Imperial College London

Department of Computing

MEng Final Year Project

Motion stabilisation for video sequences

Author:
Oliver Wilkie

Supervisor:
Professor Daniel Rueckert

June 18, 2013

Abstract

Motion Stabilisation is the art of removing unwanted observable motion from a dynamic image sequence. Computer Vision techniques serve as a purely software based means of achieving motion stabilisation by applying a sequence of techniques including calculating an original global motion estimate using feature detection, feature tracking and outlier rejection. An update transformation is derived and applied to the original video sequence and a new stiller video is created.

Minimally Invasive Heart surgery is a modern approach to heart surgery whereby the surgeon operates on the patient remotely using robotic arms to perform delicate operations within the patient. Benefits of such endoscopic surgery include less scarring, lower rates of wound infection and a faster recovery time. As patient demand for such surgery increases, the technique still poses problems for the surgeon such as having only an indirect view of the field of surgery via a small camera.

When operating on the heart, traditionally a Heart Bypass machine is employed to still the heart to ease the job of the surgeon, however sometimes serious complications can arise as a result of using such machines. Beating Heart surgery requires the surgeon to operate on the heart whilst it is still beating which requires additional effort and concentration. Motion compensation of the beating heart has been investigated by several international research groups over recent years.

In this project, we investigate the application of a recent 2D motion stabilisation method to the problem of motion compensation in minimally invasive beating heart surgery. The investigation served to assess how suitable this method is when applied to a 3D deforming surface such as the heart.

For evaluation purposes, we apply the procedure to a video of a heart phantom as well as footage of a real moving heart and quantitatively and qualitatively measure the reduction in apparent motion in each video.

The project found that 2D methods are capable of compensating non-linear 3D deforming motions to an extent.

In our real heart video we found that our global motion estimate method was able to derive a decent global motion estimate despite the high level of specular reflection and pre-existing motion blur. Consequently, the method was able to compensate a substantial amount of respiratory motion in addition to a reduction in observable cardiac motion.

Acknowledgements

I would like to thank my supervisor Professor Daniel Rueckert for providing the idea behind this project and for his guidance and advice throughout its duration. I would like to thank my friends and peers who provided subjective measures during this project and who have made the past four years at Imperial College London a pleasure. I would also like to thank the lecturers at the Department of Computing for my rigorous yet rewarding four-year education.

Contents

Chapter 1

Introduction

In this chapter we outline the underlying motivation for the research undertaken in this project and the difficulties associated with the task at hand. We discuss the benefits of pursuing the task and what work was undertaken to make progress. This section ends with a general outline of the structure of the rest of the report.

1.1 Motivation

Every day, thousands of patients undergo operations on their heart. During cardiac surgery, stilling of the heart is typically required so that the surgeon can operate effectively[24]. Cardiopulmonary bypass machines are typically used to relieve the heart of it's job during surgery however stilling the heart and using such machines can lead to a range of long-term problems which can sometimes be fatal[30]. Although traditionally cardiac surgery is performed as 'open heart surgery', more and more surgeons are opting to perform minimally invasive beating-heart surgery whereby the patient is operated on whilst his heart is still moving.

Minimally-invasive surgery (also known as MIS, keyhole surgery or Laparoscopic surgery) is a modern surgical technique that allows a surgeon to operate on a patient's internal body without having to make large incisions in the skin[1].

So that the surgeon can see his work, a small camera called a laparoscope is inserted into the patient and relays back a video feed of the area of the body under operation. Minimally-invasive surgery boasts several advantages over traditional methods including a faster recovery time during which there is less pain and less scarring.

During these surgical interventions, the dynamic image sequences obtained from the laparoscope often contain a significant amount of cardiac and respiratory motion which can make it difficult for the surgeon to navigate and perform intricate procedures on specific areas of the heart.

Operating on a moving heart is considered difficult by even the most skilled surgeons

hence applying digital motion reduction to the video feed would prove very beneficial.

Motion Stabilisation is a field of computer vision that focuses on removing unwanted movement from a dynamic image sequence. What constitutes as unwanted movement depends on the situation in which we are applying motion stabilisation. The majority of published research has focused on the stabilisation of footage captured using handheld home video cameras to remove the effects of the holder's 'shaky hand'. The field of Motion Stabilisation is also closely related to Image Mosaic Reconstruction whereby the relation between similar images is established (Image Registration) and one large picture is constructed comprised of the smaller images. This is particularly useful when capturing aerial maps of a landscape.

There are currently three general types of procedures for stabilising feed from a camera; mechanical image stabilisation, optical image stabilisation and digital video stabilisation[25].

Mechanical Image Stabilisation focuses on stabilising the actual camera as opposed to the video that the camera records. The camera rests on a gyroscope which detects movement using motion sensors and compensates for the movement accordingly. Gyroscopes offer a physical means of stabilisation and as such are typically large and consume a lot of power and are cumbersome in most scenarios.

Optical Image Stabilisation originally developed by Canon, works by varying the optical path to the CCD sensor in the chip. Like mechanical image stabilisation, OIS happens before the image is converted into digital data. To implement OIS, specialist hardware is required and the cameras tend to be bigger and heavier than their non-OIS counterparts.

Digital Stabilisation is a post processing method that stabilises video after it has been captured by a camera and converted to digital data. Digital Stabilisation is the most aggressively researched technique for motion stabilisation since software-based, algorithmic stabilisation techniques are cheaper and more compact[16] than their hardware-based equivalents.

Different approaches to digital stabilisation have been produced over the recent years, each addressing a particular usage scenario such as handheld video. Today, motion stabilisation remains an active field of study amongst the Computer Vision community. Digital stabilisation methods are the pursued means of stabilisation in Minimally Invasive surgery environments because they do not require any additional hardware to be present in the immediate environment. Unlike the physically-based options, optimal realtime performance is harder to achieve and is dependant on the complexity of the implementation and the hardware it is running on.

One of the potential benefits of motion stabilisation in MIS surgery is that if the heart's motion can be successfully compensated on the video feed, in theory it could be extended

to the robotic arm holding a surgical tool so that the tool automatically follows the area of the tissue that it was originally trained upon [13].

In this project, we investigate the suitability of using an adapted 2D motion stabilisation technique to reduce or remove motion caused by cardiac and respiratory motion during beating heart surgery. The advantage of this approach is that it would not require any domain-specific hardware and could feasibly be integrated into existing operating theatre video feed workflows where space is typically limited.

1.2 Related Work

There have been previous attempts to perform visual stabilisation of the cardiac surface by various international research groups. Some researchers have focused on tracking motion of the heart purely for synchronising surgical instruments with the periodic motion of the heart [21]. One interesting method uses strobe lighting controlled by the patient's ECG graph to reduce the apparent motion of the heart [12] however this eventually causes strain on the surgeon's concentration. One computer vision method [29] uses two cameras to recover the 3D deformation of the heart and uses augmented reality to create a virtual camera which moves to compensate the apparent 3D motion.

1.3 Objectives

The aim of this project was to investigate the suitability of applying 2D digital motion stabilisation techniques, in particular an adapted sequence of procedures originally described by Grundmann et al [14], when applied to remove cardiac and respiratory motion seen in video recorded during minimally invasive cardiac surgery. The main benefit of 2D Methods is that they would be less expensive in terms of computational power to incorporate into existing MIS workflows than their 3D equivalents.

1.4 Contributions

During this investigation we developed an open-source tailored implementation of our chosen motion stabilisation method outlined by Grundmann.

The toolkit we produce is operable via two front-ends; a graphical user interface that allows visual analysis of each stage of the stabilisation, as well as a command line interface which allows for batch video processing. All of the code is publicly available on the GitHub website in the hope that it will prove useful to others working in the same field.

The produced framework contains:

- A configurable motion stabilisation tool which performs a sequence of motion stabilisation procedures.

- An implementation of importing, analysing, processing and exporting video files.
- An interface to the OpenCV library for various optimised feature detection and tracking helper methods.
- An implementation of Local RANSAC to remove outlying feature movements from calculations.
- An implementation of optimal camera path calculation [14] using Linear Programming / Simplex tableau solving techniques with the aid of the CoinLP library.
- A graphical user interface for configuring and observing the results after each step of the stabilising procedure.
- A batch video-processing command line interface offering complete configuration of the stabiliser via arguments and flags.
- A collection of MATLAB functions callable by the Motion library to provide analysis on the videos exported by our library.

Using the framework, we present an evaluation of the attempted stabilisation method when applied to a footage of a heart phantom and footage of a real heart.

1.5 Report Structure

The rest of the report is structured as follows:

- Chapter 2 describes the relevant background information concerning the general steps involved with motion stabilisation including feature detection, feature tracking, motion models, outlier rejection, global motion estimation, optimal motion estimation and video warping.
- Chapter 3 describes the specific procedures used to perform Motion Stabilisation akin to Grundmann's method.
- Chapter 4 describes the relevant software engineering aspects of the project including tools used and an overview of the structure of our program.
- Chapter 5 outlines the experiments undertaken on three sample videos and reports the results obtained.
- Chapter 6 contains our evaluation of the results observed from the previous step as well as plausible explanations for the observed results.
- Chapter 7 provides concluding remarks and outlines potential future work.

- The Appendix consists of additional material including a User Guide for installing and running the Motion framework or clients.

Chapter 2

Background

This chapter outlines the background computer vision knowledge associated with Motion Stabilisation that was required to complete this project. The actual implementations we employ for each step are described in Chapter 3.

2.1 Basics

For the sake of conciseness we adopt the following notations and definitions used throughout the project.

2.1.1 Video

We define a video, V, to be an ordered sequence of N, equally sized, image frames $I_0, I_1, I_2, .., I_N$. Videos can be finite in length, or unbounded (in which case it is called a video stream). A video data structure contains metadata including a frame rate which specifies the number of frames to be shown per unit time. Typical videos have a frame rate of 24-30 frames per second, although a frame rate as low as 15fps can be enough to achieve the illusion of a fluidly moving scene. Uncompressed videos offer maximum quality but at the cost of a high data rate. Compression algorithms are typically employed to reduce temporal and spatial redundancy.

2.1.2 Frame

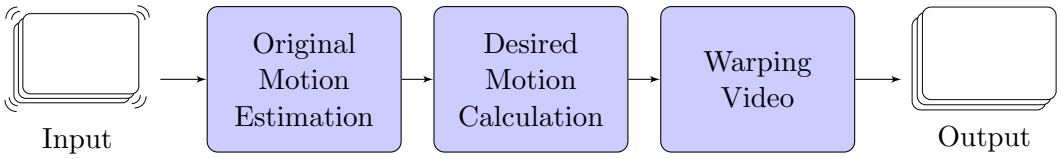
A frame is represented as a matrix of pixels of dimension WxHxC where W is the width, H is the height, C is the number of channels (a normal RGB-colour video has three channels, whilst a greyscale video has one channel). Each element in the matrix is typically an 8-bit integer and thus is capable of representing 256 different intensity values. The origin of a frame's coordinate system is traditionally located at the top left corner of the frame.

2.1.3 Inter-frame Motion

F_t represents a transformation modelling the movement of the scene from frame I_t to I_{t-1}. We model motion in this 'backwards' fashion in consideration of real-time techniques where at the current frame, the next frame isn't known yet but the preceding one is.

2.2 High-Level Overview

Most Motion Stabilisation techniques can be broken down into three distinct separate stages[25].



Original Motion Estimation A motion model is calculated which describes the *unstabilised* original motion in the video.

Desired Motion Construction A new motion model is calculated for the video which is free of unstabilised movement but preserves any desired motion from the original video as well as other constraints.

Warping/Re-rendering Video Finally, using both the old and new motion models, a warping update transformation (or equivalent) is applied to the original video. The end-result has reduced unwanted motion.

In the rest of the chapter we present background information related to these stages.

2.3 Original Motion Estimation

Estimating the original motion of an object is a very important part of motion stabilisation. There are many ways to find the parameters of a global motion model of an object [13] however from a pure Computer Vision approach we use only the images from a camera to infer motion models.

The goal of this stage is to produce a mathematical model of the motion observed in the original video which models all of the observable motion in the scene. The type of model chosen and the accuracy with which it is fitted to describe motion in the scene between a pair of frames has a large impact on the accuracy of the end stabilised video.

We define the observed elapsed motion in a video as a discrete function consisting of a series of parameterised motion models F_t mapping the scene's position in frame I_t to frame

I_{t-1}. The original motion path of the scene is thus obtained by multiplying the inter-frame motion models together.

F_t is typically calculated using the following sequence of procedures:

1. An optimal motion model is chosen that is suitable for describing the motion observed between the two frames. The parameters for the motion model are as of yet unknown.

2. An optical flow field is then calculated for some/all pixels in I_t. The flow field contains displacement vectors that map the pixels from I_t to their estimated location in I_{t-1}

3. Outlier rejection is applied to the flow field to remove displacements that go against the general direction of the other displacements.

4. Using the remaining inlier displacements, we estimate the parameters of the chosen motion model by minimising an error function.

2.3.1 Motion Models

Motion models are used to describe the overall movement of a scene between two frames. Using multiple calibrated cameras, we *can* calculate the 3D structure of the scene and can thus calculate 3D motion models (good for modelling complex movement and significant non-planar depth variation in the scene) which can then be used to perform 3D motion stabilisation [18].

The heart is an object that is perpetually rotating,translating and deforming in 3D space and thus a completely accurate model of the heart's model would need to be 3D also. There has already been some work carried out to infer the 3D deformation model of a cardiac surface and subsequently perform motion stabilisation using augmented reality [29] yet this typically involves complex equipment and computing power that is not readily available in a typical operating theatre [23].

In this project, we focused on the effectiveness of general 2D motion stabilisation largely due to the lower complexity and cost of 2D motion stabilisation which is beneficial if integrating with existing MIS workflows. It has been shown that 3D deforming surfaces can often be sufficiently tracked using 2D rigid linear approximation motion models if the image sequence has a high temporal resolution [26].

A selection of 2D motion models are available to describe the 2D motion (i.e. the projected 3D motion) of a moving scene object across a frame.

Each 2D model is parameterised and each parameter represents a degree of freedom. As you increase the number of degrees of freedom fewer restrictions govern the extent to which an original item can be warped. As table 2.1 shows, as the number of degrees of freedom increases, the number of feature characteristics retained decreases.

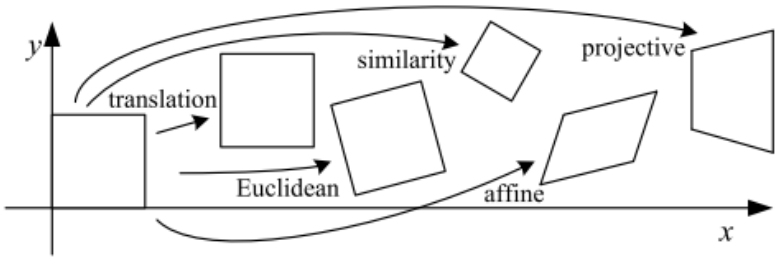


Figure 2.1: Overview of 2D Planar Transformations (Source: Richard Szeliski, 2006 [31])

Name	# Degrees of Freedom	Preserves
translation	2	orientation + ...
rigid	3	lengths + ...
similarity	4	angles + ...
affine	6	parallelism + ...
projective	8	straight lines

Table 2.1: The relationship between each 2D motion model with its degrees of freedom and the aspects of the shape it preserves.

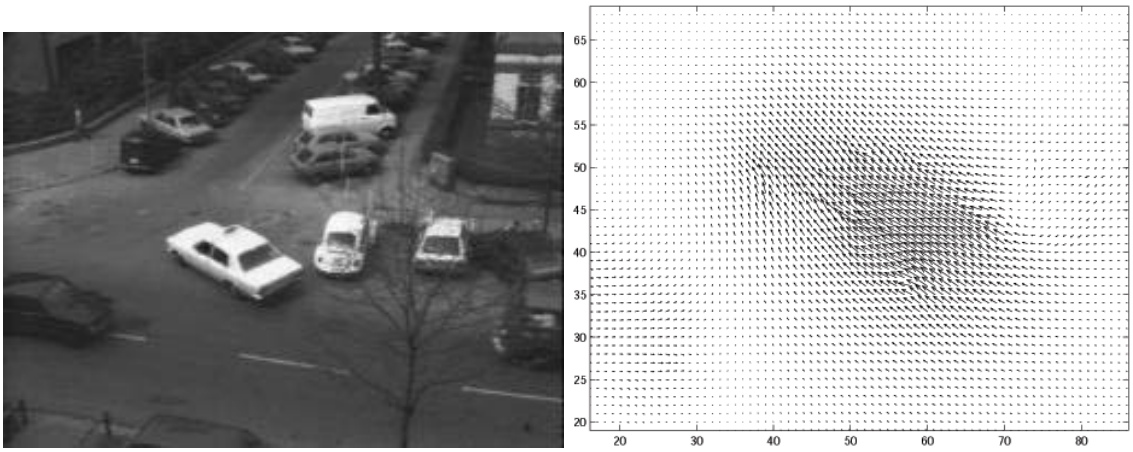


Figure 2.2: The dense optical flow field of a scene containing a Taxi. (Source: A. Borzi, 2013 [7])

2.3.2 Optical Flow Fields

An optical flow field is a field of vectors where each vector corresponds to the visible displacement $(u, v)^T$ of a pixel between two frames. When constructing an optical flow field we can choose to calculate displacements for every pixel in a frame (the resulting optical flow field is said to be *dense*) or we can calculate the displacements for a select few pixels (the resulting optical flow field is *sparse*). Figure 2.2 shows the dense optical flow field for an image containing a moving taxi.

Whether to use a dense optical flow field or a sparse optical flow field is a hot topic in the field of computer vision (see [32] and [17]). Supporters of dense fields claim that by tracking every pixel in an image, you are guaranteed to cover all the pixels moving with the overall global motion (and indeed most methods are good at 'locking' onto the global motion [17]). Despite this, the inclusion of every pixel unavoidably leads to noisy data since not all pixels are seen to move consistently with the global motion in the scene (this is especially true when specular reflection is present, or there is some temporary occlusion in the scene). These pixels (outliers) will affect later results and so the process of outlier rejection will be more complex. Those in favour of using dense optical flow fields to calculate a global motion model point out that most direct methods automatically assign weights to each pixel so that pixels corresponding to features benefit from positive discrimination whilst assigning smaller weights to pixels in homogeneous areas.

Calculation of sparse, feature-based optical flow fields are generally quicker to compute since fewer pixels are processed. Sparse, feature-based optical flow fields can produce an accurate global motion model provided that the features chosen move in accordance with the global motion of the scene. This highlights the importance of choosing the best feature detection method prior to tracking them. Many feature detection methods are *photometric*

invariant which means that the feature's displacement could still be detected even if its intensity changes. Sparse, feature-based optical flow fields have been shown to give good estimates of the global motion without the computational overhead of tracking each and every pixel [32] therefore for this project we will use feature-based sparse optical flow fields.

Feature Detection

Various feature detection techniques exist to decide which salient pixels are suitable for tracking between two frames. There is no universal definition for what constitutes a salient feature since the suitability of a feature often depends on the scenario. The most common feature detectors look for one of the following:

Edges are pixels that can be found along a gradient boundary separating two image regions. Whilst simple to detect, edges are susceptible to the *Aperture problem* [20] which means that when an edge is displaced, its shift along the boundary is not detectable.

Corners are an improvement over edges. A corner has a distinctive local two-dimensional structure and the local image gradient has two or more dominant directions. Corners are not necessarily 'traditional' corners, for example, a black speck in a blue sky is also a corner.

Blob Regions deal with areas of an image as opposed to individual points. Blobs are distinguished by comparing measurable traits such as intensity and colour across the region.

Feature Tracking and Optical Flow Calculation

Whether using all the pixels in the frame or just a selection of features, the next step is to calculate their displacements in the adjacent frame. Feature tracking and optical flow field calculation are two of the most fundamental research areas of Computer Vision.

There are various categories of optical flow estimation techniques:

Region-based matching techniques calculate optical flow fields by detecting the shift of a pixel's enclosing window between frames. They are used when differential techniques are not suitable e.g. due to noisy images. Finding the optimal shift is calculated either by using a correlation measure such as direct correlation, mean-normalised correlation, variance-normalised correlation or the sum of squared differences. It has been demonstrated [9] that the easier-to-compute measures such as SSD perform almost as well as the more intensive measures and hence are more commonly used. An example of a region-matching technique is described by Anandan et al [2].

Differential techniques are commonly used to calculate optical flow with good results using the gradient constraint equation which assumes pixel intensities are maintained between frames i.e.

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{2.1}$$

Phase-based techniques are used to detect dense optical flow fields. 2D Gabor filters are applied to images to produce signals with varying phases. By analysing the difference between the phases of adjacent frames, the optical flow field can be recovered.

Energy-based techniques consider the temporal and spatial frequencies of motion in a scene. The optical flow field is obtained by applying velocity-tuned filters in the frequency domain.

Barron et al [6] have written a comprehensive paper comparing the performances of a wide variety of such optical flow and feature tracking methods.

Often in the medical field a Magnetic Resonance Imaging technique known as MR Tagging is used to aid feature tracking [23] by encoding and scanning for specific magnetisation patterns in the heart's epidermis layer which helps find correspondences between features although unfortunately MRI machines are not typically installed in operating theatres used for heart surgery.

Laplacian pyramids A technique worth mentioning that is employed in many tracking methods is the usage of *Laplacian pyramids* and a coarse-to-fine strategy which is useful when there is a relatively large movement of features between two frames. Using Laplacian pyramids heuristically allows feature trackers to search optimally for features that have moved greatly without searching over the entire image. The laplacian pyramid of an image consists of versions of the image decreasing in resolution. Level 0 of the pyramid is the coarsest version of the image. The lower resolutions are used to help detect the larger displacements in the high resolutions. The process begins at the level with the lowest resolution where, because the image is so coarse, shifts can only happen by one pixel or less. An initial 3x3 search space thus surrounds each pixel. Each pixel in that search space is a candidate-match pixel. Feature tracking is initially performed using the coarsest version of the image L_{i-1} which provides an indicator as to what the search space should be in the following, sharper image L_i . The estimated displacement of a pixel at the previous level L_{i-1} is projected to all pixels in a 4x4 region in L_i. This means that each pixel in the current level L_i receives an estimate from 4 parent pixels in L_{i-1}. The new search area for a pixel on level L_i is a union of the 3x3 search areas centred around each of the parent estimates.

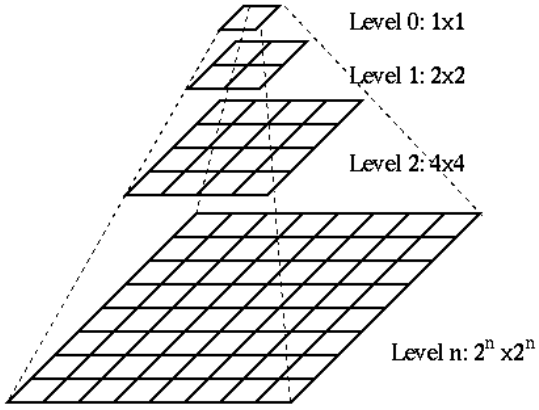


Figure 2.3: A LaPlacian pyramid of an image. Each layer represents a different level of resolution. (Source: R. Wang [33])

2.3.3 Outlier Rejection

After tracking pixels/features we now have an optical flow field of motion displacement vectors (similar to figure 2.2). There may be several observed movements taking place within the scene. We assume that there is one prevailing global motion that contains the unwanted movement that we would like to remove. All movements going against the global motion are called outliers. Before calculating the global motion estimate we therefore need to carry out outlier rejection to spot these outliers and remove them from further calculations.

In the context of Minimally-Invasive Heart surgery there are various typical sources of motion observable in the scene that we can consider as outlying motion including:

- Spilled blood from pierced blood vessels
- Rinsing fluid used to keep the surgical area clean
- Gas clouds from pierced vessels
- Specular surface highlights

2.4 Linear Programming

Our chosen motion stabilisation method involves solving a Linear Programming problem. Linear Programming (also known as Linear Optimisation) is a mathematical technique for determining parameters that will produce 'the best outcome' of an objective function. In its simplest form, a linear programming problem can be written as:

$$
\begin{array}{ll}
\text{minimise} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\
\text{subject to} & A\mathbf{x} \leq \mathbf{b} \\
\text{and} & \mathbf{x} \geq \mathbf{0}
\end{array}
$$

where vectors $\mathbf{b}, \mathbf{c}$, and matrix A are known already. The goal is to deduce the value for $\mathbf{x}$ which minimises $\mathbf{c}^{\mathrm{T}}\mathbf{x}$ whilst adhering to the constraints. Linear Programming is a very powerful means of optimisation and also offers a clear and effective means of adding and removing constraints as required. A linear program is considered infeasible if there is no possible instantiation of $\mathbf{x}$ which satisfies all of the constraints at the same time. When the problem is drawn graphically (one axis per unknown), drawing the constraints reveals a convex region which contains all the possible instantiations of the unknowns which satisfy the constraints.

Chapter 3

Procedure

This chapter describes the specific methods we employed to perform motion stabilisation in terms of each stage of the process.

3.1 Overview

The act of motion stabilisation is broken down into a series of computer vision processes. Our motion stabiliser is an implementation based predominantly on a sequence outlined by M.Grundmann [14] which has proven to be effective when removing shake from hand-held videos uploaded to YouTube. Grundmann's work focuses on removing shake but preserving intended camera movements. In this investigation we hoped to initially reproduce Grundmann's procedure and then measure its effectiveness when reducing apparent motion in scenarios where the camera is stationary but the scene is moving in a complicated fashion i.e. with cardiac and respiratory motion.

We perform Motion Stabilisation by carrying out the following steps:

- The global motion in the scene is estimated using a combination of feature detection, feature tracking, outlier rejection and motion model fitting.
- A mathematical optimisation problem is formulated that takes the calculated global motion and minimises movement wherever possible by minimising an objective function (subject to various constraints).
- A new stable video is obtained by taking a cropping frame of the original video, moving it across each frame using update transformations generated in the preceding step and cropping each frame at the crop box's new location.

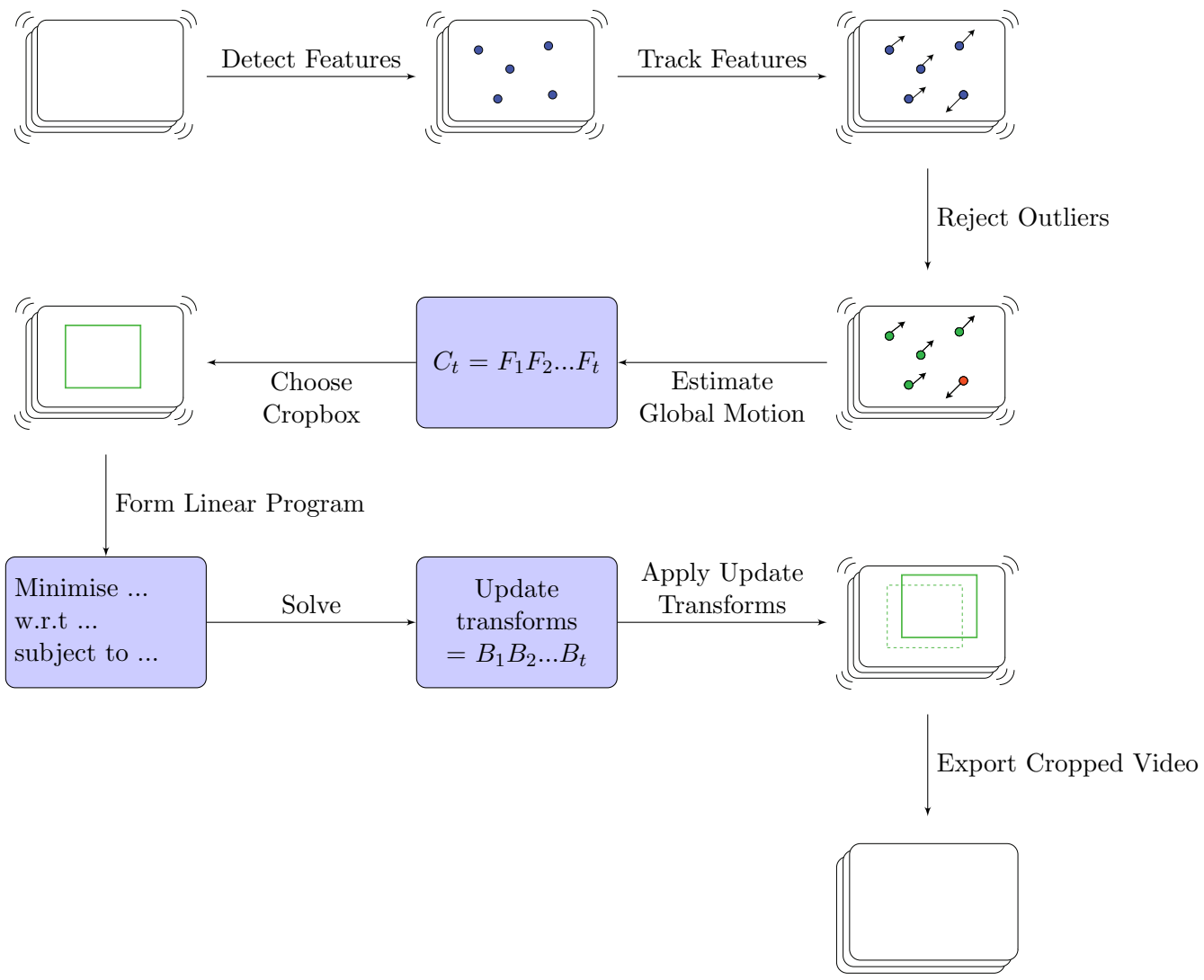


Figure 3.1: Overview of our motion stabilisation procedure.

3.2 Global Motion Estimation

We estimate global motion by first calculating a sparse feature-based optical flow field for each pair of adjacent frames. Features are detected using Shi and Tomasis Corner Detector [26]. Once found the features are tracked to the adjacent frame using the pyramidal Lukas-Kanade algorithm [19]. Outlier Rejection is performed using a Localised RANSAC [10] [14] method. Using the remaining inlier features, we estimate an affine transform that best explains the observed displacements of features between two frames. The affine transforms from each adjacent pair of frames are conjoined to form an overall discretised global motion path in the video describing motion of the scene over time.

It is important to note that the although the deformation of the heart is non-linear, it is has been shown that cardiac movements in local areas can be modelled using a linear motion model [22]. Based on encouraging results in [22] and [14] we will adopt an affine motion model to model our cardiac motion since affine transformations offer a relatively high degree of flexibility thanks to its six degrees of freedom.

3.2.1 Feature Detection using Shi & Tomasi's Corner Detector

The Shi & Tomasi Corner Detector [26] was employed to carry out feature detection because it is designed with feature tracking in mind. It has been shown that gradient-based methods such as the Shi & Tomasi corner detector are desirable for real-time usage scenarios. This particular detector is an extension of the earlier Harris-Stephens Corner Detector [15].

Each pixel in a frame is considered as a potential feature. A window of pre-defined size is placed around the current pixel. The window is then shifted by (x, y) and the *weighted sum of squared differences* (SSD) is calculated.

$$S(x,y) = \sum_u \sum_v w(u,v)(I(u+x, v+y) - I(u,v))^2 \tag{3.1}$$

where $w(u, v)$ is a window function which returns a weighting corresponding to the extent to which the pixel (u, v) lies within the window. For rectangular windows this function will return 0 or 1 however other types of windows exist such as Gaussian windows.

Using the Taylor expansion we can approximate

$$I(u+x, v+y) \approx I(u,v) + I_x(u,v)x + I_y(u,v)y \tag{3.2}$$

where $I_x(u, v)$ is the partial derivative of I in terms of x evaluated at (u,v). By introducing the approximation we can say

$$S(x,y) \approx \sum_u \sum_v w(u,v)(I_x(u,v)x + I_y(u,v)y)^2 \tag{3.3}$$

Which in matrix form is:

$$S(x,y) \approx \begin{pmatrix} u \\ v \end{pmatrix} M \begin{pmatrix} u & v \end{pmatrix} \tag{3.4}$$

Where M is a Harris matrix:

$$M = \sum w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{3.5}$$

A good corner is one that has a large S shifted by (x,y) in any direction. This quality can be examined by looking at the eigenvalues of M. If both eigenvalues are large (above a certain threshold) then the point should be considered a corner.

3.2.2 Feature Tracking using the Lukas-Kanade Algorithm

The Lukas-Kanade algorithm [19] is one of the most widely used optical flow techniques in Computer Vision[5]. Since its introduction in 1981, it has since been improved upon and extended considerably however the new techniques retain the same fundamental principles. The goal of the original LK algorithm is to find the new location of a pixel in a second image. The LK algorithm attempts to find a solution that minimises the SSD error between a small window enclosing the original pixel and a window enclosing the matched pixel in the second image.

The core LK algorithm makes three underlying assumptions of the scene:

1. Brightness constancy: The illumination of a small region in the scene remains the same even when it moves i.e. $I(x,y,t) = I(x+\delta x, y+\delta y, t+\delta t)$. We rearrange this equation using a first order taylor expansion of the right hand side of this equation

$$\begin{aligned} I(x,y,t) &= I(x+\delta x, y+\delta y, t+\delta t) \\ &= I(x,y,t) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t}\Delta t \end{aligned} \tag{3.6}$$

   thus

$$\begin{aligned} \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t}\Delta t = 0 \\ \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v = -\frac{\partial I}{\partial t}\Delta t \end{aligned} \tag{3.7}$$

2. Small Motions: The motion between two frames is small enough to allow us to solve the optical flow's differential equations

3. Uniform local movement: The LK algorithm assumes that neighbouring points belong to the same surface and thus during a small time step, the displacement (V_x, V_y) of a point $\mathbf{x}$ is the same for its surrounding neighbours i.e.

$$\begin{aligned} I_x(q_1)V_x + I_y(q_1)V_y &= -I_t(q_1) \\ I_x(q_2)V_x + I_y(q_2)V_y &= -I_t(q_2) \\ &\vdots \\ I_x(q_n)V_x + I_y(q_n)V_y &= -I_t(q_n) \end{aligned} \tag{3.8}$$

where I is the window surrounding $\mathbf{x}$, $I_x(q_n), I_y(q_n), I_t(q_n)$ are the partial derivatives in terms of x, y, t respectively, evaluated at q_n.

Equation (3.8) can be written in vector form as

$$Av = b \tag{3.9}$$

where

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, \qquad v = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \quad \text{and} \quad b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix} \tag{3.10}$$

This equation can rearranged to

$$A^T Av = A^T b \tag{3.11}$$

$$v = (A^T A)^{-1} A^T b \tag{3.12}$$

This equation can be solved by least squared methods for instance, using an iterative gradient descent method which begins with an initial estimate of (V_x, V_y) which is minimised according to the SSD error function (see equation 3.1).

In our sample videos of the heart, we typically have large inter-frame displacements which are too large for the original Lukas-Kanade algorithm to track. This is overcome by incorporating the pyramidal coarse-to-fine scheme which iteratively tracks features across coarser versions of the frames. Whilst alternative differential methods exist for calculating optical flow, the LK algorithm has been demonstrated to perform the best[6]. Furthermore, recently the LK tracking method has specifically been shown to work on endoscopic footage [11].

3.2.3 Outlier Rejection using Local RANSAC

A variant of Random Sample Consensus (RANSAC), one of the most popular methods for Outlier Rejection, is employed to identify spurious tracked features and remove them from subsequent calculations [10]. Algorithm 1 describes the RANSAC method in pseudo-code form.

We apply a slightly modified local-based version of RANSAC suggested by Grundmann et al [14] who reported optimal results using a localised version of RANSAC whereby the frame is split into 50x50 pixel grids and RANSAC is applied locally within each region, and a simple translation model is fitted to each grid. Features are then rejected as outliers within a grid if they do not move according to the grids generated translation model.

Algorithm 1 The RANSAC method

```
for N times do
    Select a random number of points (call these hypothetical inliers).
    Fit a displacement model to these hypothetical inliers.
    for all original points do
        Compare point with the new model.
        if point fits model then
            Mark as inlier
        end if
    end for
end for
Keep the saved displacement model which has the lowest error score with its inliers.
Reject the outliers of that displacement model from all further calculations.
```

3.2.4 Estimating the Global Motion from Feature Displacements

Following Outlier Rejection, we are left with a set of feature displacements which are relatively inline with each other. We estimate the parameters for an approximating affine transfer by minimising the L1 error between the actual positions of displaced features and the predicted positions of the features according to the current estimate of the affine transform matrix.

$$\mathbf{p} = \arg\min_{\mathbf{p}} \sum_{i \in inliers} |F_t(\mathbf{p}) I_t(i) - I_{t-1}(i))| \tag{3.13}$$

where $\mathbf{p}$ represents the six parameters of the affine transform F_t and $I_t(i)$ returns the inlier's position in I_t.

3.3 Optimal Path Estimation

We have so far calculated the original motion path $C_t = F_1 F_2 F_3 .. F_t$ which describes the displacement of the scene at time t from its original position at time 0. Using the novel method outlined by Grundmann we calculate an update transform B_t which is applied to a crop window within the original video to obtain our new optimal camera path $P_t := C_t B_t$. We define each update transform to be an affine transformation $A(p_t)$ whose parameter p_t is a 6x1 vector of the affine transform parameters $a_t, b_t, c_t, d_t, e_t, f_t$.

$$B_t := A(p_t) := \begin{bmatrix} a_t & b_t \\ c_t & d_t \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_t \\ f_t \end{bmatrix} \tag{3.14}$$

Our update transforms are obtained by solving a linear programming optimisation problem which minimises a cost function. We implemented both the video stabilisation method

which tries to stabilise the entire frame, and the automated stabilisation method which focuses on stabilising a selected feature within a frame. We outline here the original method and then describe how it is adapted to allow constraints concerning explicit features in the scene.

3.3.1 Cost Function

Our goal is to reduce the observed movement in a scene as much as possible thus we set our objective function (3.15) to minimise the difference between our desired smooth camera paths.

$$\mathcal{O}(P) = \sum_t |P_{t+1} - P_t| \tag{3.15}$$

Through factorisation we can reduce this minimisation objective further

$$\begin{aligned} &\text{Minimise} \sum_t |P_{t+1} - P_t| \\ &= \sum_t |C_{t+1}B_{t+1} - C_tB_t| \\ &= \sum_t |C_tF_{t+1}B_{t+1} - C_tB_t| \\ &\equiv \sum_t |F_{t+1}B_{t+1} - B_t| \end{aligned} \tag{3.16}$$

We arrange this in our final representation as a minimisation of the sum of residual errors i.e.

$$\begin{aligned} &\text{Minimise } e_1 + e_2 + e_3 + .. + e_t \\ &\text{where } -e_t \leq F_{t+1}B_{t+1} - B_t \leq e_t \end{aligned} \tag{3.17}$$

In Grundmann's algorithm he extends this smoothness constraint to allow the resulting image to contain not just periods of stillness, but also moments of constant motion and moments of constant acceleration. This is to preserve the desired actions of the filmmaker. In this investigation we emit these since our use cases do not require the preservation of such filming techniques.

3.3.2 Proximity Constraint

Proximity constraints are added to the LP problem to impose strict bounds on the amount of zoom, rotation, skew and non-uniform scale that our update transform can perform. This constraints prevents the contents of our eventual stabilised video from being warped beyond recognition.

$$0.9 \leq a_t, d_t \leq 1.1 - 0.1 \leq b_t, c_t \leq 0.1 - 0.05 \leq b_t + c_t \leq 0.05 - 0.1 \leq a_t - d_t \leq 0.1 \tag{3.18}$$

3.3.3 Inclusion Constraint

It is also important that our cropped window is not transformed to a position where part of it lies outside the original frame. With that in mind we add a hard constraint which ensures that the corner points of the original crop box, once transformed by the update transform, must have coordinates that lie within the dimensions of the original frame

$$(0,0)^T \leq CR_i p_t \leq (w,h)^T \tag{3.19}$$

where (w,h) are the dimensions of the frame, and CR_i is the Jacobian matrix of the affine transform with respect to p evaluated at the ith corner of the Crop Box.

3.3.4 Summarised Linear Program for optimal camera path

The table below shows the overall formation of the Linear Program that is solved to obtain the update transforms B_t.

Input: Frame pair transforms, F_t, Crop Box coordinates
Output: Camera update transforms, $B_t(= A(p_t))$

Minimise	e
w.r.t	$p = (p_1, ..., p_n)$
where	$e = (e_1, ..., e_n)$
subject to	
smoothness	$\begin{cases} -e_t \leq F_{t+1}A(p_{t+1}) - A(p_t) \leq e_t \\ e_t \geq 0 \end{cases}$
proximity	$lb \leq Up_t \leq ub$
inclusion	$(0,0)^T \leq CR_i p_t \leq (w,h)^T$

Table 3.1

3.3.5 Optional Saliency Constraint

Up until now our update transform is calculated without considering the movements of specific contents of the frame. We can rewrite the Linear Program to ensure that a salient feature obeys certain movement rules i.e. to always remain within our crop window.

We consider optimising the inverse W_t of our update transforms $B_t = W_t^{-1}$. W_t represents an update transform to be applied to the entire frame instead of the crop box within the frame. Previously the frame is fixed and the crop box moves, now the frame moves and the crop box is fixed. The objective is rewritten under the new formulation as minimising $|R_t| = |W_{t+1}F_{t+1}^{-1} - W_t|$. The existing constraints are also rewritten in terms of the new frame update transform. Performing this reformulation allows us to express hard or soft

constraints on the movements of salient features within the crop box when $A(p_t)$ is applied. We add new constraints to specify how far a salient point is allowed to be from a point described by (b_x, b_y).

$$\begin{pmatrix} s_t^x & s_t^y & 0 & 0 & 1 & 0 \\ 0 & 0 & s_t^x & s_t^y & 0 & 1 \end{pmatrix} p_t - \begin{pmatrix} b_x \\ b_y \end{pmatrix} \geq \begin{pmatrix} -\epsilon_x \\ -\epsilon_y \end{pmatrix} \tag{3.20}$$

where ϵ_x, ϵ_y are new slack variables to be included in our LP program and s_t^x, s_t^y are the coordinates of a chosen feature at time t. Setting b_x and b_y to equal the top left corner of our crop box results in a constraint specifying how far a salient point is allowed to be from said corner. We can add another constraint with b_x and b_y representing the coordinates of the bottom right corner. We can alternatively set b_x and b_y to be nearer the centre of the crop box. The two constraints combined prevent the feature from straying outside the crop box wherever possible. We apply a relative weighting between the residual errors and the slack variable errors to control the hardness of the salient constraints.

3.4 Warping Video

Solving the linear program returns update transforms B_t that we can apply to our specified crop box within the frame. Whilst there are various methods[8], [27] for re-rendering a video, one of the simplest and most effective is to take advantage of the laws of relative motion and create a cropped version of the video obtained by placing a crop box in the original video and warping it according to our calculated update transform. Each frame I_t represents the scene with a motion path of C_t. Applying the B_t to the crop box within the frame moves the crop box to a position such that the contents of the crop box now contain the scene with a motion path of P_t.

Chapter 4

Development

In this chapter we describe the software engineering processes undertaken to implement the procedure described in the previous chapter.

4.1 Design Overview

To perform Motion Stabilisation we developed a back-end library in C++ which performed the stabilisation. Two front-end applications were written to interface with the library. A console-based application allows for batch processing of videos without any interaction required. Meanwhile, a graphical user interface was developed which allowed for visual evaluation of each step of the stabilisation process.

4.2 Technologies Used

For efficiency, the project makes use of several established open source C++ libraries.

OpenCV C++ Framework The OpenCV (Open Source Computer Vision) framework originated from the Intel Corporation and encompasses C++ functions that perform optimally a variety of common computer vision procedures including facial recognition, gesture recognition and motion tracking. In this project, we took advantage of OpenCVs feature detection methods (including an optimised implementation of the Tomasi corner detector) and OpenCVs feature tracking methods (including an implementation of the pyramidal LK algorithm).

COIN CLP Simplex Solver The COIN-OR Linear Programming solver is an open source C++ library designed to solve large linear programming optimisation problems.

Qt C++ Framework The Qt framework is an open source library which originated from the Nokia corporation. It is an established framework for building multi-threaded applications in C++ with graphical user interfaces and event loops. It allows us to have a responsive UI whilst video processing is occurring.

Matlab The Matlab Engine is used to export data from our C++ program into the Matlab environment for efficient analysis and evaluation.

4.3 Core Library

The Core library contains all the procedures for performing motion stabilisation through the CoreApplication class and can read in a video file into an internal representation. The library can then be used to manipulate the video and return a stabilised video file. The library also exports information to Matlab for analysis of the stabilised result.

During implementation of the library, great care was required when writing the linear program (3.20) in the format required by the Coin LP Solver. Because of the relative computational intensity of video processing, all of the core library runs on a standalone thread to minimise any interference of the responsiveness of a front-end program.

NB. To perform feature detection and tracking optimally we desire that our input video is the highest quality possible. Our framework uses OpenCV during video processing and Matlab to evaluate the end result. Both needed to be able to read the video format. Ideally, we would manipulate uncompressed avi during this investigation (a format that Matlab optimally supports) however OpenCV has a long-standing bug which renders it incapable of using uncompressed avi. This bug was not fixed for the length of the project. A compromise to satisfy both OpenCV and Matlab constraints was to process and evaluate videos in the Motion JPEG format with maximal settings using the FFMPEG library.

4.4 GUI

A graphical user interface proved useful and provided us with the best means of controlling aspects of the stabilisation process whilst being able to view the immediate results. By using the graphical user interface the user is able to:

- Perform motion stabilisation step-by-step observing the outcome following each step.
- View and playback the new and original video.
- View features detected on each frame following feature detection.

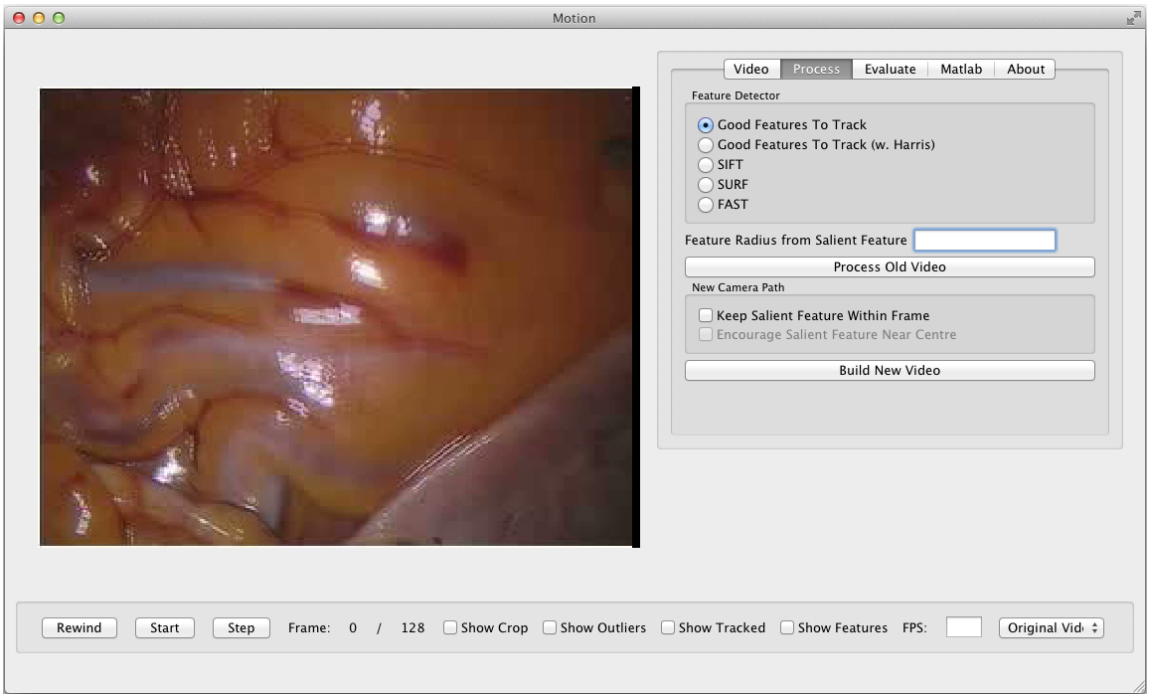


Figure 4.1: The main screen of the framework's GUI

- View disparity maps of features following feature tracking.
- View features identified as outliers.
- View the movement of the crop box across the original video following motion stabilisation.
- Manually specify the crop window position and size using click-and-drag.
- Manually track a salient feature for analysis.
- View the global motion estimate as a graph.
- Choose parameters for each motion stabilisation step (e.g. how many features to track).

4.5 Command Line Interface

The command line interface was subsequently developed as a lightweight front end configured by flags and parameters passed via the command line. It is useful for batch processing a set of videos or for replicating a previous experiment using the same parameters.

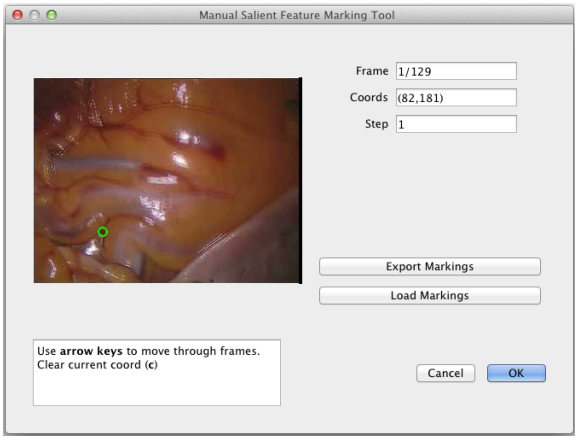


Figure 4.2: Additional screens allow us to manually track a feature across a video to provide additional information to our stabiliser.

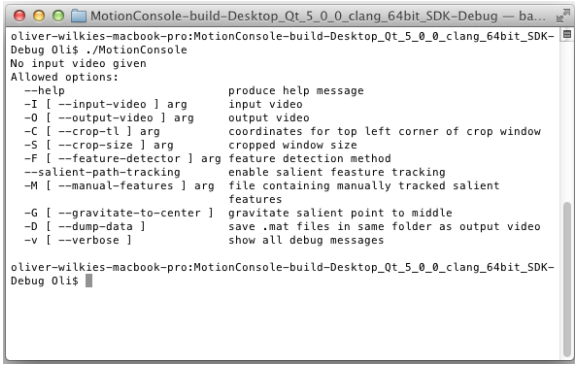


Figure 4.3: The help screen of the CLI front-end.

Chapter 5

Experiments

In this chapter we present our investigation into the performance of our motion stabilisation implementation when applied to various dynamic image sequences. We describe our experience with each of the preliminary steps of motion stabilisation before presenting the overall results of the entire motion stabilisation sequence.

5.1 Videos Used

We conducted our experiments using three videos. Each video contains a certain type of motion that we are seeking to reduce or eradicate.

The first video is a basic Handheld video from YouTube[1] and is akin to the sort of video that Grundmann's method was initially designed for i.e. an unstable camera filming an environment with stable motion. We define our unwanted movement as the universal shake caused by the cameraman. The video is taken by a person filming the back of a stationary car. Ahead of the car is a T-junction and on occasions a car passes across the road in the background. The moving cars in the background provide us with a clear means of testing outlier detection. Since we are ultimately aiming to still motion in a heart sequence, we only consider the segment of the film where the filmer's intent is a still shot.

Figure 5.1: A Handheld video containing camera shake. Grundmann's method was originally designed for this type of video.

[1]Handheld Video Source: `https://www.youtube.com/watch?v=d69k_eVJ534`

Our second image sequence was obtained by filming an artificial *ex-vivo* heart phantom[2] made from silicon rubber which mimics the beating movement of the heart. An advantage of using such a device is that imaging techniques can be tested in a controlled environment. In our video, the heart is filmed from a fixed position under controlled illumination. In this video, our undesired movement is the movement of features due to cardiac motion. Note that our heart phantom is not subject to respiratory movement nor does it show the flow of blood through blood vessels on the surface in the same way that Real Heart tissue would. The Heart Phantom contains artificial landmarks (white dots) which makes feature detection and tracking easier and helps derive a more accurate motion model.

Figure 5.2: A video of a heart phantom including cardiac motion and artificial landmarks (white).

Our third image sequence shows the active surface of real cardiac tissue moving according to both cardiac and respiratory motion[3]. It was taken from a stereo laparoscope *in vivo* and contains the rhythmic movement that we would expect to find in a real-life scenario. The sequence inevitably contains 'wet' specular reflection which is difficult to remove via post-processing to reveal the surface underneath. The video was taken with a low-frequency camera which is not ideal since blur is observable in certain frames. Despite this blur, the video was the best example we had available of both cardiac and respiratory motion simultaneously affecting a piece of heart tissue. Unlike the Heart Phantom, this scene is not manually marked with artificial features to track and as such we rely on the detection and tracking of natural landmarks in the scene.

5.2 Feature Detection & Feature Tracking

We first run our feature detector on each frame in our video set using the Shi Tomasi method and then track their displacement in the previous frame.

Figure 5.4 shows the optical flow for a single frame in our Handheld video. Notice the types of features detected include features belonging to the background car and consequentially, note they are successfully tracked to show the movement of the car. The rest of the pixels in an ideal world should be stationary and note from the drawn lines we can see

[2]Heart Phantom Source: Hamlyn Centre, Imperial College London
[3]Heart Source: Dr Eddie Edwards, St Mary's Hospital

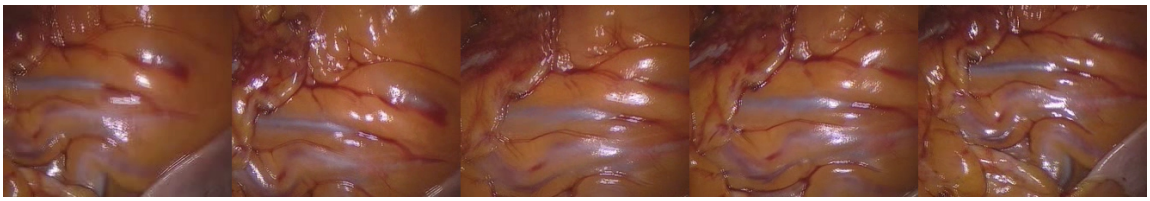


Figure 5.3: A video of a Real Heart including cardiac motion and respiratory motion. The scene only contains natural landmarks.

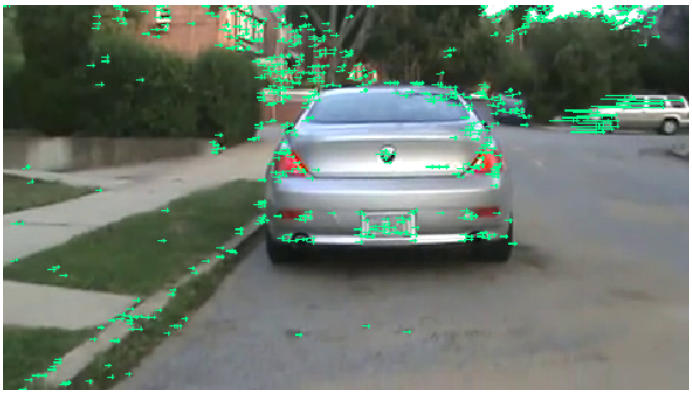


Figure 5.4: The sparse optical flow field for a frame from the Handheld video. Note the distinct outlier movement of the car in the background.

the signs of jittery movement. As we can see by figure 5.4 the Shi Tomasi corner detector avoids detecting features in large uniform areas (i.e. the road, the patches of grass or the bushes).

Figure 5.5 shows the sparse optical flow field detected in the Heart Phantom sequence. Compared to the variety of features detected in our Handheld video, there are fewer obvious salient features in this sequence. However, the phantom image has been marked with white reference points which are picked up by our feature detector as well as the purple cardiac structure near the centre of the scene. Note the scene also contains instances of specular reflection around which spurious features are identified which consequently contribute to the global motion estimate between the two frames. Again, note how a lot of the surface is ignored by the corner detector since it is too uniform in appearance.

Figure 5.6 shows an optical flow detected in a frame of our real Heart sequence. Note our feature detector is able to find many more features on a Real Heart surface than on our artificial one thanks to the sharp contrast of the surface blood vessels and variations in cardiac muscle thickness. We can see how the feature detector has detected spurious features from the specular highlights on the surface caused by the light source. These features appear to be moving differently compared to the rest of the features because the movement of the specular lobe is not bound to the movement of the cardiac surface.

The Shi Tomasi feature detector is configured by a quality parameter which specifies whether an identified corner should be deemed a suitable feature or not. The score is relative to the corner with the largest minimal eigenvalue (see previous chapter). If the best corner has the largest minimum eigenvalue of 1500, and if our quality parameter is 0.1 then corners with minimum eigenvalues between 15 and 1500 are selected as features. With a quality score of 0.1 we detect on average over 700 'features' per frame whilst if we increase the quality requirements to 0.2 then approximately 125 features are detected per frame.

Configuring the quality criteria of the Shi Tomasi feature is a sensitive issue. If we only retain the most distinct corners as features then it is plausible that these features will exist in relative isolation from one another. It is fully plausible that some of these movements may be outliers and thus when local RANSAC outlier rejection is performed in the subsequent stage, a grid containing an outlier feature may not contain enough inlier features to 'out' the outlier. There is also the computational cost of tracking more features since the feature tracking stage of our Motion Stabilisation method is currently the most time consuming of all stages.

In the Handheld video, nearly all of the motion in the scene (apart from the background moving cars) is global motion from the shake and so a feature count of about 100 should be enough to derive a similar discrete affine transformation model than if we used a feature count of 700.

We found that both the heart phantom and real heart sequences required a much lower quality ratio configuration for the Shi Tomasi feature detector to ensure that enough features (at least 150 per frame) could be detected to obtain enough inliers to reflect the

motion between two frames.

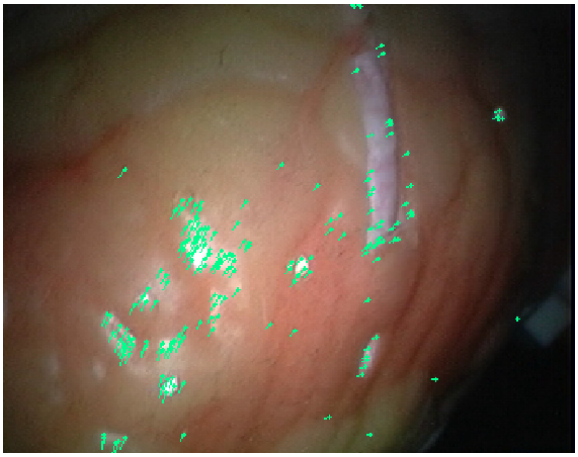


Figure 5.5: The sparse optical flow field for a frame from the Heart Phantom video.

5.3 Outlier Rejection

Following feature detection we identify and remove outlying features using local RANSAC.

Figure 5.7 shows the outcome of running local RANSAC on our Handheld video using the suggested grid size of 50x50 pixels. The pixels representing the background cars are moving contrary to the rest of the scene and so intuitively we would like them to be considered as outliers. However, the pixels of the background cars are deemed salient by our feature detector and furthermore they are deemed valid inliers under our local RANSAC method with grid size 50x50. The reason for this is that there are few other identified features in the surrounding vicinity of the background car and so most of the features that fall into the relevant grids reflect the movement of the background car.

The result is that the features of the background car have an undesirable impact on the estimation of the overall global motion which consequently has a negative impact on the remaining calculations of the optimal camera update path and the warping of our crop box to produce the final stilled video.

A solution to this specific case was to relax the sizing of the local RANSAC grids so that grids containing the background car features are more likely to also contain features moving according to the true global motion and hence more likely to be identified as outliers. Indeed with this particular video we find we need to increase our grid size towards 200x200 in order for the background car features to be correctly identified as outliers and hence discarded from all future calculations (see Figure 5.8). This shows that consideration needs to be taken into account when applying a local grid-based RANSAC as opposed to a global

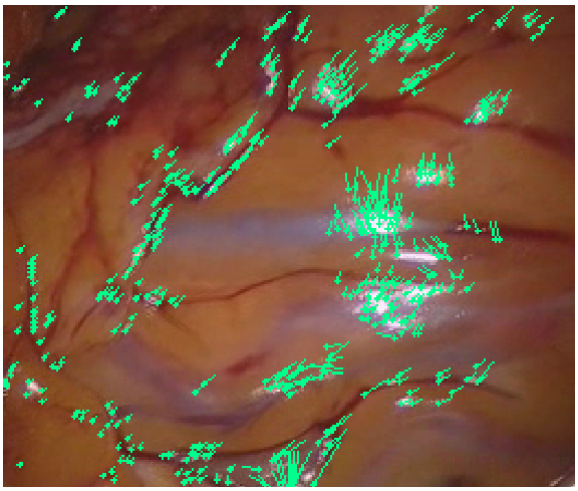


Figure 5.6: The sparse optical flow field for a frame from the Real Heart video. Note the outlier movement of features belonging to areas of specular reflection.

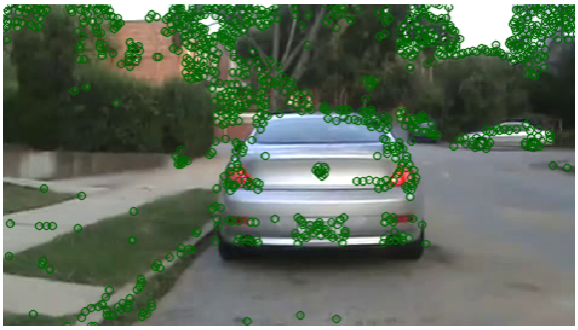


Figure 5.7: Outlier rejection in the Handheld video using the local RANSAC method and grid size 50. The method fails to identify the background car as outlying motion due to a lack of nearby inlier features.

Figure 5.8: Outlier rejection in the Handheld video using the local RANSAC method and grid size 200. The method fails to identify the background car as outlying motion due to a lack of nearby inlier features. The features on the background car are correctly classified as outliers relative to the overall global motion.

RANSAC. Local based RANSAC will be most effective when correct inliers can be found close to the salient outlier features in order for the outliers to be identified accordingly.

Figure 5.9 shows outliers detected in a frame of our Heart Phantom video. As we can see there are a significantly higher amount of inliers than outliers. It indicates that our local areas of our Heart Phantom move smoothly. so providing our salient features are correctly tracked then most features in the same area are likely to be moving in harmony also.

Figure 5.10 shows an example of outlier rejection being performed on our Real Heart video. As we can see, many features are considered to be local outliers. We can see a large pool of outliers in the top-left quadrant of the scene due to a large lobe of specular reflection that is disrupting the feature tracker. Out of the three videos, the real Heart seems to be produce the most outliers in each frame mostly due to the large specular highlights as well as the amount of local transformation taking place instead of an overall uniform global motion. Of course, the heart is a 3D deforming object which we are attempting to approximate using a 2D affine model which is likely to be sub-optimal for the RANSAC method.

Another possible reason is that our video sample was also acquired using a relatively low frequency camera and thus there are some frames which are blurred (we discuss this later in this report).

5.4 Global Motion Estimation

Using the measured displacements of the remaining inliers, we estimate the parameters of an affine transformation matrix for each pair of adjacent frames. We then concatenate

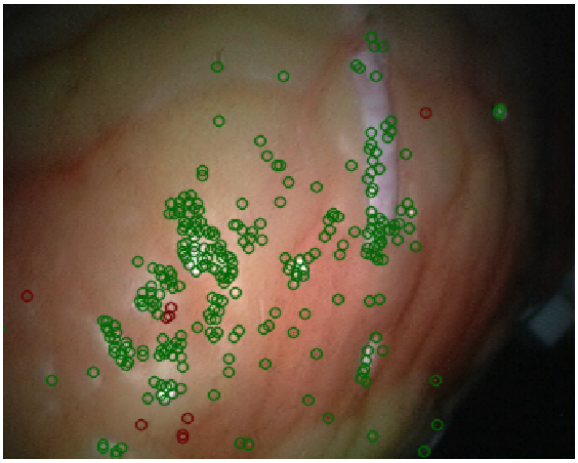


Figure 5.9: Outlier rejection in the Heart Phantom video using the local RANSAC method of 50. Few outliers (red) are detected indicating local linear displacement between frames.

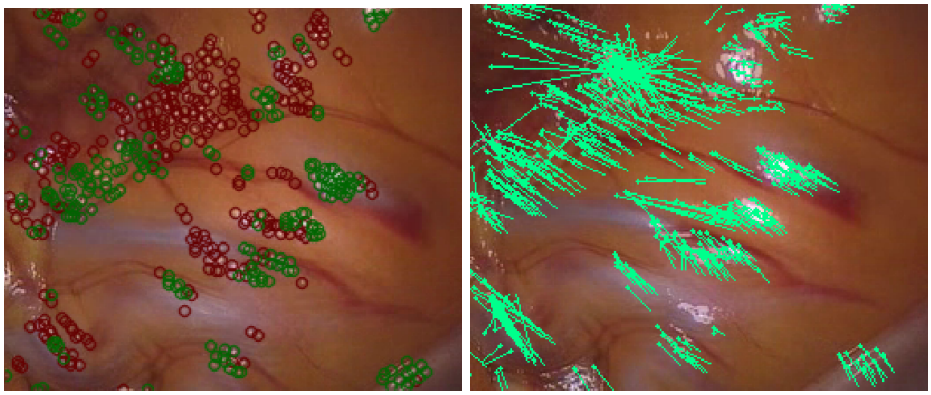
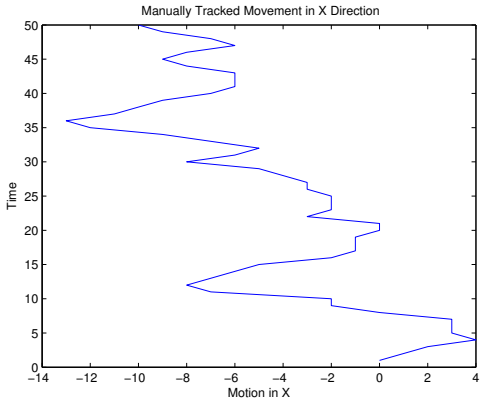
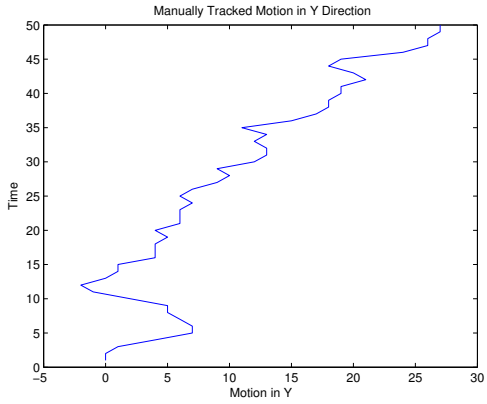


Figure 5.10: Outlier rejection in the heart video using the local RANSAC method of grid size 50x50. The abundance of spurious specular highlights results in a large amount of features being marked as outliers.

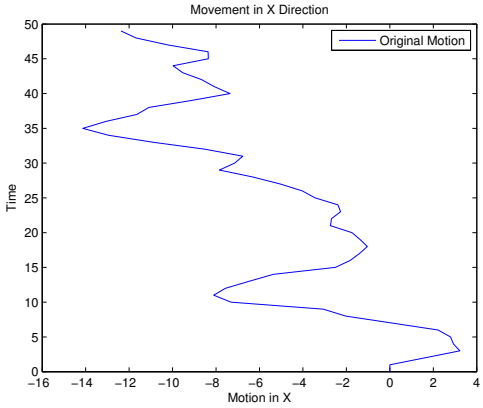
these matrices to obtain a discretised model describing the transformation of the scene. We display the results of this process by applying the discrete transforms in sequential order to an arbitrary point and plotting its displacement in both the x and y directions.



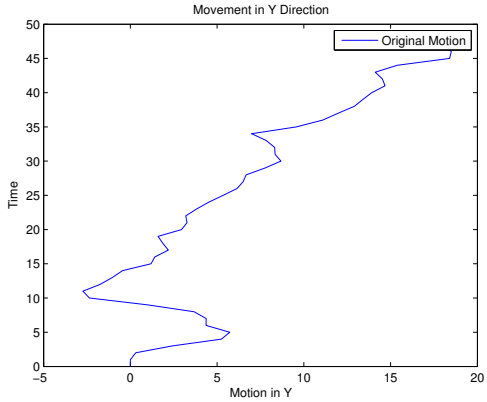
(a) Manually tracked motion in the X direction



(b) Manually tracked motion in the Y direction



(c) Estimated motion in the X direction



(d) Estimated motion in the Y direction

Figure 5.11: Manual and estimated motion plots of the Handheld Video across 50 frames.

Figure 5.11 depicts the derived global motion observed in the first 50 frames of our Handheld video. Since we know a priori that the scene itself contains only stationary objects (the moving background car has been identified as an outlier and removed) we can say that the graph represents the motion of the shaky camera. For validation, 5.11 also shows the actual displacement observed by manually tracking the jitter in the scene. As we can see, the derived global motion closely (but not perfectly) fits the true path.

Figure 5.12 shows the resulting derived motion of the beating Heart Phantom in both X

and Y directions over 50 frames. We can observe the distinct rhythmic beating of the Heart Phantom in the motion in both axes which indicates an encouragingly accurate model of the heart.

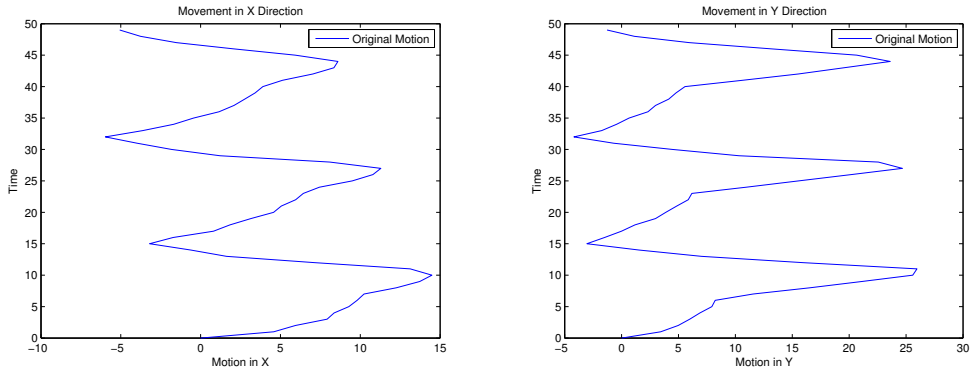


Figure 5.12: The global motion estimate of the Heart Phantom sequence across 50 frames.

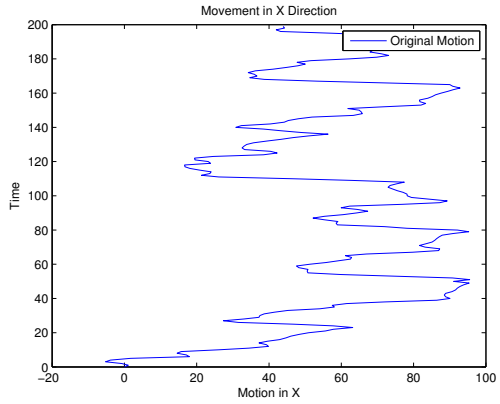
Figure 5.13 shows the detected motion in the real Heart video for 200 frames which is long enough to feature both cardiac and respiratory motion. We compare the detected motion against a smoothed version of that motion calculated using a moving average of the neighbours of each data point. Regardless of noise, we can perceive both cardiac and respiration motion. The cardiac motion is observed by the high frequency motion (repeating approximately every 28 frames) whilst the respiratory motion is observed by the lower frequency motion (repeating approximately every 114 frames).

5.5 Camera Path Recalculation and Video Warping

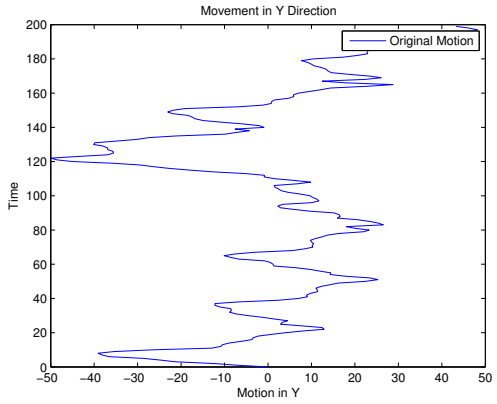
Following Global Motion Estimation we compose a linear program (3.1) with an objective function that seeks to minimise the observed motion between adjacent frames of our final video. Solving the linear program provides us with update transforms for a crop box placed at an initial location within the frame. We render the end video by transforming the crop box in frame I_t according to B_t and using the new contents of the crop box as frame I'_t in our new video.

5.6 Main Results

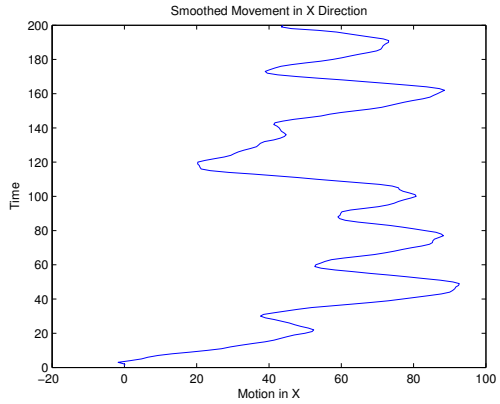
In this section we report on the generated final videos using both qualitative and quantitative metrics.



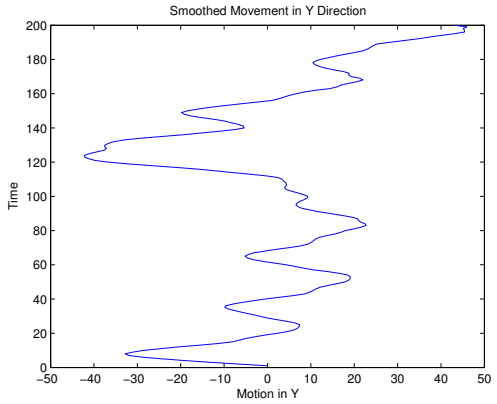
(a) Global motion estimate in X Direction



(b) Global motion estimate in Y direction



(c) Smoothed GME in X Direction (filter window size: 10)



(d) Smoothed GME in Y Direction (filter window size: 8)

Figure 5.13: The global motion estimate of the Real Heart sequence across 200 frames.

5.6.1 Measurements Used

In this project, several measurements were calculated to analyse the output generated by our Motion Stabilisation implementation. The motion in a newly generated video is always compared to the motion in the cropped version of the original video (without the update transform) of the same size.

Mean Image For each video we display the mean image calculated from all the frames in the final video. The pixels are converted into normalised greyscale intensity values stored as matrices of doubles to avoid arithmetic overflow. We take the average of these matrices and display the result. Larger movements cause larger observable blurs. Stiller videos produce clearer mean images where the underlying objects in the scene are easier to make out. The mean image serves as a qualitative measure.

$$\bar{I} = \frac{1}{N} \sum_{t=1}^{N} I_t \tag{5.1}$$

Mean Squared Error The mean squared error (5.2) is a commonly used tool for comparing video sequences. It is a favoured measure because it is robust and simple to calculate. In this project we calculated two mean squared error measures typically employed to assess the effectiveness of our stabilisation implementation.

$$MSE(I_t, I_R) = \frac{1}{M \times N} \sum_{M \times N} (I_t(m,n) - I_R(m,n))^2 \tag{5.2}$$

where M is the width of the image frame, N is the height of the image frame, I_t is current frame we are evaluating and I_R is the reference frame we are comparing the current frame against.

Global Mean Squared Error The Global MSE score compares each frame I_t in a sequence against the first frame, I_0. If the scene is completely stabilised we would expect every frame to be identical to the first frame and hence have a score of zero. We calculate the mean and variance of the Global MSE score applied to every frame in both the original video and the new video. The mean and variance give us a measure of the extent to which a video's content deviates from its opening frame.

Interframe Mean Squared Error The Global MSE score alone is not sufficient as a measure of general stability in a scene. Imagine there is a large change in frame contents after the first frame but after the big shift, total stability occurs. The Global MSE measure would not be able to spot this. A frame's Interframe MSE score, MSE_I, is calculated in the same way as the MSE_G score (5.2) but uses the preceding frame I_{t-1} as a reference frame and offers a measure of how rapidly a scene changes over time.

Opinion Score In addition to the objective global and interframe MSE measures, researchers typically obtain a subjective score from human observers (since the ultimate goal is to improve the visual experience of a human surgeon). In a survey created for this project, 67 participants were shown the three video sequences used in this project before and after motion compensation. The participants were then asked rank the extent to which they think motion was reduced using the following scoring system:

1	There is no apparent motion in the new video
⋮	⋮
10	There is just as much apparent motion in the new video, if not more

The results were collected and the average used as a measure to compare against the other measures.

5.6.2 Handheld Video

We report the outcome of applying our Motion reduction process to our Handheld video where the unwanted motion is caused by a jittery camera. Figure 5.14 shows the mean frames of the video pre-stabilisation and post-stabilisation using a crop window of 180x160 encompassing the foreground car. The jitter in the original video is evident by the strong blur in the pre-stabilisation mean image. Meanwhile, the significant reduction in blur (though not complete) in the post-stabilisation results in a clearer, more distinct mean image which indicates a much more fixed video sequence. When solving the LP we found that optimal results were obtained by constraining the update transform to four degrees of freedom (i.e. a similarity transform) so that any observable residual motion would be motion from the original video.

Table 5.1 shows the calculated means and variances of both Global and Interframe Mean Squared Error. We observe a noticeable decrease in Global MSE (including a smaller variance) which indicates that each frame in the final video has a higher fidelity to the starting frame than the original video. Even more noticeable is the greatly improved Interframe MSE score tells us that there is typically a much smaller change between adjacent frames in our stabilised image.

The median opinion score collected was *3* which indicates a general consensus that the jitter in the video has successfully been reduced.

5.6.3 Heart Phantom

Figure 5.15 shows the effect of applying motion reduction to a crop window sized 200x200 of the Heart Phantom Video. Though more difficult to visually assess than the Handheld video (because of fewer distinguishable features) we can still see a noticeable reduction in blur in the mean image of the stabilised video.

Figure 5.14: The mean frames of the Handheld video pre-stabilisation (left) and post-stabilisation process (right)

Measure		Original	Post-Stabilisation	% Improvement
Global Mean Square Error	Mean	0.0466	0.0131	71.9
	Variance	0.000455	0.0000651	85.7
Interframe Mean Square Error	Mean	0.00868	0.000602	93.1
	Variance	0.0000554	0.000000216	99.6

Table 5.1: MSE Scores for the Handheld Video

We initially restricted our update transform to have four degrees of freedom but this resulted in only a small reduction in motion in the final video. We relaxed our update transform to have six degrees of freedom allowing the crop box to be warped to the fullest extent that an affine transform allows and found a much more desirable reduction in the apparent movement of salient features.

By allowing six degrees of freedom, our final video does a better job of keeping the displacement of our salient points to a minimum at the cost of a wobble. This wobble is the result of approximating the heart's 3D transformation as a 2D planar affine transformation. Note that most of the feature-rich areas remain relatively still as these are the areas which had a greater contribution to the global motion estimate.

The median optical score collected was *5* which indicates a general consensus that motion had been reduced but not as successfully as the Handheld video example.

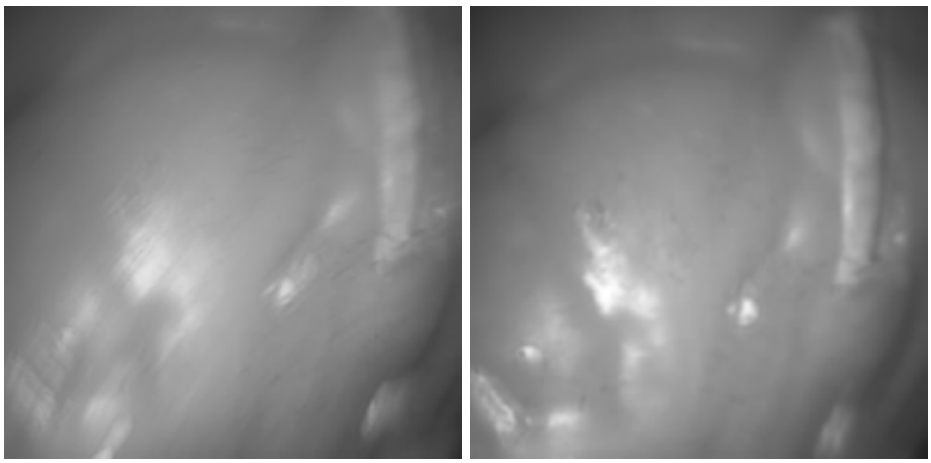


Figure 5.15: The mean frames of the Heart Phantom video pre-stabilisation (left) and post-stabilisation (right)

Measure		Original	Post-Stabilisation	% Improvement
Global Mean Square Error	Mean	0.00341	0.00245	28.1
	Variance	0.00000421	0.000000943	77.6
Interframe Mean Square Error	Mean	0.00152	0.00042	72.4
	Variance	0.000000973	0.000000148	84.8

Table 5.2: MSE Scores for the Heart Phantom

5.6.4 Heart

Figure 5.16 shows the result of applying our Motion Stabilisation tool to a cropped window sized 170x142 in our real Heart video.

Of the three videos, the heart video produces a mean image which is the hardest to visually interpret. The initial mean image is a complete blur which shows that there is a great deal of change occurring in the original video. If we look at the mean image of the video post-stabilisation we can see a slightly clearer image, for instance, blood vessels can be distinguished in the left edge of the image. The abundance of moving specular reflection has a large negative impact on the results of this experiment measure.

Looking at 5.4 we notice that there is no change in the Global Mean Square Error between the two images which tells us that there is no increase in fidelity between each frame and the first frame of the sequence. We calculated the Mean Absolute Error in addition to the Mean Square Error to see if the lack of improvement in the MSE score was due to a few individual frames that were greatly erroneous and so had a largely disproportionate impact on the MSE score than the other frames however there was no significant change in the global MAE either.

Despite the Global MSE result, we do notice an improvement in the Interframe Mean Square Error which indicates that on average there is a smaller change in scene contents between adjacent frames in the new video.

We finally collected the Opinion Score from various individuals who gave an average score of 4. Interestingly the participants generally favoured the motion reduction in the Real Heart as opposed to the Heart Phantom. Some participants commented that they found the residual wobble effect in the Heart Phantom footage off-putting.

Video	Mean	Median
Handheld	2.99	3
Heart Phantom	5.06	5
Heart	4.41	4

Table 5.3: Summary of Opinion Scores obtained for all three videos from 67 participants. A score of 1 means no residual motion whatsoever whilst a score of 10 means same amount of motion as before, if not more.

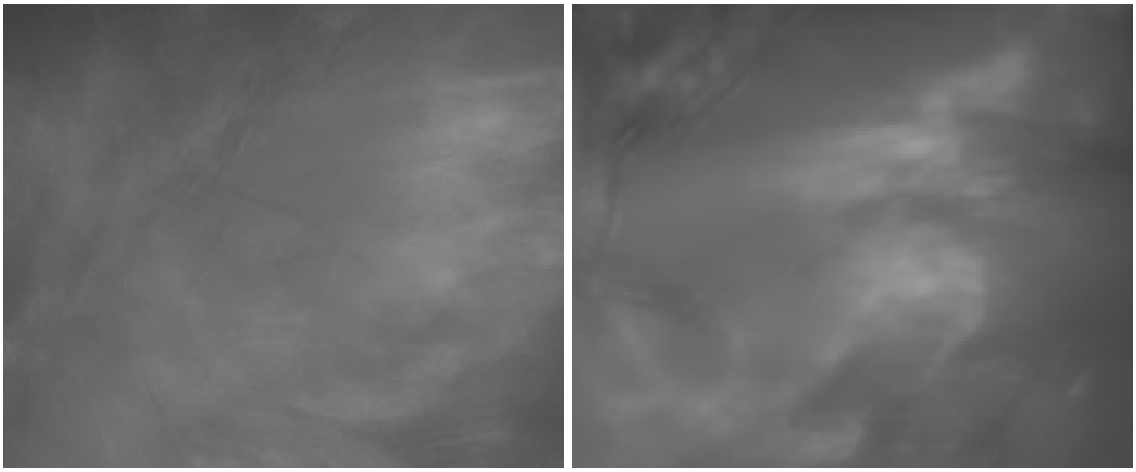


Figure 5.16: The mean frames of the first 100 frames from our Heart video pre-stabilisation (left) and post-stabilisation (right)

Measure		Original	Post-Stabilisation	% Improvement
Global Mean Square Error	Mean	0.0167	0.0160	No change
	Variance	0.0000103	0.0000125	No change
Interframe Mean Square Error	Mean	0.00818	0.00425	48
	Variance	0.00000976	0.00000847	13.3

Table 5.4: MSE Scores for the Real Heart

Chapter 6

Evaluation

In the previous chapter we showed the results of applying motion reduction to a video containing undesired shake, a video containing undesired cardiac motion and a video containing undesired cardiac and respiratory motion. In this chapter we evaluate the observed results and reason about why we observed the outcomes that we received.

6.1 Handheld Video

The goal of the handheld video was to verify that we had initially correctly implemented the motion stabilisation workflow described in Grunmann's method before enhancing and adapting it to our heart videos. Both the objective measures 5.14, 5.1 and the subjective measure 5.3 agree that there is a substantial drop in unwanted shake post-processing which implies a functioning motion stabiliser. It is worth highlighting that not all shake has been removed completely (as is indicated in particular by the mean Opinion Score and the Mean Image).

We briefly compare our observed output against the observed output of the implementation on the YouTube Video Editor which uses Grundmann's method. Note that since the YouTube Video Editor is for general use, we do not have control over the size and original placing of the crop box which makes it slightly harder to reliably compare and contrast the two. Similarly, the video needs to be uploaded to the YouTube platform prior to editing. In doing so, the video quality is automatically adjusted but the parameters are kept hidden meaning the video has a different quality setting prior to stabilisation which has an impact when measuring the MES and MOS scores. The YouTube implementation is thus effectively a black box and makes comparisons between the two implementations difficult.

Regardless of these pitfalls we present two mean images 6.1; one from a video produced via our implementation and one from a video produced from YouTube's implementation. What the mean image is not capable of showing is that there is in fact no high-frequency

(a) Our implementation (b) YouTube's implementation

Figure 6.1: The mean frames of the stabilised videos created by our implementation and YouTube's implementation.

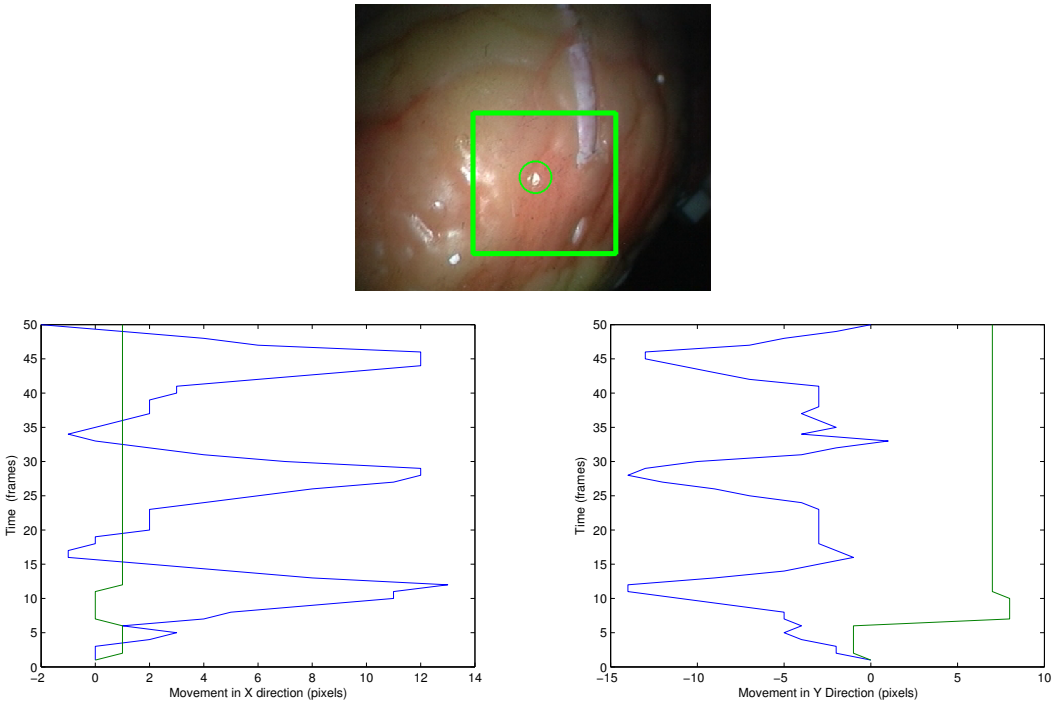
camera shake in the YouTube result. The vertical blur observable in YouTube's mean image is the result of a steady linear pan that the YouTube method falsely assumes was the original desired effect of the cameraman. This shows the limitations of using the mean image alone as a measure of reduced high-frequency jitter.

6.2 Heart Phantom

The goal of the phantom heart video was to reduce the displacement of features due to cardiac motion. Despite the relatively low number of salient features on the surface of the heart phantom, they are persistent over time and highly distinguishable. The global motion estimation process produced a well defined, regular, smooth motion path showing the rhythmic pulse of the mechanically simulated cardiac motion. The objective and subjective scores 5.15, 5.2, 5.3 all indicate a reduction in feature displacement. Indeed, for additional validation we briefly report the results of manually tracking a feature in the original video against the same feature in the new video.

As we can see by the graphs over time the feature is displaced to a much smaller extent than previously. We note that perhaps one explanation for why this feature stays so still is that there were a lot of salient features detected by the Shi Tomasi method in that area which means the local area of that movement had a greater impact on the global motion estimate which in turn ensured that the region stayed relatively still after motion reduction.

When playing the resulting video we still observe evidence of residual global motion in



(a) Movement in X Direction

(b) Movement in Y Direction

Figure 6.2: The manually tracked motion of the displacement of a salient feature (circled in green) in the original video (blue) and the new video (green).

the form of a wobble. This wobble is caused by our approximation of a 3D deformation model using only a discrete 2D planar affine transform which alone cannot easily translated into a 3D deformation model. The wobble was reduced by fixing the update transform to having four degrees of freedom (i.e. a similarity transform) but at the cost of additional movement of the salient features.

6.3 Heart

The goal of the heart video was to finally apply our method to a real example of footage from heart surgery. Compared to the results obtained from the heart phantom, the results obtained for the real heart are not as conclusive due mainly, to more complicated activity in the scene including respiratory motion and error in the global motion estimate caused mostly by visible specular highlights on the surface.

A noteworthy limitation of the footage in this experiment is the low-frequency frame capture rate of the laparascopic camera which recorded this footage. This is apparent on certain frames which show a distinct blur where the motion of the tissue is faster than the rate at which the camera can take the picture (e.g. 6.3). The Lukas-Kanade feature tracking method is unable to detect when a feature is present twice and does not linearly interpolate between the two locations to approximate the current location of the feature which would be desirable. Laparoscopes are continuing to evolve in terms of resolution and frame capture rate which means future related experiments would benefit from sample footage taken using more recent cameras.

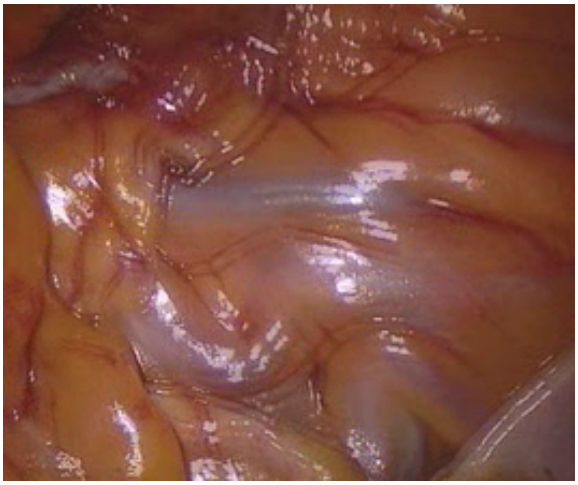


Figure 6.3: An example frame where the motion of the heart is faster than the capture rate of the camera

Alongside the drawbacks of the temporal and pixel resolution of the sample video, we also note the disruptive effect of the specular reflection caused by the 'cold' light source attached to the laparascope to light the environment. The specular highlights (observable in 6.3) have a negative impact on the tracking of features between frames and are not fixed to the underlying surface of the heart. Because of the large size of the highlights, their movements are never completely removed by outlier rejection. It would be advantageous to filter out the specular highlight and thus prevent the feature detector and tracker from producing spurious results. This process would ideally be a separate element of the MIS workflow run prior to motion reduction. The removal of specular highlights from a scene is a general Computer Vision problem and several approaches exist [4] which make use of various properties of light including the use of cross-polarised light. Some approaches exist that have been designed with endoscope footage in mind [3], [28].

We note that in our heart video there are no surgical tools in sight. In most scenarios, we should expect the surgeon to be equipped with a surgical tool which will occupy a significant portion of the screen. When calculating the global motion estimate of the scene it must not be affected by the local motion of the surgical tool. Steps must be taken to either discard any features located on the tool or to ensure that the local RANSAC grids contain more background features than foreground (tool) features to encourage the RANSAC method to label the tool features as outliers.

Despite the fundamental issues with the original video sequence we still observe results which indicate an improvement in motion compensation. Unaided, the crop window update transform is particularly successful at reducing a large proportion of the vertical displacement caused by respiration. However, cardiac motion still has a significant presence on the final video. The lack of improvement in the Global MSE score (and reduced improvement in the Interframe MSE compared to the other tested videos) supports this claim. The results suggest that a more expensive but accurate deformation model of the heart is required to accurately compensate for the deformation of salient features over time. After all, we are trying to fit a 2D planar transformation model to a surface which is deforming non-linearly in 3D thus residual motion is inevitable.

The performance of Grundmann's stabilisation method is fundamentally determined by the accuracy of our global motion estimate which (alongside the original crop box position) is the input of the linear program solved to find the update transform. Despite the glaring specular highlights, and the low temporal and spatial resolution of the laparascope footage, we find the global motion estimate to be strikingly faithful to the movement of individual features central to the scene. To prove this, we manually tracked a feature across 100 frames and compared its spatial displacement against the global estimate. We chose to manually track a fork of a blood vessel (Fig: 6.4) that is never located too close to the edge of the frame and can be located at each time step.

As Figure 6.5 shows, the Global Motion Estimate has a noticeable fidelity to the movement of the original feature.

In Figure 6.6 we manually track the feature again after having performed motion com-

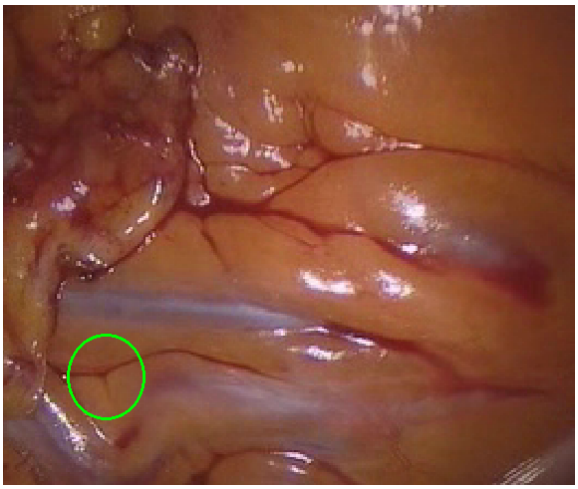
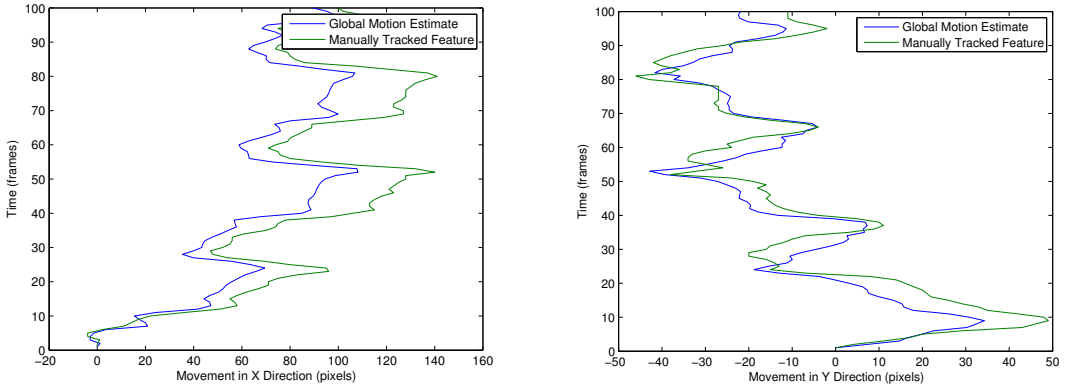


Figure 6.4: The feature circled was chosen for manual tracking because it is always visible in the original scene.

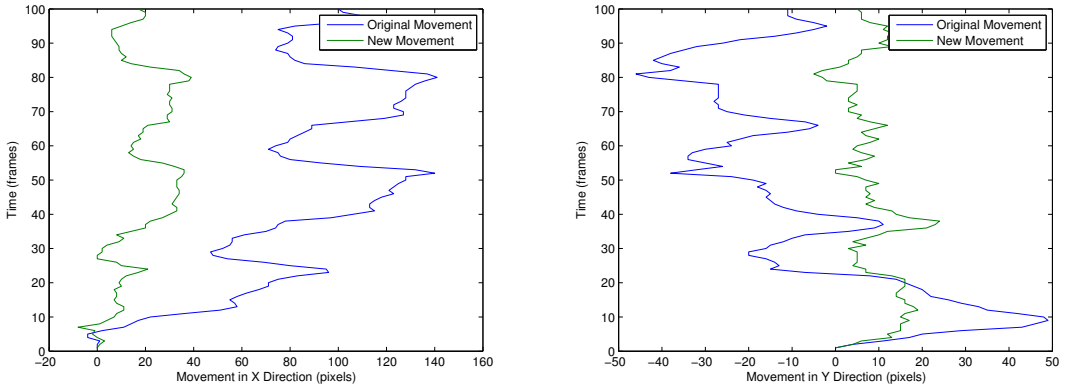


(a) Movement in X Direction

(b) Movement in Y Direction

Figure 6.5: Comparing the global motion estimate (blue) against the manually tracked displacement of a visible feature (green).

pensation using an enclosing 130x120 crop box we make several observations. The low frequency periodic motion caused by respiratory motion is greatly reduced in the new video. Whilst the movement of the feature in the new video still contains the high frequency periodic motion of the cardiac motion we note that the amplitude of the motion is significantly lower. In the new video the high frequency variation that is higher than the cardiac motion should largely be attributed to human error caused by difficulties manually marking the feature's displacement using a computer mouse.



(a) Movement in X Direction

(b) Movement in Y Direction

Figure 6.6: Comparing the manually tracked original movement of a feature against the manually tracked new movement of the feature.

Where the user decides to place the crop box has a large impact on the observable stillness of the final cropped result. Because of the camera's position and the relatively large movements caused by the cardiac and respiratory motion, not all the original features continue to stay within the frame. Naturally because of this, the system is limited to which areas can be continuously stabilised. If the crop box is placed with not enough room to move according to the global motion estimate it will stop at the edge of the frame causing the relative motion inside the frame to move considerably. Generally speaking optimal results are obtained if the crop box is placed around salient features which move but always stay within the frame boundaries.

In an ideal world, the surgeon would be able to specify a feature for the system to focus on and all the motion stabilisation process would need to do is move the crop box so that it stays continuously centred at the feature's new location. Unfortunately current natural landmark detection and tracking methods are not yet fully capable of tracking the same feature over time regardless of illumination changes, specular highlights, and temporary occlusion caused by spilt blood or cleaning fluid.

In Grundmann's paper, a reformulation of the LP equation is presented which focuses

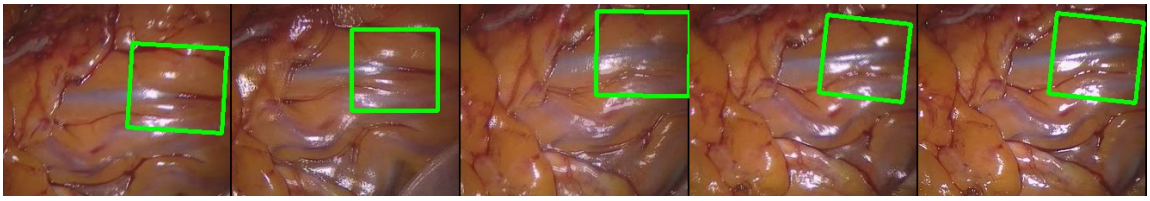


Figure 6.7: The inclusion constraint prevents our crop box from crossing the boundaries of the original frame.

on modelling an entire frame update transform against a stationary crop box (up until now we have done the opposite). The advantage of this method is that we can add soft saliency constraints to ensure that selected features remain within the crop box boundary. These constraints need to be soft because ultimately the crop box would otherwise be able to escape the confines of the frame boundary in order to ensure the constraint holds. This formulation offers the surgeon the ability to have a tighter rein on the behaviour of the crop box at the cost of needing accurate natural landmark tracking. Because constraints exist on a frame-by-frame basis you could omit the feature tracking constraints for frame pairs where the natural landmark was not accurately tracked and the result will still be satisfactory because of the other constraints.

We successfully performed the reformulation by manually providing the coordinates of a salient feature and adding soft constraints to keep the manually tracked coordinate near the centre of the crop box however automatic tracking of the same feature over time would be required in a real-time surgical scenario.

Chapter 7

Conclusions and Future Work

In this chapter, we summarise what has been achieved in this investigation and provide suggestions for future work.

Motion reduction of cardiac and respiratory motion in endoscopic surgery helps assist the surgeon by compensating for the original observable motion allowing the surgeon to focus more efficiently at the task at hand.

The aim of the project from the outset was to harness Grundmann's general automated stabilisation method for motion jitter and to assess its suitability in reducing both cardiac and respiratory motion in footage obtained during endoscopic surgery.

We were able to successfully implement the stabilisation procedure and produce a GUI application for processing and analysing individual stages of the process. An affine global motion estimate is generated for each frame using the Shi & Tomasi corner detector, a pyramidal Lukas-Kanade tracker, local RANSAC rejection. A new update transform for each frame is found by solving a linear optimisation problem and the result is applied to a crop box defined by the user.

We assessed our implementation on three videos and were able to remove a large proportion of observable motion from each.

A preliminary investigation of our implementation on a hand-held camera video with shake produced a video with most of the shake removed.

We next applied the implementation to footage of a heart phantom with artificial landmarks and found a noticeable reduction in the displacement of features due to cardiac motion at the cost of a wobble in uniform featureless areas.

Finally we applied the implementation to footage taken from a real heart tissue during endoscopic surgery and found the new video contained much displacement due to respiratory motion as well a decrease in cardiac motion but to a lesser extent.

7.1 Future Work

The implementation was applied to a dynamic sequence of cardiac images that were low in both pixel and temporal resolution. Future work should consider the effectiveness of the Grundmann method when applied to image sequences when captured from recent laparascopic cameras capable of capturing frames at a higher frequency and at a high pixel resolution. Future work would collect a large database of endoscopic cardiac surgery footage to obtain statistical validation of the results obtained in this experiment.

A large factor determining the effectiveness of Grundmann's method is the calculation of the global motion estimate. Our method calculates a global motion estimate using a sparse optical flow field subject to a lot of inevitable noise from specular highlights. The addition of a preliminary workflow process to remove specular highlights in endoscopic surgery is an active field of study [3] of which our motion reduction process would stand to benefit greatly from.

Future work would focus on refining the used method and optimising its suitability as a real-time procedure. We would also like to consider ways to facilitate automatically choosing the region of stabilisation focus based on the real-time requirements of the surgeon.

Future work would consider alternative means of original motion estimation including modifying techniques for recent cardiac surface natural landmark detection techniques to work in a near-realtime setting. An end goal would be to accurately build and maintain an accurate local 3D deformation model in realtime which could then be fed into an updated Grundmann's linear optimisation formulation and solved to obtain a 3D update to the visible scene. The biggest foreseeable challenge would be optimising this to work in realtime.

Bibliography

[1] An Introduction to Laparoscopy. `http://www.nhs.uk/conditions/Laparoscopy/Pages/Introduction.aspx`, 2011.

[2] P Anandan and New Haven. A Computational Framework and an Algorithm for the Measurement of Visual Motion. 310:283–310, 1989.

[3] Mirko Arnold, Anarta Ghosh, Stefan Ameling, and Gerard Lacey. Automatic Segmentation and Inpainting of Specular Highlights for Endoscopic Imaging. *EURASIP Journal on Image and Video Processing*, 2010:1–12, 2010.

[4] Alessandro Artusi, Francesco Banterle, and Dmitry Chetverikov. A Survey of Specularity Removal Methods. *Computer Graphics Forum*, 30(8):2208–2230, December 2011.

[5] Simon Baker and Iain Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision*, 56(3):221–255, February 2004.

[6] JL Barron, DJ Fleet, and SS Beauchemin. Performance of Optical Flow Techniques. *International journal of computer ...*, 1994.

[7] A Borzi and K Kunisch. Optical Flow Field for Taxi Scene. `http://www.uni-graz.at/imawww/invcon/`, 2013. Research group's image gallery.

[8] C. Buehler, M. Bosse, and L. McMillan. Non-metric image-based rendering for video stabilization. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2:II–609–II–614, 2001.

[9] P.J. Burt, C. Yen, and X. Xu. Multi-resolution flow through motion analysis. In *Proc. IEEE Conf. On Computer Vision and Pattern Recognition*, pages 246–252, 1983.

[10] KG Derpanis. Overview of the RANSAC Algorithm, 2010.

[11] Haytham Elhawary and A Popovic. Robust feature tracking on the beating heart for a roboticguided endoscope. *The International Journal of Medical ...*, (July):459–468, 2011.

[12] Terence J Gilhuly, Septimiu E Salcudean, and Samuel V Lichtenstein. Evaluating optical stabilization of the beating heart. *IEEE engineering in medicine and biology magazine : the quarterly magazine of the Engineering in Medicine & Biology Society*, 22(4):133–40, 2003.

[13] Martin Groeger, Klaus Arbter, and Gerd Hirzinger. Motion tracking for minimally invasive robotic surgery. *Medical Robotics, I-Tech ...*, 2008.

[14] Matthias Grundmann, Vivek Kwatra, and Irfan Essa. Auto-directed video stabilization with robust L1 optimal camera paths. *Cvpr 2011*, (1):225–232, June 2011.

[15] Chris Harris and Mike Stephens. A combined corner and edge detector. *Alvey vision conference*, pages 147–152, 1988.

[16] SC Hsu, SF Liang, and CT Lin. A robust digital image stabilization technique based on inverse triangle method and background detection. *Consumer Electronics, IEEE ...*, pages 335–345, 2005.

[17] M Irani and P Anandan. About direct methods. *Vision Algorithms: Theory and Practice*, pages 267–277, 2000.

[18] Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala. Content-preserving warps for 3D video stabilization. *ACM SIGGRAPH 2009 papers on - SIGGRAPH '09*, page 1, 2009.

[19] BD Lucas and T Kanade. An iterative image registration technique with an application to stereo vision. *Proceedings of the 7th international joint ...*, 1981.

[20] Josh McDermott, Yair Weiss, and Edward H Adelson. Beyond junctions: nonlocal form constraints on motion interpretation. *Perception*, 30(8):905–923, 2001.

[21] Y Nakamura. Heartbeat synchronization for robotic cardiac surgery. *Robotics and Automation, ...*, 2001.

[22] Tobias Ortmaier and M Groger. Motion estimation in beating heart surgery. *Biomedical ...*, 52(10):1729–40, October 2005.

[23] X Papademetris and JS Duncan. Cardiac Image Analysis : Motion and Deformation. *Handbook of Medical Imaging*, pages 675–710, 2000.

[24] Francis J. Podbielski. An Introduction to Laparoscopy. `http://www.laparoscopytoday.com/2002/01/evolving_techni.html`, 2002.

[25] Paresh Rawat, Jyoti Singhai, and Truba I E I T Bhopal. Review of Motion Estimation and Video Stabilization techniques For hand held mobile video. 2(2):159–168, 2011.

[26] J Shi and Carlo Tomasi. Good features to track. *. . . , 1994. Proceedings CVPR'94., 1994 IEEE . . .*, pages 593–600, 1994.

[27] Brandon M. Smith, Li Zhang, Hailin Jin, and Aseem Agarwala. Light field video stabilization. *2009 IEEE 12th International Conference on Computer Vision*, pages 341–348, September 2009.

[28] T Stehle. Removal of specular reflections in endoscopic images. *Acta Polytechnica: Journal of Advanced Engineering*, 46(4), 2006.

[29] Danail Stoyanov and Guang-Zhong Yang. Stabilization of image motion for robotic assisted beating heart surgery. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 10(Pt 1):417–24, January 2007.

[30] Bruce Stutz. Pumphead: Does the heart-lung machine have a dark side? *Scientific American*, 2009.

[31] Richard Szeliski. Image Alignment and Stitching :. pages 273–292, 2006.

[32] P Torr and A Zisserman. Feature based methods for structure and motion estimation. *Vision Algorithms: Theory and Practice*, pages 278–294, 2000.

[33] Ruye Wang. Computer Image Processing and Analysis Lecture Notes. `http://fourier.eng.hmc.edu/e161/`, 2013.

Appendix

Building and running the project code

What follows is a guide for building and running the stabilisation tool that was created to conduct the research in this project. The following instructions were correct at the time of the report's submission.

Prerequisites

- QT5+ (including qmake)
- OpenCV 2.4+
- Matlab 2012b +
- GCC 4.4 (required by Matlab Engine)
- The project's source code which can be checked out via GitHub at `https://github.com/ozzyoli/motion`.

Compiling

The project can be built using qmake, a tool to simplify the build process of the project across multiple platforms. Every QT project has a .pro file which contains information which the qmake tool uses to build standardised Makefiles. You will need to customise the provided .pro files to point to the relevant locations of the required libraries on your machine.

To run the GUI front-end or the console front-end you will need to first build and deploy the core framework as a static library using qmake and then update the .pro files of the GUI or console applications to the location of the deployed library. Following that you should be able to successfully build and run either application. NB. The location of the Matlab binaries must be in your PATH environment variable at runtime.