low-frequency signals to pass unimpeded, but attenuates signals with a frequency higher than the cutoff frequency. In an inertial measurement unit, it is possible to use one to filter out erroneous readings from an accelerometer, as described by [10]. This method would filter out short term changes in acceleration, and would show only long term changes, such as a constant movement or gravitational acceleration, and also a large amount of noise. Whilst effective, this method is only suitable to use on accelerometers, and adds lag to the system. It also cannot be applied to the gyroscope readings, and thus cannot correct for drift derived from these sensors.

3.2.2.2 Kalman Filter

This is the 'gold standard' filtering mechanism used by many Inertial Measurement Units [10]. However, it is also much more complex than a simple low pass or averaging filter and would require a significant effort to effectively implement it [10]. Fortunately, there are devices that are readily commercially available that combine the inertial sensors necessary and an implementation on a micro-processor of a version of the Kalman filter - the Extended Kalman Filter. An example of such a device would be the "CH Robotics UM6 Ultra-Miniature Orientation Sensor" described in [9]. For these reasons, this will be the device that is used in conjunction with a camera for implementation and experimentation throughout this work. As the Kalman Filter is relatively complex, a detailed description would be beyond the scope of this work. Further details can be found in the original paper by R. E. Kalman, [24]. Presented in this subsection is an overview of the filter's general concepts and operation.

Overview Kalman filtering is defined by [40] as an 'optimal recursive data processing algorithm', that is capable of producing an output based on several sets of data available to it. One such use case could be in order to deduce the orientation of an

object based on the information available to it from multiple sources, i.e. an orientation sensor consisting of the multiple sensors described previously in this chapter. The basic operation of such an algorithm is described in Chapter 1 of [40], and can be paraphrased for the application described in this work as such:

Suppose the angular orientation of a sensor is to be deduced by integrating the data measured by the gyroscope as described in Section 3.1.3 at time t_1, the orientation of the sensor could be estimated as z_1. The error of such an estimation, taking into account both constant bias and angular random walk errors values is such that the standard deviation of this value is σ_{z_1} (and equivalently, the variance of this measurement is $\sigma_{z_1}^2$). Therefore, the *conditional probability* of $x(t_1)$ - the actual orientation at time t_1, conditioned on the observed value of the measurement being z_1 can be described by Fig. 3.10.

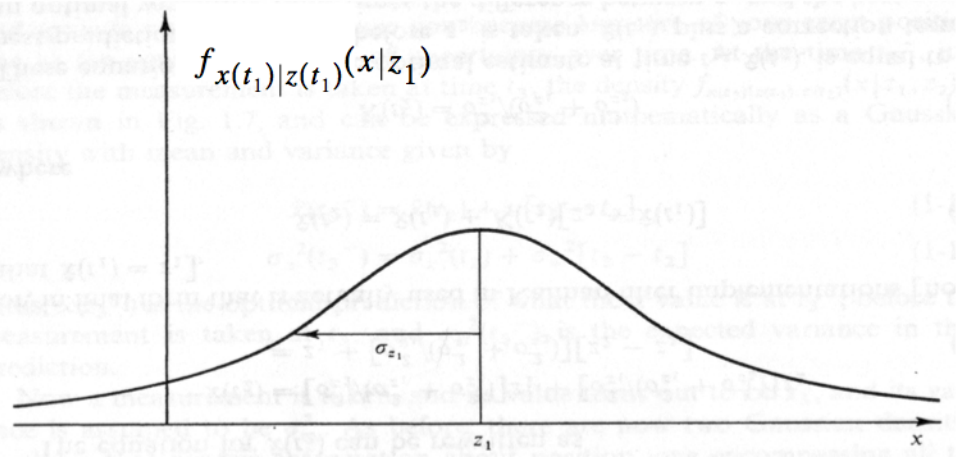


Figure 3.10: Conditional Probability Density of Position Based on Measured Value z_1 - Taken from [40]

If we compliment this reading from that of another sensor - such as the magnetometer, referred to as z_2 taken at the same time t_1, and this sensor has a higher level of accuracy (i.e, a smaller variance) - we can use this information to obtain a more reliable overall result. Such a combination is shown by Fig. 3.11.

In this graph, the position, μ has the highest probability of being the correct orientation of the device. This position, at time $t_1 = t_2$, has a conditional density which

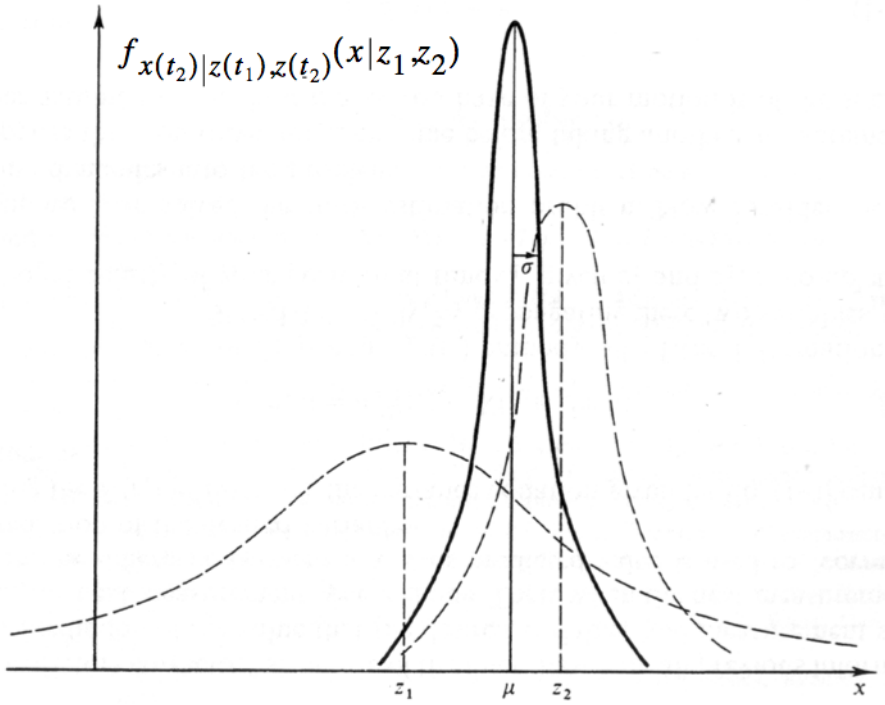


Figure 3.11: Conditional Probability Density of Position based on data z_1 and z_2 - Taken from [40]

can be described as a Gaussian density with mean μ, calculated as:

$$\mu = [\sigma_{z_2}^2/(\sigma_{z1}^2 + \sigma_{z2}^2)]z_1 + [\sigma_{z_1}^2/(\sigma_{z1}^2 + \sigma_{z2}^2)]z_2 \tag{3.13}$$

Given this density, the best estimate for the true orientation, taking into account measurements from both the gyroscope and magnetometer is equal to μ, with a variance of σ^2 where

$$\frac{1}{\sigma} = (\frac{1}{\sigma_{z_1}^2} + \frac{1}{\sigma_{z2}^2}) \tag{3.14}$$

The equation for μ can be re-written as:

$$\mu = \hat{x}(t_1) + K(t_2)[z_2 - \hat{x}(t_1)] \tag{3.15}$$

where:

$$K(t_2) = \sigma_{z_1}^2/(\sigma_{z_1}^2 + \sigma_{z_2}^2) \tag{3.16}$$

3.15 Forms the basis of the Kalman filter. The filter is capable of including inputs from any number of sources, so it is possible to combine data from multi-axis accelerometers, gyroscopes and magnetometers to produce an accurate estimation of the sensor's orientation.

3.3 Capabilities and Limitations

In this chapter, the operation and capabilities of various motion detection devices have been presented - along with a description of the types of error that can be expected from such devices and methodologies in order to counteract these. As can be seen from the preceding section, it is clear that the error produced by such devices is highly significant. 'Dead reckoning, or the process of determining the location of an inertial device alone is known to produce position errors of around 1-8 cm per second [4]. Many works have confirmed this and have attempted to develop strategies to cope with this problem, such as the augmentation of inertial sensors with additional wireless mapping and location systems[12], or by the use of advanced filtering techniques [46]. Currently, the state of the art will not allow for MEMs based inertial measurement devices to accurately and reliably measure changes in position (linear movement) over anything other than a very short amount of time [46][42][12], however, by using a series of magnetometers, accelerometers and gyroscopes in conjunction with a Kalman Filter based processing algorithm, it is possible to reliably and accurately determine the orientation of a device in terms of roll, pitch and yaw[41] [50]. These devices are small, easy to interface, and relatively inexpensive, and so would be suitable to use in conjunction with a camera for the purposes of capturing additional information at the time a photograph is taken.

4 Related Work

4.1 Overview

In the previous chapters of this work, the concepts of Epipolar Geometry and the workings of an Orientation Sensor have been described. In this section, we review methodologies of determining the Epipolar Geometry of a scene from a two 2D images, and the use of orientation sensors to assist this process.

4.2 Image Registration Methods

Section 2.2.2 and Section 2.2.3 describe how known point correspondences in multiple images can be used in order to determine the 3D geometry of an image. The process of determining these point correspondences between two images is often referred to as Image Registration. Image registration is a broad term used to describe how one image of a scene may be transformed to match a separate image of the same scene taken from a different perspective or at a different time. In this work, this transformation and the subsequent mapping of points from one image to another is represented by the fundamental matrix (described in Section 2.2.3). These matched points can be provided by a user manually selecting points to use, however, for obvious reasons of time, cost and accuracy this is impractical and automated algorithms are the most widely used methodology of determining points and their matches in a set of images. In this section, we focus mainly on how points are selected from an image and matched. Many different methods are available and a comprehensive survey of these techniques is presented in [58]. This survey defines a broad framework in which the majority of methods operate, namely:

1. Feature Detection
2. Feature Matching

3. Transform Model Estimation
4. Image resampling and Transformation

In this subsection, we focus on the first three stages of this process (as Image Resampling and Transformation is described by the Epipolar Geometry of a scene) and present a review of the most commonly used and reliable methods documented for achieving each stage.

4.2.1 Feature Detection

This stage is the initial part of the image registration process. In this stage, objects and points in both images are identified. Typically, this could be achieved by a user selecting points in the image, however the focus here is on the automated detection of objects within a scene. Many articles and pieces of literature divide the field of feature detection into two broad categories: region based and feature based. Region based methods focus mainly on the statistical matching of areas in an image to corresponding areas in a reference image as opposed to identifying any discreet features within a scene, and as such are more relevant in the Feature Matching stage of the registration process.

Traditional feature-based methods make strong use of corner and edge detection algorithms. An overview of a standard corner detection algorithm is described below:

4.2.1.1 Corner Detection Algorithms

The simplest of corner detection algorithms run as follows:

1. Place a small window over an area of pixels on an image
2. Translate the window in any direction.
3. If the image in this window changes significantly in both the horizontal and vertical translations, then the location of the corner can be approximated as the central point of the first window location.

This basic idea is illustrated in Fig. 4.1.

Of course, this is a very simplified description of a corner detector. Much work has been done on the subject and it is an large area of ongoing research. There exist several popular methodologies for corner detection, one of the most widespread in

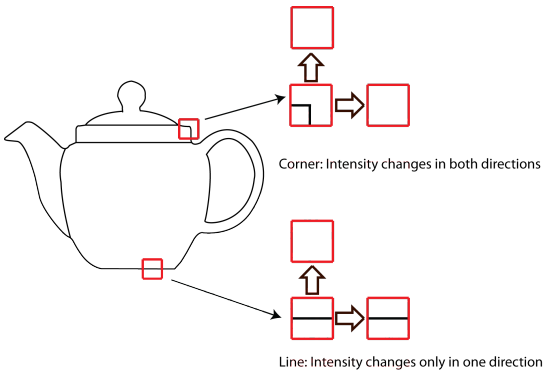


Figure 4.1: A Simple Corner Detection Algorithm

use being the *Harris Corner Detection Algorithm* described in [19]. This method makes use of *image gradients* both over the whole image and in a particular window in order to determine the location of a corner. A gradient for an image I is defined by [14] as:

$$\Delta I = (\tfrac{\delta I}{\delta x}, \tfrac{\delta I}{\delta y})^T$$

which can be estimated by observing that:

$$\tfrac{\delta I}{\delta x} = \lim_{\delta x \to 0} \tfrac{I(x+\delta x, y) - I(x,y)}{\delta x} \approx I_{i+1,j} - I_{i,j}$$

At a corner, it is expected that there should be both a large gradient, and, in a small window, the gradient of orientation should swing sharply. Corners can thus be identified by analysing the variations in gradient within a window. The matrix (defined in [14])

$$H = \sum_{window} \{(\Delta I)(\Delta I)^T\}$$

$$\approx \sum_{window} \begin{bmatrix} (\frac{\delta G\sigma}{\delta x} \times I)(\frac{\delta G\sigma}{\delta x} \times I) & (\frac{\delta G\sigma}{\delta x} \times I)(\frac{\delta G\sigma}{\delta y} \times I) \\ (\frac{\delta G\sigma}{\delta x} \times I)(\frac{\delta G\sigma}{\delta y} \times I) & (\frac{\delta G\sigma}{\delta y} \times I)(\frac{\delta G\sigma}{\delta y} \times I) \end{bmatrix}$$

gives a good representation of the behavior of the orientation of gradients in the window. If both eigenvalues of the matrix are small, then it can be assumed that the window is in an area of constant grey level. If one is large, then this can signify an edge, as one large value can be associated with the shift in gradients at an edge. At the location of a corner, it would be expected that both eigenvalues would be large. The Harris Corner Detector [19] searches for local maxima of

$$\det(H) - k(\frac{trace(H)}{2})^2$$

Where k is some constant. These maxima are then tested against a threshold, and local maxima above this can imply that both eigenvalues are large, and thus estimate the location of a corner.

The outcome of the feature detection stage of the registration process, by using feature based method detection - such as a corner detector, should be a set of points in both images that are likely to match with the points in the other image. As mentioned in [58], the use of feature based methods is adequate for a large number of applications - such as those captured from remote imaging devices or for use in computer vision applications.

4.2.2 Feature Matching

Once a set of corners, or 'features', has been recovered from each image, it is necessary to match them to one another in order to proceed with calculating the fundamental matrix. As with the original process of feature detection, this is a significant area of research in its own right, and there are many different methods by which points can be matched [58], a detailed examination of which is beyond the scope of this work. Presented here is an overview of matching points using a basic correlation technique described in [55]. This algorithm forms the basis of several widely used and robust techniques that have been developed since [57].

Matching Through Correlation Given a point determined by the corner detection algorithm described in Section 4.2.1.1, $m_1$in image 1 a *correlation window*, of size $(2n+1) \times (2m+1)$ is created centered around this point. A rectangular search area, the *search window* of size $(2d_u + 1) \times (2d_v + 1)$ is then created around this point in the second image, as shown in Fig. 4.2:

A correlation operation is then performed on a window around all points found in the search window and the *correlation window* in the first image, to produce a score of between -1 for two windows that are completely dissimilar, to 1 for correlation

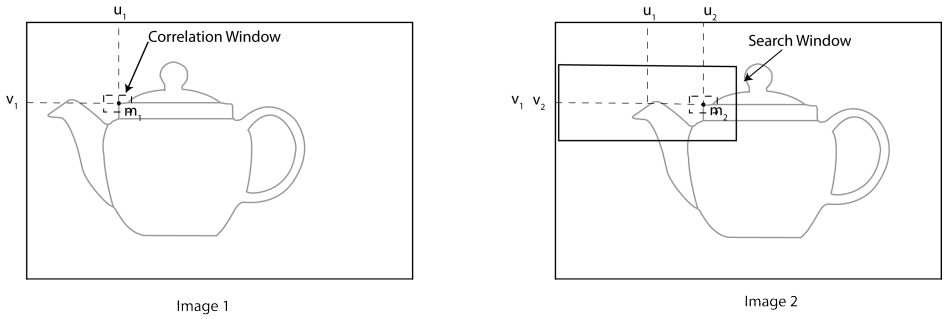


Figure 4.2: The Search and Correlation Windows of the Correlation Technique of Feature Matching

windows found to be identical. This correlation score is defined in [55] as:

$$\frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m}[I_1(u_1+i,v_1+j)-I_1(\bar{u_1},v_1)]\times[I_2(u_2+i,v_2+j)-I_2(\bar{u_2},v_2)]}{(2n+1)(2m+1)\sqrt{\delta^2(I_1)\times\delta^2(I_2)}} \tag{4.1}$$

where $I_k(\bar{u},v)$ is the average value at the point (u,v) of I_k $(k=1,2)$ and $\sigma(I_k)$ is the standard deviation of the image I_k in the neighborhood $(2n+1)\times(2m+1)$ of (u,v).

During implementation, a threshold is applied, and should a score be higher than this threshold - it can be assumed that these two points match. The size of the correlation window (n,m) and the search window (d_u,d_v) will also vary depending on implementation - with a larger value of d_u and d_v increasing the amount of points searched, and larger values for n and m decreasing the accuracy at which points are matched. This method of point matching is also considerably affected by the rotation between images (the camera's extrinsic parameters).

Robustness of the Matching Process Except in the simplest of images, it is clear that the process defined in the previous paragraph for determining point matches will yield a significant level of errors, for example, many surfaces will have areas of high repetition which will cause multiple erroneous matches to be found. In order to produce a suitable match, several algorithms are available to fulfill this need [58][43], one of the most widely used being the Random Sample and Consensus (RANSAC) algorithm first defined by [13]. The essence of the algorithm is as follows [15]:

1. Until k iterations have occurred:
    a) Detect features present in each image and match them using a correlation process
    b) Randomly select enough matches (pairs of points) to determine a transform that will align the two images (transform model estimation)
    c) Apply this transformation to all points in the first image
    d) Evaluate the number of points that now lie within a threshold area of the points in the second image
2. Return the transform model with the highest number of points matched

This algorithm relies on the use of a ***transform model estimation*** algorithm, such as one described in the following section.

4.2.3 Calculating the Fundamental Matrix from Point Matches (Transform Model Estimation)

In his work on epipolar geometry and the essential matrix, Longuet-Higgins describes the methodology for determining the essential matrix by the use of a set of matching points. [31] This method is known as the '8 Point Algorithm' and can easily be extended to include the Fundamental Matrix. The Fundamental Matrix, as described previously, is defined by the equation

$$\tilde{m}_r^T F \tilde{m}_l = 0$$

for any pair of matching points $\tilde{m}_l \leftrightarrow \tilde{m}_r$. The 8 Point Algorithm stipulates that this definition can be re-written as a homogenous linear equation, with the following definitions:

$$\tilde{m}_r = \begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix}, \tilde{m}_l = \begin{bmatrix} x_l \\ y_l \\ 1 \end{bmatrix}, F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$$

$$x_l x_r f_{11} + x_l y_r f_{12} + x_l f_{13} + y_l x_r f_{21} + y_l y_r f_{22} + y_l f_{22} + x_r f_{31} + y_r f_{32} + f_{33} = 0$$

If f is denoted by the 9-vector made up of the entries of F in row-major order, then the previous equation can be expressed as a vector inner product:

$$(x_l x_r, x_l y_r, x_l, y_l x_r, y_l y_r, y_l, x, y, 1) f = 0 \tag{4.2}$$

It is then possible to solve for F linearly by the use of at least 8 matching point coordinates (hence the algorithm's name).

Whilst simple and easily implemented, this algorithm is extremely susceptible to noise, or inaccurate point location and matches. One way of reducing this problem is to use the algorithm developed by Hartley and described in [21]. In this algorithm, the image coordinates are *normalized* before the linear solution is applied to the system of equations. This normalization is the translation and scaling of each image so that the centeroid of the set of points lies at the origin of the image, and the RMS of the distance of points from the origin is equal to $\sqrt{2}$. The justification for this, as described in [21] is that, should the first values (the u values) of a set of point coordinates in the first image be relatively close to one another (i.e. $\{1001.5, 1005.6, 988.7, ...\}$etc), by translating by 1000, these would become $\{1.5, 5.6, -1.3\}$. This is useful, as before translation, the significant values of these coordinates are obscured by the coordinate offset of 1000 and as such is only found in the 3rd or 4th significant, and can have negative effects on calculating the Fundamental matrix. This is due to the fact that in equation 4.2, the terms $x_l x_r, y_l y_r, y_l y_r$ would likely end up having magnitudes of 10^6should the points be un-normalised. By reducing the points' average distance from the origin to $\sqrt{2}$ we make the system less susceptible to noise from incorrectly matched and outlier points.

In the normalised version of the algorithm, we thus apply a transformation T to the points before computing the matrix. This means that formula 2.11 must then be re-written as:

$$\hat{m_l}^T \hat{F} \hat{m_r} = 0$$

where

$$\hat{m_l} = T_l m_l \ \hat{m_r} = T_r m_r$$

where T_l and T_r are the normalization transformations applied to both sets of points individually. F can then be retrieved from $\hat{F}$ by $F = T_l^T \hat{F} T_r$.

4.2.4 Image Rectification

Image rectification is the process whereby two images taken of the same scene are transformed into a standard coordinate system, by projecting them on to a common image plane. Usually, the image is transformed in such a manner as to align the epipolar lines to become parallel to one another and the horizontal axis. Once an image is rectified, the search for point correspondences is simplified as 1D search [49], greatly enhancing the feature matching process. In its simplest form, in order for images to be rectified, it is necessary to know both the intrinsic and extrinsic parameters of the cameras, however, there exist several techniques that are capable of rectifying multiple images from uncalibrated cameras ([27],[22]) . In all cases, image rectification can be described as a projective transformation, or warping of the images[49]. For example, if the extrinsic and intrinsic parameters are known, the transformation can be computed as follows[41]: A plane is defined that is an intermediate between the two image planes for the two poses. Using R to denote the rotation matrix between the two camera extrinsic rotations - such a plane can be produced by rotating the first image with rotation R', which has the same rotation axis as R but is of half magnitude of R. The images can then be rectified onto this plane by creating a transformation based upon a homography, or projective transform defined as

$$H = BR'A^{-1} \tag{4.3}$$

where B is the camera matrix of the resulting plane.

4.2.5 Advanced Methodologies

Described in the preceeding subsections is an overview of the basic process of registering a pair of images and determining the epipolar geometry (encapsulated in the Fundamental Matrix) of a scene using point correspondances in the 2D images. This alone is an area of significant research interest, and much work has been done on increasing the accuracy of the obtained geometry.[56] presents a comprehensive survey of a variety of methods detailed in the literature.

Whilst the '8-Point' method as described above can produce good results [21], it has

also been critisized by some [48] for being innacurate and unstable, especially with regard to images that have undergone certain transformations.

Presented in ([20] pp. 284 - 285) is method considered by the authors to be the 'Gold Standard' for producing an estimate for the Fundamental Matrix. In this method a value for F is determined using the standard 8 point algorithm as described and is then optimised to reduce the cost function 4.4 using Algorithm 4.1.

$$\sum_i d(m_i, \hat{m}_i)^2 + d(m_i', \hat{m}_i') \tag{4.4}$$

Where in Equation 4.4, d is the geometric distance between the two points, and $m_i \longleftrightarrow m_i'$ are corresponding 'measured' points (i.e. Selected manually or via an automated matching process) and $\hat{m}_i \longleftrightarrow \hat{m}_i'$ are corresponding 'true' points that satisfy $\hat{m}_i'^T F \hat{m}_i = 0$ exactly for the rank-2 matrix F, the estimated fundamental matrix. Selecting F with the smallest geometric distance cost should therefore produce highly accurate results, even in the case where points have been manually selected - as it is usually impossible (or at least highly impractical) to select known point correspondances at sub-pixel level.

Algorithm 4.1 The Gold Standard Algorithm for estimating F. Taken from

1. Compute an initial rank 2 estimate of $\hat{F}$ using a linear algorithm (in this case the 8 point algorithm)
2. Compute an initial estimate of the subsidiary variables as follows:
   a) Choose camera matrices $P = [I|0]$ and $P' = [[e']_\times \hat{F}|e']$ where e' is an epipole obtained from F.
   b) From the correspondance $m_i \longleftrightarrow m_i'$ and $\hat{F}$ determine the point $\hat{M}_i$ - the point m_i in 3D space.
   c) The correspondance consistant with $\hat{F}$ is obtained as $\hat{m}_i = P\hat{M}_i$, $\hat{m}_i' = P'\hat{M}_i$.
3. Minimise the cost: $\sum_i d(m_i, \hat{m}_i)^2 + d(m_i', \hat{m}_i')$ over $\hat{F}$ and $\hat{M}_i$ for $i = 1, ..., n$.

Algorithm 4.2 A sparse Levenberg-Marquardt Algorithm [30][35] used to calculate the Fundamental Matrix taken from ([20] pp. 605 & 608)

Given: A vector of measurements $\mathbf{X} = (X_1^T ... X_n^T)^T$, with a covariance matrix of Σ_X an initial estimate of the parameter vector $\mathbf{P}$ where $\mathbf{P}$ is partitioned such that $\mathbf{P} = (a^T, b_1^T, ..., b_n^T)$, such that $\partial \hat{X}_i / \partial b_j = 0$ for $i \neq j$

Output: A set of parameters $\mathbf{P}$ that minimises $\epsilon^T \Sigma_X^{-1} \epsilon$ where $\epsilon = \mathbf{X} - \hat{\mathbf{X}}$

1. Initialise a constant $\lambda = 0.001$
2. Compute the derivative matrices $A = [\partial \hat{X}_i / \partial a_i]$ and $B = [\partial \hat{X}_i / \partial b_i]$ and the error vectors $\epsilon_i = X_i - \hat{X}_i$. Where A and B are defined as the Jacobian submatrices $J = [A \mid B]$ where $J = [\partial X / \partial P]$.
3. Compute the intermediate values:
   a) $U = \sum_i A_i^T \Sigma_{x_i}^{-1} A_i$
   b) $V = \text{diag}(V_1, ..., V_n)$ where $V_i = B_i^T \Sigma_{x_i}^{-1} B_i$
   c) $W = [W_1, W_2, ..., W_n]$ where $W_i = A_i^T \Sigma_{x_i}^{-1} B_i$
   d) $\epsilon_A = \sum A_i^T \Sigma_{x_i}^{-1} \epsilon_i$
   e) $\epsilon_B = (\epsilon_{B_1}^T, ..., \epsilon_{B_n}^T)^T$ where $\epsilon_{B_i} = B_i^T \Sigma_{x_i}^{-1} \epsilon_i$
   f) $Y_i = W_i V_i$
4. Augment U and V by multiplying their diagonal elements by $1 + \lambda$
5. Compute δ_a from the equation

$$(U^* - \sum_i Y_i W_i^T) \delta_\alpha = \epsilon_A - \sum Y_i \epsilon_{B_i}$$

6. Compute each δ_{b_i} in turn from the equation

$$\delta_{b_i} = V_i^{*-1} (\epsilon_{B_i} - W_i^T \delta_a)$$

7. Update the parameter vector $\mathbf{P}$ by adding the incremental vector $(\delta_a^T, \delta_b^T)^T$ and compute the new error vector.
8. If this vector is less than the previous error vecotor, either accept the parameters and diminish the value of λ by a factor of 10 and repeat from step 2 or terminate.
9. If the new error is greater than previously then revert to the old parameter values, increase the value of λ by a factor of 10 and repeat from step 4.

The method described by the authors of [20] for determining an estimate for F which with a minimum cost function is a non-linear method known as the Levenberg-Marquardt (LM) algorithm originally described in [30] and [35]. This algorithm is relatively complex and a detailed description and analysis is outside of the scope of this work, however it is useful in a variety of computer vision tasks. Briefly, in this application it is used as following Algorithm 4.2, as described in ([20] pp. 609-610).

Camera matrices $P = [I|0]$ and $P' = [X|t]$ are defined. In addition 3D points M_i are defined. These can be deduced with an initial estimation of F and point correspondances using a linear method. As shown by Equation 2.3 in Section 2.1.2 , a 3D point M maps to a 2D camera point m with the relationship $m = PM$. Thus, for the corresponding point $m' = P'M$. P' and the points M_i are varied so as to minimize the error expression. The parameter vector $\boldsymbol{P}$ is described as $\boldsymbol{P} = (a^T, b_1^T, ..., b_n^T)^T$ where $a = p'$ is made up of the entries for camera matrix P' and $b_i = (X_i, Y_i, T_i)$ is a 3 vector parameterizing the i-th 3D point $(X_i, Y_i, 1, T_i)$. The measurement vector $\boldsymbol{X}$ is described as $\boldsymbol{X} = (M_1^T, M_2^T, ..., M_n^T)^T$ where each $M_i = (m_i^T, m_i'^T)^T$, the measured location of the i-the point. Algorithm 4.2 is then applied. The winning parameters $\boldsymbol{P}$ are then used to compute P' and M from which the optimal value of F can be calculated.

4.3 The Use of Orientation Sensors in The Process of Determining Epipolar Geometry

So far, an overview of the capabilities of orientation sensors have been presented, alongside the standard methods of determining the epipolar geometry of a scene from two 2D images. An active area of research is currently dedicated to the process of augmenting this process with the use of data obtained by these additional hardware units. It is this this research area that this work intends to build upon.

One of the most significant issues in the process of determining scene geometry is the process of determining robustly point correspondences [41][43][58][15]. It should be clear at this point that an orientation sensor would be useful in this problem as the data provided by it can be used to determine the extrinsic parameters of a camera and therefore reduce the amount of point correspondences necessary to determine the fundamental matrix [41]. By reducing the amount of point correspondences

necessary, it follows that there should be a smaller likelihood of erroneous point matches causing an unreliable result.

In [41], Okatani and Deguchi present a method of determining the fundamental matrix of two images from a calibrated camera equipped with an orientation sensor. The orientation sensor provides details of the rotation of the camera between capturing two images. As described in Section 2.2.3 and Section 4.2.3, the relationship between two homogenous matching points x,x' from two images will be $x'Fx = 0$ where F is the fundamental matrix. Where R and t represent respectively the rotation and translation of the camera between photographs, and A and A' are the 3×3 matrices containing the cameras' intrinsic matrices, [41] state that $x' \propto A'[R|t]X$, and thus, the fundamental matrix F is given as

$$F = A'^{-T} T_x R A^{-1} \tag{4.5}$$

where T_x is a matrix such that $T_x p = t \times p$ for an arbitrary 3-vector p. If the intrinsic parameters are known, and also rotation R between the two views, 4.5 can be re-written as a constraint on the unknown translation t in the following form [41]:

$$a(x, x'; A, A', R)^T t = 0 \tag{4.6}$$

The scale of t cannot be determined due to scaling ambiguity, and so t has only two degrees of freedom. By imposing an additional constraint, $|t| = 1$, t can be determined from only two matches of points. By using an orientation sensor, the images can be rectified easily by way of the transforming them by the formula defined in 4.3.

In order to determine these two points, and therefore the translation and fundamental matrix, Okatani and Deguchi use a 'voting' method in order to determine point matches. In this process, a Harris corner detector is used to detect feature points in both images. Given a correct pair of matches, there is one equation for the unknown t, as is shown by Equation 4.6. Because of the scaling ambiguity - t is described in [41] as $t = [1, t_2, t_3]^T$. Therefore, for two points, x and x', the equation for t will draw a line in the t_2, t_3 plane - otherwise known as the 'voting space'. If all points are compared with one another in this way, then it is possible to detect peaks (the points at which most lines will cross) and the point at which the most lines

will cross will be located at the coordinate (t_2, t_3) - which will be the true values for t_2, t_3 of t. Results obtained by Okatani and Deguchi in [41] show that there is significant improvement in using this method of determining the necessary point correspondences, and thus determining the Fundamental Matrix over using random sampling without the use of an orientation sensor.

5 Proposed Method

In this work, we aim to compare the methods of determining the Epipolar Geometry of a scene presented in previous sections, along with a novel approach. In this chapter, an overview of the selected methods are presented.

5.1 Methodology

In this work, tests will be carried out on a set of image pairs over three different scenes. These scenes are designed to provide a variety of conditions for the algorithms to work on. Image Set 1 Consists of a highly repetitive pattern and a relatively small rotation and translation between images. It is expected that this will provide many good feature points to be detected by all algorithms, however the repetitive nature of the image is expected to present a challenge for the original algorithm that does not use an orientation sensor - and hence has to perform the search for matching features over the entire image - to find a set of correct matches. The second scene is a low-light scene that may cause for only a small number of features to be detected, and so it is likely that data from the orientation sensor will be needed in the computation of the Fundamental Matrix. The third scene consists of many easily detectable feature points, however also has a large rotation and translation magnitude between images. It is expected that all methods should compute a satisfactory value for the Fundamental Matrix, and is included in order to test the accuracy of the different methods.

The process of testing the images and performing the experimentation was as follows:

- The methods were implemented in and some cases tested with synthetic data
- The camera was callibrated using a method based on [54] and implemented by a third party in [5]. The camera used for each image set was the same, and the focal length, and other parameters (image size, ISO Rating) remained constant throughout.

- The results were calculated and recorded - and evaluated using the method described in Section 7.5.

5.2 Harris - Normalised Cross Correlation - RANSAC Method

In this approach, the Epipolar Geometry of a scene shall be determined using the image registration methods described in Section 4.2, specifically the following algorithm:

1. Feature points from both images are extracted using a Harris Corner Detector [19].
2. Candidate matches are created using the process of Normalised Cross Correlation [57]
3. Valid match pairs from these candidates are obtained using the RANSAC algorithm, with hypotheses for the correct fundamental matrix obtained using the 'Normalized 8 Point Algorithm'.
4. The winning hypothesis from the RANSAC stage is presented as the Fundamental Matrix for the image pair.

5.2.1 Feature Point Extraction

Using the Harris operator described in [19] a set of corner points from the image will be identified. A window around each point shall then be created, and this window will be used as the feature for matching in the normalised cross correlation step. This stage of the process corresponds to the feature detection stage as described in Section 4.2.

5.2.2 Feature Point Matching

In this implementation, prior to the use of RANSAC to determine a transform model, speculative matches are created using the normalised cross correlation algorithm described in Section 4.2.2. This will produce a map of potential matches, which

can be a one-to-many relation, i.e. A point in one image can have more than one potential match in the other image. This stage of the algorithm corresponds to the Feature Matching stage of the Image Registration Process described in Section 4.2.

5.2.3 Estimation of the Fundamental Matrix with RANSAC

In this stage, a set of potential point matches are randomly sampled - and used to produce an estimation of the Fundamental Matrix using the Normalised 8-Point Algorithm. In order to determine a consensous as to the most suitable candidate Fundamental matrix, *all* candidate point matches (x corresponds to x') are then sampled, and the candidate for F which produces the most matches for the equation $x^T F x' < t$ is taken to be the winning value for F, where t is a threshold parameter.

5.3 Okatani & Deguchi's Method with an Orientation Sensor

In this method, the Fundamental matrix will be calculated using the algorithm presented in [41] and described in Section 4.3. At the time of image capture, the orientation of the camera will be recorded using an orientation sensor mounted on top of the camera - specifically, a CH Robotics UM6 Orientation sensor, detailed in [9]. This sensor will be rigidly connected to the camera, and electronically synchronised to the shutter release as described in the following section. As this sensor gives a value for the camera's orientation described in the 'North-Up-Down' frame, where rotation around the vertical axis (yaw) is described with reference to magnetic north - a delta is taken for each reading over the set of two photographs, and this is used to compute the rotation matrix corresponding to the rotation between each image. Further details of the implementation of this method are given in the following section.

5.4 Modified Okatani & Deguchi

The process introduced by Okatani & Deguchi in [41], and subsequently developed in [28] to include more than two images, allows for the computation of the Fundamental

Matrix using only two correct point matches from an image by determining the translation across two images - computed up to a scale factor. Okatani & Deguchi use a method of voting in order to determine this translation. However, this method of determining the Fundamental Matrix relies on the use of data obtained from the orientation sensor directly in the calculation for determining the fundamental matrix. For reasons mentioned in Section 3.2, it can be assumed that this sensor may produce some error, particularly due to the electromechanical noise introduced by the workings of a camera.

This method explores whether methods used in traditional computer vision techniques, and in particular, Random Sample and Consensus may also produce robust results. In [41], Okatani & Deguchi compare their method of using a voting system to determine the translation of the camera to using RANSAC, and encodes the rotation directly from the orientation sensor in the Fundamental Matrix, and determine that the voting mechanism produces favorable results. In this method, this work will explore the results obtained by using only the point match candidates obtained by the restriction of a search along horizontal lines (with a vertical threshold) across the rectified images, and the RANSAC method of determining the fundamental matrix from this. Unfortunately, this will have the implication that, as with the current traditional methods of computing fundamental matrix, at least 8 correct point matches will need to be found. This process is described in detail in Algorithm 5.1

It is expected that this will give more precise results compared to the Harris-Normalised Cross Correlation-RANSAC method and Okatani and Deguchi method, however may be less robust than the Okatani and Deguchi method as more point correspondences will be required to compute a Fundamental Matrix. It is however expected that it will still be more robust than the Harris-Normalised Cross Correlation-RANSAC method as by restricting point match searching to horizontal lines in a rectified image, there will be a smaller chance of outlying point matches being used in the computation of the Fundamental Matrix.

5.5 Gold Standard (Ground Truth) Method

For all images pairs, the 'Gold Standard' algorithm described in Section 4.2.5 will be used to calculate a value for the Fundamental Matrix. As it is expected that

Algorithm 5.1 Proposed Method for Determining Fundamental Matrix based on modifications to [41]

1. Record the Roll-Pitch-Yaw orientaion of the camera from the orientation sensor at the time of each photo capture and hence the rotation, R, between the two images.

2. Compute a rectification Homography H where $H = BR'A^{-1}$ where B is the camera matrix for the rectified image, R' is a rotation of the same axis as R but with half the magnitude as R and A is the intrisic parameters of the camera. As per [41]

3. Run the Harris Corner detection algorithm as described in Section 4.2.1.1 on both images.

4. Rectify both images onto a common plane using the homography computed in item 2. Transform the corners detected in item 3 by this homography.

5. Extract features surrounding the corner points in both images, and perform Normalised Cross correlation described in Section 4.2.2 on features in the corresponding image - limiting the search area to points with the same vertical position (within a threshold t)

6. Transform candidate point matches back to their original positions using the inverse of H.

7. Apply the RANSAC based method for computing F (described in Section 4.2.3) to these point pairs.

the automated matching method used to detect point correspondances may not produce the required 8 point matches necessary for this algorithm to compute a match, it will be run upon a set of manually selected known correspondances to produce an estimate that will be taken as the 'ground truth' value for F. This will allow the algorithms to be tested against an optimum value acheivable for F and thus provide a benchmark for the various algorithms evaluation. Furthermore, in situations where 8 or more point correspondances are found by automated methods, these will be used as input into the Gold Standard algorithm in order to further evaluate the effectiveness of automated-matching techniques.

.

6 Implementation

In this chapter, an overview of how the necessary hardware and software systems were developed for testing is presented.

6.1 Orientation Sensor and Camera

As mentioned previously, the orientation sensor used throughout this work was a CH Robotics UM6 sensor. This is a powerful device that contains a MEMS accelerometer, gyroscope and magnetometer for each axis (x, y & z) alongside a 32 bit processor that is able to process the data from these systems. Running on the processor is an Extended Kalman Filter - used to calculate the orientation in North, Up, Down relative Euler angles and Quaternions, and the device is able to output the orientation and other information over a serial interface. North, Up, Down refers to the reference vectors used to calculate the orientation of the sensor - with the X axis aligned to magnetic north at the origin [9]. A mount was made for this sensor in order to rigidly connect it to the camera - using the standard 'hot-shoe' that is found in nearly all cameras to mount an external flash, as shown in Fig. 6.1.

6.1.1 Synchronisation

During development of the orientation sensor mount to the camera, it was discovered that on shutter release, two terminals on the hot-shoe will become connected. If a small current is supplied across these two, the circuit will when the shutter is open. A very simple circuit was constructed in order to hold the 'Clear To Send' pin on a USB to RS232 serial adapter at logic level high (5v) using a resistor. Whilst the shutter is open, this pin is then tied to logic level low (0v). This can be detected by the host computer application, and at this point, the sensor can be queried for the its orientation. It is true that slight movement of the camera during the shutter's

Figure 6.1: The Camera with an Orientation Sensor Mounted on the 'Hot-Shoe'

opening will not be recorded using this method, however for the purposes of this work it is assumed that a reasonably fast shutter speed would be used in order to photograph a well lit subject, and that motion during exposure shall be avoided. It is also not known how much of a delay takes place between shutter opening and the camera's sensor recording an image - and thus the amount of time between orientation being sampled and an image being recorded, although it is assumed that this will likely be a very short amount of time and the affect of this on the results to be negligible.

6.1.2 Data Retrieval

In order to read information from the camera, a simple C++ application was developed in order to read the state of the serial port, and issue and read commands to and from the device. By using a USB to serial converter cable, it was also possible to obtain power for the device from the host system, further simplifying the design and operation of the unit. One disadvantage of this method is that the rig must remain connected to a computer during use.

6.1.3 Calibration of the Orientation Sensor

As the sensor computes its orientation based on the use of magnetometers, it is necessary to take into account the effect of the surface it will be mounted on and the possible interference that may take place, as discussed in Section 3.1.4. Particular note should be given to the fact that an SLR camera contains moving parts that are operated electronically through the use of electromagnets and motors that will also have an effect on the magnetometers during operation. The manufacturer of the orientation sensor provides an application that can be used in order to calibrate the magnetometer when it is mounted on the camera. Unfortunately, it is not possible to calibrate the orientation sensor for the effects of the electromagnetic field created by the motor in the shutter release mechanism (as this will not be constant). This limitation should be taken into account during the analysis stage of the results obtained.

6.2 Computer Vision Methodologies and Translation Estimation

The methods described in the previous chapters are all implemented in the MATLAB environment. Originally, the intention was to implement these methods in the C programming language in order to take advantage of the graphics libraries such as OpenGL which may have lead to some performance gains, however, in the time available, this would not have been feasible - as MATLAB contains an extensive array of pre-built utilities, such as the Image Processing and Computer Vision 'Toolboxes' [39][38] that allow for many of the algorithms introduced in previous chapters (Corner Recognition, RANSAC algorithm etc) to be implemented with ease.

During implementation, there were a number of challenges. One significant issue was related to image size and the amount of computation required to search an image for points and potential matches. It should be noted that within a large amount of literature, images are often down-scaled to a size of around 900 by 600 pixels, whilst modern cameras are capable of capturing images with resolutions of upwards of 3,888 by 2,592 pixels [6]. Searching for feature points on images of this size takes a significant amount of time and memory[34], especially when implemented

in a technology such as MATLAB which has been designed to be optimised for prototyping speed as opposed to execution efficiency. This meant that throughout our experimentation - images were down-scaled to 972 by 648 pixels. It would be an interesting potential future research project to determine and quantify the effect of this on the quality of detected and matched features. Furthermore, none of the algorithms that are used in this work exploit the use of colour in the images. More modern methods of image registration and feature detection and matching (such as [36]) make extensive use of colour and texture from an image. For this reason, images were also converted into grayscale before being processed. Again, it would make for interesting discussion and further research to analyse improvements to the various algorithms by using a feature detector and matcher that exploited the information available from the image in terms of colour.

The implementation for the Voting Algorithm as described in [41] was implemented in MATLAB using a system based on an Accumulator Array - in a similar fashion to how the 'Hough Transform' (a feature detection algorithm that aims to extract features from objects falling within a certain class of shape by the use of a voting procedure - further described in[44] and [17]). This required some effort, however conveniently, the implementation could be tested with known synthetic point correspondences and also a known rotation matrix and translation vector, whilst also allowing for the results to be visualised, as show in Fig. 6.2, whilst developing and testing the algorithm.

The 'Gold Standard Method' for calculating ground truths as described in Section 4.2.5 is relatively complex, however many reliable implementations have been made readily available. In this work, the implementation by the authors of the method (in [20]) available from [1] and using functions from [7] is used to compute values for F.

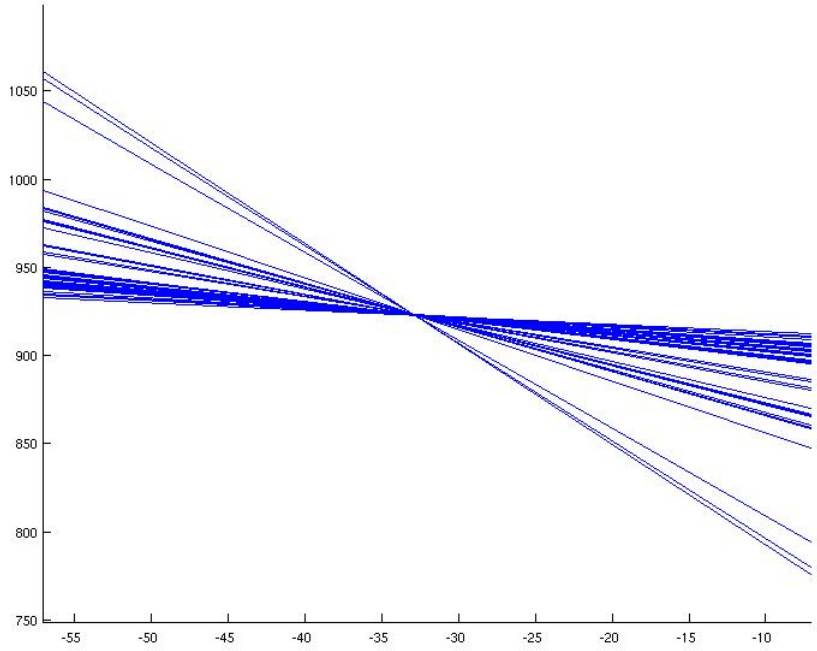


Figure 6.2: Visualisation of results obtained for calculating $t_1$and t_2 (x and y axis respectively) whilst implementing the voting algorithm with a set of synthetic points and known rotation and translation.

7 Results

In this chapter, the results and methodologies for testing the approaches described in chapter 5 and implemented as described in chapter 6 are presented.

7.1 Method of Testing

The 3 methods described in chapter 5 are implemented on three sets of two 2D images of a 3D scene. Between each image of a scene, a rotation and translation of unknown amount will have taken place. The camera used is calibrated according to Section 2.1.3 and the intrinsic parameters are known. The 3 sets of images are designed to provide differing types of images and different conditions for the methodologies on test, shown in Fig. 7.1:

Figure 7.1: Set of images used for testing. Left to Right: Set 1, 2 and 3

- **Image set 1 - Checkerboard** Consists of a small change in orientation over a surface that has a number of repetitive but easy to detect feature points.

- **Image set 2 - Chairs** Consists of a reasonable change in orientation over a scene that has poor lighting and difficult to detect points (few corners and several rounded edges)
- **Image set 3 - Outdoors** Consists of a large change in camera orientation and reasonable texture to allow for the identification of many feature points in good lighting.

Although these images were taken with a colour camera, they were converted to grayscale, and resized in order to perform testing on. Where arbitrary constants are used, such as in the threshold indicator in the Harris corner detection algorithm [19], these are selected manually in order to maximise the number of 'good' corners, i.e. clearly defined features present in both sets of images, detected by the algorithm. Whilst these will vary over the different images, they shall remain the same for each set of images throughout testing each of the three methods. As all methods will use a set of point correspondences, Fig. 7.2 shows the results obtained of applying a Harris corner detector to the image pairs.

Figure 7.2: The image set with feature points highlighted as found by a Harris corner detector

In order to evaluate the results of the differing methods - we use the Fundamental matrix produced to calculate the Epipolar lines - Fm_r for each pair of point m_r in the 'right' image. As stated previously, the corresponding point in the 'left' image should lie along this line. In reality, it is expected that there would be some error and the point will in fact lie a short distance away from this line. The Root Mean Square distance from the lines to the corresponding point is recorded, and this value is used in order to compare the success of the 3 methods of computing the

Fundamental matrix. In order to perform this evaluation, the root mean squared distance is calculated for a selection of manually matched points across both images.

7.2 Image set 1

7.2.1 Harris - Normalised Cross Correlation - RANSAC Method

Despite there being approximately 80 points available to the system in each image, it was found that when un-rectified and hence un-constrained to the epipolar lines, only two candidate matches for points were produced. It is clear from looking at the image (Fig. 7.3) that neither of these points are correct matches and therefore that it will not be possible to solve the epipolar geometry using any method - as a minimum of 8 correct point matches are required for the fundamental matrix to be calculated.

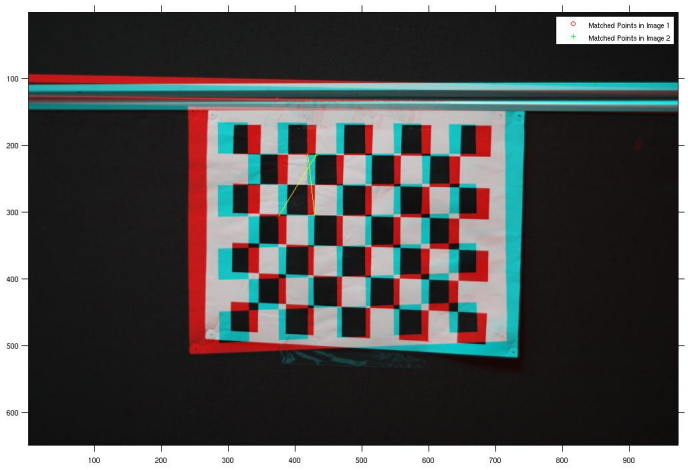


Figure 7.3: The candidate point matches produced by the Normalized Cross correlation of features detected. No correct point matches are found.

7.2.2 Okatani & Deguchi Method with Orientation Sensor

In this method, the images are first rectified using the data obtained from the orientation sensor as shown in Fig. 7.4. This should mean that any point match candidates will lie along the same horizontal line, and it is with this constraint that matches are searched for using a normalized cross correlation algorithm, with

constants identical to the ones used in the previous test. Point match candidates from this stage are shown in Fig. 7.5.

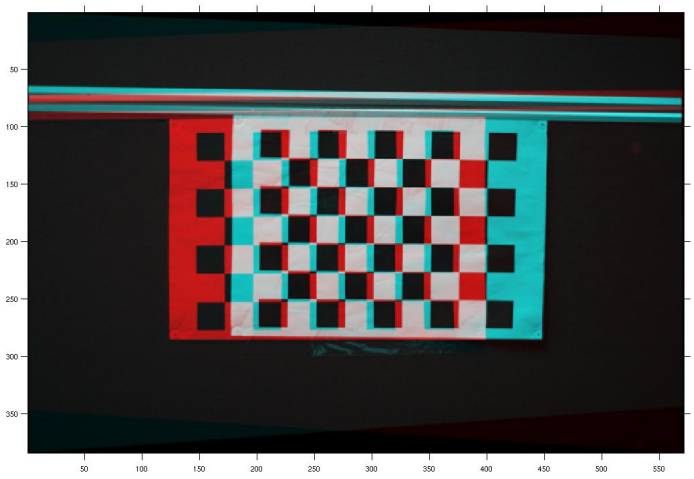


Figure 7.4: Image Set 1 Rectified with Rotation information obtained from the Orientation Sensor

It is clear from this stage that the method of using an IMU is a much more effective way of deducing candidate point matches. From the large number of matches available, it should be easy to compute a reliable estimate of the fundamental matrix. These sets of correspondences are then used in the estimation of the translation between the two rectified images, which is estimated to be $[-56.9281, 6.6801, 1]^T$ in pixels. Using the equation described in [41]. The fundamental matrix produced for this image is found to be (to 4 decimal places):

$$F = \begin{bmatrix} 0.0000 & -0.0000 & 0.0118 \\ 0.0000 & 0.0000 & -0.0144 \\ -0.0141 & 0.0129 & 0.9996 \end{bmatrix} \tag{7.1}$$

Furthermore, analysis of the point matches shows that out of the match candidates shown in Fig. 7.5, 65.4% of matches were to correct points identified manually in the image.

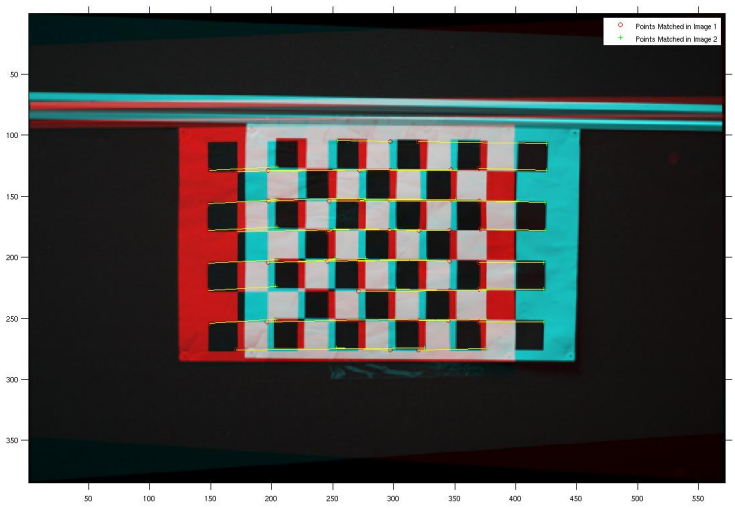


Figure 7.5: Point Match Candidates Determined from the Epipolar Constraint of the Rectified Images

7.2.3 Modified Okatani & Deguchi Method

In this method the point correspondence candidates are rotated back to the original image by inverting the rotation applied in order to rectify the images originally. Using a RANSAC algorithm on these points produces the Fundamental matrix:

$$F = \begin{bmatrix} -0.0000 & -0.0000 & -0.0003 \\ 0.0000 & 0.0000 & -0.0325 \\ -0.0014 & 0.0275 & 0.9991 \end{bmatrix} \tag{7.2}$$

7.2.4 Gold Standard (Ground Truth) Method

Using a set 8 of manually selected known point correspondences, the Fundamental Matrix computed using the Gold Standard method was calculated to be:

$$F = \begin{bmatrix} 1.156\text{e}-6 & -2.521\text{e}-5 & 0.007 \\ 2.422\text{e}-5 & 4.049\text{e}-7 & -0.015 \\ -0.007 & 0.015 & 0.2619 \end{bmatrix} \tag{7.3}$$

7.3 Image Set 2

In this image set, the aim is to provide a challenging situation for the corner detection and correlation algorithms to operate in. The photographs were taken in low light and there were few textured surfaces and clear corners in the image. The corner detection algorithm was able to identify 74 corners in the first image and 67 in the second.

7.3.1 Harris - Normalised Cross Correlation - RANSAC Method

Despite there being a reasonable number of points detected in this image, using the unconstrained normalized cross correlation method was only able to match one set of points correctly as shown in Fig. 7.6. Similarly to the first image set, this is not sufficient to compute the fundamental matrix from.

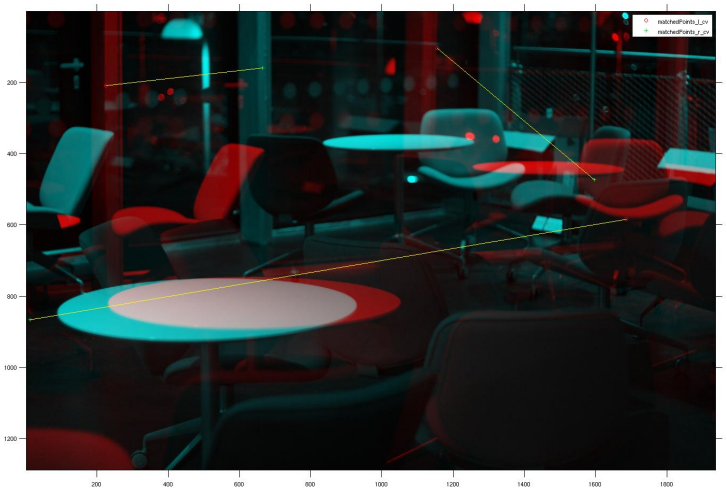


Figure 7.6: The Results of Unconstrained Normalised Cross Correlation Matching of Point Pairs

7.3.2 Okatani & Deguchi Method

By using the rotation information obtained from the orientation sensor, the images may be rectified onto a common plane, as shown in Fig. 7.7.

Constraining the point matching algorithm to produce candidate matches only along horizontal lines produces the result shown in Fig. 7.8.

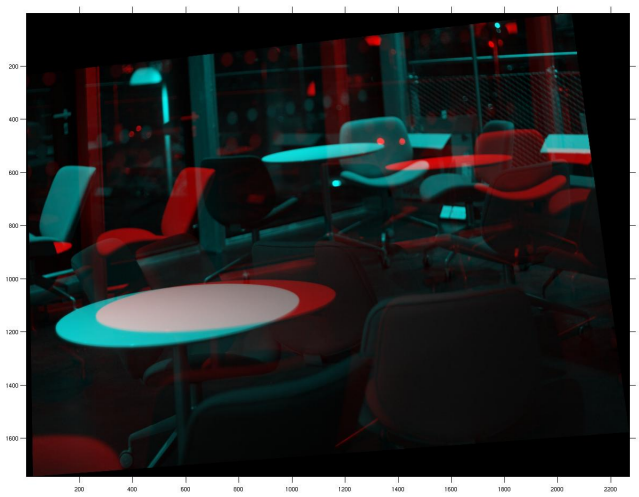


Figure 7.7: The Result obtained from Rectifying The Images using Data Obtained from the Orientation Sensor

This result produces 9 candidate matches, of which 4 are determined to be correct, a score of 44%. This is sufficient to determine the fundamental matrix for the two images (after the point matches have been rotated back to the original image planes) by the voting method to determine the translation between points in the rectified images. Thus, the fundamental matrix was determined to be (to 4 decimal places):

$$F = \begin{bmatrix} -0.0000 & 0.0000 & -0.0013 \\ -0.0000 & -0.0000 & 0.0486 \\ 0.0009 & -0.0314 & 0.9983 \end{bmatrix} \tag{7.4}$$

As this image only contains 4 correct matches, the novel algorithm is also unable to be completed on this image.

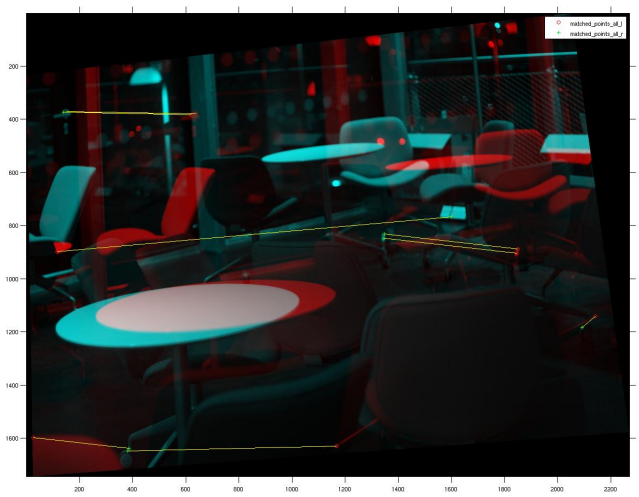


Figure 7.8: Point Match Candidates Determined from the Epipolar Constraint of the Rectified Images

7.3.3 Gold Standard (Ground Truth) Method

Using a set 8 of manually selected known point correspondences, the Fundamental Matrix computed using the Gold Standard method was calculated to be:

$$F = \begin{bmatrix} -5.148\text{e}-7 & -2.4101\text{e}-6 & 0.0039 \\ 1.297\text{e}-6 & 5.9245\text{e}-7 & 0.0013 \\ -0.0030 & -0.0033 & 0.9268 \end{bmatrix} \tag{7.5}$$

7.4 Image set 3

In this set of images, a large number (200) of points were identified from an outdoor scene containing a large amount of texture and feature points.

7.4.1 Harris - Normalised Cross Correlation - RANSAC Method

Shown in Fig. 7.9 is the result of point matching using an unconstrained normalised cross correlation method over all the points on both un-rectified images.

This method produces 11 candidate matches, 8 of which are correct (87.5%). This

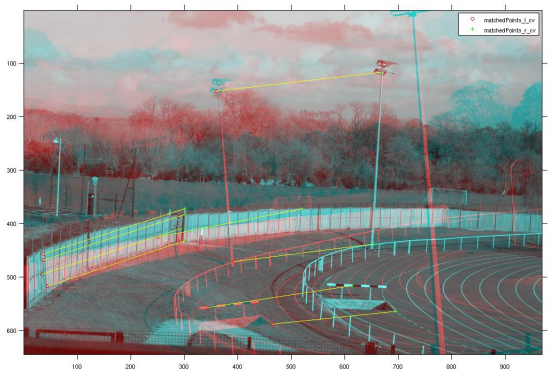


Figure 7.9: The Results of Unconstrained Normalised Cross Correlation Matching of Point Pairs

is sufficient to produce an estimate for the Fundamental Matrix of:

$$F = \begin{bmatrix} 0.0000 & 0.0000 & -0.0020 \\ -0.0000 & 0.0000 & 0.0042 \\ 0.0010 & -0.0054 & 1.0000 \end{bmatrix} \tag{7.6}$$

7.4.2 Okatani & Deguchi Method

As for the previous image sets, the image pair is rectified prior to point correspondences being searched for, the process being constrained to horizontal lines. Fig. 7.10 shows the result of the image rectification process with data obtained from the orientation sensor.

Fig. 7.11 shows the point correspondence candidates produced by searching from these images.

Of these 8 candidate matches, all appear to be correct (100%). This allows for the translation to be calculated as $[298.5897, -11.8101, 1]^T$ pixels. Thus, combined with the rotation obtained from the orientation sensor, the fundamental matrix is calculated to be:

$$F = \begin{bmatrix} 0.0000 & 0.0000 & -0.0024 \\ -0.0000 & 0.0000 & 0.0044 \\ 0.0006 & -0.0074 & 1.000 \end{bmatrix} \tag{7.7}$$

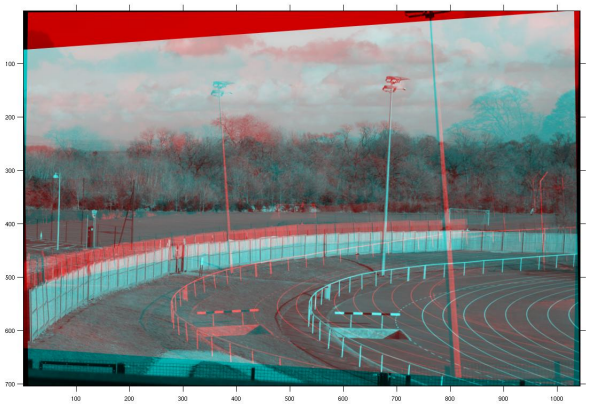


Figure 7.10: The Result obtained from Rectifying The Images using Data Obtained from the Orientation Sensor

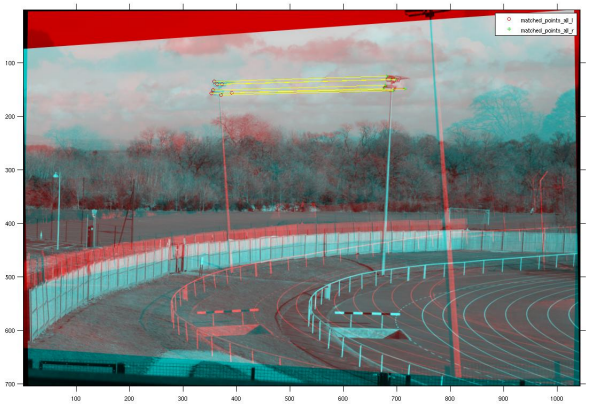


Figure 7.11: Point Match Candidates Determined from the Epipolar Constraint of the Rectified Images

7.4.3 Modified Okatani & Deguchi Method

As all 8 points appear to be correct, it would be expected that a good estimate of the fundamental matrix can be computed using the RANSAC approach from the point matches determined from the rectified image. The Fundamental matrix using this method was computed to be:

$$F = \begin{bmatrix} 0.0000 & -0.0000 & -0.0029 \\ 0.0000 & 0.0000 & 0.0041 \\ 0.0028 & 0.0012 & 1.0000 \end{bmatrix} \tag{7.8}$$

7.4.4 Gold Standard (Ground Truth) Method

Using a set of 8 manually selected points, the following value for F was obtained:

$$F = \begin{bmatrix} 3.716\text{e}-7 & 6.053\text{e}-7 & -0.0019 \\ -1.778\text{e}-6 & 4.112\text{e}-7 & 0.0060 \\ 0.0013 & -0.0062 & 0.9374 \end{bmatrix} \tag{7.9}$$

As the feature matching algorithm was able to determine 8 potential point matches from this image pair, the Gold Standard algorithm could be run on the resulting image pairs, resulting in the following vale for F.

$$F = \begin{bmatrix} -4.6381\text{e}-7 & -7.235\text{e}-6 & 3.5116\text{e}-4 \\ 6.9183\text{e}-6 & -1.4219\text{e}-6 & 0.0015 \\ 9.9176\text{e}-5 & 0.0012 & -0.0508 \end{bmatrix} \tag{7.10}$$

7.5 Evaluation and Discussion

Presented here (Fig. 7.12, Fig. 7.13 and Fig. 7.14) are graphs depicting the RMS values for distance from a point to it's corresponding epipolar line. (For space purposes, The Harris-Normalised Cross Correlation - RANSAC method is referred to as the 'Standard' method)

These graphs and results indicate that the combined method of using orientation data in the feature matching stage provides good results as long as there is an adequate number of features extracted from the image in order to estimate the Fundamental Matrix using standard techniques. The results from this experimentation show that the method described by Okatani and Deguchi is successful in estimating a Fundamental Matrix in a wider variety of situations than any of the other methods - and therefore can certainly be considered to be a more robust method, however would also appear to suggest that, given sufficient feature points, the state of the art in computer vision techniques tends to produce higher quality results. Furthermore, in every case, using the 'Gold-Standard' technique appears to always produce better results, even in the case of image set 3 where enough points are found automatically for both the standard 8-point algorithm and the novel algorithm to operate, the

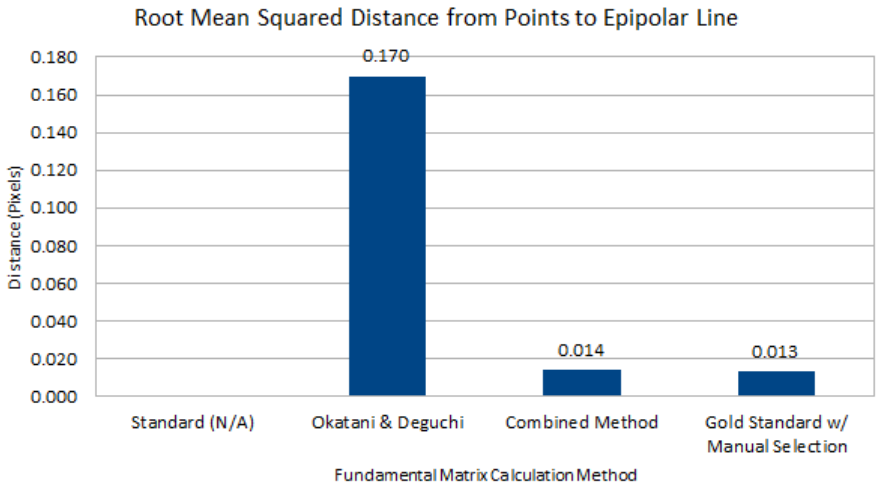


Figure 7.12: RMS Values for Distance from point to Epipolar Line - Image Set 1

'Gold Standard' algorithm produces a more accurate result. It would be interesting to investigate further what the potential results of modifying the novel method to estimate the epipolar geometry utilizing the gold standard approach instead of the standard 8-point method with its point matches.

The current state of the art in consumer-level MEMs orientation sensors is not sufficient for the readings to be considered noise-free, as discussed in chapter 3, and therefore it should be expected that noise and errors from the readings obtained from sensors will propagate into the calculation of the Fundamental Matrix if used directly. This is a conclusion that is supported by the results of this experiment. One anomaly is that in the third image set, it would be expected that the accuracy of the Fundamental Matrix would be higher - however this is not the case. This appears to stem from the fact that fewer points are matched after the image has been rectified. The reason for this is unclear, however, it is likely due to the re-projection of the images having an effect on the correspondence scores of the feature-points. This is a problem that is known in the literature, and more recent and advanced techniques exist that are able to more robustly match feature points that have undergone an affine transformation, such as the methods presented in [37] and [33]. This is also a significant limitation of the modified Okatani and Deguchi method - and is not present in their original algorithm. It would be worthwhile to repeat this experiment with more robust feature matching processes.

This project does have limitations, particularly that it has been carried out solely

on 'real' images, and it is therefore not possible to quantify any exact gains in performance of the original methods given precise inputs. It would be valuable for further work to be carried out in order to ascertain with greater certainty the advantages of the methods described here and how they could be improved on. These are discussed further in the following section.

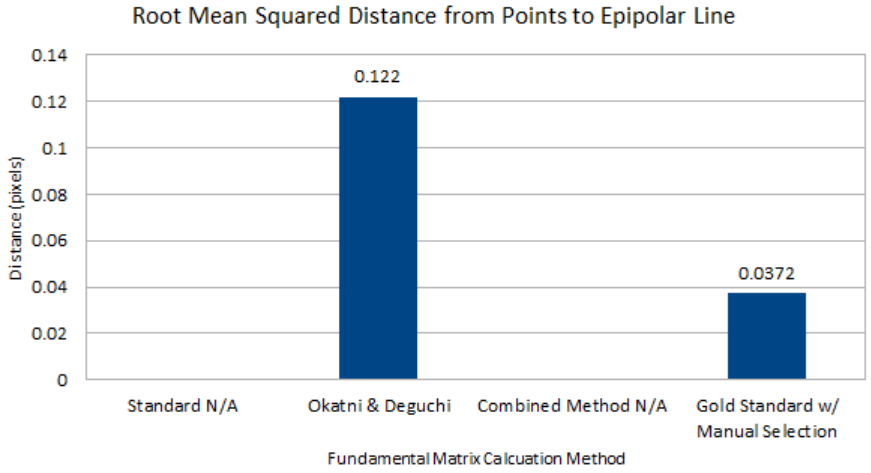


Figure 7.13: RMS Values for Distance from point to Epipolar Line - Image Set 2

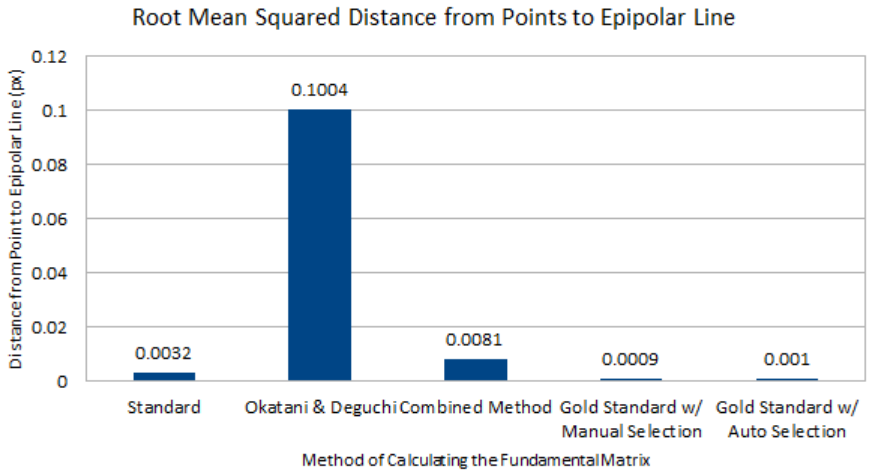


Figure 7.14: RMS Values for Distance from point to Epipolar Line - Image Set 3

8 Conclusions and Further Work

In this final chapter, an overview of the outcomes from this work is presented, along with suggestions for potential future work to be carried out on this subject.

8.1 Project Outcomes

This work has produced the following outcomes:

- A description and implementation of the current state of the art machine-vision techniques for determining stereo geometry from 2D images.
- Research was undertaken into micro-electromechanical sensor based orientation and movement sensors, and a system for interfacing a commercial device with a DSLR camera and computer was designed and built.
- A technique described in [41] was implemented in order to combine the use of inertial sensors and computer vision techniques.
- This technique was developed to use a larger set of potential point matches in order to calculate the Fundamental Matrix (and hence scene geometry) of a set of image pairs taken from significantly differing viewpoints.

This work has highlighted the use of data obtained from camera mounted orientation sensors as a valuable resource for the computation of 3D scene geometry. It has also highlighted the fact that it is still necessary to use a combination of techniques from both computer vision and additional hardware based systems in order to produce the best results. It is not yet practical for any sensor by itself to measure to a sufficient degree of accuracy the usually small movement between two camera viewpoints in order to reconstruct the scene on this information alone, and the field of combining visual and inertial sensors is currently a highly active area of research. Integrating data from a camera within the process of filtering the input from various sensors (for

example, using a Kalman filter as described in 3.15), in order to increase the accuracy of such Inertial Measurement Units, is a popular proposition. [26] [53] explore this area - with particular application to the field of Augmented Reality. Furthermore, as the costs of both high quality vision and inertial sensors continue to fall, many consumer grade devices are being equipped with such sensors as standard - which will continue to drive forward this research area.

With regards to the outcome of this work, and indeed others before it, it is clear however that there is a trade-off between the amount of points that need to be collected and the quality of the quality of the resulting calculation. One significant advantage of relying on data from the orientation sensor, as presented in [41] is that the number of correctly matched points needed to solve for the Fundamental matrix falls from the previous minimum of 8 to 2, which creates a significantly more robust system for determining the scene geometry of a particular scene - especially when such point matches that would be found using purely automated computer vision means. On the other hand, the use of an orientation sensor to help with the process of determining point matches appears to always increase the number of correct matches identified by the system. This is a useful result, as it shows that even the addition of a low-cost sensor with limited accuracy can have a positive effect on the process of automatically analysing data from an imaging device.

8.2 Further Work

This work concentrated largely on the use of orientation and image data gained from 'real' photographs, and relying on a number of factors and techniques (such as adequate camera calibration, correct determination of the camera's orientation and good selection of parameters for use in corner detection and feature matching) to produce results. It would be valuable to analyse this process numerically, for example, by projecting a set of points into a space using known rotation and translation parameters, and then adding noise to these readings. Alternatively, the photographs used could be taken of a known calibration target with feature locations and the translation and rotation of the camera known to a high degree of accuracy. It is possible to recover rotation and translation information from the Fundamental Matrix, and a comparison of values obtained from this could easily be performed against the known values.

Further research areas could also focus on the use of data from an orientation sensor in other ways - such as stabilisation of an image and also estimating magnitude and direction of movement during the time in which a shutter on a camera would be open, which could have applications to de-blurring processes. Other uses might include the segmentation of an image taken by a moving camera of a scene containing moving objects, and also in determining 'real-world' distances of objects or points in a scene.

As mentioned in Section 7.5, it was discovered that re-projecting the images causes a significant affine transformation of the feature points, which considerably reduces the accuracy of the feature matching process. More recent methods of feature matching are able to compensate for this problem ([33] and [37]), and so it would be of interest to repeat the experimentation described in this work with different feature matching processes in order to see what improvements could be made by their use, and how effectively they may be augmented with the use of an inertial sensor.

8.3 Final Conclusions

In conclusion, this work has highlighted the promissing contributions that the use of additional orientation sensing hardware alongside cameras and computer vision methodologies can have. However, it has been shown that there still remain reasonable sources of error in this process - and that the best and most reliable results are obtained when vision and inertial sensing are used to complement one-another in order to formulate a strong estimation of a scenes geometry.

Bibliography

[1] Omid Aghazadeh. Fundamental matrix computation. Online, May 2010.

[2] Naomi Allen. *MEMS sensors for wall shear stress and flow vector measurement.* PhD thesis, Durham University, 2008.

[3] Apple Inc. iphone 4s Technical Specifications, 2012.

[4] B. Barshan and H.F. Durrant-Whyte. Inertial navigation systems for mobile robots. *Robotics and Automation, IEEE Transactions on*, 11(3):328 –342, jun 1995.

[5] Jean-Yves Bouguet. Camera calibration toolbox for matlab, July 2010.

[6] Canon Inc. Canon EOS 400D digital slr camera. Online, 2012.

[7] David Capel, Andrew Fitzgibbon, Peter Kovesi, Tomas Werner, Yoni Wexler, and Andrew Zisserman. Matlab functions for multiple view geometry. Online, Nov 2012.

[8] M.J Caruso. Applications of magnetoresistive sensors in navigation systems. *Sensors and Actuators*, SAE SP-1220:15–21, Feb 1997.

[9] CH Robotics. Um6 ultra-minature orientation sensor datasheet, Feb 2012.

[10] Shane Colton. The balance filter: a simple solution for integrating accelerometer and gyroscope measurements for a balancing platform, 2007.

[11] Digi-Key Corporation. Stmicroelectronics mems gyroscopes, 2012.

[12] Lei Fang, P.J. Antsaklis, L.A. Montestruque, M.B. McMickell, M. Lemmon, Yashan Sun, Hui Fang, I. Koutroulis, M. Haenggi, Min Xie, and Xiaojuan Xie. Design of a wireless assisted pedestrian dead reckoning system - the navmote experience. *Instrumentation and Measurement, IEEE Transactions on*, 54(6):2342 – 2358, dec. 2005.

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

[14] David Forsyth and Jean Ponce. *Computer Vision A Modern Approach*. Pearson, 2012.

[15] David Forsyth and Jean Ponce. *Computer Vision A Modern Approach*, chapter 10.4 Robustness, pages 329–336. Pearson, 2012.

[16] Freescale Semiconductor. Mma7361l technical data. `http://www.freescale.com/files/sensors/doc/data_sheet/MMA7361L.pdf`, 2008.

[17] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[18] G. Hanning, N. Forslow, P. Forssen, E. Ringaby, D. Tornqvist, and J. Callmer. Stabilizing cell phone video using inertial measurement sensors. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1 –8, nov. 2011.

[19] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–152, 1988.

[20] R Hartley and A Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[21] R.I. Hartley. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(6):580 –593, jun 1997.

[22] Richard I. Hartley. Theory and practice of projective rectification. *International Journal of Computer Vision*, 35:115–127, 1999. 10.1023/A:1008115206617.

[23] Jie Yu Andrew Gallagher Jiebo Luo, Dhiraj Joshi. Geotagging in multimedia and computer vision - a survey. *Multimedia Tools and Applications*, 51(1), Janurary 2011.

[24] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[25] Alexandre Karpenko, David Jacobs, Jongmin Baek, and Marc Levoy. Digital video stabilization and rolling shutter correction using gyroscopes. Technical Report CSTR 2011-03, Stanford University, Sept 2011.

[26] G.S.W. Klein and T.W. Drummond. Tightly integrated sensor fusion for robust visual tracking. *Image and Vision Computing*, 22:769–776, 2004.

[27] S. Kumar, C. Micheloni, C. Piciarelli, and G.L. Foresti. Stereo rectification of uncalibrated and heterogeneous images. *Pattern Recognition Letters*, 31(11):1445 – 1452, 2010.

[28] Martin Labrie and Patrick Hebert. Efficient camera motion and 3d recovery using an inertial sensor. Forth Canadian Conference on Computer and Robot Vision, 2007.

[29] J.E. Lenz. A review of magnetic sensors. *Proceedings of the IEEE*, 78(6):973 –989, jun 1990.

[30] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, II(2):164–168, 1944.

[31] H.C Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[32] Loughborough University. *An Introduction To MEMS*. Prime Faraday Technology Watch. PRIME Faraday Partnership, January 2002.

[33] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[34] P. Mainali, Qiong Yang, G. Lafruit, R. Lauwereins, and L.V. Gool. Lococo: low complexity corner detector. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 810–813, 2010.

[35] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):pp. 431–441, 1963.

[36] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, May 2004.

[37] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *In British Machine Vision Conference*, pages 384–393, 2002.

[38] MathWorks. Computer vision toolbox. website, 2013.

[39] MathWorks. Image processing toolbox. website, 2013.

[40] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. 1979.

[41] T. Okatani and K. Deguchi. Robust estimation of camera translation between two images using a camera with a 3d orientation sensor. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 275 – 278 vol.1, 2002.

[42] Oliver J. Woodman. An introduction to inertial navigation. Technical Report UCAM-CL-TR-696, University of Cambridge, August 2007.

[43] Joquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. A review of recent range image registration methods with accuracy evaluation. *Image*, May 2006.

[44] Linda Shapiro and George Stockman. *Computer Vision*. Prentice-Hall, 2001.

[45] ST Microelectronics. L3g4200d datasheet, December 2010.

[46] M. Stakkeland, G. Prytz, W.E. Booij, and S.T. Pedersen. Characterization of accelerometers using nonlinear kalman filters and position feedback. *Instrumentation and Measurement, IEEE Transactions on*, 56(6):2698 –2704, dec. 2007.

[47] D. Titterton and J. Weston. *Strapdown Inertial Navigation Technology*. The American Institute of Aeronautics and Astronautics, 2004.

[48] P. H. S. Torr and A. W. Fitzgibbon. Invariant fitting of two view geometry or in defiance of the 8 point algorithm. Technical report, British Machine Vision Conference, 2002.

[49] Alessandro Verri Trucco. *Introductory Techniques for 3D Computer Vision*. 1998.

[50] Yun Xiaoping, E.R. Bachmann, and R.B. McGhee. A simplified quaternion-based algorithm for orientation estimation from earth gravity and magnetic field measurements. *Instrumentation and Measurement, IEEE Transactions on*, 57(3):638 –650, march 2008.

[51] Gang Xu and Zhengyou Zhang. *Epipolar Geometry in Stereo Motion and Object Recognition*. Kluwer Academic Publishers, 1996.

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

[53] Suya You and Ulrich Neumann. Fusion of vision and gyro tracking for robust augmented reality registration, 2001.

[54] Zhengyou Zhang. A flexible new technique for camera calibration. Technical report, Microsoft Research Microsoft Corporation, 2000.

[55] Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, October 1995.

[56] Zhengyou Zhang and T. Kanade. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27:161–195, 1998.

[57] Feng Zhao, Qingming Huang, and Wen Gao. Image matching by normalized cross-correlation. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 2, page II, may 2006.

[58] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21:977–1000, 2003.