DUAL-MODE OPTICAL TACTILE SENSING

Espen Knoop (Engineering Mathematics)

Project Report submitted in support of the degree of Master of Engineering

Supervisor: Dr Jonathan Rossiter, Engineering Mathematics

May 2012
Department of
Engineering Mathematics

Dual-Mode Optical Tactile Sensing

Espen Knoop

Project Report submitted in support of the degree of Master of Engineering
Abstract

Tactile force sensing and compliance are key elements of safe and natural-feeling human-robot interaction. In this project we have developed and tested a complete optical tactile sensor system, in the form of a compliant elastomer ‘fingertip’ tracked by a high-speed low-resolution image sensor with on-board signal processing. The sensor relies solely on an 18×18 pixel camera to track the deformation of the elastomer ‘fingertip’ and thus estimate force.

We have developed a dual-mode bio-mimetic control strategy, mimicking reflexes and voluntary actions in organisms with reflex and explore sensing modes.

The scope of the project has included mechanical design and manufacture of the sensor as well as an automated test rig. We have developed suitable tracking algorithms, and written embedded software for low-level interfacing.

We demonstrate sensing of normal force in both modes of operation, showing that in reflex mode we are able to rapidly detect the presence of forces and compute approximate force estimates while in explore mode we estimate force more precisely. We are able to detect shear displacement with low error in both modes of operation.
# Contents

1 Introduction .................................................. 5  
  1.1 Motivation ........................................... 5  
  1.2 Optical Tactile Sensing ................................. 5  
    1.2.1 Bio-Mimetic Dual-Mode Sensing Scheme ............ 6  
  1.3 Previous Research ..................................... 7  
  1.4 Tasks and Applications ................................ 8  
    1.4.1 Picking Up Objects ................................ 8  
    1.4.2 Assessing Objects ................................ 8  
    1.4.3 Detecting 3D Shape ................................ 9  
    1.4.4 Object Manipulation in Unknown Environments ...... 9  
  1.5 Aims ..................................................... 9  
  1.6 Scope of Work .......................................... 10  
  1.7 Report Structure ....................................... 10  

2 Sensor Design and Manufacture .......................... 11  
  2.1 Camera System .......................................... 11  
    2.1.1 ADNS-2620 Optical Tracking Sensor ................. 12  
    2.1.2 Optical System Properties .......................... 12  
    2.1.3 Optical System Modelling ........................... 13  
    2.1.4 The Developed Optical System ...................... 15  
  2.2 Mechanical Design and Manufacture ................. 16  
    2.2.1 Elastomer Fingertip ................................ 16  
    2.2.2 Skin ............................................... 17  
    2.2.3 Tracking Pattern .................................. 18  
    2.2.4 Body .............................................. 18  
  2.3 Electronics ............................................. 19  
  2.4 Test Rig ............................................... 19  
  2.5 Software ............................................... 22  
    2.5.1 Program Structure ................................ 22  
    2.6 The Developed Sensor ................................ 23  

3 Tracking ..................................................... 26  
  3.1 Tracking Algorithms .................................... 26  
    3.1.1 Tracking with Image Statistics ....................... 27  
    3.1.2 Optical Flow Methods ............................... 27  
    3.1.3 Differential or Direct Tracking ..................... 28  
  3.2 Tracking Patterns ....................................... 29
### CONTENTS

3.2.1 Regular/Irregular ............................................. 29  
3.2.2 Feature Size ................................................. 30  
3.2.3 Practical Considerations .................................... 30  
3.2.4 Possible Pattern Designs .................................... 31  
3.3 Reflex Mode Algorithm ........................................ 32  
3.4 Explore Mode Algorithm ....................................... 35  

4 Characterisation and Testing .................................. 37  
4.1 Force/Displacement Relationship .............................. 37  
4.2 Reflex, Normal Force ......................................... 37  
4.3 Explore, Normal Force ......................................... 38  
4.4 Explore, Force Tracking Test ................................... 39  
4.5 Reflex, Shear ................................................... 39  
4.6 Explore, Shear .................................................. 40  

5 Conclusions ....................................................... 42  
5.1 Key Skills and Challenges ..................................... 43  
5.2 Further Work .................................................... 43  
5.3 Commercialisation of Sensor Design ......................... 44  
5.4 IROS 2012 Paper ............................................... 44
Chapter 1

Introduction

1.1 Motivation

Efficient human-robot interaction requires compliant tactile force sensing. Robots that are able to interact with humans in a natural way would have uses in a wide range of areas, from healthcare to virtual reality simulators. A robot designed for human interaction could have a humanoid form factor, or it could be a robotic arm or manipulator, or indeed be any other shape or form. The most natural means of interaction would seem to be a gripper or robotic hand.

Conventionally, robots are fabricated from hard and rigid materials, such as metals or plastics, which offer great strength and precision. While this is useful for heavy-duty or high-precision tasks, it is less suitable for interaction with humans. We can class human interaction with such robots as hard-soft interaction. The conventional robot is not naturally compliant, so that fast and precise force sensing and position control of joints is required to pick up an object with a desired level of force. Interacting with humans, there is a risk of the robot applying excessive force. Furthermore, the interaction would be perceived as mechanical and artificial, and tasks requiring cooperation between the human and the robot would be difficult.

If robots instead could be made soft and compliant, this would allow for more natural human-robot interaction. Soft-soft interaction would be safer due to the compliance of the robot, and lower the requirements on position control and force sensing. It would also seem that interaction with soft robots would be perceived as far more natural and ‘organic’, making it preferable to hard-soft interaction. Chorley et al. [1] argue that robot manipulators should mimic the human finger to gain the same desirable properties of compliance, conformability and mechanical strength.

Compliance and soft-soft interaction is also preferred in medical and therapeutic applications, for medical devices such as tools or prosthetics, where hard-soft interaction can lead to tissue damage [2].

Force sensing is essential for any robot manipulator interacting with humans, to detect contact and be able to grip objects with a suitable force. Traditional methods of force sensing include capacitive sensors [3], piezoelectric materials [4] and strain gauges [5]. However, these approaches cannot be readily coupled with soft and compliant materials.

1.2 Optical Tactile Sensing

We propose a tactile sensor design based around an optical system, depicted schematically in Fig. 1.1.

The force-sensing fingertip is made of soft, compliant and transparent elastomer, with a durable and compliant opaque skin. A pattern is applied to the inside surface of the sensor skin, which is tracked by
CHAPTER 1. INTRODUCTION

Opaque, compliant skin

Camera

Pattern (on inside surface of skin)

Transparent, compliant elastomer

Figure 1.1: Schematic diagram of the sensor design. The fingertip is made from transparent elastomer, with an opaque rubber skin. A pattern is applied to the inside of the skin, tracked by a camera mounted in the sensor base.

a camera mounted in the base of the sensor. The base also houses an array of LEDs for illumination of the pattern.

Tracking the motion of the sensor skin will allow for estimation of displacement and thus force. This task is computationally intensive, in particular if high precision is required.

1.2.1 Bio-Mimetic Dual-Mode Sensing Scheme

Biological organisms commonly have multiple sensory pathways. When performing voluntary actions such as exploring an environment, sensory information is processed by the brain. This is done at relatively low speed, due to the high cognitive load. Motor commands are then sent to the muscles from the cerebral cortex. On the other hand, organisms are able to respond much more rapidly to stimuli with reflex actions. Here, the control loop passes through the spinal cord only, so that the higher-level cognitive parts of the brain are bypassed [6, pages 91, 94].

We propose a robotic control scheme mimicking this dual-mode system, presented schematically in Fig. 1.2. In explore mode (Fig. 1.2b), the sensor outputs a continuous stream of data which is processed by the central controller. This is analogous to higher level cognitive processing of sensory input in humans. In reflex mode (Fig. 1.2a), basic processing of the sensory input is carried out by the sensor itself. This allows for fast notification of the central controller and for the sensor to send fast commands directly to actuators given some condition on the sensory input, analogous to reflexes bypassing the cognitive parts of the brain in living organisms. Features of the two modes are summarised in Table 1.1.

This dual-mode system presents a number of advantages. Local processing of sensory input removes computational load from the central controller, allowing for a large number of sensors to be implemented in a robotic system and freeing up computational power for other tasks. Bypassing the central controller and sending commands directly from the sensors to the actuators will allow for more rapid responses to stimuli, so that the robot can safely carry out actions faster. A decentralised control structure will also be more robust: the robot can respond to stimuli even in the event of a failure of the central control system.

Table 1.1: Comparison of the sensor’s two modes of operations.

<table>
<thead>
<tr>
<th>Reflex Mode</th>
<th>Explore Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>High speed</td>
<td>Low speed</td>
</tr>
<tr>
<td>Low level of detail</td>
<td>High level of detail</td>
</tr>
<tr>
<td>Detect presence of force, rough magnitude estimation</td>
<td>More accurate estimate of force, force components</td>
</tr>
<tr>
<td>Computed by sensor</td>
<td>Computed centrally</td>
</tr>
</tbody>
</table>
It can be envisaged that the concepts presented here of a multi-level control structure could extend to form a hierarchy of controllers, e.g. a humanoid robot could feature individual sensors in the fingers linked up to respective finger actuators, a central hand controller combining the sensory inputs from all the fingers to allow for the whole hand to respond to reflexes, further connected through multiple levels of control to the central robot controller. This would greatly reduce the cognitive load of the central controller to allow it to focus on higher-level cognitive tasks, and would present a parallel computational scheme mimicking that in living organisms, however the implementation of such a control structure would present challenges due to the increased complexity.

1.3 Previous Research

Optical tactile sensing has been studied previously by Kamiyama et al.\[7, 8, 9\], with the GelForce. The sensor is based around the principle already presented, using a camera to track the motion of the sensing surface. Force is then estimated from the motion. The sensing surface is an elastomer pad, and the pattern is formed of coloured markers suspended in the elastomer. Kamiyama develops the sensor for use as a tactile user interface, a ‘touch pad’. A ‘finger’ for robotic applications, based on the same concept, is also developed.

Obinata et al.\[10\] have developed an optical elastomer ‘fingertip’ sensor where a camera is used to track the motion of markers suspended in the elastomer. Normal force is here estimated from the size of the contact area on the sensor rather than directly from the tracking of a pattern. Shear forces as well as applied moments are also measured. Obinata proposes a grip force control system based around the optical elastomer sensor.

Chorley et al.\[11, 12\] have developed a similar tactile sensing system, where the sensor is formed of a fluid-filled rubber finger with papillae markers protruding on the inside of the sensor skin, and have in particular studied tactile detection of edges. The papillae amplify the movement of the markers under the application of force, thus improving the precision of the sensor.

The method of using a camera to track the displacement of an elastomer is also applied in the
GelSight system developed by Johnson et al. [13] used for capturing microscopic surface geometry. An object is pressed against the elastomer, such that the elastomer surface conforms to the surface of the object. A high-resolution camera captures images of the elastomer surface when illuminated from different directions, and 3D surface models of the object are then constructed from the images.

Alternative tactile sensing approaches have also been researched. An optical tactile sensor using a waveguide probed by compliant sensing elements is presented by Ohka et al. [14]. Due to the rigid waveguide, this design would seem to be less compliant and robust.

Conventional semiconductor pressure sensing methods have also been embedded in elastomer to form compliant tactile force sensors [15].

The sensor developed here is based on the designs of Kamiyama and Obinata, however we introduce the concept of a dual-mode sensing scheme allowing for faster reactions and putting less computational load on the central computer. Furthermore, the work described above uses high-resolution cameras for more precise tracking, creating higher computational loads as well as increasing complexity and cost. We believe that for many applications a low-resolution optical system will provide sufficient precision. Lower resolution will also allow for higher frame rates given a fixed data rate.

1.4 Tasks and Applications

Before specifying the design of our sensor, it seems natural to take a step back and consider possible tasks and applications it could be used for.

1.4.1 Picking Up Objects

Picking up objects would seem an essential task. The robot is required to grip with appropriate force, such that the object is neither dropped nor crushed. For gripping delicate objects, a fast control loop is required. This could be a possible application of the reflex mode presented above.

Appropriate force levels could be determined by identifying the object visually and comparing it to a database of objects for which required gripping forces are known. Alternatively, monitoring the force/displacement characteristics of the gripper could tell the robot if the object is deforming — clearly only suitable for compliant objects. A further possibility could be detection of slip. This would allow for a control strategy where initially a low gripping force is applied and the robot tries to lift the object, ramping up the gripping force until no more slip occurs. Sliding contact between the elastomer fingertip and the object would result in low-amplitude vibration of the fingertip, which could be detected using a high-speed camera.

1.4.2 Assessing Objects

Once an object is picked up, it could be useful for the robot to analyse it. A gripper sensing the complete reaction force and moment vectors could evaluate the weight of the object, as well as the location of the centre of gravity. If the reaction forces and moments are monitored while the object is moved and rotated through space, it will also be possible to evaluate the inertia tensor. This information could be used for object identification, as well as identifying the orientation in which an object is held. Static evaluation of object weight would not require high-speed sensing, however dynamic evaluation of the inertia tensor would require high-speed force sensing coupled with precise movement control.
1.4.3 Detecting 3D Shape

In addition to sensing force, the sensor could estimate the 3D surface structure of encountered objects from the deformation of the pattern. This is the primary aim in the work done by Chorley et al. [11] and Johnson et al. [13].

Algorithms could estimate the full 3D structure, or they could search for straight edges, useful for exploring and determining the size and shape of objects, or they could detect points, e.g. for reading braille.

Detection of 3D structure is a computationally intensive task requiring a high-resolution imaging system.

1.4.4 Object Manipulation in Unknown Environments

The task of picking up an object becomes increasingly challenging if the position of the object is unknown and the object must be localised using touch.

Tactile localisation of the object would require a control loop with low latency to allow for the robot to respond rapidly to collisions. We believe the dual-mode scheme developed here is well suited for this type of task.

Consider the following example of a robotic gripper carrying out this task. As the gripper is moving and trying to localise an object, the force sensor operates in fast reflex mode and will halt the actuators if any sensory input is detected i.e. if the gripper has collided with an object. At this point, the sensor would switch to the slower explore mode, and the gripper would move more slowly in the neighbourhood of the detected object to investigate its shape and size. Once sufficient information about the shape has been obtained, the object could be gripped with a specified force and picked up.

1.5 Aims

Having established some of the applications of a tactile force sensor, as well as the fundamental sensing methodology, we can draw out the requirements of an optical tactile sensor suitable for safe human-robot interaction.

**Compliance.** The sensor should allow for soft-soft human robot interaction. The compliance should be similar to that of a human finger.

**Speed.** For safe interaction, the sensor is required to be fast to allow for fast reactions. Again, the sensor should mimic human performance.

**Force Sensing.** The sensor should be able to detect both normal and shear forces of low magnitude, as would be encountered when interacting with human hands. The sensor should be suitable for manipulation of delicate objects. A suitable test could be to pick up an egg.

**‘Finger’ form factor.** The sensor should be suitable for mounting on the tip of a robotic gripper (‘finger’), in direct contact with the object being manipulated.

**Low cost.** We aim to design a mechanically simple sensor that could be manufactured at a low price-point. A human-like robot would require 10 force-sensing fingers, further emphasising this.

**Robustness.** Hands are exposed to mechanical damage, such as nicks and cuts, and hostile environmental effects such as heat and moisture. A mechanically robust sensor is therefore important.

The aim of this project is to develop an optical tactile sensor meeting the above requirements.
1.6 Scope of Work

The scope of our work will cover design, manufacture and testing of the optical tactile sensor, including:

- Development of a dual-mode bio-mimetic control strategy.
- Mechanical design, including:
  - Design of the optical system.
  - Development of a pattern suitable for tracking.
  - Design of the sensor hardware, packaging the concept into a compact sensor unit.
- Simulation of the optical system, for use as a design aid.
- Manufacture of the optical tactile sensor, including:
  - Moulding of the elastomer fingertip.
  - Machining of the sensor body.
  - Making of circuit boards for the on-board electronics.
- Design and manufacture of an automated test rig for the sensor.
- Software development, including:
  - Embedded programming, for low-level interfacing of sensor and test rig hardware.
  - Programming of higher-level control algorithms and test procedures.
- Development of tracking algorithms for both modes of operation.
- Testing of the force sensor and analysing the results.

1.7 Report Structure

The report is structured approximately in chronological order.

**Chapter 1, Introduction:** We outline the project motivation and present the sensing concept, reviewing previous work. The dual-mode sensing scheme is presented, and we consider possible applications. We also outline the project scope.

**Chapter 2, Sensor Design and Manufacture:** This chapter covers the selection of a camera system, mechanical design of the sensor, development of a servo-actuated test rig and development of electronics and software for interfacing with and controlling the sensor and test rig.

**Chapter 3, Tracking:** We consider the general problem of tracking a pattern in a sequence of images to infer movement. We discuss the properties of different patterns, and how the choice of tracking algorithm affects the choice of tracking pattern. We design the tracking pattern for the sensor. We also develop the reflex and explore mode tracking algorithms for the sensor.

**Chapter 4, Characterisation and Testing:** We first determine the force/displacement characteristics of the sensor. We carry out tests detecting normal force and shear displacement in both modes of operation. We demonstrate that the optical tactile sensor can track force and displacement with relatively low error.

**Chapter 5, Conclusions:** We discuss the project findings and show that the project aims have been achieved successfully. We list some of the challenges encountered in the project. Suggestions for further work are given. We also present a cost estimate for a production-model sensor based on the concept developed in this project.
Chapter 2

Sensor Design and Manufacture

This chapter covers the development of the optical tactile sensor and test rig, including hardware and software.

2.1 Camera System

The choice of camera system for the sensor has a great impact on the entire sensor design, making it a natural starting point for the development process.

The response time of the sensor will be of great importance in many applications. In particular for the dual-mode scheme presented here an essential element of the reflex mode is a rapid response time. The frame rate of the optical system forms a lower bound on the response time — the achievable response time will be higher due to signal processing overhead. For example, an imaging chip with a frame rate of 30 Hz will have a minimum response time of 33 ms. According to Stiehl [16], humans can detect tactile stimuli as two separate events if they are spaced in time by 5 ms; a sensor with similar properties would require a response time of less than 2.5 ms demanding a frame rate of at least 400 Hz. This clearly limits the choice in camera systems, in particular as we are aiming to develop a low-cost device.

The spatial resolution of the tactile sensor will be largely determined by the spatial resolution of the image sensor as projected on to the sensor skin. This is directly proportional to the resolution of the imaging chip, but is also affected by the optical system. There is a trade-off between the spatial resolution of the tactile sensor and the sensing area: a smaller sensing area will allow for higher spatial resolution with the same image chip resolution. For reference, the spatial resolution of the human fingertip with regards to separating tactile stimuli is 2 mm according to Stiehl [16].

The Avago ADNS-2620 [17] optical tracking sensor chip is a low-cost, low-resolution, high-speed image sensor with on-board Digital Signal Processing (DSP) designed for tracking application. This makes it highly suited for the optical tactile sensor, meeting the majority of the requirements specified in the introduction. The on-board DSP is particularly well suited for the dual-mode sensing scheme introduced previously. The achievable sensor precision will be limited by the low resolution of the chip, however we believe the chip offers sufficient resolution for many force-sensing applications.

The ADNS-2620 is an image sensor, not a complete camera unit, so the design of a custom optical system is required. It also requires interfacing to a microcontroller.

The following section describes the ADNS-2620 in detail.
2.1.1 ADNS-2620 Optical Tracking Sensor

The ADNS-2620 optical tracking sensor features an image sensor as well as on-board DSP. The image sensor is 18×18 grayscale pixels with 6-bit Analog-to-Digital Conversion (ADC). The default frame rate of the sensor is 1500 Hz, but this can be increased up to a maximum of 3000 Hz at the expense of image contrast.

Communication with the image sensor is done over a two-wire half-duplex synchronous serial port, with a clock wire and a data wire.

The on-board DSP computes basic image statistics. The image data as well as the image statistics are available at a rate of one statistic per image frame. The computed statistics include:

- **Shear Displacement.** The displacement in the x- and y-directions is computed by the DSP. We expect this to be done using cross-correlation, however the algorithm is proprietary and thus not known to us. When the sensor is requested for x- or y-displacement, it returns the total displacement since the last time that statistic was requested.

- **Maximum and minimum pixel value.** The pixel value of the brightest and darkest pixel are available.

- **Average pixel value.** The average pixel value, scaled by a constant factor. The average is computed as the pixel sum (15-bit unsigned integer) of which the 7 most significant bits are returned.

- **Image quality.** Image quality returns a measure of the number of features visible to the sensor in the current frame. Again, the algorithm is proprietary and not known to us, however we believe it could be related to the entropy of the image.

- **Pixel value.** Entire images can be extracted from the sensor, but are only available one pixel at a time. 324 frames are thus required for each image, yielding an image frame rate of 1500/324 ≈ 4.6 Hz.

The statistics returned by the sensor are listed in Table 2.1.

2.1.2 Optical System Properties

The selected image sensor requires a custom optical system. This section introduces the key elements of optical systems. An in-depth discussion of camera optics is presented by Szeliski [18], from which we present key results.

The optics of a camera can largely be characterised by the following properties.
Field of View. The viewing angle of the camera. A larger field of view will capture a larger area, while a smaller field of view will give more detail on objects in the centre of the area.

Depth of Field. How quickly objects go out of focus as they move away from the focal distance. A small depth of field will cause only objects at distances close to the focal distance to appear sharp while a large depth of field will make the system less sensitive to the object distance.

The properties are mainly determined by the parameters of the optical system, as follows.

Focal Length. The focal length of the lens (or compound lens) determines the distance required from the image plane to the lens for an object at a particular distance to appear in focus. A longer focal length also increases the depth of field.

Aperture. The aperture is the hole through which light travels to reach the CCD. The ratio of focal length to aperture diameter is denoted the f-number of the camera, commonly written as $f/#$ where # is the ratio, and determines the depth of field. A smaller aperture increases the depth of field, however it also requires a longer shutter speed for the image to achieve the same brightness level.

The size of the area on the sensor skin visible to the image sensor is determined by the field of view as well as the distance from the camera to the sensor skin: moving the camera further away will make a larger area visible. In general it would seem appropriate to select a field of view and camera position such that the entire sensing area of the skin is visible to the camera. However, with the selected imaging sensor, such a field of view would make the spatial resolution on the sensor skin very low: each pixel in the image would correspond to a large area of the sensor skin, greatly degrading tracking performance.

For most tracking algorithms, it is beneficial to design the depth of field of the optical system such that the pattern is in focus for any applied force. However, a larger depth of field requires a smaller aperture which demands higher lighting levels and thus increases power consumption. It will be seen that there are also less sophisticated tracking applications where it may be desirable to have a low depth of field, essentially allowing for the normal displacement to be estimated based on the level of defocus in the image.

### 2.1.3 Optical System Modelling

To aid in the design of a custom optical system, we required a model of the optical system.

The simplest camera model is a pinhole camera, with zero aperture and no lens (i.e. infinite focal length). However, this does not model blur due to focusing. To better analyse the performance of the optical system, a more sophisticated model was required.

An improvement on the pinhole model is the thin-lens model. Assuming the lens is of negligible thickness, we can use the lens law equation to estimate the distance an object must be at to appear in focus given the focal length of the lens and the distance from the lens to the image plane, given by

$$\frac{1}{z_o} + \frac{1}{z_i} = \frac{1}{f} \tag{2.1}$$

where $f$ is the focal length of the lens, $z_o$ is the distance from the lens to the object and $z_i$ is the distance from the lens to the image plane.

This allows us to design a system focusing at a particular distance, however it does not give information about the depth of field i.e. how far objects can be from $z_o$ and still appear in focus.

Using the thin-lens assumption, it is possible to evaluate the depth of field, and other distortion effects can also be modelled. However, to make the design process more intuitive and accurate the complete
optical system was modelled from first principles using ray tracing. This allowed for the generation of images as they would be detected by the image sensor, given the parameters of the system.

Ray tracing is commonly used in high-quality rendering of 3D computer-generated graphics. The method is based around modelling the path of individual light ‘rays’ from the object of interest, through the lens system including a set of lenses and an aperture, and onto the CCD array. See [19] for an in-depth description of the technique.

To improve computational efficiency, rays are traced ‘backwards’, i.e. they are emitted from the CCD array. This means that we only evaluate the rays that are ‘seen’ by the camera.

Rays are emitted from the centre point of each pixel in the CCD array for a range of directions distributed uniformly. The pixel colour value is then taken to be the mean colour value of all the rays emitted from that pixel.

Rays not passing through the aperture of the camera are ignored, as well as rays passing through the aperture but not passing through a lens.

We model lenses as spherical, and include the thickness of the lens in the model. Rays passing through a lens are computed by first evaluating the refraction at the air-lens boundary on the near side of the lens and then evaluating the lens-air boundary on the far side of the lens.

As described in [19], denoting the unit direction vector of the ray before refraction by $i$, the unit normal vector of the lens surface by $n$ and the unit direction vector of the ray after refraction by $o$, the law of refraction is

$$ o = \frac{n_1}{n_2} i + \left( \cos(\theta_2) - \frac{n_1}{n_2} \cos(\theta_1) \right) n \tag{2.2} $$

where

$$ \cos(\theta_1) = -n \cdot i $$

$$ \cos(\theta_2) = \sqrt{1 - \left( \frac{n_1}{n_2} \right)^2 (1 - \cos^2(\theta_1))} $$

and the surface normal is in the direction of the ray of the ray i.e.

$$ i \cdot n > 0 $$

The parameters $n_1$ and $n_2$ are the refractive indices of the first and second media respectively.

For the ray tracing tests, we modelled the dot pattern as a planar image positioned in space. The colour value of rays landing on the image were taken as the colour value of that pixel in the image. As the optical sensor is grayscale, the model was designed to generate grayscale images. Extensions of the model to allow for colour images would however only require minor modifications.

Functionality was added in the ray tracing model to allow for compound lenses, i.e. lenses with multiple elements, to be modelled.

The operation of the ray tracing model is demonstrated in Fig. 2.1, where the rays are shown along with the image and CCD array as well as the lens and aperture. Note that the number of elements in the CCD array as well as the number of rays per pixel were decreased for clarity.

We now required the ray tracing model to simulate the behaviour of the actual camera system, including the image sensor and optics. Accurate measurement of the parameters of the camera system was not feasible, so instead approximate measurements and estimates were used to initially set the parameters of the ray tracing system. The parameter values were then fine-tuned by comparing the images generated by the ray tracing model to the images obtained by the real camera system under the same conditions. A dot pattern was created for testing, and a high-quality image of the same dot pattern was used in the ray tracing model.
2.1.4 The Developed Optical System

The camera is required to focus at short distances, so that the tactile sensor can be small. This requires an optical system with a short focal distance i.e. a strong lens. The size of the optics is also clearly of importance as it must fit inside the tactile sensor.

These constraints limited the selection of available optical systems and lenses. The small budget of this project further constrained this selection.

For the low-cost low-precision system developed here, a single-element lens was deemed sufficient. Higher-precision image sensors would clearly benefit from more sophisticated optics.

We obtained a small acrylic lens, with an overall diameter of 5 mm and an estimated focal length of 4 mm. The small size and short focal length made it suited for use in the optical system. We wanted to maximise the field of view, and also for the level of focus to change minimally with movement of the sensor skin. Focusing the system at infinity, i.e. with a distance of 4 mm between the lens and the image plane, maximised the field of view while providing a sufficient level of focus over the range of positions of the sensor skin. The field of view was still sufficiently small to achieve a suitable spatial resolution on the sensor skin. The low resolution of the imaging chip means that small amounts of defocusing will not be noticeable.

The imaging chip featured a relatively large aperture, with a diameter of ~1 mm. While reducing the aperture would have increased the depth of field, the depth of field obtained with the built-in aperture was sufficient for imaging at the distances required for the tactile sensor.
CHAPTER 2. SENSOR DESIGN AND MANUFACTURE

Figure 2.2: Comparison of images from the optical sensor with images generated using the ray tracing model. The images on the left were taken experimentally with the optical sensor. Note that \( z \) is the distance from the pattern to the image plane. There is seen to be very good correspondence between the model and the optical sensor, including blur and field of view for both distances. The ray tracing does not model uneven lighting: the left side of the image in Fig. 2.2c is brighter and this is not seen in Fig. 2.2d.

2.2 Mechanical Design and Manufacture

The tactile force sensor developed here is a prototype and a proof-of-concept intended to demonstrate the dual-mode sensing scheme. The design was heavily constrained both with regards to funds and time available, however we still strived to develop a compact and functional sensor unit. We focussed on a design suitable for mounting on a robotic gripper, for ready practical application. Clearly, a production model of the sensor could be better tailored in terms of dimensions and mounting for the required application.

Fig. 2.3 shows a computer model of the designed tactile sensor. A single-piece body houses the imaging sensor and lens, with the elastomer fingertip in front of it, and a rubber skin covers the fingertip and body.

2.2.1 Elastomer Fingertip

The material of the fingertip clearly has a great impact on sensor characteristics. We require a material with the desired level of compliance that melts at moderately high temperatures (i.e. a thermoplastic) for simplified manufacture. Some elastomers cure chemically, which would make the moulding more difficult. We found a Terpene Phenolic Resin with the desired properties, including a compliance similar to that of human skin. Although the elastomer is red in appearance, it was found to appear transparent to infra-red light, which can be readily detected by the image sensor.

The elastomer was melted and moulded into the shape of a fingertip. This process required great care, both during the melting and pouring processes, to ensure a fingertip free of air bubbles. Air bubbles could make tracking impossible, and large air bubbles would also alter the force/displacement characteristics.
To ensure good optical and mechanical performance, we require a transparent rigid backing plate supporting the elastomer. An air pocket between the elastomer fingertip and the sensor body would deform under load, not allowing light to pass straight through. Furthermore, an exposed elastomer surface would rapidly see a build-up of dust and contamination which would severely affect optical performance. An acrylic disc, the same diameter as the sensor body, was made for this purpose. This can be seen as the layer between the sensor body and the elastomer in Fig. 2.3. The plate was placed on the molten elastomer before it set, bonding the two. A circular pattern of slots was cut near the outside of the disc to improve adhesion between the elastomer and the smooth plate. Again, great care was required when placing the plate on the elastomer to ensure no air bubbles.

Fig. 2.4 shows photos of the fingertip production process.

2.2.2 Skin

An opaque compliant skin is required to cover the elastomer fingertip. We obtained a ready-made rubber skin of suitable dimensions, made to cover the human fingertip and used e.g. by office workers for sorting through sheets of paper.

Initially, attempts were made at painting the tracking pattern directly on to the skin before pouring the molten elastomer into the skin to form a fingertip with skin attached. However, it was found that when applying shear forces to the sensor with low levels of normal force delamination between the elastomer and the skin occurred i.e. the skin moved without the elastomer. This would suggest that air pockets between the elastomer and the skin were introduced at manufacture or during use, and will degrade optical performance. For tracking purposes as well as consistent force/displacement characteristics it is
essential that the skin remains attached to, and moves together with, the fingertip. A possible solution could have been the use of an adhesive between the elastomer and the skin, however this would have to be transparent to not obstruct the dot pattern as well as soft and compliant when dry in order to not alter the sensor characteristics.

Instead, the delamination of the elastomer and skin was used to our advantage by using the obtained rubber skin as a mould for the elastomer, see Fig. 2.4a. The molten elastomer was poured into the skin, ensuring no air bubbles, and allowed to set, before the solidified elastomer was removed from the mould. The tracking pattern was then applied directly to the elastomer before a black liquid rubber skin (Plastidip) was applied in multiple coats. The liquid rubber was found to adhere well to the elastomer, eliminating the problem of delamination. Furthermore, the liquid rubber skin is more compliant than the original skin, making the sensor better suited for low-force applications.

2.2.3 Tracking Pattern

The design of the tracking pattern is discussed in Chapter 3 as it is closely linked to the design of tracking algorithms.

A semi-random pattern of \( \sim 1 \) mm white dots spaced \( \sim 2 \) mm apart was hand painted onto the elastomer before the rubber skin was applied.

A more controlled method of application, such as printing or spraying with a stencil, would have allowed for more sophisticated pattern designs.

An alternative to painting the tracking pattern is to suspend markers in the elastomer, as done by Kamiyama et al. [7]. This would make the pattern more robust against mechanical damage. However, it would increase the complexity of the manufacturing process and for the sensor developed here the pattern painted on the surface of the elastomer performed well.

2.2.4 Body

The sensor body is required to house the electronics and optics as well as defining the sensor form-factor. The electronics include the optical sensor as well as a circuit board with LEDs for illumination. The required dimensions of the optical system constrain the design of the body.

We machined the body from aluminium, as for a single prototype this was the most efficient means of production. Aluminium was chosen as it is easily machineable while offering good mechanical strength.

We also explored the alternative of using a Rapid Prototyping Machine, or 3D printer, to fabricate the sensor body. This would have readily allowed for more complex body geometry, however the design we developed did not require this and was well suited for machining. While in general the cost of machining
is far greater than the cost of 3D printing, the fact that we could do the machining ourselves made this the faster and less costly alternative.

The production drawing of the body has been included in Fig. 2.5.

### 2.3 Electronics

For communication with the ADNS-2620, a microcontroller board from RIKEN BMC based around the Microchip dsPIC30F6012A was used. This was also used to control the test-rig (Sec. 2.4). The dsPIC microcontroller board featured PC communication by means of a USB connection with a virtual serial port.

The two-wire serial communication with the ADNS-2620 was implemented using bit-banging with general-purpose I/O-ports.

For illumination of the dot pattern, a LED-board was made. This was designed with a circular layout to fit with the form factor of the tactile sensor and with the LEDs evenly distributed for uniform lighting. High-power surface mounted infra-red LEDs were used, as the image sensor was found to respond well to infra-red light. Individual current-limiting resistors were used for uniform brightness. In order to control the overall brightness, LED-boards configured with 3, 4 and 6 LEDs were tested. The configuration with 6 LEDs was found to perform well.

The wiring schematic for the tactile sensor is presented in Fig. 2.6. Note that the LED-board and ADNS-2620, along with the crystal and capacitors, were housed in the sensor ‘finger’ with the microcontroller being external.

### 2.4 Test Rig

In order to calibrate and test the developed sensor, we required controlled movement and precise control of position and force. While manual control and measurement of position would have been possible, it was decided to automate this to allow for streamlined experiments collecting large numbers of data points.
Ideally, we would have mounted the tactile sensor on a 6-axis load cell and measured the full force and moment vector. A robotic manipulator could have been used for actuation, allowing for realistic gripping tests as well as application of any force and moment to the sensor. However, the limited budget of this project necessitated a much more restricted testing approach.

A test rig for the tactile sensor was developed, as shown schematically in Fig. 2.7. This figure also defines the coordinate axes as used in the tests done later. The test rig has two configurations: for normal force tests the normal force exerted on the tactile sensor is measured and the normal (z) direction of motion is actuated. The force gauge has a resolution of 0.01 N. For shear force tests, the sensor is actuated in the normal direction and pushes against a plate actuated in the shear (y) direction. The tactile sensor, and the plate it pushes against in the shear configurations, are mounted on linear sliders that constrain the motion while allowing for smooth low-friction movement. The linear sliders also feature position measurement.

It would have been preferable to simultaneously measure shear force and shear displacement to allow for a more complete set of tests to be done, but this would have greatly increased the complexity of the test rig and was not feasible.

For actuation of the linear sliders, we used Radio Control (RC) servos. RC servos provide low-cost position-controlled actuation at sufficient force levels, and are also relatively simple to interface. The servos require a 5 V supply, and the position is controlled by a Pulse Width Modulation (PWM) signal with a fixed period of 20 ms and a duty cycle length between 0.5 ms and 2.5 ms specifying the position. While the servos are position controlled, backlash in the servo gears and linkage meant that using the servo position to determine displacement would have introduced significant error.

For position measurement, digital vernier calipers were mounted on the linear sliders. The caliper mounts had to be rigid in the direction of motion with minimal backlash for precise position measurement, however some flexibility of the mount was required for smooth low-friction operation. Initially a rigid mount was tried, however this caused the caliper to bind and not move smoothly.

The calipers provided position measurement with an error of 10 \( \mu \text{m} \). We conservatively estimate that the error in measuring the position of the tactile sensor and shear displacement is less than 50 \( \mu \text{m} \), with the increase in error arising due to the caliper mounting.
In addition to a 7-segment display showing the position, the digital vernier calipers output the position over a 2-wire serial connection with a signal level of 1.5 V and a refresh rate of approximately 3 Hz. While this refresh rate is comparatively low, it still is a large improvement over taking readings manually. It was assumed that the tests carried out would mainly involve motion along one axis (y or z) with the position of the other changed incrementally, so it was seen as sufficient to interface one of the calipers to the microcontroller, switchable between y and z, and taking readings off the other manually.

The caliper and servos were interfaced to the microcontroller allowing for automated experiments with minimal time lag, for which a circuit board was made. The wiring schematic is shown in Fig. 2.8.

The servos required a higher current than what was available from the microcontroller, so an external power supply was needed. The PWM control signal was generated using the output compare module of the dsPIC microcontroller. Support for velocity control of the servos was provided by ramping the PWM duty cycle up or down to the desired value, with the slope of the ramp determining the servo velocity. This simplified the design and execution of tests.

For the caliper, the 1.5 V serial signal needed to be converted to 5 V for interfacing with the micro-
controller, for which transistors were used as shown in Fig. 2.8. This conversion also inverts the signal. Note that internal pull-up resistors on the relevant inputs of the dsPIC were enabled. The data format on the serial port is somewhat crude: the caliper generates the clock signal and every ~300 ms a single signed 24-bit integer is sent with the position of the caliper. Communication with the caliper was again done using bit-banging, and the reading of the clock signal was done with a change notification pin.

2.5 Software

For the experiments carried out in this project, MATLAB was used to communicate with the dsPIC over the virtual serial port. Tests and algorithm development was carried out using MATLAB, as was implementation of the explore mode algorithm. The reflex mode algorithm was implemented directly on the microcontroller. Low-level interfacing to the test rig was also done with the dsPIC. The microcontroller was programmed in embedded C.

2.5.1 Program Structure

The dsPIC was essentially used as an interface between MATLAB and the ADNS-2620 and test rig. This allowed for the majority of testing and development to be carried out in MATLAB, simplifying and speeding up the development process considerably.

An encoding of commands for the dsPIC was developed, presented in Table 2.2. The commands were sent as ASCII text over the serial connection. No start or stop characters were used, requiring the command format to be followed exactly including the number of digits. A set of MATLAB wrapper functions were written to carry out the serial communication, ensuring strict adhesion to the command format.

For both the PWM output and the reading of the caliper serial port the dsPIC was required to act independently of the commands issued by MATLAB. This was achieved using timer interrupts and change notification interrupts.
Table 2.2: List of commands for controlling the optical tactile sensor and test rig. Commands are sent as ASCII text over a serial connection. Numbers (#) are also encoded as ASCII text.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optical Sensor Commands</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>Get MAX</td>
<td>8-bit unsigned int</td>
</tr>
<tr>
<td>N</td>
<td>Get MIN</td>
<td>8-bit unsigned int</td>
</tr>
<tr>
<td>X</td>
<td>Get DX</td>
<td>8-bit signed int</td>
</tr>
<tr>
<td>Y</td>
<td>Get DY</td>
<td>8-bit signed int</td>
</tr>
<tr>
<td>Q</td>
<td>Get SQ</td>
<td>8-bit unsigned int</td>
</tr>
<tr>
<td>S</td>
<td>Get AVG</td>
<td>8-bit unsigned int</td>
</tr>
<tr>
<td>I</td>
<td>Get entire image (uses 324 frames)</td>
<td>324 × 8-bit unsigned int</td>
</tr>
<tr>
<td>R</td>
<td>Soft reset of optical sensor</td>
<td>—</td>
</tr>
<tr>
<td>F###</td>
<td>Set sensor frame rate to ### (range 500-3000 Hz)</td>
<td>—</td>
</tr>
<tr>
<td>D##</td>
<td>Get ## complete sets of statistics (equivalent to ## × C, X, Y, M, N, Q, S ).</td>
<td>See respective commands.</td>
</tr>
<tr>
<td><strong>Servo Commands</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A###</td>
<td>Move servo S1 to ### (0—255)</td>
<td>—</td>
</tr>
<tr>
<td>B###</td>
<td>Move servo S2 to ### (0—255)</td>
<td>—</td>
</tr>
<tr>
<td>U###</td>
<td>Set servo S1 velocity to ### (0—255)</td>
<td>—</td>
</tr>
<tr>
<td>V###</td>
<td>Set servo S2 velocity to ### (0—255)</td>
<td>—</td>
</tr>
<tr>
<td>a</td>
<td>Get current servo S1 position (0—255)</td>
<td>8-bit unsigned int</td>
</tr>
<tr>
<td>b</td>
<td>Get current servo S2 position (0—255)</td>
<td>8-bit unsigned int</td>
</tr>
<tr>
<td><strong>Caliper Commands</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Wait for caliper to refresh, return new caliper position (unit: 0.01mm).</td>
<td>16-bit unsigned int</td>
</tr>
<tr>
<td>Z###</td>
<td>Set servo S1 position to ###, wait for move, return final caliper value.</td>
<td>16-bit unsigned int</td>
</tr>
</tbody>
</table>

Data sent from the *dsPIC* to *MATLAB* was encoded as binary numbers rather than ASCII characters for improved communication speed.

An alternative program structure, tried initially, featured the *dsPIC* generating a continuous stream of data. This would have been expected to improve the data rate, removing the overhead of *MATLAB* having to continuously request data. However, a continuous data stream would be difficult to integrate with the sensor switching between reflex and explore modes. Using the *dsPIC* for control of the *ADNS-2620* as well as the test rig would most likely have required hard-coding of the test procedures on the *dsPIC*. Interpreting the continuous serial data stream with *MATLAB* was also difficult; data bytes were frequently missed while *MATLAB* was processing or plotting data. Overall, the protocol of *MATLAB* requesting data from the *dsPIC* was found to work far better, and the data rates achieved were very close to the maximum data rates specified by the optical sensor.

### 2.6 The Developed Sensor

A photo of the finished optical tactile sensor is shown in Fig. 2.9.

In Sec. 1.5, we stated that the compliance of the sensor should be similar to that of a human fingertip. Fig. 2.10 shows the deformation of the sensor (Figs. 2.10c and 2.10d) and a human finger (Figs. 2.10a and 2.10b) when a normal force is applied, and it can be seen that the compliance is similar.

Photos of the test rig in normal and shear configurations are shown in Fig. 2.11.
Figure 2.9: The developed optical tactile sensor.

(a) Human finger, 0 N.  
(b) Human finger, 5 N.  
(c) Tactile sensor, 0 N.  
(d) Tactile sensor, 5 N.

Figure 2.10: Comparison of a human fingertip and the developed tactile sensor under normal force. It can be seen that the deformation of the sensor is similar to that of the human finger.
(a) Test rig configured for normal force testing.

(b) Test rig configured for shear testing.

(c) Tactile sensor mounted in test rig.

(d) Normal and shear force applied to the tactile sensor.

Figure 2.11: Photos of the developed test rig.
Chapter 3

Tracking

In this chapter, we initially discuss tracking algorithms in general, classifying them into 2 main groups. We then go into more detail with regards to the tracking algorithm design. Next we consider tracking patterns, and how different patterns are suited for different tracking algorithms. We also design the pattern used in the tactile sensor. Finally we develop the tracking algorithms for the reflex and explore modes of the tactile sensor.

3.1 Tracking Algorithms

Different tracking algorithms may be preferred depending on the application as well as the optical and mechanical design of the sensor.

We can class tracking algorithms into 2 groups. For the first group, we make the assumption that force is applied uniformly to the sensor, which would be the case when manipulating approximately planar surfaces. In general, the sensory input will then be completely described by a force vector \( \mathbf{F} = (F_x, F_y, F_z) \), with normal force and two components of shear force, and a moment vector \( \mathbf{M} = (M_x, M_y, M_z) \) with the applied moments about the coordinate axes.

Alternatively, we can track the force distribution across the surface of the sensor. We note that the elastomer fingertip will couple nearby points: an applied point force will result in the displacement of a larger segment of the sensor skin. In the most general case a force field \( \mathbf{F}(x, y) \) could be tracked, as we can express applied moments as force distributions. Practical applications would require the extraction of higher-level features, such as point forces or lines/edges, from the force distribution.

We can class previous research in this way. The Gelsight system [13] determines the 3D structure of an object pressed into the elastomer, essentially the data required for a full normal-force distribution. Chorley et al. [11] focus on the detection of edges as well as other higher-level features such as the distribution resulting from a door knob. Kamiyama et al. [7] look at uniform shear forces and moments applied to the sensor.

Tracking the motion of a dot pattern allows for estimation of displacements, so in order to estimate force a relationship between force and displacement is required. As the elastomer body undergoes large deformations, this mapping is likely to be highly non-linear. Moreover, there will be a cross-coupling between the elements of the force vector. For example, in the case of a uniformly applied force, the magnitude of the normal force will have an effect on the relationship between shear force and shear displacement. For smaller forces a reasonable assumption may be independence of the force components, which would greatly simplify characterisation of force-displacement properties. The force-displacement characteristics could be determined experimentally or analytically, although with the assumption of cross-coupling characterisation based solely on experimental data could be difficult.
For the case of a force distribution, the task of inferring force-displacement characteristics accurately appears very complex if not intractable. However, we note that for a large number of applications the displacement distribution on its own may well be sufficient. A displacement distribution would allow for the 3D shape of the object to be inferred, as well as detecting point loads, moments and edges.

A tactile sensor could utilise multiple algorithms, e.g. estimating the displacement distribution as well as an overall force vector.

3.1.1 Tracking with Image Statistics

As a simple and computationally inexpensive tracking method, image statistics may in some cases be sufficient for estimation of normal force.

A very basic example could utilise a single pixel, a light source and no pattern. The overall brightness level would be expected to correlate with normal displacement: the sensor surface appears brighter when it is closer to the light source.

If the optical system was designed to have a small depth of field, the level of blur in an image would correlate well with normal displacement. Estimation of the amount of blur could be done in the frequency domain by looking at the presence of high-frequency components, or by evaluating the image gradient and finding the maximal values, or otherwise. For this method to be reliable, the pattern must be such that the value of the image statistics are independent of the shear displacement of the pattern.

Image statistics are less suited for tracking of shear force. It could be possible, if a pattern was created where properties of the pattern varied with position. Note however that for tracking of normal displacement the requirement was for the pattern properties to be position invariant. One possibility could be to use a colour camera along with a pattern coloured by position and then use the colour level to estimate shear and a monochrome version of the image to estimate normal force.

It would seem that for a computationally inexpensive reflex mode algorithm the statistics generated by the ADNS-2620 are suitable for tracking.

3.1.2 Optical Flow Methods

Here we present a brief overview of optical flow methods, condensed from the analysis by David J. Fleet [21].

The concept of optical flow allows for estimation of displacement between frames and thus the velocity of objects in the image. It is based on the assumption that pixel intensities are conserved between frames.

We assume a pinhole camera and small z-displacements, see Szeliski [18] for details on camera modelling. Furthermore, we assume that pixel intensities are transferred from one image to the next such that

\[ I(x_{im}, t) = I(x_{im} + u_{im}, t + 1) \] (3.1)

where \( I \) is pixel intensity as a function of space and time, \( x_{im} = (x_{im}, y_{im}) \) is position in the image plane, \( u_{im} \) is the velocity in the image plane and \( t \) is time. Note that we write image plane variables with superscript \( im \).

A Taylor-series expansion of Eq. 3.1 gives the Optical Flow Equation (OFE)

\[ \nabla I \cdot x_{im} + \frac{\partial I}{\partial t} = 0. \] (3.2)

To use the OFE for velocity estimation, an assumption on the velocity distribution in the image is required as pixel intensities will not be transferred perfectly. A general form of the velocity distribution
is assumed, and the most likely velocity distribution of that form is found using optimisation. The form assumed greatly influences the computed velocity field, as well as the computational cost.

The simplest velocity model assumes the velocity field is constant over the entire image, however this will only be true for pure shear motion. A more useful model assumes an affine velocity distribution over the image, which can be shown to represent the motion of a plane. For an optical tactile sensor where a uniform force is assumed, a moving plane would seem to capture the motion of the fingertip well. An affine model is assumed in the explore mode algorithm developed here.

For estimation of velocity distributions, further extensions are required. One option is to parametrise the velocity distribution surface and use optimisation methods to select the best parameters. More generally, the Horn and Schunck method of Global Smoothing finds the optimal velocity distribution with the constraint of continuity. As the surface of the fingertip is guaranteed to be continuous, the Horn and Schunck method would appear highly applicable for medium- to high-resolution tactile sensors. For the low-resolution sensor developed here, there would not appear to be sufficient information in the image to estimate a velocity distribution reliably.

More sophisticated optical flow methods exist, but at a higher computational cost. The problem of tracking a pattern applied to an elastomer is a specific case of motion estimation where the motion is constrained and the overall variation in the images is small as we are always ‘looking at’ a pattern, and is thus far simpler than the general motion tracking problem.

### 3.1.3 Differential or Direct Tracking

A solution of the optical flow equation will give the velocity of the motion field between frames. If consecutive frames are used, this is equivalent to finding the velocity of the pattern. We wish to track the displacement of the pattern, so tracking using the velocity can be classed as a differential approach. The displacement must be computed by numerical integration of the velocity, making it prone to numerical error and drift. The effect of integration drift is that if a force is applied to the sensor and then removed, such that the sensor surface is momentarily displaced from its rest position, the tracking algorithm may not return to zero. This has severe implications for the performance of the sensor: a robotic manipulator handling a delicate object requiring constant force might, due to integration drift, apply increasing force (crushing the object) or decreasing force (dropping the object) over time.

As an alternative to evaluating the motion field between consecutive images, we propose considering the motion field between the current image and a base image taken with the sensor in rest position. Thus, the obtained ‘velocity’ will instead be the displacement between the two images so that the tracking is done directly. Absolute, rather than relative, tracking of position means that integration drift is eliminated, making it preferable. However, evaluation of the motion field between images will only produce meaningful results if the motion is sufficiently small so that a significant number of features are visible in both images. With differential tracking, this will be the case unless the velocity of the pattern is very high or the frame rate is very low. Direct tracking, however, will only work for sufficiently small displacements of the dot pattern from the rest position. An optical system with a larger field of view, such that a larger proportion of the fingertip is shown in the images, would allow for larger displacements to be tracked directly.

A diagram illustrating this difference is shown in Fig. 3.1.

The algorithm developed for the prototype switches between differential and direct tracking, attempting to combine robust tracking with zero integration drift.
3.2 Tracking Patterns

The design of the tracking pattern is closely coupled with the development of the tracking algorithm and the available camera system. In this section we discuss general pattern properties, we present some possible pattern designs and we select a pattern for the developed tactile sensor.

3.2.1 Regular/Irregular

If a regular pattern was implemented, such as a chequerboard or a grid of lines, the tracking algorithm could be built around a priori knowledge of the pattern. Changes in the size of pattern features would allow for accurate estimation of the normal displacement of the pattern, both for the whole surface moving together and for segments of the surface being tracked separately.

However, a self-similar regular pattern is prone to errors when determining the sideways displacement: the tracking will be prone to ‘aliasing’ if the pattern moves more than half the pattern width. This is also true for rotations: tracking of a square pattern will fail if the angle of rotation is greater than 45° between frames.

An irregular pattern is resistant to errors due to aliasing, as it is not self-similar. Thus, it is preferable for tracking of shear displacements. If the entire pattern was known to the tracking algorithm, accurate absolute tracking of the pattern would be possible: given an image of a pattern segment it would be possible to determine the absolute location of the sensor surface.

However, tracking surface segments individually with an irregular pattern would be a more computationally intensive task. If an image sensor with sufficiently high resolution was used, a possible approach could be to consider each segment separately, assume planar segments and for each segment find the position of the pattern with the best correspondence. For sensors with lower resolution, there will clearly be a limit as to how well it is possible to track the surface, however a tracking performance may be improved by using the property that the surface is continuous and modelling the surface shape using continuous functions such as polynomials.
3.2.2 Feature Size

The size of pattern features will have an effect on the achievable resolution of the optical sensor. A lower bound on the feature size is given by the spatial resolution of the image chip: by the Nyquist limit the size of pattern features in the images captured by the image sensor must be greater than one pixel. Thus, the minimal feature size will depend on the resolution and field of view of the camera as well as the maximal distance between the camera and the pattern.

Blur due to the pattern being out of focus is a spatial low-pass filter, so for the pattern to be distinguishable when not in focus it is desirable to have a pattern with significant low-frequency components i.e. large features.

If we consider the application of tracking normal force distributions, assuming no shear force, we can represent the force distribution in the spatial frequency domain. Then the maximum frequency detectable will be determined by the size of the pattern features. An absolute upper bound is again given by the Nyquist limit, however in practice the maximum frequency will be far lower as a reasonably large region is likely to be required for the tracking algorithm to work reliably.

A further consideration is the spatial frequency response of the elastomer. Consider the displacement of the sensor skin under the application of a point force, i.e. the spatial impulse response of the elastomer. We assume that the point force does not pierce the sensor skin. The point force will deform a large area of the sensor skin in the region around it. The elastomer is thus functioning as a low-pass filter, removing high frequencies from the force distribution. This thus puts a further physical limit on the resolution achievable with the tactile sensor.

3.2.3 Practical Considerations

The tactile sensor design presented here has the potential to be robust with regards to mechanical damage. A soft robotic manipulator would be prone to cuts and nicks if it was used for handling rough objects. However, the sensitive components i.e. the optics and the electronics are housed in the base of the sensor, and the exposed elements are the skin, the elastomer and the pattern.

If a sensor was designed around a regular pattern, known a priori to the tracking algorithm, then any changes in the pattern due to mechanical damage would make the tracking algorithm prone to failing. The use of an irregular pattern along with a tracking algorithm not relying on having the pattern hard-coded would make the sensor more resistant to damage to the pattern.

The manufacture of the pattern presents a further challenge. Depending on how the fingertip and skin are manufactured, it could be possible to print or paint the pattern on to either the outside of the elastomer fingertip or the inside of the skin. To obtain a regular pattern, automated printing would be preferable but spray painting using a stencil could also be possible. A random or semi-random pattern is far easier to manufacture, as precision in the application of the pattern is not required.

It is essential that the contrast in the pattern is as great as possible, so that the dynamic range of the imaging device is exploited. Heavily pigmented opaque white paint on a black background was used for the prototype developed here.

An alternative offering increased resistance to mechanical damage, used by Kamiyama et al. [7], is to suspend coloured markers in the elastomer fingertip close to the surface. This could have an adverse effect on tracking performance as the motion of the dots will not exactly match the motion of the sensor surface, however provided the dots were sufficiently close to the surface this effect should be small. Manufacture of the fingertip would also be more complex, with one possibility being to lightly attach the markers to the surface of the fingertip mould before moulding the fingertip. It could also require moulding in multiple stages. However, for a production model where robust performance is important this approach could be preferable.
3.2.4 Possible Pattern Designs

Here we present some possible pattern designs for different tracking applications.

Regular Patterns, Uniform Force

A very simple pattern design is shown in Fig. 3.2a. The tracking algorithm could use a Hough transform to locate the two lines, from which the absolute shear displacement could be found. The pattern is self-similar when rotated 90°, so we can track rotations of less than 45° between frames. However, the pattern is scale invariant assuming that we cannot detect the line thickness accurately, so that it does not offer any means of tracking the normal displacement.

To allow for tracking of normal displacement, we can add a circular feature to the pattern as shown in Fig. 3.2b. This maintains the properties of the previous pattern, but the size of the circular feature can now be used to track normal displacement.

Both patterns so far have been formed of lines, which are easily detectable provided the camera has sufficient resolution. For low-resolution cameras, a better alternative could be the pattern shown in Fig. 3.2c. This pattern is self-similar when rotated 180°, so rotations of less than 90° between frames can be tracked. Absolute tracking of shear will be possible, but again the pattern is scale invariant so that normal displacement cannot be tracked. A generalised Hough transform could be suitable for tracking of this pattern.
Regular Patterns, Force Distribution

The three patterns above require the assumption of a uniform force. For tracking of force distributions, the patterns in Fig. 3.2d and Fig. 3.2e are perhaps the most basic. Both offer a way of determining how the elastomer is stretched and displaced over the fingertip. The latter is similar to the patterns used by Kamiyama et al. [17] and Chorley et al. [11], and it is possible that dots are more readily tracked than the grid of lines. Both patterns are translation invariant so that relative tracking will be required with a frame rate such that the displacement of the pattern between frames is less than half the pattern spacing.

By making the central pattern feature unique (Fig. 3.2f) we eliminate the translation invariance of the pattern so that absolute tracking of shear displacement is possible. With regards to tracking of force distributions with regular patterns, this pattern could be a good candidate.

Irregular Patterns

As discussed previously, irregular patterns do not have aliasing problems and changes in the pattern (from manufacture or through mechanical damage) will affect tracking performance to a lesser extent. Irregular patterns are suited for tracking with optical flow methods or image statistics.

A natural irregular pattern to consider is a random dot pattern. Dots could be uniformly sized (Fig. 3.2g) or of multiple sizes (Fig. 3.2h). A fixed dot size, selected based on the camera resolution for optimal tracking, would ensure crisp representations of dots in the images, while multiple dot sizes would allow for greater variation between areas of the pattern ensuring no self-similarity and error-free absolute position tracking.

The Developed Pattern

It would have been preferred to apply the dot pattern in an automated printing process, to allow for accurate control of the pattern design, however this was not possible within the budget and time constraints of this project.

For the low-resolution camera system used here, and to maximise robustness, an irregular pattern seemed better suited. Tests were done with random dot patterns, both with random dot sizes and with a fixed dot size.

With fixed-size dots, a semi-random pattern of $\sim 1$ mm diameter white dots spaced $\sim 2$ mm apart hand painted on a black background was found to performed well.

With random-sized dots, the pattern was applied by spraying the finger very lightly using spray paint, reducing the air pressure in the spray to create sputter. While this simplified the manufacturing process, the lightness of the coat reduced the colour contrast in the pattern as the dots were not fully opaque.

Tests were done with both patterns, and while both patterns were tracked reasonably well the pattern with fixed-size dots was found to yield better results and was thus used in the sensor. This pattern is shown in Fig. 3.2i.

3.3 Reflex Mode Algorithm

We develop both the reflex and explore tracking algorithms to estimate normal and shear displacement, rather than force. The normal force/displacement characteristics are then found experimentally (Sec. 4.1). It is more natural to estimate displacement from images, and this divide is also better suited for testing given the functionality of the developed test rig.

For the fast operating mode, we wish to mimic reflexes and involuntary actions in living organisms. The sensor should therefore respond to stimuli as quickly as possible and output an approximate estimate
of the magnitude of the applied force. The force estimate must be computed by the sensor, requiring the force estimation algorithm to be computationally inexpensive.

We use the image statistics computed by the DSP and supplied by the ADNS-2620 (see Table 2.1) to estimate normal and shear displacement, and then estimate normal force from normal displacement with an empirical relationship (Sec. 4.1).

The force estimate from the reflex algorithm is not expected to be precise, as discussed previously. For many applications, in particular where the reflex algorithm is used to detect collisions, it will be sufficient to detect the presence of a force. We thus split the reflex algorithm into detecting the presence of a force, where low error rates are essential, and estimating the magnitude of the displacement, where precision is less critical.

For shear, the $\text{DX}$ and $\text{DY}$ statistics are expected to be suitable for detecting both the presence of a force and the magnitude of the shear displacement. We estimate the displacement with numerical integration, computing $\sum \text{DX}$ and $\sum \text{DY}$. Note that this makes it prone to integration drift. Assuming a pinhole camera, by similar triangles, the relationship between $\sum \text{DX}$ and $x$ is expected to be proportional to the distance from the image plane to the pattern so that

$$x = k_r (Z - z) \sum \text{DX}$$

$$y = k_r (Z - z) \sum \text{DY}$$

where $k_r$ is a constant, $Z$ is the distance from the image plane to the dot pattern with no force applied and $z$ is the normal displacement of the dot pattern (positive towards the camera). The value of $k_r$ will depend on the image sensor size as well as the scaling in the pattern used to compute $\text{DX}$ and $\text{DY}$ and must be found experimentally.

We wish to determine the normal displacement from the remaining 4 statistics. Fig. 3.3 shows the variation of the image statistics with sensor position for a dataset of 1000 points taken with the sensor undergoing slow reciprocating motion ($\sim 2$ mm/s) of the fingertip into a flat surface. Note that zero displacement is taken to be when the sensor first comes into contact with the flat plate. While some correlation is seen between the statistics and normal displacement, the data is very noisy. The sensor statistics have not been designed to track normal displacement, so high noise levels are expected. Thus, when approaching this problem, we must take care not to over-fit the data. This makes mathematically simple estimation methods preferable.

Detecting initial contact and the presence of a force can be considered a binary classification problem: we wish to classify data points as either ‘force present’ or ‘no force present’. Because the algorithm is required to be computationally simple, we decided to use a decision tree for the classification. Decision trees allow classification of data points using a small number of logical tests, making them computationally simple. For an in-depth explanation of decision trees see [22].

Using the dataset presented in Fig. 3.3 a decision tree was generated using built-in MATLAB functions. Some pruning of the tree was done, to further speed up point classification. The resulting decision tree is presented in the following pseudocode.

```matlab
if (MAX>40.5 OR (MAX>38.5 AND SQ<81.5))
    Force Present
else
    No Force Present
end
```

We notice that classification is done primarily based on the value of $\text{MAX}$, and from Fig. 3.3 it is seen that this statistic features the highest correlation with displacement.
As will be seen in Sec. 4.2, the decision tree achieves very low error rates despite its simplicity. The simplicity also ensures that we are not over-fitting the data: we would expect the results to hold for minor changes in sensor and pattern properties.

If a data point is classed as ‘force present’, we wish to then estimate the magnitude of the force. This must also be carried out in a computationally efficient manner. Force was taken to be a linear function of the statistics, and minimising RMS error on the dataset presented in Fig. 3.3 the optimal function was found to be

$$F = -1.26 + 0.32 \text{MAX} - 0.70 \text{MIN} + 0.11 \text{SQ} + 0.11 \text{AVG}$$

The force values were estimated from the $z$-displacement using the empirical relationship given in Eq. 4.1, see Sec. 4.1.

Note that this was fitted to the data points with force present — points with no force present will be rejected by the binary classifier. Furthermore, the normal force is always positive so we can improve our force estimates by setting any negative force estimates to zero.

The relationship seen in the dataset in Fig. 3.3 between the statistics and the normal displacement would appear nonlinear, so that the overall trend could have been better captured by more sophisticated regression techniques such as kernel methods and Ridge Regression. However, the high noise levels in the data put a relatively low upper bound on the achievable precision of a displacement or force estimation algorithm. As discussed previously, we consider the simpler approach of linear regression to be preferable due to improved robustness.
CHAPTER 3. TRACKING

3.4 Explore Mode Algorithm

For the more precise force estimation algorithm, the images from the optical sensor are processed by an external computer. Again, we develop the algorithm to estimate displacement and consider separately the task of estimating normal force from the displacement.

We have included sample images from the optical sensor in Fig. 3.4.

We use the optical flow equation (Eq. 3.2) to estimate the movement of the image.

The sensor skin can be modelled as a plane parallel to the image plane that can translate in the $x$, $y$ and $z$-directions. This is assuming uniform force and no moment. If the motion is constrained in this way, it can be shown that the motion field of the image can be expressed as

$$
\mathbf{u}^{im} = \begin{bmatrix}
    x^{im} \\
    v^{im} \\
    z^{im}
\end{bmatrix} =
\begin{bmatrix}
    v_d^{im} & 0 \\
    0 & v_d^{im} \\
    v_x^{im} & v_y^{im}
\end{bmatrix}
$$

where $v_d^{im}$ is the dilation rate which is dependent on the $z$-velocity and $v_x^{im}$ and $v_y^{im}$ are the base velocities in the $x$ and $y$-directions.

Given a sequence of images, the most likely values of $v_d^{im}, v_x^{im}$ and $v_y^{im}$ are found for each frame using least-squares. Integration of the velocity components then yields the displacements of the sensor skin, from which the force is estimated. This is an implementation of differential tracking, discussed in Sec. 3.1.3 and is thus prone to integration drift.

Now, the assumption that pixel intensities are transferred can be extended inductively to $I_0(x^{im}) = I(x^{im} + \Delta x^{im}, t)$ where $I_0(x^{im}) = I(x^{im}, t_0)$ is a reference image. Making the same assumptions on the motion of the sensor skin, we can express $\Delta x^{im}$ as

$$
\Delta x^{im} = \begin{bmatrix}
    x^{im} \\
    t_d^{im} \\
    t_x^{im} \\
    t_y^{im}
\end{bmatrix}
$$

where $t_d^{im}$ is now the dilation which is dependent on the total $z$-displacement and $t_x^{im}$ and $t_y^{im}$ are functions of the total $x$ and $y$-displacements. Let us define the vector $\mathbf{T}^{im} = \begin{bmatrix}
    t_d^{im} \\
    t_x^{im} \\
    t_y^{im}
\end{bmatrix}$. With reference to Sec. 3.1.3 this is direct tracking.

The correlation between 2 images, $\text{corr}(I_1(x^{im}), I_2(x^{im}))$, is a normalised measure of the distance be-
between them taking into account overall changes in brightness levels that can arise as the sensor skin moves closer to the light source, and is thus preferable over conventional distance metrics such as Euclidean.

In order to estimate the displacement of an image relative to a reference image we thus pose the maximisation problem

$$\arg \max_{T_{im}} \text{corr}(I(x_{im} + \Delta x_{im}(T_{im}), t), I_0(x_{im}))$$  \hspace{1cm} (3.8)

Note that for this analysis we have assumed $I(x_{im}, t)$ to be a continuous function such that the elements of $T_{im}$ take on real numbers. The images are sampled at discrete pixels, so linear interpolation was used to make $I(x_{im}, t)$ continuous. This means that a closed-form solution of the problem no longer exists. Instead, the maximisation problem is solved using the simplex method. There is a risk of the simplex method converging to local optima, however for semi-random dot patterns the likelihood of this occurring is low.

For sufficiently small $T_{im}$, i.e. sufficient overlap between the current image and the reference image, this method is expected to find a strong peak in the correlation, but it breaks down for large $T_{im}$.

The value of

$$\max_{T_{im}} \text{corr}(I(x_{im} + \Delta x_{im}(T_{im}), t), I_0(x_{im}))$$  \hspace{1cm} (3.9)

serves as an indication of the performance of this method: a correlation close to 1 indicates images with good overlap and thus good results whereas a correlation close to 0 indicates failure of this approach.

This gives us two methods for motion estimation. We can use differential tracking, where we estimate the velocity between frames and integrate to find displacement. This is robust, but prone to integration drift. Alternatively we can use direct tracking, where we estimate displacement relative to a reference image. This gives absolute position but breaks down for large displacements.

A motion estimation scheme combining direct and differential tracking is thus proposed. For each image, we first solve the maximisation problem in Eq. 3.8 and evaluate the correlation at the optimum. If the correlation is above a threshold, we use $T_{im}$ for our current position estimate. If the correlation is below this threshold, the motion field of the current image relative to the previous image, $u_{im}$, is computed and added to the current position estimate. The scheme will thus use the absolute motion estimate for small displacements, switching to the differential method for larger displacements, and crucially switching back to the absolute method once the sensor skin comes close to its original position thus eliminating any integration drift.

For the ideal pinhole camera model and small displacements, and for the motion considered here, it can be shown that the relationship between $t_{im}$ and $z$ is given by

$$z = Z(1/t_{im} - 1)$$  \hspace{1cm} (3.10)

where $Z$ is again the distance from the image plane to the sensor skin at rest. To account for uncertainties in manufacturing of the sensor as well as discrepancies between the pinhole camera model and the real camera, we determine the value of $Z$ experimentally.

Similarly to Eq. 3.4 for the reflex mode algorithm, assuming a pinhole camera, the relationship between the image statistics and the shear displacements will be given by

$$x = k_e (Z - z) t_{im}^{x}$$  \hspace{1cm} (3.11)
$$y = k_e (Z - z) t_{im}^{y}$$  \hspace{1cm} (3.12)

where in general $k_e$ will not equal $k_r$ due to the use of different tracking algorithms. Again, we determine the value of $k_e$ experimentally.
Chapter 4

Characterisation and Testing

We first determine the force/displacement characteristics of the sensor. We then carry out normal force tests with both the reflex and explore algorithms, and finally we carry out basic shear displacement tracking tests with both algorithms.

4.1 Force/Displacement Relationship

To estimate force from displacement we require a model of the relationship between normal force $F$ and normal displacement of the fingertip $z$. An empirical approach was taken where a function was fitted to a set of data points generated experimentally.

Fig. 4.1 shows the experimental results along with the fitted exponential function

$$F = 1.1293 \left( e^{0.4677z} - 1 \right)$$

found by minimising the Root Mean Square (RMS) error. The function describes the relationship well, as seen from Fig. 4.1 with a RMS error of 0.14 N.

4.2 Reflex, Normal Force

Fig. 4.2 shows the result of applying the decision tree and the force estimation to the dataset presented previously in Fig. 3.3. The large size of the dataset combined with the low dimensionality of the data

![Figure 4.1: Force against displacement for elastomer fingertip, with least-squares curve fitted.](image-url)
points and the simplicity of the algorithms imply that overfitting of the dataset is very unlikely. The error rate of the decision tree classification is 2.1%, with the rate of false positives (‘no force present’ classified as ‘force present’) being 0.7% and the rate of false negatives (‘force present’ classified as ‘no force present’) being 1.4%. However, importantly, false negatives are only seen for very low force values: in this case with a displacement no greater than 0.21 mm resulting in a force no greater than 0.09 N. For practical applications, this means that any force larger than this value will be detected accurately.

To a first approximation, the linear force estimation is seen to perform well. Estimates are high for low and intermediate forces, and at higher forces there is a greater spread in the force estimates. A RMS error of 0.143 N is achieved overall, with a RMS error of 0.30 N for forces smaller than 1 N. A more sophisticated force estimation algorithm would be expected to improved performance here, but at a higher computational cost and with a risk of overfitting.

4.3 Explore, Normal Force

A sequence of 1000 images was generated with the sensor undergoing slow (~2 mm/s) reciprocating motion into a flat plate. The direct/differential motion estimation scheme was then applied to compute $T^{nm}$. Again, we are studying normal motion so the feature of interest is $t_{im}^{m}$. The correlation threshold for switching to the relative mode was set to 0.95. Fig. 4.3 shows the computed value of $t_{im}^{m}$, the image dilation, for each data point together with the $z$-displacement of the sensor skin. Points are colour-coded by the motion estimation mode used. It can be seen that for small displacements the direct motion estimate is primarily used, and for large displacements the differential motion estimate is used. It is also seen that there is no integration drift in the force estimates; at zero displacement $t_{im}^{m}$ returns to 1.

Using the method of least-squares, the optimal value of $Z$ in Eq. 3.10 was found to be $Z = 16.03$ mm, achieving a RMS error of 0.39 mm. The resulting relationship between $z$ and $t_{im}^{m}$ has been plotted in Fig. 4.3, and it can be seen that there is good agreement between the calculated $z-t_{im}^{m}$ relationship and the data.
4.4 Explore, Force Tracking Test

Combining the force-displacement relationship from Eq. 4.1 with the $z$-$d^m_t$ relationship from Eq. 3.10, we can estimate force from a sequence of images using the explore mode algorithm.

This test simulates a real-life situation where the sensor is used to track normal force, such as picking up an object.

A dataset was generated for displacement ramping up and down, with force and displacement measured. Fig. 4.4 shows the $z$-displacement and force estimated from the images together with their true values. It is seen that both the displacement and force estimates follow the true values closely, with a maximum displacement error of 0.80 mm and a maximum force error of 2.14 N. The optical tactile sensor is thus able to successfully estimate the normal force, using only the sequence of images, with little error.

The exponential force-displacement characteristics of the sensor tip means that for smaller forces the relative force error decreases. This is useful for many applications, allowing for precise and delicate manipulation as well as higher-force manipulation within a small range of sensor deformations. It can be seen in Fig. 4.4 that for forces smaller than $\sim$2 N the force error is very small.

This result demonstrates that the developed sensor is able to detect normal force with a precision sufficient for many robotic applications, including handling of delicate objects with low forces and manipulating larger objects with higher forces.

4.5 Reflex, Shear

Extensive testing of shear and normal/shear force characteristics was not possible with the developed test rig, as shear and normal force cannot be measured simultaneously, however tests were carried out demonstrating that the tracking algorithms can successfully track shear displacement. In this test we demonstrate tracking of $y$-displacement however tracking will perform equally well in the $x$-direction.

We wish to verify Eq. 3.4 and determine the value of $k_r$. We fixed the normal displacement of the sensor at approximately 1, 2 and 3 mm, and varied the shear displacement while collecting the sensor statistics as well as the normal and shear displacement.

Now, in Sec. 4.3 the value of $Z$ was found to be 16.03 mm. We used least-squares regression on the
collected dataset to determine the optimal value of $k_r$. The optimal value was found to be $k_r = 0.21$, yielding an RMS error of 0.1 mm. Note that we estimate a single value of $k_r$ and use that for all three relationships, with the difference in gradient arising from the variation in $z$. Fig. 4.5 shows the data points as well as the relationships as computed with Eq. 3.4. It can be seen that there is some noise in the dataset, much of which arises due to the quantisation of $DY$; it only takes on integer values. This quantisation is clearly a limitation of using the ADNS-2620 DSP for shear displacement estimation, however for fast low-precision shear displacement estimates the performance is good.

### 4.6 Explore, Shear

We carry out a similar test with the *explore mode* algorithm, fixing the normal displacement at approximately 1, 2 and 3 mm, and varying the shear displacement. Here we wish to verify Eq. 3.12 and determine the value of $k_c$. A dataset was generated, and the *explore mode* algorithm was used to compute $T^y_{im}$. We wish to study the $y$-displacement, so we are interested in the values of $t^y_{im}$. Using least-squares regression, the optimal value of $k_c$ was found to be $k_c = 0.013$, achieving an RMS error of 0.05 mm. Again, we are only estimating a single value of $k_c$, that is used for all three relationships. The fit here is improved due to $t^y_{im}$ not being quantised. This error is of the same magnitude as the error in the position measurement system, so the estimated RMS error is thus an upper bound.

As we have already demonstrated the tracking of normal displacement with little error, this result means that the explore algorithm is able to track both normal and shear displacement with little error. While we have separately demonstrated normal and shear tracking for clarity of the results, tracking of combined shear and normal displacement is not expected to be different.
CHAPTER 4. CHARACTERISATION AND TESTING

Figure 4.5: Tracking of shear displacement with the normal displacement fixed at approximately 1, 2 and 3 mm, using the reflex mode algorithm. The experimental data follows the trend predicted by the pinhole camera model with some noise.

Figure 4.6: Tracking of shear displacement with the normal displacement fixed at approximately 1, 2 and 3 mm, using the explore mode algorithm. There is good correspondence between the relationship predicted using Eq. 3.12 and the experimental data.
Chapter 5

Conclusions

Optical tactile sensing has numerous advantages over traditional force sensing methods, including inherent compliance as well as resilience to mechanical damage. The compliance of the sensor allows for soft-soft human-robot interaction. The use of a dual-mode sensing scheme mimics the way biological organisms achieve fast reaction times or precise force estimation depending on the situation. Commercialisation of the technology could see sensors being manufactured at very low cost due to the low mechanical complexity and low on-board computational requirements.

We have successfully developed and tested a complete low-cost low-precision dual-mode optical tactile force sensor. This has included optical design (Sec. 2.1), mechanical design (Sec. 2.2), manufacture (Sec. 2.2), polymer moulding (Sec. 2.2.1), electronic design (Sec. 2.3), low-level interfacing of digital systems (Secs. 2.3, 2.5), manufacture of an automated test rig (Sec. 2.4), algorithm development (Chapter 3), development of test procedures (Chapter 4) and data analysis (Chapter 4). We have demonstrated sensing of normal and shear displacement as well as normal force with relatively low error.

We have developed a bio-mimetic dual-mode sensing scheme suitable for robotic applications, with a reflex mode analogous the reflex actions in living organisms and an explore mode where the sensory information is processed by the central controller. The explore mode is analogous to living organisms processing sensory information in the higher-level cognitive parts of the brain.

In Sec. 1.5 we drew out a list of aims for the project which we believe have been successfully fulfilled, as follows:

**Compliance.** The compliance of the optical tactile sensor is similar to that of a human finger (Fig. 2.10).

**Speed.** With the reflex mode algorithm, and monitoring $\Delta X$, $\Delta Y$ and $\text{MAX}$ so that we can detect normal and shear forces, the refresh rate is 500 Hz. This is similar to human performance (400 Hz, Sec. 2.1).

**Force Sensing.** The developed sensor achieves normal and shear displacement sensing along with normal force estimation, with low error (normal RMS error: 0.39 mm; shear RMS error: 0.05 mm).

**‘Finger’ form factor.** The sensor is similar in shape and size to a human finger, see Fig. 2.3.

**Low cost.** We estimate the cost of a production-model sensor unit to be £3.38, see Table 5.1.

**Robustness.** The developed tracking algorithms should be robust with regards to changes in dot pattern due to mechanical damage. All sensitive components (optics, electronics) are housed in the sensor body, away from the sensing surface.
5.1 Key Skills and Challenges

This project has encompassed tasks in many fields, requiring a broad skill set. The development of a complete sensor system required strong hands-on practical skills and good analytical skills for the development of tracking methods, as well as consideration for the interplay between practical and analytical issues.

The following lists outline some of the challenges encountered in this project.

Optics. I started this project with a very limited understanding of optics. Development of the ray tracing model and design of the optical system required me to develop working knowledge of basic optical principles and systems.

Microcontroller programming. I had not previously come across programming of embedded systems. The project required interfacing hardware over serial (master and slave) connections as well as PWM. I had to acquire working knowledge of embedded programming, including the following concepts: timers; serial communication; bit-banging of serial connections; interrupts.

Elastomer Moulding. The project required the use of the Composites lab for moulding of the elastomer, for which practical lab skills had to be acquired. In particular, moulding a high-quality void-free fingertip proved difficult.

Machining. I manufactured the sensor body, requiring both milling and turning operations.

SMD Soldering. This project required circuit board manufacture, as well as soldering of miniature surface-mounted (SMD) components and connectors (connection area down to 0.2 mm$^2$). I had experience soldering larger connections, however soldering of SMD components proved challenging.

Algorithm Design. The noisy low-resolution data of the optical sensor made the algorithm design difficult. While the explore mode algorithm is based on optical flow methods, the requirement of zero integration drift meant that conventional image tracking methods were not sufficient.

Data Analysis. Again, the experiments had to be designed around the limitations of the test rig (e.g. mainly looking at displacement rather than force), where some system parameters were difficult to measure and had to be determined from the experimental data.

5.2 Further Work

The tests in Sec. 4 were all carried out under controlled conditions using the developed test rig. Controlled conditions are clearly essential for initial development, calibration and testing, however they do not realistically represent real-world applications of the sensor. More realistic tests could be carried out by introducing noise into the tests e.g. pushing the sensor against objects rather than a flat plate.

A larger-scale project could involve testing the sensor by attaching it to robotic grippers and manipulating objects. It could also be used as a robotic force sensor for robots interacting with humans.

A natural extension to the project would have been to study the force/displacement characteristics of the sensor in shear as well as with combined normal and shear forces. This would require a test rig with shear force sensing, which is why it was not covered here. We expect the force/displacement characteristics to be highly nonlinear with a high degree of cross-coupling between the shear and normal components, however for low-precision assuming independence could be sufficient.

The explore mode algorithm could easily be modified to also detect rotation about the z-axis by adding a further parameter to $T_{im}$, but again testing of this would require a more sophisticated test rig.
Table 5.1: Cost estimate of optical tactile sensor. We assume a production volume of ≈1000 units.

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost (£)</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optical Tracking Sensor</td>
<td>0.54</td>
<td>ADNS-5095 (ADNS-2620 obsolete). <a href="http://www.mouser.com">www.mouser.com</a></td>
</tr>
<tr>
<td>Microcontroller</td>
<td>0.74</td>
<td>Microchip PIC16F1824-I/P. <a href="http://www.mouser.com">www.mouser.com</a></td>
</tr>
<tr>
<td>LEDs, resistors, crystal</td>
<td>1.00</td>
<td>Estimated.</td>
</tr>
<tr>
<td>Body unit (with skin, elastomer)</td>
<td>1.00</td>
<td>Will change greatly with scale of production.</td>
</tr>
<tr>
<td>Lens</td>
<td>0.10</td>
<td>Single-element acrylic lens</td>
</tr>
</tbody>
</table>

| Estimated Unit Cost | 3.38 |

It may be possible to improve the performance of both tracking algorithms by using longer sequences of images or statistics for position estimation. Techniques such as Kalman filtering could be applied to improve tracking performance. However, the comparatively low refresh rate of the *explore mode* would suggest this to be more applicable to the image statistics. If ‘noise’ in the statistics is not random, such that the statistics vary smoothly with distance, this could improve performance.

Detection of surface texture, such as edge detection (see Sec. 3.1) would add to the functionality of the tactile sensor. We do not expect the $18 \times 18$ pixel images generated by the *ADNS-2620* to provide sufficient information for extensive detection of such features, but have not attempted to verify this by experiment. Using a skin with papillae similar to that used by Chorley et al. [11] could improve texture tracking performance.

An imaging system with higher resolution would allow for more sophisticated tracking including detection of surface texture features, but at lower frame rates and with higher cost and most likely requiring an external microcontroller for DSP. Investigations into higher-resolution systems could allow for the development of higher-precision force sensors.

The bio-mimetic dual mode reflex/explore sensing scheme seems suitable for a variety of robotic sensing and control applications where the robot must respond to sensory inputs with minimal latency. In this report we have outlined it conceptually, however further study could include developing a more rigorous framework for the implementation of such a hierarchical control system.

5.3 Commercialisation of Sensor Design

In a commercialised model, the fingertip, skin, dot pattern and body could be manufactured as a single unit fabricated in a multi-stage injection moulding process. A single electronics board could hold the optical sensor as well as LEDs for illumination and the microcontroller. We have estimated the cost of a complete sensor unit to be £3.38, see Table 5.1. This means that a humanoid robotic hand could be instrumented with compliant tactile force sensing fingers at a cost of £16.90.

The cost is sufficiently low that tactile sensing could be integrated in products from toys to running shoes (making the core of the sole transparent and detecting the normal and shear forces on the sole for analysis of running style).

If a tactile force sensor replaces an existing ‘finger’, such as in robotic applications, it will be possible to implement tactile force sensing with virtually no added cost.

5.4 IROS 2012 Paper

We have written a paper on the topic of this project, submitted to the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems: E. Knoop and J. Rossiter, “Dual-Mode Compliant Optical Tactile Sensor for Safer Human-Robot Interaction”.

44
Bibliography


