PHD

Sensitive Skin for Robotics

Pollard, Frederick

Award date:
2011

Awarding institution:
University of Bath

Link to publication

Sensitive Skin for Robotics

Frederick Norman Pollard

A thesis submitted for the degree of Doctor of Philosophy

University of Bath

Department of Electronic and Electrical Engineering

May 2011

Signed: …………………………………………………………………….

Contents

Acknowledgements **1**

Abstract **2**

Chapter 1: Introduction **3**

1.1. From Safety to Sensing in Robotics...3

1.2. Necessity of Tactile Sensing 4

1.3. Outline of this work...5

Chapter 2: State-of-the-art in Tactile Systems **8**

2.1. Tactile Systems and Biology 9

2.2. Current Developments in Tactile Sensors.......................................20

2.2.1. Resistance..21

2.2.2. Capacitance..29

2.2.3. Piezoelectricity..36

2.2.4. Optics and Imaging..41

2.2.5. Miniaturisation Techniques 47

2.2.6. Conclusions..52

2.3. Tactile Data, Imaging and Control 53

2.3.1. Tactile Data and Manipulation...54

2.3.2. Environmental Mapping for Robotics 61

2.3.3. Conclusions..72

2.4. Discussion...72

Chapter 3: Initial Work with QTC **75**

3.1. Commercially Available QTC..75

3.2. Array-based QTC Development 82

3.3. Discussion and Conclusions...90

Chapter 4: QTC Matrix – Design and Characterisation............................ **92**

4.1. Design and Experimental Methods...92

4.2. Results and Discussion – Application to Robotics 95

4.2.1. Isolated Tactels.....95
4.2.2. Multiple Tactels.....104
4.2.3. Conclusions and Discussion.....108
4.3. Results and Discussion – Electrical Characteristics.....109
4.3.1. Background and Comparison to QTC Pills.....109
4.3.2. Alternatives to the Fowler-Nordheim Conduction Model.....114
4.3.3. Conclusions and Discussion.....122
4.4. Conclusions.....123

Chapter 5: Bitworld – A Bit-Based Representation of Tactile Data 126
5.1. Introduction.....126
5.2. Description of a Compact Representation.....127
5.2.1. Bit-Based Spatial Representations.....127
5.2.2. Bitworld – An Extension for tactile data.....129
5.2.3. Interim Conclusions.....133
5.3. Image Manipulation – Techniques and Performance.....134
5.3.1. Tactile Images and Manipulation.....134
5.3.2. Bitworld Notation.....136
5.3.3. Methodology and Performance Metrics.....143
5.3.4. Image Translation.....145
5.3.5. Image Rotation.....148
5.3.6. Conclusions.....151
5.4. Conclusions.....152

Chapter 6: Super-resolution Bitworld Imaging..... 153
6.1. Introduction.....153
6.2. Super-resolution Methods and Implementation.....155
6.3. Image Combination – Results and Discussion.....160
6.3.1. Straight Edges.....161
6.3.2. Curved Features.....170
6.4. Conclusions.....179

Chapter 7: Bitworld – Combining Noisy Bitworld Images..... 181
7.1. Introduction.....181
7.1.1. Effects of Noise on Image Combination.....181
7.1.2. Motivating Example.....182

7.1.3. Assumptions and Definitions....183

7.2. Removing Noise from Bitworld....185

7.2.1. Bitworld combined with Bitworld....185

7.2.2. Improvements using Counting....189

7.2.3. Simulation....194

7.2.4. Discussion....197

7.3. Two-bit Linear Optimisation....198

7.3.1. Extension of Matrix Description....198

7.3.2. Implementation for one-bit measurements....202

7.3.3. Implementation using intermediate knowledge....205

7.3.4. Discussion....209

7.4. Two-bit Brute-Force Optimisation....210

7.4.1. Optimisation Approach....210

7.4.2. Results....211

7.4.3. Simulation....214

7.4.4. Conclusions....217

7.5. Discussion and Conclusions....217

Chapter 8: Conclusions and Further Work.... 220

8.1. Conclusions....220

8.2. Further Work....224

References.... 228

Appendices.... 239

A5.1. Bitworld and MatrixWorld class code listings....240

A5.2. StopWatch class code listing....248

A6.1. BitworldMatrix class code listings....249

A7.1. Worked Example of Positional Noise in a Multi-Link Robot....253

A7.2. Proof of Limited Applicability of 1-bit Bitworld....255

A7.3. Derivation of k-value threshold....257

A7.4. Results of 1-bit Linear Optimisations....255

A7.5. Results of 1-bit Linear Optimisations using k-value thresholds....259

A7.6. Results of 2-bit Linear Optimisation....261

A7.7. MATLAB Linear Optimisation Routines....262

A7.8. Comparison of brute-force and k-value approaches....269

A7.9. Incremental updates after brute-force optimisation....................................271

A7.10. Converting Optimised Matrices into Boolean Functions 273

Acknowledgements

I would like to express my appreciation to my supervisors Dr. Duncan Allsopp and Dr. Peter Shepherd for their support and guidance during the course of this work. To describe their advice, encouragement and support as invaluable is to understate how much I have learnt from them during the course of these studies.

I would also like to acknowledge the joint financial support of the EPSRC and OCRobotics for this work. Specific thanks go to Dr. Rob Buckingham, Managing Director of OCRobotics for helping to establish the funding and for many invaluable discussions before and during the work. I would also like to thank Swisstulle UK Ltd for their support in the form of providing access to the necessary materials and technical information.

The support of friends and family throughout the production of this work has been incredibly helpful to be and ranges from allowing me to ramble on about my work in conversation to actually reading the thesis and discussing matters of research with me. I would particularly like to thank my mother, who acted as a sounding board for me as I tried to come to grips with certain aspects of cognitive psychology!

The largest thanks of all go to my wonderful wife, Gillian. She has provided me with the unwavering love and support I needed at both the highs and lows that are common in research. She has also shown immense patience, accepting late nights at the office and the periodic conversion of our home into a branch of the University as part of my adventure into the world of robotic skin. Without all of this and her unceasing encouragement, this thesis would not be and I am incredibly grateful to her for all of this.

Abstract

This thesis explores two novel ways of reducing the data complexity of tactile sensing. The thesis begins by examining the state-of-the art in tactile sensing, not only examining the sensor construction and interpretation of data but also the motivation for these designs.

The thesis then proposes two methods for reducing the complexity of data in tactile sensing. The first is a low-power tactile sensing array exploiting a novel application of a pressure-sensitive material called quantum tunnelling composite. The properties of this material in this array form are shown to be beneficial in robotics. The electrical characteristics of the material are also explored.

A bit-based structure for representing tactile data called Bitworld is then defined and its computational performance is characterised. It is shown that this bit-based structure outperforms floating-point arrays by orders of magnitude. This structure is then shown to allow high-resolution images to be produced by combining low resolution sensor arrays with equivalent functional performance to a floating-point array, but with the advantages of computational efficiency. Finally, an investigation into making Bitworld robust in the presence of positional noise is described with simulations to verify that such robustness can be achieved.

Overall, the sensor and data structure described in this thesis allow simple, but effective tactile systems to be deployed in robotics without requiring a significant commitment of computational or power resources on the part of a robot designer.

Chapter 1
Introduction

1.1. From Safety to Sensing in Robotics

On January 25th 1979, an assembly line worker in the United States earned the dubious honour of becoming the first person to be killed by a robot. Robert Williams was struck in the head by the arm of a robot that was designed to collect parts from a storage unit where Williams was working. It was a costly mistake, not only in terms of the loss of life, but also in damages amounting to $10m against the robot manufacturer [1].

Williams' story is a salutary lesson in the need for safety when robots and humans interact. On modern assembly lines, industrial robots are equipped with a vast array of discrete devices such as light gates, proximity alarms and emergency stop systems that limit the possibility of humans and robots occupying the same space. Structured environments such as these can be constructed specifically for a robot, limiting the possibility of unsafe interactions with personnel and achieving automation efficiency at the expense of developing precision systems.

Current trends in robotics are exploring deployment of systems in unstructured environments, particularly in the context of emerging social problems such as an ageing population. This has led to an examination of the possible use of service robots that must be able to operate in a domestic environment that is unstructured and in which interaction with humans cannot be avoided or is even required [2]. In this emerging area, where the fixed safety features of the industrial domain cannot be practically or economically deployed to prevent injury or damage, equipping a robot with its own sensor systems becomes a necessity.

Providing robots with sensor systems additionally allows more complex interactions with the environment and to this end, as well as the assurance of safety, researchers have developed a variety of systems over the past few decades to emulate the animal senses of vision and touch in robotics.

The focus of the current work is in the area of touch sensing. This is also known as tactile sensing, a nomenclature that that will be used here and throughout. Tactile sensitivity is essential in the context of safety, since it allows a robot to determine if it hits unexpected objects in the environment.

1.2. Necessity of Tactile Sensing

Although it would appear that vision or proximal sensitivity would be a better choice, in that contact with objects or people can be prevented before they occur, these modalities do not always operate effectively at short ranges [3]. Vision also suffers from problems related to occlusion, the requirement for external sources of light and complex data processing.

The provision of tactile sensing can replace or complement vision systems to afford robots a degree of semi-autonomy. It can also insure against failures in the potentially more information-rich vision systems, where hardware failures or a loss of light would render the robot incapable of making decisions.

As will be explored in the review of Chapter 2, work in this field has concentrated on examining various means of transducing a tactile stimulus to a measurable electrical quantity. Despite extensive work in this area, tactile sensing has not seen widespread take-up in robotics. This stems from difficulties in deployment and integration, for which this work seeks to provide solutions.

An advantage for vision systems is that emulating the sense of sight can be achieved in a single, discrete sensor. As a result, researchers and commercial developers are in a position to procure off-the-shelf sensors and deploy these using common interfaces and by developing software techniques that are independent of the inner workings of the device. By contrast, tactile sensing is less well understood and it is unclear exactly what features need to be measured to facilitate particular tasks, such as gripping or environmental exploration. This means that a single design of sensor has been unable to dominate sufficiently to allow the wider integration challenges to be resolved in a manner sufficient to facilitate significant deployment of tactile sensing in robotics. Repeated efforts have been made to devise generalised tactile skins, but none have achieved widespread acceptance to-date.

This work rejects this sensor-driven approach to tactile system design. Although it is possible to attribute the success of vision to the development of a common sensor, it is more likely to be the ability of systems designers to treat the camera independently of its implementation. This “black box” approach has not been pursued in tactile sensing, with researchers in all areas tailoring their work towards a specific sensor implementation. Consequently, pertinent issues in tactile sensing such as handling, processing and representing the resulting volume of data have not been developed in a sufficiently general fashion. This makes integration with robotic systems complicated, as the data processing requirements currently pose a significant computational burden [4].

1.3. Outline of this work

In Chapter 2, the current state-of-the art in the field of tactile data will be presented. As part of this examination, a different design philosophy focussing on the interface between the sensor and the sensing system is developed. This is what will be referred to as a *data-centric* approach and provides a set of criteria against which existing systems can be assessed and new systems designed.

Application of these criteria show that many existing systems are deficient in some way, and that an interesting material with the potential to satisfy the majority of them has not been deployed in a form suitable for tactile sensing. This material is a pressure-sensitive composite called QTC™ and Chapter 3 will examine the properties of this material. It will also describe the development and characterisation of two prototypes for its development into a novel array form, highlighting some practical difficulties in doing so and showing how these can be overcome.

Chapter 4 presents the final development of this novel deployment in a manner suitable for wide-scale use of tactile sensors. This low-resolution sensor also makes novel use of a conductive fabric with properties that are conducive to making the sensor easily deployable over the non-planar surface of a robot. The sensor is characterised under a series of loading conditions, including the effects of loading across multiple sensing cells. The dynamic response and the variations in the force-resistance relationship of the sensor when it is forced to conform to a curved surface are also examined to assess the suitability of the sensor in robotic applications.

At all stages, the measured properties of the sensor are assessed against the data-centric criteria that are developed in Chapter 2. Although the goal of this work as a whole is to show how the development principles of tactile systems can be shifted to a data-centric rather than sensor-centric approach, it is an important development to have a sensor that satisfies these criteria as fully as possible whist still being practical.

Since the sensor is shown to be endowed with useful properties by the QTC™, it is desirable to facilitate continued development of the material for tactile sensing. To that end, Chapter 4 also describes the electrical behaviour of the material and its origins. Although the behaviour of the material has been the subject of other work [5, 6], this represents the first time that the current-voltage characteristics have been examined directly in order to postulate a theory of conduction for this class of materials.

Returning to the idea of separating the sensor from the processor using the data representation, Chapter 5 outlines the framework and language of a novel data representation for tactile data called “bitworld”. This is based on a grid structure of individual bits representing tactile data, represented by easily manipulated computer words. This representation reduces the bandwidth required within a robot system, and it will be demonstrated that it offers significant computational advantages over comparable representations when the resulting images require manipulation. The separation of the central processing component of a robot from the implementation of the sensor is thus achieved, effectively converting it into a black box comparable to the camera in the field of vision.

Tactile data is sensitive to measurements of robot pose, since the location of objects and obstacles will often require translation to a common set of axes. Chapter 6 extends bitworld so that it can be made robust in the presence of noise in these measurements of pose. It is shown that representing grid cells with only a single bit is generally insufficient to protect against the effects of noise and shows how noise can be reduced in a two-bit system with low computational complexity during operation.

Having devised these two data-focussed components of a tactile system, they are combined in the work of Chapter 7. Multiple images of objects produced by the low-resolution sensor are combined using the bit-based structure to show that higher resolution images can be obtained. The functional and computational performance of these techniques is compared to a more typical representation and to a hybrid of the two that is shown to afford certain advantages at limited additional cost. Using these images, simulations are performed to show the effects of noise on the image combination routines and the impact of using the methods of Chapter 6 in reducing these effects. Finally, Chapter 8 presents the overall conclusions to this work and lays out the requirements for further work in these areas.

The novelty of this work is therefore three-fold; a new approach to tactile system design, a novel deployment of materials to manufacture a tactile sensor array and a data representation that has low computational cost but is still robust in the presence of real-world effects. The combination of these contributions in Chapter 7 highlights the utility of these particular methods and thus of the data-centric approach itself.

Chapter 2
State-of-the-art in Tactile Systems

This chapter examines the current state-of-the-art in tactile sensing, incorporating the specification of tactile systems, the developments in tactile transducers and the use of the information derived for them. Tactile sensors are commonly defined as a device or system that can measure the presence of an object or a property thereof through physical contact [7]. This is a wide-ranging definition that incorporates sensors of force, pressure, temperature, texture and a variety of others.

The underlying property that all of these sensors measure, however, is the presence of an object adjacent to the sensor. The simple act of detecting an object is, in many domains, also of more immediate and practical importance, since it has implications for safety and the prevention of damage to the robot or other elements of its immediate environment. On this basis, the focus of this work will use a more limited definition of tactile sensing that relates solely to detecting the presence of an object.

Tactile sensing is considered as distinct from proximity sensing, whereby objects can be detected prior to contact. For the purposes of this work, however, systems permitting proximity sensing will be considered if they also permit a direct contact to be detected in some form.

This chapter begins with an examination of the broad principles underlying current work in the field, which typically focus on biological mimicry. This provides a means of assessing the utility of existing work in sensing systems, followed by an examination of how tactile data is exploited. The transmission, processing and use of tactile data are crucial to the understanding of how to construct tactile systems. The final section of this chapter will review how these often independent streams of work can be combined to satisfy the needs of tactile systems in general.

2.1. Tactile Systems and Biology

Robots generally consist of structures that mimic the characteristics of animals. Robotic arms manoeuvre objects or tools to a desired location, and grippers act as hands to manipulate objects in a precise way. Even less sophisticated robots lacking manipulators may use cameras, acting as the robotic equivalent of eyes to examine the environment. Such structures are easily comprehensible to designers and researchers, since they can intuitively relate the desired properties of the relevant robotic systems to the functionality and operation of their own biological equivalents.

It is not then surprising that human beings are often used as the basis of comparison when researchers are attempting to define desirable qualities of tactile systems. It is therefore useful to examine the basic structure and role of biological tactile sensors. For human beings, this is a subset of the total role of skin, which also acts as a protective layer and performs functions relating to homeostasis.

Human skin can be divided into two major structures: the outer epidermis and the underlying dermis. The epidermis acts to mediate the interaction between the internal structures of the body and the environment, protecting the systems from the effects of external elements [8]. The dermis supports the main structures associated with skin, including the mechanoreceptors that afford tactile sensitivity. There are multiple types of sensor, each with a fundamentally identical operating principle, whereby the tactile stimulus produces a deformation in nerve endings [9]. This deformation affects the ionic permeability of the membrane surrounding the nerve cells, generating a potential across the receptor that travels via the peripheral nervous system to the central nervous system, where it is processed [9, 10]. The mechanisms underlying human tactile sensing are not yet completely understood, but five key structures have been identified, each providing access to different tactile sensations.

Meissner's corpuscles are elongated receptors formed by a connective tissue capsule comprising several lamellae of cells. Changes in the shape of the capsule stimulate nerve endings in the centre. The structure of the corpuscle maintains an innate tautness that appears to allow for greater responsiveness to mechanical stimuli [11]. This allows this receptor to detect touch, including dynamic pressure changes of between 30Hz and 50Hz close to the lower surface of the epidermis [9]. The Pacinian corpuscles have an onion-like structure, where the nerve ending is surrounded by multiple layers of lamellae. These layers vary in shape from cylindrical at the core to spherical at the outer edge [12]. This change in shape leaves voids between layers that are filled with fluid, allowing vibrations of between 250Hz and 350Hz to be detected. These sensors reside in the adipose tissue, which is relatively deep within the skin [8]. Merkel's disks are disk-like cells containing nerve endings that respond to static pressure. These are located close to the epidermis and are particularly prevalent in the fingertips of human beings since they play a significant role in the static discrimination of shapes, edges and textures, which is regarded as essential for gripping [8, 9].

The remaining two systems do not contribute significantly to touch sensing, but are nonetheless vital to tactile sensation. Ruffini corpuscles are elongated capsules that are orientated along the stretch lines of the skin. This orientation allows stretching of the skin to be detecting, affording human beings the sense of proprioception [9]. This sensation allows the human body to locate the position of joints and fingers, and so to accurately locate the tactile sensors located across the surface of the skin. Finally, there are sensors with no encapsulating structure that are simply free nerve endings, providing the body with sensations of pain and temperature.

These sensing systems form the human tactile perception system, but their distribution is non-uniform across the surface of the skin. This can be discerned using the two-point test, which is where a single point or two spatially separated points are placed on a subject's skin, and the subject is asked how many points are present. The test shows that there some areas of the body require larger separations of the two points than others to be discernible. In the fingertips, this distance is only 1mm, whilst on the arm the distance is typically 20mm or more [8]. This lower density of sensors proves to be quite manageable, even though it may become lower over time. For example, up to 80% of Meissner's corpuscles are lost before human beings reach an advanced age, but the absence of these sensors does not contribute to a loss in sensation because the sensors in intermediate areas are likely to be unaffected [11].

In engineering terms, therefore, human skin is robust to damage, capable of self-diagnosing faults and uses a variety of transduction methods. Generally, each sensor measures only one stimulus, with two being measured only in the case of free nerve endings. In all cases, information from these sensors is transmitted to the central nervous system. This system consists of the spinal cord, where involuntary reflex actions may be processed, and the somatosensory cortex of the brain where the signals can be processed to allow higher levels of awareness and planning to occur [10].

The signals transmitted vary according to the strength of the stimulus, but they also have a temporal variance. The signals are generally classified into two types; rapidly-adapting (RA) and slowly-adapting (SA). Within these, two further subgroups can be identified, denoted by the numerals I and II and their characteristic signals, also called pulse trains, are shown in Figure 2.1.

As can be seen from these signals, rapidly-adapting receptors respond primarily to changes in the stimulus so that the strength or frequency of the pulse train decays rapidly in the absence of changes in stimulus [13, 14]. Slowly-adapting receptors encode a static stimulus, meaning that the frequency of the pulse train increases while an object is in contact with it, but this frequency will decay slowly over time as the sensor becomes accustomed to the object's presence [14].

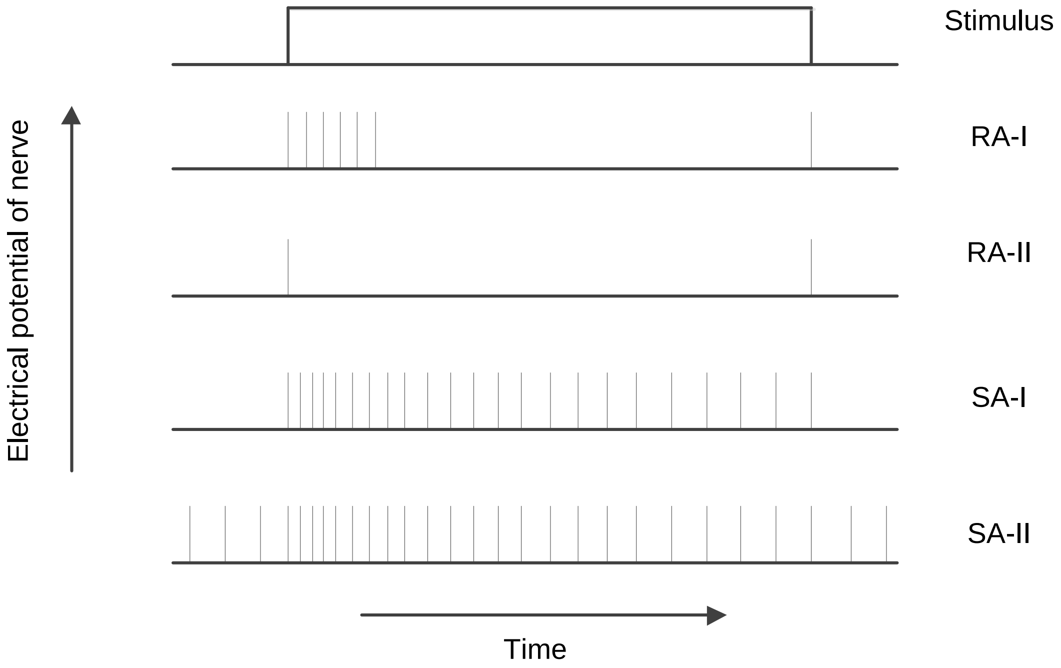


Figure 2.1: Illustration of the four different signal patterns in response to mechanical stimulation of mechanoreceptors. RA-I and RA-II refer to the two types of rapidly adapting sensor profile, and SA-I and SA-II refer to their slowly-adapting counterparts. Adapted from [13].

The capabilities of human tactile sensing have been highly influential in the study of its robotic counterpart. Evidence of this is the inclusion of similar reviews of human tactile capabilities in review articles within the field such as those of Howe [15] and the recent survey by Dahiya *et al.* [4]. The latter paper took the anthropomorphic approach to sensor design to the extent of providing detailed specifications based on human sensations. As a specific example, the authors proposed that sensor resolutions in tactile sensor arrays of between 1mm and 5mm be sought [4].

The acknowledgement that in robotics, the spatial resolution of the sensors should vary based on body site has been relatively uncommon within the field, with most focus being given to a single resolution of 1mm. This is exemplified in the approach of Lumelsky *et al*. who proposed that a single, generalised tactile skin system be sought with a resolution of 1mm [16]. This work gives no specific justification for this choice, although in a later work, Lumelsky justifies a high skin resolution by identifying that if a robot is to use touch to move in an unstructured environment, its sensor resolution will directly limit its range of motion [3]. The justification here is in the name of generality so that a single sensor can be applied to a significant range of tasks and still be manufactured in a manner that exploits the economies of volume manufacture.

However, in what are the first published efforts to identify the specifications for tactile sensing in robotics, the 1mm resolution was justified by the specificity of the perceived field of application, without reference to the anthropomorphic approach [17]. This was the work of Harmon, who conducted a survey of a relatively small group comprising industrialists, academics and other interested parties in the early 1980s [17]. This survey indicated that a spatial resolution of 1-2mm was a necessity for tactile sensing. The focus was on industrial applications, where grippers equipped with sensors could identify parts extracted from bins on a production line to enable some cost-effective reductions in the structuring of the environment. Tactile sensing would then be effective in identifying the orientation of parts to allow the robot to adjust them before placing them [18]. It has latterly been acknowledged that this application area has not materialised in the manner Harmon's work suggested [19], yet it completes the picture of a specification in resolution within the field that relies variously on anthropomorphic considerations, a desire for generality and the needs of specific applications.

In order to adequately assess the utility of the current state-of-the-art in tactile sensors, as is intended in the next section, it is important to examine the validity of these considerations. In doing so, a benchmark specification can be generated against which the different developments can be assessed.

It is clear that inspiration can be drawn from biological structures in robotics, drawing not simply from human experience but also from other animals. Human beings are not the only animals that experience touch [20] and other animals utilise similar structures to those already described, often augmented by additional structures to achieve greater sensitivity or an otherwise improved response. More commonly referred to as whiskers, these sensors act as short-range distance sensors to obstacles and can be considered as indirect tactile sensation since the interaction with the mechanoreceptors is mediated by the hair follicle [21].

Other biological structures exist that enhance the effect of a stimulus on a touch-sensitive receptor. One example of this is the campaniform sensillum, which is a structure coupled to strain-detecting tactile sensors in insects and arachnids to magnify the strains induced by contact to allow for easier detection [22]. Artificial versions of these structures have been simulated and implemented by boring small holes into a plate, coupled by some mechanical linkage to a strain sensor. The holes can be aligned to select the component of force that is of interest and this has been demonstrated by finite element modelling to amplify applied strain by a factor of four [23].

However, whilst biological structures offer an obvious model for robotic tactile sensing, it is not obvious that direct mimicry is appropriate, nor that the capabilities of animals must be replicated in their entirety. Proponents of a broadly anthropomorphic approach to analysing tactile requirements such as Howe and Dahiya *et al.* highlight the fact that there is little knowledge of what tactile data is actually required for given tasks [4, 15].

Howe goes further, highlighting the fact that the signals produced by biological tactile sensors obscure much of the information available from them [15]. However, because the focus of his work is on gripping applications, he rightly indicates that the control loops required to control a grasp must make use of detailed information, at least based on the present control theories. This highlights an incongruity in the anthropomorphic approach in that sensors are specified according to biological resolutions and capabilities but that the data and control systems must be defined by the limits of what is possible under current control and data transfer techniques. Research in the field must attempt to tackle the deficiencies in integration simultaneously with developing sensors or adapt the sensors to the limits of what is possible in integrating systems. The alternative is to realise that biological systems cope without precise information of the kind these specifications demand and so reconsider the specifications themselves.

Overall, attempting to realise biological constructs in robotic systems should be treated with caution. Natural systems were designed neither in the same manner nor with the same constraints as mechanical systems, but by a process of trial and error over a considerable period of time. The sensation of touch in humans is also dependent on upbringing, as the response to touch can be conditioned by cultural considerations [24], which highlights how much the biological system is dependent on a lengthy process of learning and calibration.

This is by no means to assert that biology is not a useful source of inspiration, but simply that biological systems evolved in a particular way with their own particular restrictions that may not be applicable, relevant or desirable for a man-made system with fundamentally different goals. The important lesson of biological sensors is actually that sensing should be task-based. This assertion is based on results such as the two-point tests previously discussed, highlighting the differing resolutions in different areas of the body. The fingertips require a high density of sensors for their manipulation tasks, but the arms and legs are afforded a far lower spatial resolution because these areas of the body do not require anything better.

Developing tactile sensors for particular tasks was an approach promoted by one of the landmark reviews of Lee and Nicholls in pursuit of industrial applicability [7]. The key point is that tactile systems must be requirements-driven, rather than focused on the promotion of a particular technology. Although the authors also used this as a justification for considering alternatives to tactile sensing, the same argument applies equally to the specification of the same. By this logic, one approaches the specification of tactile systems based on individual tasks and does not seek to generalise the sensors at all.

The counter-argument to this could be offered by the work of Lumelsky *et al.*, who created a specification for a general tactile skin, ultimately asserting that the then state-of-the-art sensors would be, in themselves, insufficient to meet these requirements [16]. Devising a general approach to tactile sensing is a laudable goal. In the parallel field of computer vision, the field has moved on from issues of image capture and is instead concentrating on processing images and extracting useable information [7]. This has facilitated a more rapid advance in the deployment of vision systems in robotics, both in academic and commercial settings, than has been seen for wide-scale tactile systems.

Unfortunately a general-purpose sensor specification is unsatisfactory for a number of reasons. Creating a sensor for as many systems as possible requires either under- or over-specification for the majority of systems. Under-specified systems will suffer from an inability to perform certain required functions, whilst over-specified ones will have added complexity, power requirements and data bandwidth requirements that impose additional and unnecessary constraints on the robotic system as a whole for no additional benefit. It is then apparent that, in terms of the sensor design, the task-based approach is preferable.

However, the advantages of generality are the ability to deploy systems in robotics easily without being concerned about the low-level measurement of stimuli in the same manner possible in the field of computer vision. In essence, this represents an abstraction of the sensor into a "black box", with a defined general interface that the robot's central processing system or controller interacts with. This represents a more general form of modularity, which others have proposed in relation to sensor systems to promote the possibility of rapid deployment and customisation [25-27], but isolates the sensor entirely from the robot controller. The generality is therefore contained in the interface to the system, a significant component of which is the data that it presents at these interfaces.

There are biological parallels to this in human beings, which are explored by the active research field of cognitive science. One aspect of this research is into the processes that are involved in humans and animals using sensory data to form perceptions about their environment. Cognitive psychology in particular regards the human mind as an information gathering and processing unit that transforms sensor information into common forms that can be used by some computational process [28]. A brief examination of these processes is beneficial, since it provides insight into the means needed for processing robotic data and highlights a more modular approach to data acquisition. This review will concentrate only on an abstract analysis, ignoring the details of neuroscience and other biological mechanisms in accordance with the tenants of cognitive psychology [29].

Cognitive psychologists regard the brain as a system receiving sensory information from the environment, processing it in the context of previously acquired experience and goals and then performing actions based on the outcomes of these computations [30]. Three main structures arise in the consideration of this abstract system, in what is called the information processing model: the sensory register, the short-term storage or working memory, and long-term storage [29]. The sensory register is a very short-term, high-capacity memory separated into iconic, echoic and haptic partitions to respectively accept sensory signals from the visual, auditory and somatosensory systems [30-32]. The raw incoming signals are only stored for a very limited time, on the order of tens of milliseconds before the information decays, such that only the most recent stimuli can be recalled by subjects in experiments [33, 34]. These early studies suggested that the sensory register might be an unaltered snapshot of sensor state at the moment of acquisition, but later work showed that the register performs encoding and applies transformations to the data according to past experience [35]. It is believed that the register transforms the data into so-called cognitive codes, that are transferred to the working memory if the cognitive system allocates resources for this purpose prior to the decay of the information [29].

The working memory is the location in the cognitive system where information from all memory structures can be combined to perform a mental task [36], and its existence has been verified through experiments requiring subjects to compare images at varying time intervals [37, 38]. These codes can be transferred to long-term storage by a process of reinforcing the codes through repetition or rehearsal [39], after which the codes are essentially retained in perpetuity, unless access is inhibited by the presence of other codes [29]. The total capacity of the working memory is unclear, although it is regularly quoted as being 7±2 items based on a study by Miller [40]. The true capacity appears to be dependent on the nature of the information being stored, with sequences of greater than seven letters having been shown to be easier to remember if they contain sub-sequences that occur frequently in the English language [38].

What is clear, however, is that the capacity is limited, much as the capacity of computer memory is limited. Indeed, a computer system finds many analogues with these cognitive structures, such as processor registers, cache memory, RAM and permanent magnetic or optical storage devices. In the pursuit of high density tactile skins, Lumelsky *et al.* highlighted the need for acquiring and processing the large amounts of data that such devices would produce [16]. The lesson from cognitive science and from the neural signals shown in Figure 2.1 is that the means of handling large amounts of sensor data lies in manipulating the raw sensor signal into a common, easy-to-manipulate form that can be passed around within the processing system.

In order to provide benefits in limiting the amount of data bandwidth required in the system, a problem identified by both Lumelsky *et al.* [16] and Dahiya *et al.* [4], some of this processing will have to take place as close as possible to the sensor. A general guideline for evaluating sensors can therefore be formed by focusing on the requirements of a data system. This data-centric approach rejects directly imposing requirements on modality, resolution, compliance or other matters that are in the domain of the robot system, not the tactile system. These requirements are:

1. *Ease of translation*: this imposes a requirement that it be easy to translate the measured sensor quantities to the tactile parameter that they represent. Clearly an involved translation is more time-consuming and so less desirable.

2. *Ability to process locally*: it is desirable to be able to perform processing locally [4] to reduce the load on the central processor and, ideally, to reduce the data bandwidth required in the system.

Some features of sensors may impact the ability to process data locally, such as a computationally intensive process.

3. *Easy access to sensor data*: this is a data-centric redefinition of the more commonly named "wiring complexity" problem. The question here is how easy it is to access the information or to enact the measurement of the sensor.

4. *Easy to reject data*: it is desirable to have a simple way to identify redundant data at a local processor and not transmit it so as to reduce data bandwidth and to limit the burden on the central processor to only relevant information.

5. *Ease of replacement*: this is a return to the desire for modularity. Tactile sensors will wear out faster than other sensor types due to their repeated contact with the environment so will need replacing. In an ideal system, replacement would not require changes to the robot controller or the data interface, although only the former is a strict requirement.

6. *Low power consumption*: This is the most non-data centric requirement, in that the sensor should not draw large amounts of power. This is a benefit to the robot in reducing wiring complexity, and enhanced battery life in mobile scenarios. The minimisation of power consumption also applies to the local processing of sensor data and so overlaps with the requirement for the ability to process data locally.

All other specifications regarding sensors must be considered in the context of possible application domains, but are not suitable for general statements of worth. This is not to negate the importance of their consideration, merely to contextualise it with the wider goal of large-scale uptake of tactile systems in robotics. Certainly, a system that met these requirements but could not detect tactile sensation with any accuracy would not be a good sensor.

In the next section, therefore, the current state-of-the-art in sensor systems will be assessed according to the data-centric criteria developed here. Additionally, the sensors will be assessed for their utility and for the scope of their application in robotics. In the subsequent section, existing strategies in processing and using tactile data will be assessed in the context of the data-centric approach. It is difficult in advance to assess the requirements of these higher-order processes, but it will be important that they meet the requirements of coping with the possible outputs of a data interface that enforces the data-centric requirements outlines above. In order to assist with this, a brief discussion on how human beings exploit their cognitive codes will also be useful.

In conclusion, this section has explored biological sensing and its impact on the specifications for tactile sensors within the literature. The conflicting desires for generality and a task-based approach to tactile sensing have been resolved by an approach where sensors are task-focused and the data interface to the robot controller is generalised. In this context, new criteria for assessing the utility of tactile sensors have been developed for use in evaluating the current state-of-the-art.

2.2. Current Developments in Tactile Sensors

This section examines the currently available technologies in tactile sensing, with reference to the principles established previously. Tactile sensors have been constructed using a variety of transduction methods, each with their own advantages and disadvantages. It is along these lines that the taxonomy of tactile sensors has been arranged, following the example of the reviews of both Lee and Nicholls [7, 41].

The only exceptions to this arrangement are the tactile sensors that have been constructed using miniaturisation techniques through the construction of microelectromechanical systems (MEMS) or semiconductors. These may exploit properties of other transduction techniques, but have common fabrication processes that are somewhat independent of the transduction technique that they exploit. Consequently, they are discussed separately in section 2.2.5. The different techniques will be reviewed independently of each other, and a relative assessment of the various techniques will be deferred until the conclusions of section 2.2.6.

2.2.1. Resistance

Sensors in this category operate by changing the resistance of a device in response to a tactile stimulus. This resistance is measured during operation and then typically compared to a known behaviour of the sensor, allowing the stimulus to be quantified and then acted upon.

One approach to the production of such devices is the use of conductive elastomers, where conductive particles are embedded into a non-conductive elastic polymer. These have a relatively high resistance when uncompressed since the conductive particles are separated by the polymer [42]. As the composite is compressed, however, the conductive path has been modelled by percolation theory as arising from the conductive particles forming chains across the material, increasing conductivity and so providing an electrical response to force and pressure [42, 43].

In robotic tactile sensing, the most common conductive filler used is carbon in the form of graphite. One of the earliest implementations is that of Russell, who used a graphite-loaded silicone to detect strain [44], although the idea of using such materials in artificial skins goes back at least as far as 1977 when it was proposed for use in prostheses [45]. The composite rubber was arranged in strips across the desired sensing area and then individual conducting tracks were introduced to intersect the strips at various points. These connections terminated with fixed resistors to produce potential dividers [44]. This allowed the resistance to be measured, and so the contact profile to be established, but required *NM* connections to produce an *N* x *M* array [44].

For a small scale array, this is not an obstacle but becomes an increasing data-access problem as the array grows. More recent work has also employed this type of material, by stitching wires directly through the conductive material [46]. This was an iteration on earlier work in which the electrodes were imprinted on a substrate connected directly to the rubber [47]. The problem with the latter configuration was that tangentially applied forces cause peeling of the rubber from the electrodes. Stitching the wires directly into the transducer was explored as a means of improving durability [46]. The stitching pattern used by Shimojo *et al.* in [46] is illustrated in Figure 2.2.

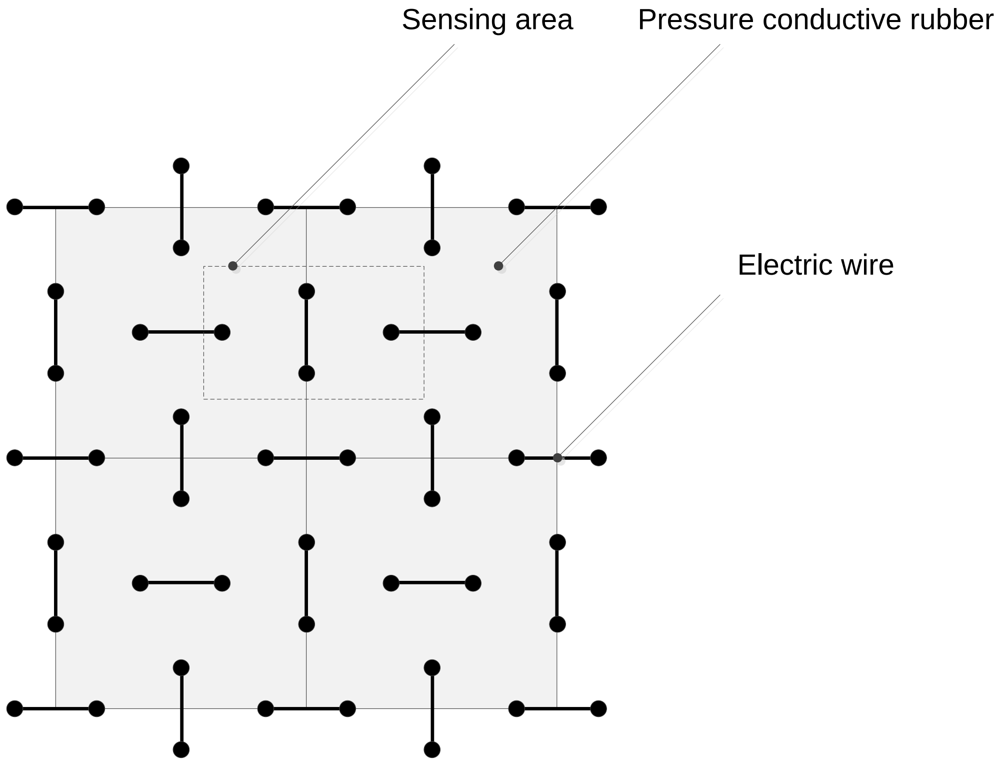


Figure 2.2: Diagram showing the stitching configuration employed by Shimojo *et al*. in [46].

The electrodes were made of beryllium copper wire coated with gold, with the wire stitched at 3mm intervals in both the horizontal and vertical orientations. The results of applying pressures were typical of those for this class of sensor, namely a non-linear relationship between pressure and resistance, and a pronounced hysteresis effect produced by the stress-strain characteristic of the elastomer. Linearity is regarded as being desirable in tactile sensing [4], but Hillis did not find the non-linearity inherent to his design to be a significant obstacle to object recognition [48] so this appears to be a variable requirement depending upon the application domain.

Hillis' work was designed for fingertips and consisted of a monolithic array of 256 sensors, each with an area of less than 0.01mm^2. Such an array is scanned by applying a fixed voltage to cross each tactile cell in turn, whilst all other connections to the array are held at ground in order to restrict the alternative paths and the associated crosstalk [48, 49].

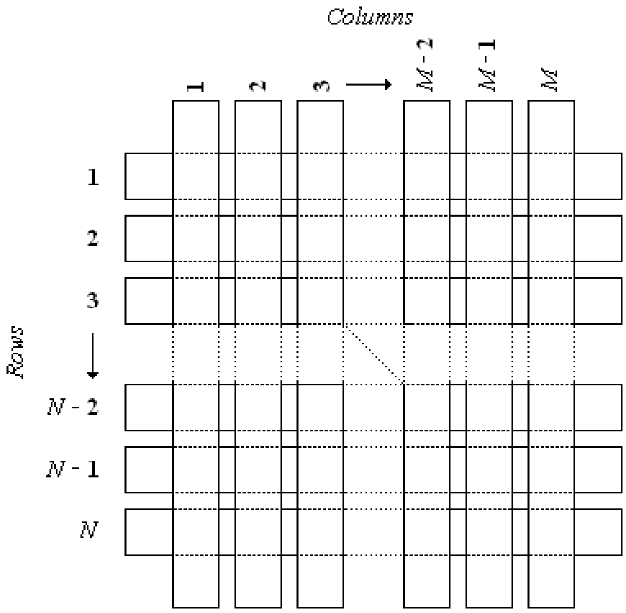


Figure 2.3: Illustration of the tactile array produced by researchers such as Hillis [48] to reduce the number of wires needed to address each cell.

This configuration, illustrated schematically in Figure 2.3, reduces the wiring complexity of a tactile array with *N* rows and *M* columns by only requiring $(\boldsymbol{M} + \boldsymbol{N})$ connections. This configuration is common for tactile sensing arrays, and will recur repeatedly throughout this review.

For such conductive composites, beneficial properties are generally considered to be good repeatability and low hysteresis, as well as easy detection of changes in resistance with pressure in an analogue with the requirements developed previously in this chapter [41, 49]. Unfortunately, graphite suspensions in silicone rubber in tactile applications have until now suffered from non-uniformities such that the chains of carbon atoms have already formed in production, limiting the useful range of the sensor and introducing greater variance in resistance profiles between sensors, an effect which is exacerbated with reducing sensor sizes [49]. Nonetheless, work on improving these mixtures continues, with work such as that of Hussain *et al.* who proposed a wet-mixing method to combine carbon black with a polymer to produce a pressure-sensitive device [50]. This method appears to improve uniformity by allowing near-homogeneous dispersion of the carbon black in the polymer. This enabled the researchers to achieve resistivity changes from 10^4Ω.cm when unloaded to 10^1 Ω.cm when loaded to a pressure of 1.2MPa [50]. Although lacking a robot-specific implementation, work such as this highlights on-going improvements in this area of work.

For thin skins, however, this dynamic range is relatively low compared to efforts involving suspensions of nickel in place of carbon. This type of suspension has been manufactured in various ways such as that by Chang *et al*. [51], but the most widely seen in the robotic application domain is the composite described by Bloor *et al*. and marketed as a quantum tunnelling composite (QTC) by Peratech Ltd [5]. The name is given from the postulation by Bloor *et al*. that under the influence of applied pressure, the nickel particles in the composite move closer together allowing electrons probabilistically to tunnel across the insulating polymer. This supposedly arises from the uniquely spiky morphology of the nickel particles, coupled with a mixing process that intimately coats each particle with the elastomer, avoiding the problem with chain formation that occurs in carbon-filled polymers [5].

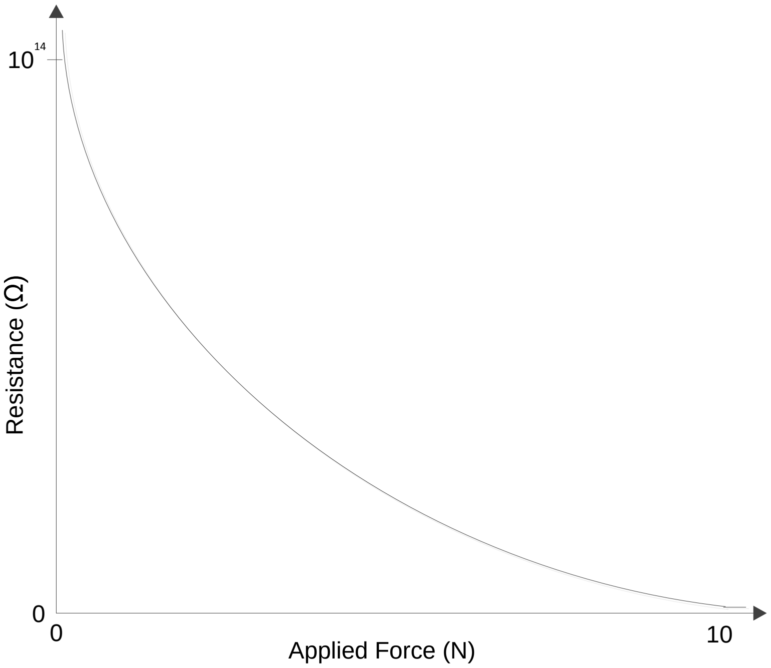


Figure 2.4: Force-resistance curve for quantum tunnelling composite deployed of the NASA/DARPA Robonaut project. Adapted from [52].

This elastomer has found use as an isolated sensor in two robotics applications; a teddy-bear robot produced by Stiehl *et al*. that was able to react differently to different levels of touch [53] and a robotic astronaut developed by NASA [52]. Both of these sets of researchers saw great appeal in the wide dynamic range of the sensors from $10^{14}\Omega$ when unloaded to as low as 1Ω when loaded [5, 52]. The typical force-resistance curve is shown in Figure 2.4, which has been adapted from the NASA work.

Stiehl *et al*. were able to differentiate between three levels of contact force, which they denoted soft, medium and hard by multiplexing a connection to the QTC to form three separate potential dividers to produce different thresholds for the different levels [53]. NASA used the QTC as part of a gripping application, where they were trying to make a humanoid robot grip delicate objects without dropping them. In order to boost the sensitivity, plastic beads were placed above each sensor on the robotic fingers in order to enhance stimuli from contact [52]. What had most appealed to the researchers was the sensitivity and flexibility of this composite, which caused them to move on from their earlier approaches with ink-based Force Sensing Resistors (FSRs) manufactured by Interlink Electronics [52]. Although not explicitly stated, previous work by Stiehl *et al*. also utilised this technology [54, 55] but subsequently replaced it with QTC.

An FSR consists of a semiconducting ink printed onto a flexible substrate mounted on a separate flexible substrate with electrodes printed onto it. These layers are separated by a flexible spacer. Pressure applied to the flexible semiconductor changes the resistance between electrodes, which can be measured to detect the stimulus. The range of pressures detectable is from 0.1 to 10kg.cm^2 with a resolution of 0.5% of the full range [54].

The FSR also comes in a variety of packages and package sizes, including linear potentiometer configurations, where both the position and applied force along a strip can be detected. As such, it has found extensive use as a testing ground for the data analysis techniques explored in Section 2.3. Generally, this type of sensor is deployed in small-scale two-dimensional arrays with isolated connections to each cell [56, 57], although other patterns such as a hexagonal [58] array for improved filling factor of the sensors, and a triangular [59] deployment have also been used.

Deployment on a larger scale can become prohibitive in terms of wiring complexity, or ease of accessing the data. On a small scale, this is not a significant issue, and may even be beneficial as the array scanning may be parallelised. For wide-scale deployment, however, custom designs must be considered. These methods typically involve screen printing rows and columns onto separate substrates and then combining them together in order to produce an array [60, 61].

Dario *et al*. applied this technique to a miniaturised fingertip to produce an 8x8 array with a higher resolution of 1mm in the centre of the sensor than the 4mm resolution utilised at the periphery [62]. These choices were based on an anthropomorphic argument without reference to a specific robotic task, but do in part reduce the data that must be extracted from the entire fingertip. Kawasaki *et al*. went further in their development of a robotic hand, which deployed an ink-based array with 624 sensing points although these authors found some difficulty with attaching the sensor to the hand [63]. The problems with these types of sensors is that the inconsistencies introduced by the screen-printing process means that there can be significant cell-to-cell differences in behaviour [60]. Furthermore, the sensors are subject to bending fatigue and so their ability to be conformed to a surface without degrading the sensor is questionable.

Other researchers have criticised these sensors for their high non-linearity and significant dependence on temperature, instead proposing the use of strain gauges as an alternative [64]. As force is applied to a strain gauge, its length changes, altering its resistance, which can then be measured. Up until now, the methods considered have only examined measurements of the normal component of force. In the work of Jong-Ho *et al*., however, four strain gauges were used in a tactile sensor, where the orientation of the gauges allowed all three components of force to be measured [65]. This is of particular use in gripping applications, where the additional components can aid in the detection of slip.

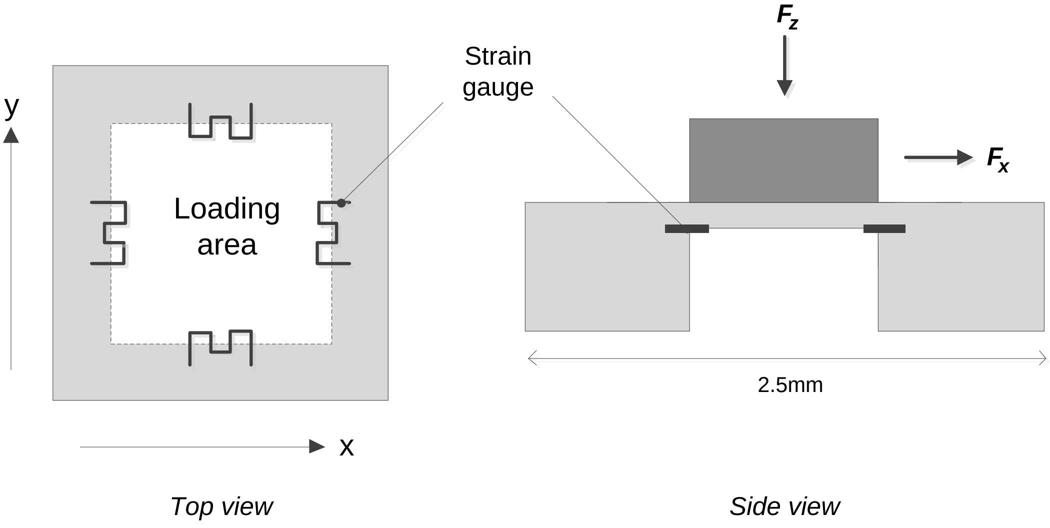


Figure 2.5: Illustration of the construction of a three-component force-detecting tactile sensor, as developed by Jong-Ho *et al*. [65]

Figure 2.5 shows how this sensor was constructed, which yielded a sensor with a force capacity of 0.6N. Given the small size of the cell, this is not a significant issue. However, the measurement method requires up to ten electrical connections per cell, which limits the practical area for which such a sensor could be deployed [66]. A less ambitious approach used a single strain gauge to measure the normal component of force up to 0.6N over a 2mm x 2mm sensor [64]. Nickel chromium was used to produce the strain gauge, which was sufficiently thin to produce a flexible skin when mounted on a flexible polymer. This loses the benefit to gripping applications, but does improve the ease of measurement. One concern with this method, however, is the low force measurement range when the sensing sheet is curved around a surface. This application will induce stresses in the material, and there is no discussion as to the effect this might have in prematurely saturating the measurement capacity of the sensor. Strain gauges find significant usage as the components of MEMS-based sensors, so will be discussed further in Section 2.2.5.

Other techniques in resistance-based tactile sensing have endeavoured to reduce the wiring complexity. A notable example of this is the so-called "hybrid" resistive sensor proposed by Hong and So [67]. This sensor consisted of two resistive surfaces of known resistivity with electrodes attached to both sides of each sheet. The two sheets were then arranged so that the electrodes were orthogonal to each other. When unloaded, the sheets are normally separated but application of force caused the surfaces to come into contact, forming separate voltage dividers across each surface. By taking two measurements, the *x*- and *y*-coordinates of the contact can be located [67]. This has the advantage of minimising the wiring over a large area, but there is an inherent inability to resolve multiple points of contact on the sensor, and there is no function of force. This limits the range of applications in which this sensor can be used and highlights the trade-offs that must be made in sensor design according to the intended application. With a similar motivation, Shimojo *et al*. proposed a wide-array mesh of force-dependent resistors contained between two grid-layers, as illustrated in Figure 2.6 [68].

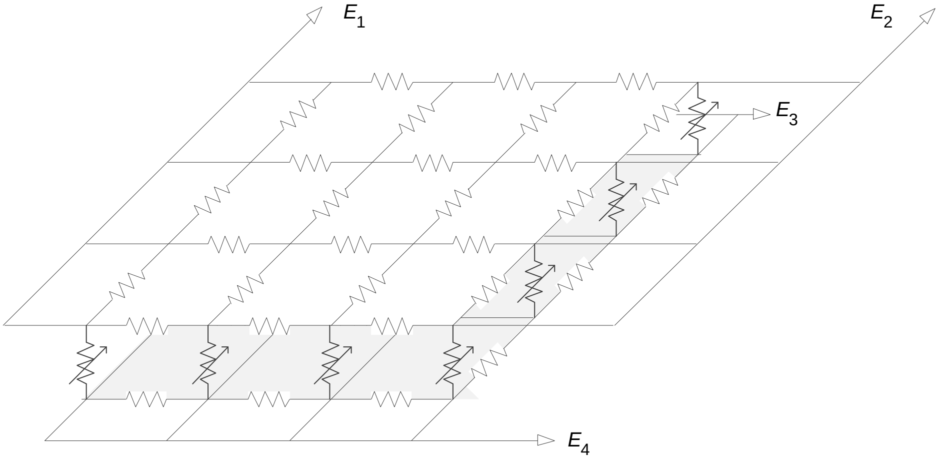


Figure 2.6: Illustration of the sensor design proposed by Shimojo *et al*. to minimise the number of connections to a tactile array [68].

This method only used four electrodes to measure voltages from the array, using these values to determine the force profile of an arbitrary tactile array. This was done by exploiting knowledge of the equivalent electrical circuits within the array, and allowed the determination of the centre of applied loads. However, this position detection was dependent on the loads applied, with errors in these values increasing significantly with increasing loads and numbers of contacts. Novel work continues on several fronts, including the replacement of polymer-based composites with conductive gels [69] and the impregnation of fabrics with pressure-sensitive inks and composites [70]. These have significant implementation challenges at present that limit their use, but the basic operating principle of detecting changes in resistance remains the same.

Tactile sensing through resistance changes is quite popular, principally because it has a simple operating principle. For high resolutions, however, many of the methods presented in this section would be inadequate as the drawbacks would outweigh the advantages in deployment. Further work exploring the possibilities of miniaturisation will be explored in Section 2.2.5.

Measured against the data-centric goals developed in Section 2.1, these methods broadly perform well. The sensors presented show monotonic profiles in their pressure-resistance dependency, and the general trend is that translation between measured resistance and stimulus is simple, with no computational complexity in all bar some isolated cases. The variations introduced by manufacturing methods may, however, mean calibration is required, limiting the ease of simple replacement of the sensor. Power consumption is likely to be a problem for resistance-based methods, since $\boldsymbol{P} = \boldsymbol{V}^2 \,/\, \boldsymbol{R}$. The problem is that most tactile sensors on a whole-body sensor will be unloaded for much of the time. If there is a low unloaded resistance, however, each unloaded cell will still draw significant power, which is undesirable. From this perspective, many of these sensors are adequate but not ideal. The exception to this is QTC, which has an extremely high unloaded resistance, changing by several orders-of-magnitude in response to contact. For an isolated sensor, this is ideal, but no array version has been found in the literature. It is concluded that the resistance-based methods broadly satisfy the data-centric criteria, but may offer limited capabilities in some application areas.

2.2.2. Capacitance

As with the resistance-based methods of tactile sensing, this category of sensors relies on the tactile stimulus altering an electrical property of the sensor. In this case the property is capacitance, but unlike resistance, this is used to detect one of two modalities: touch and proximity. As highlighted previously, this review is focusing on touch sensing, but the mechanism by which object proximity is detected can influence touch sensing in a negative way so will be examined.

This method has found the widest commercial deployment in recent years, with the touch screens used in the devices manufactured by Apple Inc. utilising touch sensing technology based on capacitive techniques [71]. The success of this is attributed by Dahiya *et al*. to the availability of commercial capacitance to digital conversion chips making their integration more simple [4]. This technology is not, however, pressure-sensitive and for that reason alone would be difficult to envisage for significant use in robotics.

Nonetheless, the capacitance-based approach in general has been widely used in robotics. The simplest touch sensor of this type is simply formed from two conducting plates surrounding an elastic dielectric material, but these sensors have been shown to suffer from parasitic capacitances formed by the wires connecting the sensor to the readout electronics [49]. This effect is particularly problematic as sensor resolution increases, since the total capacitance of the device is reduced accordingly. This means that the parasitic capacitance becomes increasingly significant. The simple device structure can be augmented with additional conducting plates to allow a three-axis measurement of force to be achievable. In the work of Rocha *et al*. this sensor was composed from a single plate on the top layer and four plates on the base layer [72]. This is illustrated in Figure 2.7.

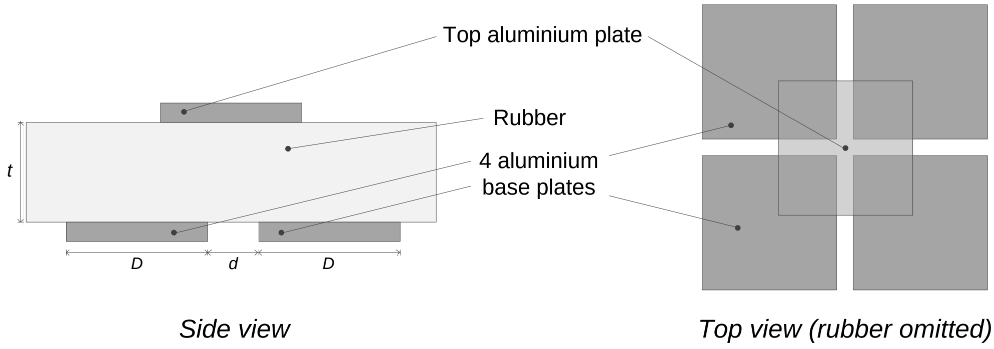


Figure 2.7: Illustration of a capacitive sensor capable of measuring three components of force. Adapted from [72].

This construction produces four capacitors from five conductors. As the rubber is deformed by a tactile stimulus, the area of the top plate that covers the base plates will vary with applied shear force. In a steady state, the four capacitances are equal to

$$C_1 = C_2 = C_3 = C_4 = \frac{\varepsilon_r A}{t} \qquad (2.1)$$

where

$$A = \left(\frac{D - d}{2}\right)^2 \qquad (2.2)$$

and where $\boldsymbol{\varepsilon_r}$ is the relative electrical permittivity of the dielectric, and *D, d* and *t* are the lengths of features shown in Figure 2.7. As shear force is applied, the area term changes and the capacitances are no longer equal. A series of simultaneous equations can be used to determine the applied force [72], but these equations do not yield unique solutions as the variables are over-constrained. The authors' approach to remedy this is to average the results of solving different subsets of the equations, but this appears computationally inefficient, particularly if applied over a large area.

Moving into applications in the domain of arrays, the work of Shinoda *et al*. describes a dual structure consisting of two layered capacitors, one using a soft dielectric and the other utilising a hard one [73]. Measuring the changes in capacitance produced by contact, the authors discovered a correlation between the change in capacitances of the two layers and the applied force and contact area of the impacting object.

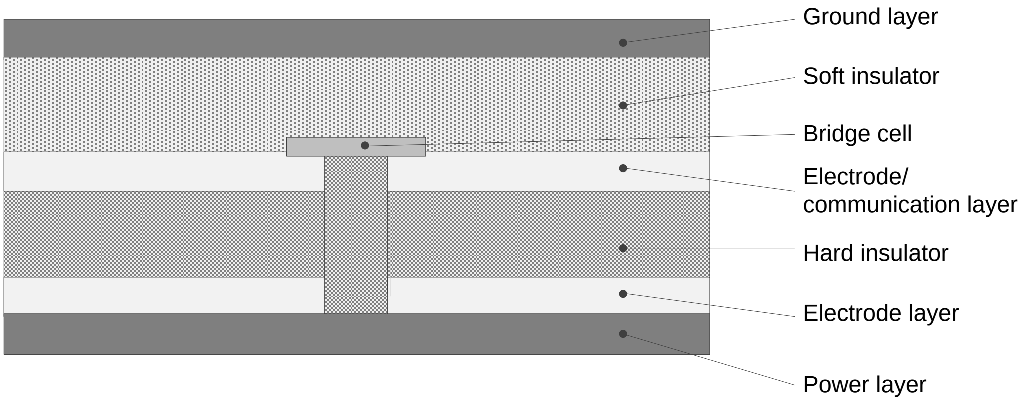


Figure 2.8: Illustration of the dual-capacitor structure described by Shinoda *et al*. in [73].

This system relies on relatively low sensor resolutions, and the experiments performed show that it was only possible to estimate the contact position when it crossed sensing cells, due to the impossibility of measuring its movements within one. This is not a significant problem provided the sensor resolution is acceptable for a given application, especially since the contact area can be estimated from the data. One novelty in this application is the data transmission method. As shown in Figure 2.8, a so-called bridge cell connects the separate sheets of the topmost electrode layer together. This cell consists of a chip that takes measurements of the capacitances, and then uses the electrode/communication layer to transmit the information to the robot's controller by hopping the data through the other bridge chips within the skin [73].

This allows the structure to exist without lengthy wires, although the complexity of the device overall is increased since a power source for the chips must also be introduced. It does, however, neatly open the possibility for satisfying the local processing and data redundancy criteria established earlier. A more conventional approach to array formation was adopted by Hyung-Kew *et al.*, who formed an array by overlapping two copper meshes at orthogonal angles [74]. The structure formed is shown in Figure 2.9 and is similar to a structure proposed by Son *et al*. but with greater capabilities, as will now be discussed [75].

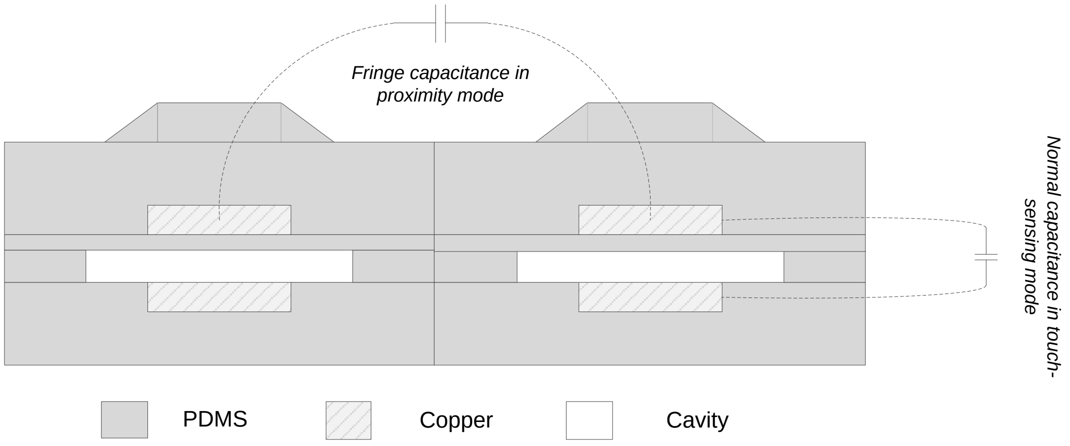

Fringe capacitance in proximity mode
Normal capacitance in touch-sensing mode
PDMS
Copper
Cavity

Figure 2.9: Capacitive sensor array produced by MEMS processes by Hyung-Kew *et al*. in [74].

Although these are formed using MEMS processes, which are principally discussed in Section 2.2.5, the transduction method is so ubiquitous in this area of tactile sensing that it will be discussed here. The fabrication process is best understood from their earlier work producing tactile sensors [76]. In brief, however, the result is copper electrodes embedded in polydimethylsiloxsane (PDMS), with PDMS used to produce surface bumps to enhance transmission of the stimulus to the electrodes.

As shown in Figure 2.9, the sensor is capable of making measurements using two capacitances. By using the top electrodes as the capacitor, the change in fringe capacitance can be measured which allows objects up to 17cm away to be measured [74]. Reconfiguring the connections to measure capacitance between the top and base plates allows contact force to be measured from the change in separation. This is therefore a dual-mode sensor in a highly compact form that offers significant advantages if both modes are desired. The system measures picofarad changes in order to measure these characteristics, but it is not clear what measures are proposed to eliminate noise and stray capacitances. Without this real-world implementation data, the usefulness of this sensor is difficult to assess. For the sake of completeness, it is worth highlighting the work by the same authors in producing a three-axis force measurement sensor using the same fabrication processes, and employing the same measuring processes described previously [77].

Cotton *et al*. sought a means of producing a capacitive skin that could easily stretch to conform to the surface of objects or to move with the joints of a robot without damage [78]. This was done through appropriate choice of materials, with thin gold films separated by PDMS, which were then embedded in silicone rubber. These materials have an appropriate mechanical compliance to be deformed without lasting damage.

The authors describe three possible modes of operation; strain, touch and pressure, although their definition of touch appears to be the same as the proximity sensing of Hyung-Kew *et al*. [74] The key difference is the addition of electrode interconnects attached to the conducting plates being used as source of capacitance detection for the fringe detection effect. The problem with so many modes integrated into a single, yet simple, device structure is that the various modes are interdependent: thus, for example, the effect of strain is hard to differentiate from the effect of pressure. This is evidenced in the experimental results, where the error bars at increasing levels of strain show that the effect of increasing pressure can be masked [78]. The size of the sensing cells is also so low that the device is constrained to measuring difference on the order of tens of femtofarads, which may be difficult to distinguish from noise or require extensive signal processing.

A simpler approach to these micromachining techniques is to use low-cost flexible fabric substrates surrounding a dielectric material, as described by Sergio *et al.* [79]. The conducting fibres of the fabric were used to form a row and column matrix around the dielectric material. The authors considered the possibility of achieving conductivity through various means such as conductive ink impregnation and weaving conducting fibres into the textile [79]. The problem with these stretchable designs is that they are intended to allow sensors to move over the joints of the robot. However, no discussion is made in regard to locating these dynamically located sensors. Consequently, a detected tactile stimulus cannot be located relative to the environment.

In all of the above, the elastomers used introduce non-linearities and hysteresis, which at the small capacitances being used can have a more profound effect than in the case of the resistance-based methodologies. This problem seems to be exacerbated by reducing the size of the sensors, as was encountered by Gray and Fearing, who manufactured a sub-millimetre resolution array and discovered severe hysteresis beyond that attributable to the effect of the elastomer [80]. It was postulated that this resulted from the effect of imperfect formation of the sensor structure and the interfaces between materials at this scale [80].

One effort to avoid this problem entirely was described by Peer *et al.*, who constructed a sensor from two conductive membranes, as in a microphone

[81]. Only one membrane was able to move in response to contact, and the other was fixed as shown in Figure 2.10.

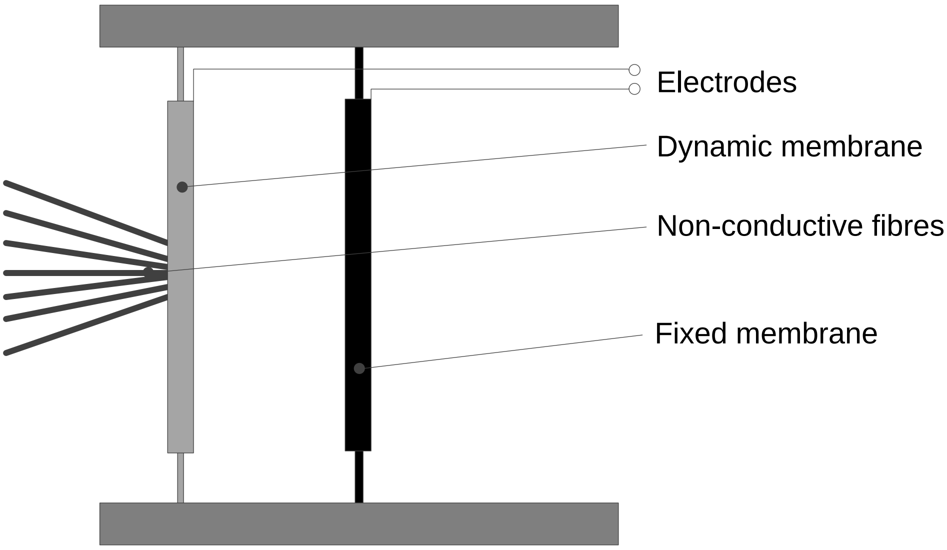


Figure 2.10: Illustration of the sensor constructed by Peer *et al.* that does not use any elastic materials. Adapted from [81].

The dynamic membrane had fibres attached to it to prevent the capacitor from coupling from the environment. These fibres come to rest quickly, and this along with the absence of viscoelastic material limits the hysteresis in the sensor. However, it is difficult to see how this could be extended to a wider array and the need to spend 300ms recharging the device after a contact limits the applicability of this method in practice [81].

Capacitance-based methods of tactile sensing afford touch sensitivity at a similar level of implementation complexity to the resistance-based methods. The technique is well developed and is used in commercial sensors [71, 82]. However, there are significant issues such as the difficulties scaling the sensor resolution up that relate to the impact of stray capacitance. The ability to perform proximity sensing is useful if multimodal operation is desirable, but it also provides a lesson for tactile sensing that local conditions can disturb the electric field of capacitors, whether this is desired or not [83]. There is also the possibility that the capacitors may couple with their environment, particularly if obstacles are metallic and the signal conditioning circuitry and subsequent processing would need to account for this potential interference. This introduces a data-processing problem that has no counterpart in the resistance-based methods. Nonetheless, capacitance-based methods are well-established and cannot be dismissed when considering deployment for robotic applications.

2.2.3. Piezoelectricity

Mechanical stress induces charge generation in piezoelectrical materials, rendering them suitable for detecting applied force and pressure. The charge generated produces a voltage across the material, which can be used to quantify the level of stimulation. Films of polyvinylidene difluoride (PVDF) are the material of choice for tactile sensor designers because they are able to withstand harsh environmental conditions, are flexible and are chemically stable [4, 84]. Non-film piezoelectric materials have also been discovered, notably the polyelectrolyte gel described by Sawahata *et al*., which exploits the piezoelectric properties of the gel to power a circuit to generate light in response to tactile stimulus [85]. It should, however, be noted that such materials respond only to changes in applied force and pressure, and cannot directly measure the static component of force. This means that the resulting sensors are a form of dynamic tactile sensing that is particularly suitable for detecting other object features such as surface texture [86].

For this reason, in one of the earliest implementations by Dario *et al*., a sensor was constructed consisting of two PVDF layers with a pressure-sensitive elastomer, as illustrated in Figure 2.11 [84].

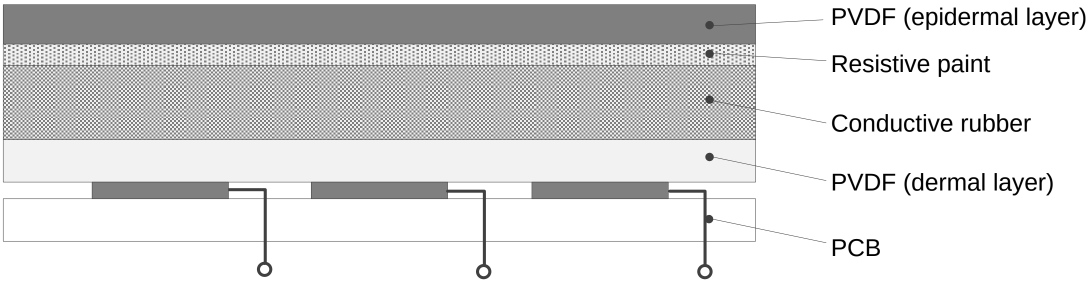


Figure 2.11: Illustration of the sensor design by Dario *et al*. incorporating PVDF layers for dynamic feature detection [84].

In an effort to reproduce human-like qualities in the sensors, the lower layer of PVDF was intended to emulate the sub-dermal receptors of the human skin, with the upper layer used to directly detect textural variations when the sensor moves across a material. The elastomer provided the measurement of static force by the processes described previously. Piezoelectric materials may also exhibit pyroelectric effects, and in this case, the resistive paint layer allows portions of the PVDF to be heated to a known temperature. Changes in charge will result on contact with a cooler body as heat dissipates, and the level of charge build up provides a measure of thermal conductivity [84]. This is a similar operating principle to other more conventional sensors devoted to thermal sensing [87]. Yuji and Sonada exploited this property in developing a multimodal touch sensor using a capacitance technique for the static force and PVDF for the temperature measurement in what was a derivative of the work of Dario *et al*. [88]. They overcame the piezoelectric effect of contact by using a material for their touch-sensitive capacitor that has a much lower stiffness than PVDF so that the effect of pressure was absorbed and the PVDF thus simply acts as a thermal sensor. It is unclear, however, that a structure could be made to isolate the PVDF from temperature effects in this way without it being embedded within another material.

Dario *et al*. showed that there was a high sensitivity to texture, even in the sub-dermal layer where it was anticipated that the elastomer would damp most of the vibrations [84]. Nonetheless, the paper presents some unfortunate drawbacks to the use of these materials in sensors. The most significant is that it is difficult to manufacture them into a dense matrix form such as that for resistance- and capacitance-based methods in order to reduce wiring complexity. This is because the charge generated by a single sensing point would become distributed across the rest of the material, making location of the contact impossible [84]. Multiplexing with limited crosstalk is possible, but does not resolve the fundamental connectivity issue.

Despite this, PVDF has been exploited in many tactile sensing devices. Kolesar and Dyson used it as the dielectric material for a capacitor, which allowed the charge build-up to be measured directly as a voltage across conducting plates [89]. The relationship between the surface charge $\boldsymbol{Q}$ and the applied force $\boldsymbol{F}$ can be written as $\boldsymbol{Q} = \boldsymbol{SF}$, where $\boldsymbol{S}$ is the charge sensitivity constant. Combining this with the standard relationship between voltage, capacitance and charge gives the relationship between force and voltage in this configuration as

$$V = \frac{SF}{C} \tag{2.3}$$

where $\boldsymbol{C}$ is the capacitance.

Kolesar and Dyson produced 0.5mm resolution sensors in an 8 x 8 array based on a silicon integrated circuit. The results were impressive, showing a linear relationship between output voltage and applied force up to 1.35N, whereafter the polymer saturated [89]. However, these results were for single tactels and in the images taken by the sensor it was clear that inhomogeneities in the sensing cells caused irregular readings when these cells were not stimulated. This occurred despite specific efforts by the authors to avoid it through a pre-charge stabilisation technique that takes 0.1s to complete [19, 89]. Crucially, these experiments were also conducted in an isothermal manner to limit pyroelectric effects, which means the tests are not a good indicator for real-world performance.

Generally, piezoelectric techniques are exploited for detecting dynamic events, so it is unsurprising that considerable focus has been given to their use in gripping applications where slip detection is regarded as essential to the task of grasp maintenance. Cranny *et al*. demonstrated the use of a piezoelectric material for detecting slip in a device intended for deployment in a prosthesis [90]. The material used was lead-zirconate-titanate (PZT), which is a material with better piezoelectrical properties than PVDF but is difficult to miniaturise [91]. The operating principle for detecting slip is simply that if an object is being gripped, any vibrations detected by the PZT imply that the object is slipping [90, 91]. The same principle was employed using PVDF in a robotic application by Byungjune *et al*. in combination with a piezoresistive sensor for contact force detection [92]. This was later deployed in a robotic hand and shown to be effective in maintaining grasp [93].

Other sensors exploit the ability of piezoelectric materials to detect vibration in parallel with their ability to be made to oscillate through the inverse piezoelectric effect.

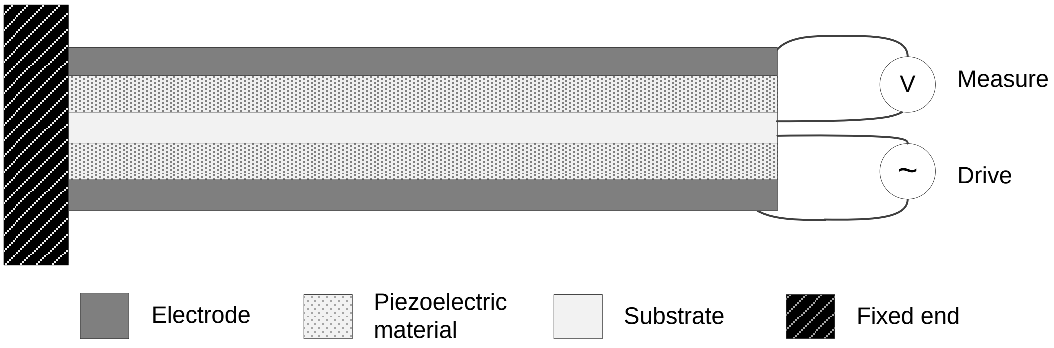


Figure 2.12: Illustration of a tactile sensor using two layers of piezoelectric material as described in [94].

Motoo *et al*. constructed the sensor illustrated in Figure 2.12, where one of the piezoresistive elements is driven by an alternating current to cause it to vibrate and the other element is used to measure vibration in the entire sensor [94]. Applied forces change the mechanical impedance of the sensor, leading to a measurable change in the vibration of the sensor. By applying different elastic body coatings to the device, the range of impedance change can be widened, enhancing the range of force detection. Although a simpler structure involving only a single piezoelectric element for detection is feasible, this dual structure is claimed to allow some distinct advantages such as this improved range of force detection. One interesting advantage is the capacity for self-detection of faults, as a failure in either piezoelectric element can immediately be detected.

In another application involving detecting induced vibration with PVDF films, Shinoda and Ando describe detecting ultrasonic acoustic waves produced by contact with a hemispherical structure [95]. The operating principle is that contact events produce ultrasonic emissions that are transmitted towards the centre of the device. A 2 x 2 array of PVDF films can detect the vibrations and the relative intensities of the detected stimulus can allow the exact location of the contact event to be detected [96]. This is ideal for isolated sensors, although somewhat complex for fabrication into a wider-scale array.

Other methods relying on the deformation of a cavity on contact have been proposed using ultrasonic emitters and receivers to ascertain that cavity deformation has occurred and to quantify it in order to determine touch [97]. These do not use piezoelectric elements, and are also difficult to fabricate and require two separate devices to emit and receive the ultrasonic signals, which would require significant power consumption over a wide area. The mechanical resonant frequency of piezoelectrical materials also changes if they are in contact with objects, and this feature has been exploited to measure force and pressure [98]. Krishna *et al*. sandwiched PZT between a matrix of electrodes to form an array of independently resonatable sites and measured the resonant frequency of each site. Applied strain altered the resonant characteristics, and this change could be measured allowing force to be profiled.

Finally, the work of Kimoto *et al*. in creating a multifunctional PVDF-based sensor should be noted [99]. In this work, PVDF film was sandwiched between two electrodes and different circuit configurations used to measure different properties of a material such as hardness and viscosity, and with the addition of a second identically constructed sensor, the capacitance of the object could be measured through the fringe capacitance change on two of the electrodes [99]. This allowed ten different materials to be discerned, but still requires multiple sensors to make a single determination, which may prove difficult to scale. Kimoto *et al*. intend to improve this sensor through the use of additional films [99], which will further increase the measurement complexity.

The piezoelectric methods described here clearly exhibit useful traits in tactile sensing, particularly in the domain of dynamic tactile sensing where features such as slip can be detected more easily than the resistance- or capacitance-based methods where such detection is difficult or impossible. However, piezoelectric films cannot directly measure static contact forces and the indirect means of doing so either requires bulky structures that mediate static contact into a dynamic event or require additional piezoelectric elements that induce oscillations that may vary with static contact. The former approach is satisfactory only in the case where a limited number of sensors are required, since localisation of the contact events for a single cell requires multiple sensors. The latter requires continuous power supply and so could have a detrimental impact if widely deployed.

It is therefore concluded that piezoelectrical devices are supplementary to direct detection of contact, but are ill-placed to be an alternative. It is for this reason that Dario *et al*. [84] and other researchers have tended towards including additional structures for detecting static contact forces. As devices, there are also significant issues when compared to the data-centric requirements developed previously. The wiring complexity needed in the field makes accessing the information difficult. The simple cases of use, such as using PVDF as a dielectric in a capacitor, are easy to process but other structures and the inhomogeneities in the film structures implies the requirement for a greater level of post-processing with a commensurate increase in required computational complexity.

Piezoelectrical devices should therefore, in general, be confined to limited areas of a robot's surface. Given the significant need for dynamic sensing in gripping applications, small-scale deployment in the form of isolated sensors or limited arrays on grippers or fingers would be most appropriate. However, the limitations in technology on this scale render it difficult to facilitate the wide-scale of deployment required for a whole-body skin.

2.2.4. Optics and Imaging

The use of optical methods in tactile sensing has many appealing attributes for researchers. By using sensors that manipulate the transmission of light, the problems associated with electronics such as stray capacitances or thermal noise can be avoided. Additionally, since many advances have been made in processing images for cameras, the exploitation of light-based devices allows techniques from this domain to be re-used in the tactile area, even affording the possibility of simple fusion between the two sensor types.

The most rudimentary example of this was the work by Saga *et al*., where a silicone rubber sheet was used as a reflective surface upon which an image was projected from below the surface [100]. This reflection was recorded by a camera and then machine vision techniques were applied to determine the location and depth of contacts that would distort the rubber, and therefore the image reflection. This technique yielded good results in simulation, but poorer results under experimental conditions, where the sensor was unable to resolve a broad planar object due to insufficient definition of the edges [100]. This implementation suffer most from the use of a camera located so close to the device, which requires multiple cameras to cover a larger area or a more distant camera to have a wider field of vision, both of which are bulky and costly solutions.

Instead of locating a camera directly beneath the surface, fibre optic cables can be used to transmit the surface image back to a camera in a central location, as demonstrated by Yamada *et al*. [101]. Their sensor is illustrated in Figure 2.13, and consists of regularly distributed reflector chips embedded in the surface of a deformable, transparent gel illuminated from below.

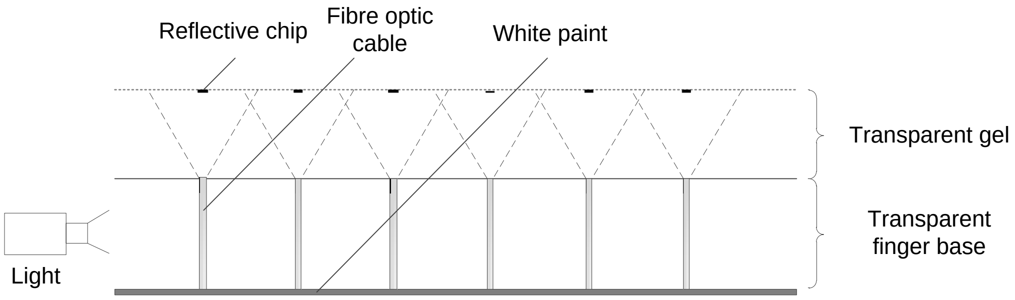


Figure 2.13: Illustration of fibre-optic cables used to extract image of reflector chips embedded at the surface of a tactile sensor, as shown in [101]. Dashed lines indicate the field of view of each fibre optic grating.

The image is detected by a camera at a central location, with the image guided by fibre optic cables. Applied force deforms the surface of the gel and so redistributes the reflector chips, and the image can be analysed to determine the 3D depression caused by the stimulus [101]. The complexity in this sensor derives from the image processing, which Yamada *et al*. extended in a later work in order to identify the orientation of objects based on models [102]. The problem with the technique was the easy deformity of the silicon gel, which under certain conditions cause the chips to deform outside the range of their corresponding image guides. The choice of material was also considered to be inappropriate for gripping applications due to hysteresis [102], but the deployment of one optical fibre for each reflecting chip raises further questions about the ease of deployment for such a technique over a wider area than a fingertip.

Kamiyama *et al*. proposed a similar device, but used two layers of coloured markers imaged from below by a camera based on a charge-coupled device (CCD) [103]. The surface was covered with a black opaque layer of the same silicone rubber used to produce the sensor and lit from within by white LEDs to limit the effects of ambient lighting. The two layers of markers consisted of red and blue markers, with a single colour for each layer. Using models of the elasticity of the rubber and its related behaviour under the influence of applied force, similar analysis to that of Yamada *et al*. could be performed to identify distribution. However, Kamiyama *et al*. stated that a single layer was insufficient and the second layer at a different depth allowed the depth and therefore magnitude of the stimulus to be assessed [103]. Although the application domain for this work was not directly in the field of robotics, it does suggest a possible inexpensive approach.

The use of CCD cameras in touch sensing for robotics is not unknown, and an approach similar to the previous work was described by Ferrier and Brockett, who used one to image an array of dots printed on the underside of a membrane that encapsulated an incompressible fluid [104]. In all cases, these sensors are dependent on implementing a complex inverse model to map the observed movement of the dots to deformation of a mechanical structure.

A different approach is to use a transducing material that generates light in response to tactile stimulus to be detected by the camera. Such a device, based on a 100nm thick structure of metal and semiconducting layers was reported by Maheshwari and Saraf [105]. This device consisted of alternate layers of gold and cadmium sulphide monolayers separated by dielectric layers, where tunnelling via the Fowler-Nordheim process into the cadmium sulphide layer produces an electroluminescent effect. According to the paper, the dielectric layers are around 16nm thick when unloaded, allowing tunnelling, and the imposition of stress narrows the dielectric allowing greater tunnelling currents through the cadmium sulphide layer. This leads to a roughly linear relationship between stress and electroluminescent intensity [105].

The overriding concern with using computer vision and the associated hardware such as camera is that the sensor becomes increasingly large and difficult to deploy in robotics. The methods also rely on trying to turn tactile sensing into another branch of computer vision, but this negates the advantages of tactile data as a less-detailed and complex data provider by imposing the data requirements of a vision system upon it. The use of photo-detectors in this form was also proposed by Ohmura *et al*., who constructed an array consisting of the electrical circuit similar to that shown in Figure 2.14 mounted beneath a urethane foam [106].

When the urethane foam is compressed, the light emitted by the LEDs is scattered and detected by its accompanying photo-detector and the stimulus is therefore quantified by the intensity of detected light. The problem was that each LED consumed 50mA of current, so the circuit in Figure 2.14 constitutes a revision designed to allow only particular banks of LEDs to be active at a given time so as to lower overall power consumption [106].

A similar scheme created isolated sensors where a tactile stimulus would act as a plunger-like system, obscuring the light from an LED that pointed directly at a photodiode in a predictable way [107]. These configurations do not, however, improve the situation over cameras since they are also generally large in size. Additionally, even if these problems can be mitigated by miniaturisation, the advantage of the optical approach is lost by the almost immediate and local conversion back into the electrical domain. This means there is increased complexity and power consumption for no substantial gain.

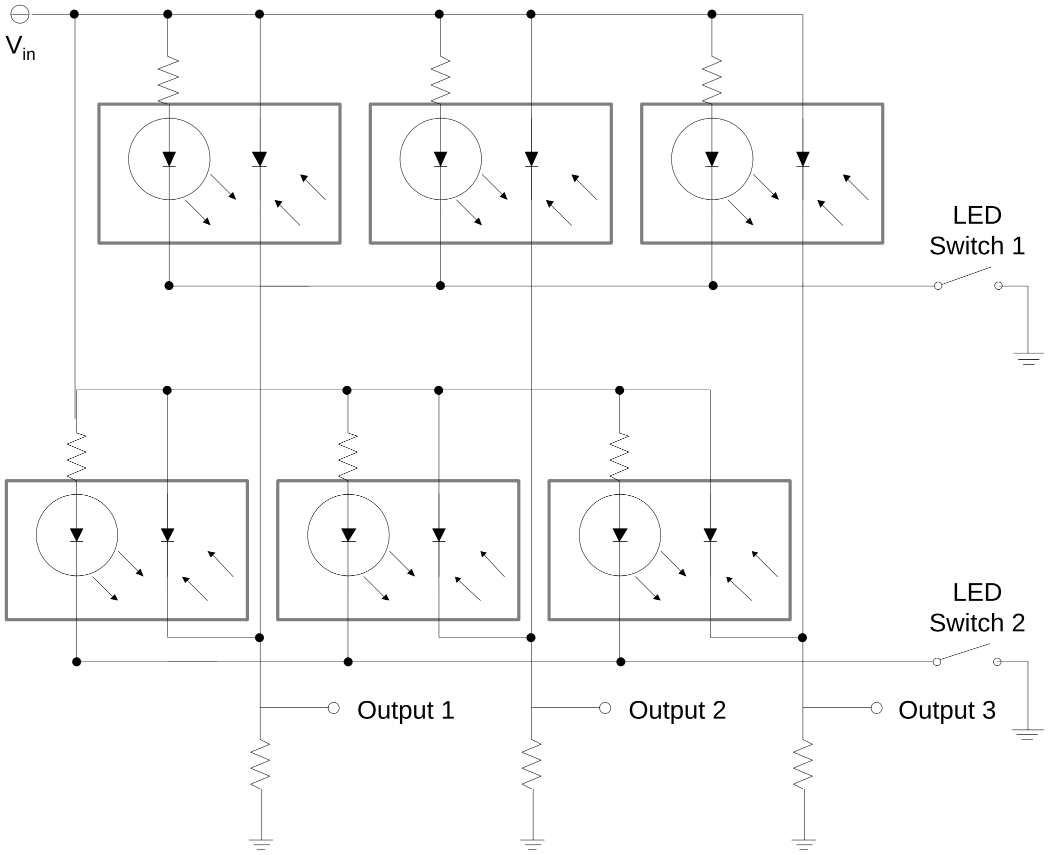


Figure 2.14: Circuit diagram of the revised LED-photodiode array devised by Ohmura *et al.* for tactile sensing in [106].

In response to this, one final class of optical methods must be considered. These are based on fibre Bragg grating sensors, which are made by manufacturing an axially periodically-varying refractive index along the core of a fibre-optic cable [108]. When broadband light is directed through such a fibre, the grating will reflect a narrow band of light with a central wavelength given by

$$\lambda = 2nT \quad (2.4)$$

where $\boldsymbol{n}$ is the effective refractive index of the fibre core, and $\boldsymbol{T}$ is the period of the grating. As the fibre is placed under strain, the period of the grating changes, which alters the Bragg wavelength. A sensor based on this method was shown to correctly discern between being loaded or unloaded with no errors, compared to a 70% success rate for the same test applied to a resistive strain gauge [109]. For robot fingertips, these fibres have been used with multiple Bragg reflectors embedded along the core with different steady state wavelength embedded within them to allow multiple strains to be measured from the application of a single broadband pulse [110].

One issue with these sensors is their sensitivity to temperature, since thermal excitation will also change the period of the gratings. One method proposed is to use a second, identical fibre that is not mechanically deformable in order to measure the wavelength shift induced by temperature [110], but this requires a second set of readings and interpretation. An alternative is to design the structures such that temperature changes do not affect them, for example by the inclusion of meshes made of materials with high thermal conductivity, such as copper [111].

The principle advantage of these fibres is that the sensors are also the data transmission medium, creating compact structures within the skin. However, this may be balanced by the significant space required by processing where lasers and optical processors are often required. Sensors have a resolution limited by the number of sensors embedded in the fibre, but the computational intensity of de-multiplexing the sensors increases significantly with increasing numbers of sensors. Consequently, their use may prove prohibitive in terms of time, space and cost in a range of applications.

The conclusion is that optical methods of transduction may not be appropriate for wide-area deployment, because they do not readily satisfy a number of the data-centric criteria defined previously. In the case of fibre-based systems, the data is difficult to process locally because of the complex interrogation procedures and optics required. This complexity also limits the scope for data rejection and simplification before the information reaches the robot's controller. In other optical methods, the need to generate a light source imposes a significant power burden on a system. Camera-based methods also have a difficult method for translating the images into force-data based on inverse mechanical models of the sensor as well as practical problems in locating the camera in a suitable location.

Further advances in this area, including advances in miniaturisation may mitigate these effects. At present, however, detection using these methods is limited by the difficulties in accessing, processing and simplifying their output data.

2.2.5. Miniaturisation Techniques

This section will examine miniaturisation techniques employed in pursuit of tactile sensing, but this does not involve introducing new means of transduction. Nonetheless, such techniques are increasingly employed in pursuit of high-resolution, low cost tactile array and so are worthy of consideration.

One early approach using integrated circuits in a silicon wafer was presented by Raibert and Tanner [112]. This used conductive elastomer placed over discrete sensing sites implemented using very-large-scale-integration (VLSI) techniques in a wafer to enhance resolution and to allow a high degree of local processing if necessary. Nonetheless, the underlying structure in such a sensor is brittle and difficult to make conform to the curved surfaces of a robot and consequently has limited application. Other, more recent, approaches have built the tactile transducer directly into the semiconducting structure, such as the piezoelectric oxide semiconductor field-effect transistor (POSFET) arrays developed by Dahiya *et al*. where piezoelectric materials were spin-coated directly onto the silicon wafer [113, 114]. The brittle nature of the substrate was, however, acknowledged and these devices were therefore principally proposed for fingertips.

Many current developments in sensors involve MEMS in some way, with constructions both on silicon and plastic. Vásárhelyi *et al*. described a micro-machined porous Si-membrane that was capable of shear-force sensing [115]. In this sensor, a central plate is suspended over a cavity by four bridges. A piezoresistor is connected to each bridge and a measurement of the different bridges couples with an analysis of the response under strain can provide a measurement of normal and shear force. This is a relatively common implementation layout, with additional support structures also used to amplify the strains in the piezoresistors [116].

The use of piezoresistivity in this way has been demonstrated in array form by Kane *et al*. who produced a 64x64 element array of these suspended bridges using CMOS-compatible techniques [117]. The advantage here was that circuitry such as sensor selection for array scanning could be collocated with the sensitive cells, reducing the amount of external circuitry that was required. With resolution equal to or greater than that of the human fingertip and a broadly linear response to normal and shear stress, such a sensor has significant deployment scope in the area of robotic fingertips, but suffers from an inflexible silicon substrate.

This array was also coated in an elastomer to prevent damage to the delicate structures of the sensing array, which have low flexibility. Alternative structures to the bridge structure also exists, using layered cantilevers with attached piezoresistive sensors that are responsive to strain [118]. The cantilevers used here were also embedded in elastomer and the significant hysteresis that results is a driving force to avoid the use of elastic polymeric materials. Such an approach can also be criticised because it introduces a low-pass filtering effect in the material. Elastic materials also make it difficult to determine the contact force distribution because there are multiple force configurations through an elastic medium that can produce a particular set of tactile sensor readings.

A novel alternative to this was proposed by Takao *et al*. by building miniaturised piezoresistive strain gauges on a thin diaphragm made of bulk silicon, as illustrated in Figure 2.15 [119]. The pressure within the diaphragm is determined by pressurised air provided through a hole on the base of the structure, allowing the stiffness of the array to be controlled by air pressure [119]. By controlling the pressure to include an alternating component, Takao *et al*. were also able to deduce hardness properties in a multi-modal version of their sensor [120, 121]. What is missing from these papers is a consideration of the additional requirements imposed on a robot from the need to provide a pressurised air supply, which causes some reviewers to dismiss it as an approach [4]. On the contrary, however, this may be a viable application for some probing applications, particularly in industrial robotics where the problems of providing and transporting an air supply are minimal and where the level of sensitivity achievable using this silicon-based method is desirable.

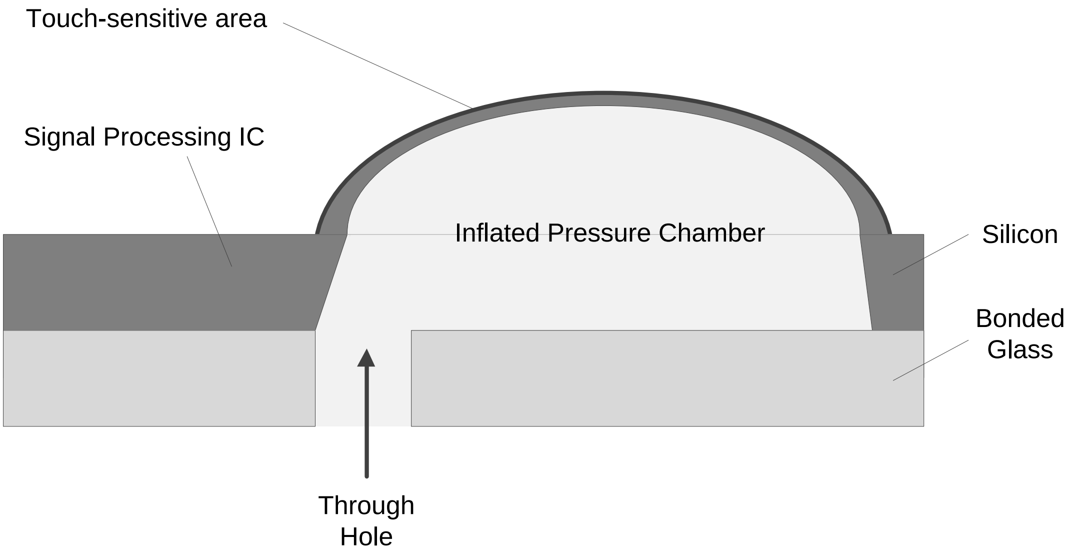


Figure 2.15: Illustration of inflated diaphragm approach to MEMS flexibility, as advocated by Takao *et al*. in [119].

Although these methods all rely on resistance-based methods, MEMS techniques also exploit capacitance including the work of Lee *et al*. discussed in Section 2.3.4 [76]. Chu *et al*. produce a somewhat different structure, where each cell consists of a silicon bump attached to a flexible membrane over a cavity formed in a layer of glass [122]. On the top of the glass in the cavity, and on the underside of the silicon are two electrodes, aligned to form four capacitors that afford sensitivity to normal and shear forces. These were implemented in an array, and showed similar behaviour to other capacitance methods although with less sensitivity due to the inherently lower capacitances achievable at this high resolution. Some work has also been done in exploiting the ability easily to create complex circuitry at a low level to eliminate common problems in capacitance-based methods. Such work includes that of Leineweber *et al*., who constructed an additional non-pressure-sensitive capacitor for each pressure-sensitive one to act as a reference capacitance and avoid issues with parasitic capacitances because the magnitude of this interference would be common to both [123].

Although the performance of these sensors is comparable with their non-miniaturised counterparts, there are problems with achieving flexibility for a wide-area skin. One more recent approach has been to use plastic substrates with MEMS-based techniques. One such approach by Engel *et al*. resulted in a sensor capable of variously measuring hardness, thermal conductivity, temperature and contact force using only polyimide polymer and metallic imprinting [66]. With four sensors per sensing pixel, however, the wiring complexity becomes difficult and in the absence of materials that can simultaneously be used to create processors, this imposes a high burden on a system.

Another approach to miniaturisation that seeks to use polyimide with metallic imprints is that of Eun-Soo *et al*. who spurn silicon because of the difficulties, expense and low yields that they claim are associated with constructing the silicon diaphragms used in other techniques [124]. This method simply used strain gauges layered underneath a polyimide bump that amplifies and distributes the applied force. Multiple gauges allowed the detection of shear and normal forces, but the sensitivity was low because of both the use of polymer and the relatively insensitive metal strain gauges [124]. The use of materials does, however, mean that the sensor is more durable that the Si-micromachining techniques, and was proposed for use as a ground-force sensor.

One way of avoiding the inherent fragility of these semiconducting structures is to use organic semiconductors. Someya *et al*. explored the use of such materials in conjunction with a pressure-sensitive rubber and produced a flexible 32 x 32 array with a resolution of 2.54mm [113]. The motivation to this work was the ease with which such devices can be fabricated, contributing to faster development and lower costs over comparable silicon-based techniques. A further advantage was realised in a later development by the same researchers, who moved towards customisation by aligning the wiring to the array such that the sensors could be cut to any size and integrated onto a robot [125]. This was a significant advance towards modularity, and conforms to the goals set out earlier in this chapter.

However, as the authors and others acknowledge, the performance of organic semiconductors is poor relative to their non-organic counterparts [4, 113] and the effect is that each pixel has a response time of 30ms and the achievable scanning rate is relatively low. Nonetheless, whilst this is an obstacle for deployment on fingertips where response time is paramount, deployment over other areas of the robot body may not be so limited. More recent work in this area has seen improvements through the use of structured PDMS films in conjunction with organic field effect transistors (OFETs) to produce sensors that claim immediate response times and relaxation times on the order of milliseconds [126]. However, the matrix form of this work is in an early stage and there are some limitations such as the effect of the elastic properties of the plastic substrates to be overcome [126].

One final technological area that may be of future interest is the use of thin-film technologies, such as those that have found widespread use in consumer electronics. By depositing silicon-based circuits onto elastomeric substrates, it has been possible for stretchable and foldable integrated circuits to be manufactured [127, 128]. One alternative approach has been in printing germanium- and silicon-base nanowires onto a substrate, coated with a pressure-sensitive material to provide a flexible array based on these inorganic semiconductors [129]. Although this offers a promising approach, the results of deploying this technique in a matrix array suggest that more development work is required. They suggest a significant number of defective tactels produced by the method. The images produced when known objects are placed on the sensor array, such as Figure 5 in [129], do not clearly show the outline of the shape and signals from adjacent tactels appear to be influencing non-contacted tactels.

Miniaturisation techniques offer significant scope to improve the resolution of tactile sensing arrays and to facilitate easier and more direct integration with modern electronics through similarities in material construction and scale. This in turn offers the possibility for processing and conditioning circuitry to be placed locally to each cell and the scope for using these circuits to satisfy the data-centric specifications developed earlier is immense.

There are, however, two significant issues to be countered. The most significant is the general lack of flexibility or durability in many of these devices, which can be mitigated through the use of other structures. These additional structures, however, tend to lessen the benefits of miniaturisation. The resolutions of these sensors are also extremely high, which may not be appropriate for a wide-area skin. It is clear, however, that this kind of sensor offers current advantages for use in fingertips or grippers and future developments using flexible materials may render these techniques suitable for a wider range of uses.

2.2.6 Conclusions

This section has examined a variety of transduction methods, and shown the various classes of implementation methodology within them. Of the five categories considered here, an evaluation based on the data-centric specifications developed in Section 2.1 suggests that for touch-detection, the resistance- and capacitance-based methods are best for wide-area sensing, with miniaturisation techniques facilitating deployment of various transduction methods in the domain of gripping applications. This is principally because of the simplicity of the data-collection technique and the ease of translation between an electrical quantity and stimulus. It is worth noting, however, that the precise implementation of the transduction methods affects their utility for tactile sensing. For example, the ten-connection sensing cell based on strain gauges by Jong-Ho *et al*. [65] is difficult to justify over a large area because of the complexity of individual addressing, compared to the relatively simple arrays formed from intersecting strips of electrodes. The trade-off, however, is in the amount of data extracted from the environment and this will vary based on application.

One material of particular interest in the resistance class of materials is QTC, given that it has an extremely high unloaded resistance. For a robot skin spread over a large area, it is reasonable to assume that most sensors will be unloaded for most of the time. Having a high resistance under these circumstances is beneficial because the power consumption will be very low. In the case of QTC, the change in resistance is by orders-of-magnitude when loaded and this gives a sufficiently clear response to load. There is a lack of a matrix implementation using this material where each cell does not require its own connection to the scanner.

It can be concluded that there are a wide-variety of tactile sensors available for deployment in robotics, and one trend is the increasing recognition by researchers that their sensors have limited deployment potential. This is particularly evident from papers designing sensors specifically for fingertips and grippers. The other half of a data-centric examination of tactile sensing is the way that data is used. That is the subject of the next section, whereafter it will be possible to take a rounded view of the field as a whole and draw some conclusions.

2.3. Tactile Data, Imaging and Control

Information from tactile sensors can be used for one of three broad classes of task: manipulation, exploration and reaction. In the first category are the applications where a robot may grasp an object in order to move it or control it in some way. Exploration is a task devoted to discovering new information about an object or about the environment in general. Reaction is simply to define a response to contact events from the environment, and may be used for the purposes of safety or for functions relating to the posing of robots by direct interaction. This third class is not particularly pertinent as programming a response to the existence of a tactile stimulus such as a stop motion is more dependent on the robotic system involved than on the tactile system.

This leaves manipulation techniques, which have tended to revolve around identifying contact points on gripping fingers, object recognition and exploration which additionally involves building maps of the environment. The remainder of this section will examine the common approaches in these two areas.

2.3.1. Tactile Data and Manipulation

Research into the use of data from tactile sensors has principally revolved around their deployment on gripping fingers. In almost all of the sensors described in Section 2.2, images were produced to highlight the ability of the sensor to detect a tactile stimulus. These will not be repeated here, and consequently this section will focus on the use of this image data.

There are two types of tactile sensing to be considered: intrinsic and extrinsic [7]. Intrinsic tactile sensing involves the derivation of contact information from internal force and torque measurements. Extrinsic tactile sensing utilises sensors of the type explored in Section 2.1 mounted externally on the robotic surface to detect contact events. One of the earliest descriptions of the intrinsic system was made by Salisbury, who showed that an inverse Jacobian transform, produced as a function of joint positions, could be used to identify the location of a contact on a fingertip [130]. These approaches often produce ambiguous information sue to a lack of a one-to-one relationship between contact and joint configuration, so a six-axis force/torque sensor is typically integrated within the fingertip of the robotic device and the models of contact force and location are adjusted based on this additional information [130, 131].

However, Salisbury's work precluded soft coatings on the fingertip or elastic objects being grasped in order to produce a simplified model [130]. The model was extended by Bicchi, who was able to derive the contact centroid for soft finger contacts using loose assumptions related to the solid mechanics of the system [132]. This model was also valid in the hard-body case, so constitutes a generalisation of the methodology. In a review of these methods, however, De Rossi highlighted the inability of such techniques to uniquely locate line or planar contacts without a series of small exploratory motions whilst maintaining contact [133].

Galvez *et al*. deployed the principle of intrinsic tactile sensing in a pipe crawling robot to optimise the force distribution of the feet [134]. This measured three components of normal force and two of the resultant torque to maintain stability, and was able to do this using similar knowledge to other intrinsic tactile routine and linear optimisation techniques. Although not ideal because the mechanical model of the feet was incomplete, it does represent a non-gripping application of intrinsic tactile sensing. A related technique was the work of Tsujimura and Yabuta who used a force/torque sensor attached to the base of a flexible polymer probe [135]. This touched its way around an object measuring the coordinates of contact to obtain an image of a telephone receiver.

In their comparison of intrinsic sensing and using tactile arrays for contact localisation, Son *et al*. showed that intrinsic sensing was fast and often accurate, but required high levels of calibration and is affected by transient forces that produced large errors [75]. As part of the comparison, the authors described three means of localising isolated contact on a tactile sensing array for a gripping application. These methods were partly based on the techniques and fingertip structure described by researchers like Nicolson and Fearing [136]. The general structure is shown in Figure 2.16, with sensors laid on a solid substrate and coated by a compliant elastomer.

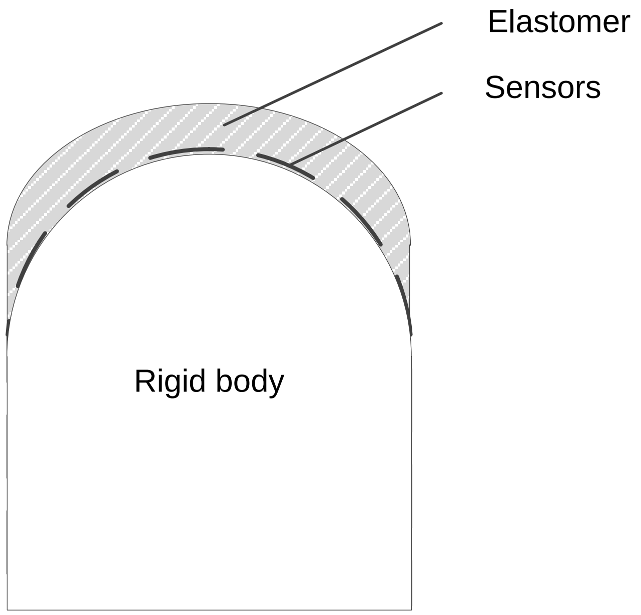


Figure 2.16: Illustration of the general fingertip sensor used as a basis for contact modelling. Adapted from [136].

Essentially, the force is transmitted mechanically through the elastomer and detected by the sensor elements. The mathematical problem tackled by many researchers in this area is how to invert the sensor data to produce surface force and contact location data. Son *et al*. applied three different methods for localising contacts [75]. The first was simply a weighted average of location with measured pressure as given in equations (2.5) and (2.6), where $(\boldsymbol{x_c}, \boldsymbol{y_c})$ is the calculated centroid, $\boldsymbol{x_{ij}}$ and $\boldsymbol{y_{ij}}$ are the locations of the cells with index $\boldsymbol{i}$ and $\boldsymbol{j}$ and $\boldsymbol{p_{ij}}$ is the pressure measured at the same place.

$$x_c = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}p_{ij} \Bigg/ \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij} \tag{2.5}$$

$$y_c = \sum_{i=1}^{m}\sum_{j=1}^{n} y_{ij}p_{ij} \Bigg/ \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij} \tag{2.6}$$

The second method repeated this, but a threshold was applied to the sensors to remove noise in the sensor. The final method is a simplification of the most pervasive method in this area of tactile data processing. The tactile element with the maximum strain is found and a quadratic curve was fitted to the three points surrounding this one to find a peak in the strain with a greater resolution than the tactile array spacing [75].

Nicolson and Fearing developed an inverse model based on the solid mechanics of the elastomer in order to achieve accuracies of 0.03mm in contact location [136]. Fearing managed to achieve a resolution of 0.01 tactels by fitting a continuous function to the two-dimensional pressure data from an 8 x 8 capacitive tactile array [137]. This used a complex interpolation process to produce a function

$$f(x,y) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} z_{ij}\frac{\sin\pi(x-i)\sin\pi(y-j)}{\pi^2(x-i)(y-j)}w(x-i)w(y-j) \tag{2.7}$$

where $\boldsymbol{z_{ij}}$ is an array formed from the detected percentage deflection in the array $\boldsymbol{w}(\blacksquare)$ is a Hanning window designed to limit the effects of the interpolation. Although this gives a good resolution, it is not good for determining the orientation. Son *et al*. further criticised these methods for their high complexity, noting that they were much slower to process that their intrinsic counterparts. Fearing and Binford were able to apply similar techniques with a 4 x 4 tactile sensor to determine the radius of curvature and orientation of the impacting object simultaneously [138]. This gives good detail for the number of tactile sensors, although both methods suffer from increasing errors if low forces are applied.

A somewhat different, but related representation of tactile information was presented by Ozaki *et al*. who defined a three-dimensional object as consisting of a stack of two-dimensional plane-closed curves [139]. Each curve was described as a pattern of unit vectors called a Unit Vector Distribution (UVD). These were used to identify known objects by comparing distributions with known patterns. The commonality with the solid-mechanics-centred techniques is the use of least-squares fitting methods to determine the curves from tactile data.

Kamiyama *et al*. formulated their inverse problem in terms of a discrete traction field of different forces at different points across the sensitive area [103]. This allows the continuous solutions to be mapped to discrete pixels within the tactile image. Nowlin comments on the fact that such methods often rely on ill-conditioned inverse matrices, meaning that noise in the sensor data will be amplified [140]. The ill-conditioned nature of the matrices is in part due to the discretisation of the forward model that is ultimately inverted. He proposed stabilising the inverse using Bayes' theorem to precondition the permissible solutions, and so improve the robustness of the algorithms [140].

This form of data processing is significant for tactile sensing on fingertips and grippers, but is less important on a wide-area skin where the necessary level of compliance may be lower. Most of these models are also based on tight cylindrical fingertip structures such that only a single contact centroid can or needs to be identified. This is inadequate for larger area coverage when an actual image of the object may be necessary. The focus of this section will now turn to using tactile data to encode object features.

Allen considered the data that could be generated by employing several anthropomorphically-inspired exploratory procedures to identify the attributes of an object [141]. These were:

1. *Grasping by containment*: by containing a shape in a grasp, multiple finger contact points can be obtained from an encompassing hand. This gives sufficient points to fit a surface and give an estimate of the object volume.

2. *Lateral extent*: this involves searching for the object boundaries by tracing a finger across a plane until a discontinuity is detected.

3. *Contour following*: a gradual pinching motion around a generalised cylinder followed by noting the Cartesian coordinates of the contact points allows the definition of cylinder-like constructs in the environment.

These three procedures generate multiple data representations: the first produces a super-quadric mesh, the second produces a set of planes and vertices and the third produces cylinder information [142].

An example of using quadric data structures to identify objects was presented by Petchartee and Monkman, who were able to train a system on four objects and then have a high degree of accuracy when those objects were subsequently re-presented to the system with different forces, positions and orientations [143]. Shapes were represented by a quadric surface by writing it as a second-order polynomial

$$P^T Q P = 0 \tag{2.8}$$

where

$$Q = \begin{pmatrix} a & d & f & g \\ d & b & e & h \\ f & e & c & j \\ g & h & j & k \end{pmatrix} \tag{2.9}$$

$$P = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{2.10}$$

and all *a...k* are constants to be determined. By exploiting properties relating to the eigenvectors of $\boldsymbol{Q}$, it was possible to produce a minimisation that correctly identified objects with a very low percentage error. Abdullah *et al*. describe a contour following process, which is based on making contact with the surface and then moving across it whilst maintaining constant normal forces [144]. Okamura and Cutkosky sought to define tactile features in terms of contours and curvatures [145] This allowed for surface features such as cracks, bumps and ridges to be detected by tracing over a surface with a tactile-equipped robotic finger.

Producing feature-based representations of tactile data has been at the core of other work as well. Browse stated that interpretation of tactile data should involve both proprioception, the measurement of pose, and the extraction of tactile features [146]. Although the results of his work suggest the utility of this approach, the necessity for planar contacts and the inability for agreement in the field as to what features are necessary for tactile data [4] calls this approach into question without further study.

Other notable work in this area includes that of Gaston and Lozano-Perez, who used local surface normals and contact location as their extracted features [147]. High resolution requirements were imposed by this work, with 256 sensor elements per 1cm^2 being cited as necessary as well as specific mechanical properties for a sensor. Despite these shortcomings, their object identification process was centred on a tree of features so that gripping would allow possible candidates to be eliminated in the object search. This was a method also explored by Luo and Wen-Hsiang [148].

All of these methods suffer somewhat from a data-centric standpoint in that they take tactile data and then create more data in varying formats that add complexity. From this perspective, the use of artificial neural networks (ANNs) is of interest, because the inputs to these networks are the values of the tactile cells without intermediate feature extraction. Petrui *et al*. used an array of FSRs to train and identify values of a pseudo-random array embossed on the surfaces of multiple elements [56]. The ANN was trained using back-propagation and added synthetically noisy data. The addition of the synthesised noise is reported to have improved the error rate from 2.1% to 0.6%. This was extended in the work of Payeur *et al*. where the FSR array was placed on a passive-compliant wrist and linear displacement transducers were used to measure the sensor orientation [149]. In addition to identifying the patterns of a pseudo-random array, the orientation and type of screw heads were identified. After 1000 epochs of training, the ANNs were able to identify all but a cross-head screw. This failure is attributed to the relatively low resolution of the tactile array [149].

However, ANN classification methods have been criticised both for the need to normalise the shape data [143] and the fact that their performance is strongly dependent on the training data supplied to them. A novel classification technique was proposed by Nakamoto *et al*. who used a multi-fingered robot hand with pressure-sensitive rubber on its fingers [150]. The measurement technique relied on continuously rolling an object between the thumb and fingers and measuring the kurtosis, or sharpness of pressure, at each point yielding a time-dependent set of values. Edges have high kurtosis and faces have a low kurtosis. Over the course of a series of rotations, the time-dependent profile can be compared to reference patterns in order to identify objects. This exhibited high rates of success for identifying a hexagon, octagon and cylinder although the authors believed greater success rates could be achieved with an improved sensor [150].

This idea of active motion in order to extract tactile measurements has also been exploited by Charlebois *et al*. who utilised rolling and gliding motions to extract curvature information [151]. The most successful approach appears to have been to sweep multiple fingers of a dextrous hand over an object, and then to fit a surface to the locus of the sensing points with a smoothing spline.

Phung *et al*. examined the capabilities of active touch with low resolution sensors to identify the size and edges of convex objects [152]. For edge following, the four cells arranged into a 2 x 2 array can measure 13 different combinations, so that the algorithm can easily be specified in advance. In a separately proposed scanning mode, the sensors are moved along until all the sensors lose their signals and then the position is recorded. This had the disadvantage of requiring many steps to achieve and had a low tolerance to sensor noise [152]. In a similarly motivated effort to reduce the number of sensors required to perform imaging tasks, van den Heever *et al*. proposed a tactile array consisting of FSRs [57]. This was for a medical application to identify growths in the necks of patients. Consequently, the sensor was moved by hand across the surface of the neck and multiple low resolution images were taken. As a result of the lack of knowledge about the alignment of the sensor, image registration techniques were deployed to align the multiple images. These were then combined to produce a higher resolution image than the sensor array alone could provide [57].

These pieces of research broadly cover the categories of tactile data. The predominant theme is the use of data from grippers to achieve compliance and object recognition, with little or no focus on wide-area skin data handling for which many of these techniques are inappropriate. In order to complete the current picture of robotic sensor representations, it is necessary to consider the means by which environmental data derived from sensors is integrated into internal maps. That is the subject of the next section.

2.3.2. Environmental Mapping for Robotics

One of the basic necessities for autonomous and semi-autonomous robotic systems is the ability to explore unstructured environments. This can be broken into a series of sub-problems, such as localisation, planning motion, collision avoidance and safety [153, 154]. All of these tasks are strongly dependent on the accuracy of the internal map of the robot's environment [155].

Given the strong relationships described in Section 2.2, it is worth exploring the internal representations of the environment employed by humans and other animals. For human beings, this follows on from the discussion of cognitive science in Section 2.1. Researchers in this field first employed a concept called a cognitive map to describe the internal spatial representation of rats following complex paths within a maze [156]. The rats were trained to find food within the maze over a series of trials, with both the foodstuff and the structure of the maze remaining constant. In subsequent tests, the rats were introduced to the same general setup, but the trained path was blocked meaning that the learnt route was no longer accessible. The noteworthy result was that the majority of rats chose new paths that were oriented with the true direction of the food, suggesting that the rats held a more comprehensive representation of the environment than a simple list of turning points for navigating the maze. This representation allowed them to identify the direction of the food from any other point within the map.

Human behaviour in navigating the environment is also considered to be based on cognitive maps that define our knowledge of spatial features relative to the current location [157]. Experiments with blindfolded individuals trained to walk from a fixed origin to three locations demonstrated that people quickly learnt the spatial relationships between all four locations [158]. These studies did, however, indicate that judgements of the scale of these relationships were impaired in adults who suffered from early blindness who consequently had little or no visual experience.

The cognitive processes of the visually impaired are of interest when trying to draw comparisons with a robot using tactile skin as its sole means of exploration information. There are three schools of thought when considering the cognitive mapping capabilities:

- *Deficiency theory:* the lack of visual experience renders humans incapable of internalising spatial relationships [159].

- *Inefficiency theory:* the capacity for internalisation exists, but is impaired by the limitations of auditory and tactile data [159].
- *Difference theory*: the capacity is the same as that for sighted people, but differences in performance are dictated by external variables [160].

In the case of a person who is visually impaired, the tactile and auditory senses become more relevant suppliers of information and these have been shown to be sufficient for maps of the environment to be generated by those who are congenitally blind [159]. Other studies have indicated that blind adults and children are still able to accurately estimate the distances between objects within familiar environments [161]. This suggests that a valid approach to cognitive mapping in the visually impaired can be represented by either inefficiency theory of difference theory.

Although research in cognitive mapping has centred on information derived from vision, this research shows that the human brain can construct an internal representation of the environment without the use of vision and use this for navigation. The need for this ability is clear in the case of dark or occluded environments or in the event of temporary or permanent loss of visual capability. These problems also exist in the robotics domain and expose a need to be able to develop internal representations from similar data sources.

All of this research indicates the existence of a cognitive map in humans and animals, but it gives no indication of how the map is constructed. There are two broad approaches: the metric approach where the relative distances, angles and orientations of objects are used to represent an environment; and the other, non-metric approach uses a structure based on relative relationships between landmarks in the environment with no specific values such as distance used to encode these [162]. Both of these approaches can be generalised as a set of images that can be manipulated to organise the spatial qualities of the environment [163].

In examining the metric approach, the concept of cognitive distance is of use. This is the distance between objects that people can estimate from their internal representations. It has been found that the cognitive distance invariably differs from the physical distance by both under- and over-representation of the latter [164]. In other experiments that used virtual reality to distort a perceived environment, it was shown that these inaccuracies failed to induce errors by the participants [163, 165]. Both of these suggest that the human cognitive map is composed of a connected graph of landmarks.

Further evidence of this was offered by experiments in which participants were required to draw maps of their urban environment in an effort to examine their cognitive maps directly. One of the outcomes was that participants drew road junction intersections at right angles, despite the wide range of intersection angles that exist in reality [166]. This suggested that little or no vector information conforming to distance and angle exists within the cognitive map, which instead has a looser network structure consisting of landmarks connected together.

Humans are clearly capable of navigating their environment and yet appear to perform this task with limited detail within their internal representation. However, this representation may be necessary given the structure of the brain. Certain structures within the brain are believed to be responsible for the mapping and localisation functions which have been identified, such as place cells that fire when in the vicinity of particular objects or locations [167, 168]. Other cells, called grid cells have been identified and it is posited that the brain uses these to maintain a map of the environment on some form of regular grid [169].

An underlying theme of these representations is that apparently complex information from the human sensors is manipulated into more easily processable forms without regard for the loss of detailed information in the process. Given the finite resources of any computational system, either biological or artificial, the manipulation of data into the most easily processed form is a highly relevant paradigm to the field of robotics.

These cognitive map structures have counterparts in the map representations in robotics, where the different types can be divided into three areas: topological, metric and hybrid. A topological map may be thought of as an artificial version of the cognitive map in structure, in that it encodes the environment in terms of the location of obstacles and the relationships between them [170]. This is typically done by representing obstacles as nodes on a graph with arcs linking nodes where a free path exists between them, but specific coordinates for the nodes are not necessarily required [171]. The identification of appropriate features is a source of frequent discussion in this field of research, since it is not a simple problem to usefully partition an arbitrary environment [172].

Engelson and McDermott used a topological approach in order to correct errors that enter a map as a result of defective data or analysis [173]. This is an extension of the work by Kuipers and Byun, who made one of the earliest introductions of the topological concept in robotics, but without the post-acquisition error-correction capabilities [174]. In these representations, the arcs between nodes represented the actions that a robot would have to make in order to traverse from one node to the next. Engelson and McDermott's extension was to incorporate geometric information into the arc in order to describe its shape. This was to allow the robot to make a distinction between perceptually similar locations in addition to the inherent navigational benefits.

For mobile robots, localisation is an important issue because of error accumulation from the use of odometry. In order to localise the robot, Engelson and McDermott's system maintained a set of tracks that identified the possible locations of the robot at a given time. Matching current observations to expectations in each possible location allowed either localisation or admission of an unknown locale [173]. Most studies focus on the use of vision or proximity sensors in order to identify landmarks, limiting the scope for applying this work to tactile sensing. Some work by Zimmer involved constructing a qualitative topological model that combined data from both passive light and tactile systems [175]. This fusion was achieved by weighting the contributions from both sensor systems to produce a vector that represented an arc in the topological graph. There is no information, however, regarding the type, composition or responsiveness of the tactile sensors used apart from a comment that their limited angular resolution required a filtering of the data obtained.

Yamauchi and Beer described what they call an adaptive place network, which has a similar structure and presented a mathematical means of updating the map [176]. The network is defined as consisting of two sets of information: a set of place units that are three-tuples $\langle \boldsymbol{i}, \boldsymbol{x}, \boldsymbol{y} \rangle$ with an index *i* and the *x*- and *y*-coordinates of the unit, and a set of place links that are four-tuples $\langle \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{\theta}, \boldsymbol{c} \rangle$ where *i* and *j* are the indices of the linked place units, $\boldsymbol{\theta}$ is the approximate heading from one to the other and *c* is a level of confidence in the information. As in other implementations, the graph is extended by the robot moving to a new place unit and generating new members of the two sets that comprise the network. On subsequent transits, the map is updated using the relationship

$$\theta_{new} = \arctan\left(\frac{\lambda_{link} \sin\phi + (1 - \lambda_{link}) \sin\theta_{old}}{\lambda_{link} \cos\phi + (1 - \lambda_{link}) \cos\theta_{old}}\right) \qquad (2.9)$$

where $\boldsymbol{\theta_{new}}$ and $\boldsymbol{\theta_{old}}$ represent the updated and previous headings within a single place link, $\boldsymbol{\lambda_{link}}$ is the link learning rate to control the rate of updating and $\boldsymbol{\phi}$ is the new heading measurement. Unlike other work in this field, however, Yamauchi and Beer do not use this topological construction for localisation, preferring instead to use the separate paradigm of occupancy grids.

The occupancy grid is a metric map that divides the environment into tessellated cells, each of which has an associated value representing the belief in the occupancy of the cell, usually in the form of a probability [177]. This is the dominant representation used in robotics [178, 179], although this dominance may be attributable to the relative recency of topological representations. As with topological maps, most of the work done with occupancy grids relies on vision and sonar data. The predominant algorithm for using such structures is to take data from multiple sensor measurements, use inverse sensor models to produce probabilistic maps of individual grid cells from each measurement and then combine them to yield a single map [180]. The differences between various implementations lie predominantly in the methodology for updating the map based on sensor data.

In an early implementation by Moravec and Elfes, a sonar detector was used with readings transferred to an inverse model to produce probabilistic maps of the empty space and the occupied region [181]. This produces two grids, representing probabilities that the cells are occupied or empty using equations (2.10) and (2.11).

$$P_E(x,y,z) = Prob\big((x,y,z) \text{ is empty}\big) = E_r(\delta)E_a(\theta) \tag{2.10}$$

$$P_O(x,y,z) = Prob\big((x,y,z) \text{ is occupied}\big) = O_r(\delta)O_a(\theta) \tag{2.11}$$

Here, $\boldsymbol{\delta}$ is the distance of the cell from the sensor and $\boldsymbol{\theta}$ is the angle between the centre of the sonar beam and the cell. $\boldsymbol{E_r(\delta)}$ is the probability that the cell is empty based on its distance from the sensors, whilst $\boldsymbol{E_a(\theta)}$ is based on the angle of the cell from the beam centre. $\boldsymbol{O_r(\delta)}$ and $\boldsymbol{O_a(\theta)}$ have identical definitions, except that they relate to the probability of a cell being occupied instead of empty. All of these functions are determined by the physical models of the sensors, which are omitted here for succinctness.

In this representation, the final integrated map had a range of values for each cell from -1 to 1 where -1 represents definitely empty and 1 represents definitely occupied. Two intermediate maps $\boldsymbol{E}$ and $\boldsymbol{O}$ are involved in this process that hold the data for all cells based on the probabilities determined by equations (2.10) and (2.11). The failing of this particular scheme is the update of these two intermediate maps, which was performed incrementally using the equation

$$E_n(x,y,z) = E_o(x,y,z) + E_k(x,y,z) - E_o(x,y,z)E_k(x,y,z) \tag{2.12}$$

where new map *n* is updated using old map *o* and some sensor reading *k*. The problem is that erroneous readings can cause cell $\boldsymbol{(x,y,z)}$ in the intermediate maps to tend towards one. It thus becomes difficult, and at full convergence impossible, to adjust the value of the cell based on subsequent evidence. Subsequent work by Elfes and Matties sought to achieve similar results using a single map directly with no intermediate maps [182]. The same sensor model was used, but the update was performed by using the Bayesian update formula

$$P(s_i|e) = \frac{P(e|s_i)P(s_i)}{\sum_j P(e|s_i)P(s_i)} \tag{2.13}$$

where $\boldsymbol{s_i}$ is one of the states being estimated, e is the relevant evidence relating to the state, $\boldsymbol{P(s_i)}$ is the probability of the system being in that state already, and $\boldsymbol{P(e|s_i)}$ is the probability that the evidence would be found if the system were in that state. Using the definition of the map, *M*, that

$$M(x,y,z) = O(x,y,z) = 1 - E(x,y,z) \tag{2.14}$$

it is possible to rewrite (2.13) to update the map using the relationship shown in (2.15).

$$M_n(x,y,z) = \frac{P_E(x,y,z)M_o(x,y,z)}{P_E(x,y,z)M_o(x,y,z) + \big(1 - P_E(x,y,z)\big)\big(1 - M_o(x,y,z)\big)} \tag{2.15}$$

This simplifies the procedure of updating the map and introduces a more probabilistic approach to the map-building process, but this method still has the problem that as $\boldsymbol{M_o(x,y,z)}$ approaches an extremal value of either 0 or 1, it becomes apparent that the convergence problem of [181] remains.

Other update algorithms have been proposed, which avoid this convergence problem, but these were defined by very specific environments and sensors [183]. Other approaches, such as that proposed by Thrun, utilised neural networks as a means of updating the grid cells [184]. Thrun's approach proposed two different neural networks; a sensor interpretation network and a confidence estimation network. The former network was designed to allow a mapping between sensor data and expected reward within a set of occupancy grid cells. This exposes a weakness in this update system, since neural networks are adaptive structures that require training in order to converge on a series of internal parameters that allow the network to correctly interpret subsequent inputs. In the context of environmental interpretation, however, this training cannot easily be performed in advance. Although Thrun suggested that the adaptive nature allowed rapid adaptation to a variety of unexpected circumstances [185], his interpretation suggests that the robot would need to know in advance the environment that it was being placed in and so know that it needed to retrain the network. This suggests that such a network ultimately encodes specific environmental characteristics, reducing generality and so leading to reductions in performance [155].

Thrun later introduced a more radical approach by not using an inverse probabilistic model to map sensor readings to environmental causes, but instead using a forward model that mapped occupancy to sensor readings [180]. He highlighted that the inverse models used would typically update grids on a cell-by-cell basis with little or no influence from neighbouring cells, which is congruous with the findings of this literature review. This update procedure relies on the assumption of the independence of the probability of individual grid cells being occupied based on single readings of the data. Since sensors tend to make measurements that can be influenced by multiple cells, the probabilities on the individual cell are not independent of those of the surrounding cells, so Thrun asserts that the assumption is flawed. In this research, Thrun described the means of implementing a forward model paradigm for sonar sensors, the specifics of which are of little relevance here and so are omitted. The results of this development do, however, indicate that forward models provide more accurate occupancy grids than inverse models [155, 180]. There are problems with this methodology due to its reliance on algorithms that force the sensor data to be assessed multiple times, rendering the approach inappropriate for real-time applications [180]. This approach is also sensitive to transitory changes, such as human movement, that render it less effective at producing entirely useful static maps.

Elmenreich criticises all of these classes of method for failing to be sufficiently tolerant of sensor faults, specifically criticising their intolerance to intermittent and permanent failures in the sensor [186]. The criticisms are justified in the sense that there has been an intense focus on eradicating errors induced by the nature of sensors, but not on the real-world aspects of sensor failure. Elmenreich's approach to correcting this deficiency was to append additional data that allows either a voting mechanism between multiple sensors measuring the same cell or a heuristic determination of malfunctioning sensors over the course of multiple, non-simultaneous measurements [186]. Although the results of this work indicate good elimination of error compared to other methods, it is possible to criticise the addition of more data to cells due to the increased difficulty in processing the structure within the software and the resultant increased memory requirements.

Of the two approaches, the occupancy grid approach is more applicable than the topological approach. This is because of both the common grid-like structure of the arrays and because the tactile data is heavily localised to a specific area, making a topological approach less valid that in the case of vision data where a large range of objects may be identified over a wide field of view.

In the literature, tactile data is often mentioned as a possible source of data for fusion onto an occupancy grid, but there are almost no examples of actually using it. One exception to this is the work of Faria *et al*. who extracted tactile data from grippers and used it to produce an occupancy grid that gives an approximation of the shape of the object [187]. The sensor gave a Gaussian noise distribution of 0.2 in locating the occupied cells and a Bayesian approach was employed to develop the belief values in the occupancy grid. The benefit of this is that it facilitates integration of tactile data with data derived from vision and similar systems in robotics, which was proposed for future work.

Occupancy grids require significant storage space because beliefs must be stored as integer or floating-point representations with a single datum for each cell. This is exacerbated by high resolution maps of the environment that are often desired. Other problems such as efficient planning and localisation are cited as concerns, particularly the latter as it inhibits effective updates of the occupancy grid. Such criticism would make topological maps seem the better choice in robotics overall, since planning is efficient and their qualitative nature precludes the necessity for accurate determination of the position of the robot. This class of map is, however, deficient in other respects where grid-based methods are superior such as updating the grid when sensor data is inconclusive, partitioning the environment and identifying a location when it is already recorded on the map.

As a result of the deficiencies in both these techniques, a hybrid architecture has been proposed to combine the occupancy grid and topological paradigms [185]. In this approach, the map is learnt using an occupancy grid and this is then partitioned into areas of interest. These partitions then form the nodes in a topological graph, allowing the easier construction of maps using grid-based methods to create more efficient topological representations. Thrun acknowledges, however, that this does not eliminate the problems of high memory usage or the necessity for accurate localisation since the topography of the map will be strongly influenced by the grid-based map. Regardless, it can be envisaged that tactile data would be represented as a grid within such a hybrid structure. However, the quantity and scale of tactile information requires consideration of the means by which touch data is converted onto a grid-based representation.

It is clear that no mapping approach has been effectively applied to tactile sensing as an exploratory mechanism. The three paradigms also have problems, but the most significant from a data-centric approach is the fact that all schemes are necessarily memory-intensive. Although this criticism has been levelled primarily at occupancy grids, the implementation of the topological graphs described in software would entail the use of memory pointers, arrays of link objects or arrays of integers to represent the links between the nodes. None of these is inexpensive in terms of memory usage, particularly in a more cluttered environment where there may be many identifiable connections between nodes. If tactile data is to find significant use in robotic mapping, consideration will have to be given to reducing the memory consumed by its data structures, in order not to be forced to compete with mapping systems such as those described here, since these are of greater importance to robots in unstructured environments.

2.3.3. Conclusions

This section has examined the current methods in tactile data manipulation and environmental mapping. It is clear that research in tactile data representation has focused extensively on contact location and object recognition using grippers, with no work concentrating on general representations for wide-area skins. The derivation of these parameters is generally dependent on the existence of manipulators, which will not be the case for a more general system.

In the realm of environmental mapping, very little exists in the domain of tactile sensing with the only paper to do so concentrating on manipulators. The final section of this chapter will consider this apparent gap in current research and consider how it might be addressed.

2.4. Discussion

This chapter has examined the state-of-the-art in tactile systems and the general finding is that there has been a significant concentration on devices and techniques relating to gripping applications. Wide-area skins for covering other areas of the body have generally been neglected as a result, discussed only in principle and with little serious effort devoted to integrating them in systems.

As part of this review, a specification for tactile sensing was developed that attempted to abstract the tactile sensor away from the robot upon which it is deployed. This was done by specifying a system in terms of data considerations. From this, the easily translated methods involving resistance- and capacitance were deemed to be best, particularly when implemented as part of a matrix to minimise wiring complexity. Miniaturisation techniques are also favourable, but there are significant disadvantages with current techniques relating to durability, fragility and speed. In devising the data-centric specification, the issue of sensor resolution was dispensed with as part of the effort at abstraction. However, the higher the resolution of the sensors, the more data needs to be transmitted within the system so it cannot be ignored entirely.

The work of van den Heever *et al*. [57] and Phung *et al*. [152] discussed in Section 2.3.1 suggested that, despite low resolutions, it was still possible to perform as well as higher resolution techniques. This is particularly noticeable in the super-resolution techniques described by van den Heever *et al*. [57]. High resolution sensors do not afford a robot any degree of control over the acquisition of data. This is because a high-resolution sensor must produce a large amount of data regardless of whether that data is required. For all the possible techniques to prevent transmission, there is no avoiding the need to scan each cell at some point in the system. A lower resolution sensor offers a lowest-common-denominator for the robotic control system in that the amount of data is significantly reduced. The obvious downside is that if the robot needs higher resolution information at some point, it is not able to acquire this from its tactile sensors.

Yet the work of van den Heever *et al*. suggests a compromise in this domain, even though their work was not applied to robotics. If the robot makes a low-resolution reading of an impact, there is the possibility that it could decide to explore that object further by taking multiple tactile images and fusing them together. The clearest way to improve further the data transmission problems associated with large numbers of tactile sensors is to reduce the size of the data representation. This review has shown that for gripping applications, a range of features can be extracted but these are designed for the needs of gripping applications and may not be appropriate in other domains. This is particularly the case given that these feature extractions would appear to require more memory than even a simple record of just the raw measurement would occupy.

Instead, it is proposed that the paradigm of occupancy grids be adapted to contain the data from the inherently grid-like tactile sensing arrays. This needs to be balanced against the needs of the quantity of tactile data that might need representation, limiting the appropriateness of using an integer or floating-point value for each cell in the representation. In order to reduce this, a reduction in the resolution of the data will also be required. This specification is distinct from the resolution of the sensor, allowing the data structure to be independent of the sensor resolution that should be at the choice of the robot's designer. However, it will be necessary to verify that in the case of both low resolution sensors and low resolution data, the tactile system can be made to exceed its apparent capabilities, such as by producing higher-resolution images. The data structure will additionally have to be robust in the presence of various forms of noise.

The remainder of this work will therefore focus on demonstrating a low-resolution sensor interacting with a novel data structure designed for compactness and efficiency. In order to do this, a tactile sensor must be developed that satisfies the data-centric specification. For this, the choice is QTC because of its simple measurement method and its low unloaded power consumption. Since this has not yet been implemented in a matrix form, such an implementation will also be presented in addition to some novel techniques allowing the design to be scaled up or down to different resolutions in an effort to provide a design in the spirit of modularity.

The next two chapters will deal with the development and characterisation of these sensors, followed by the development of a tactile data structure and ultimately a combination of these novel developments in order to demonstrate the ability to obtain high resolution images from low resolution sensors and low resolution data at the discretion of a robotic system.

Chapter 3
Initial Work with QTC

At the end of the previous chapter, it was concluded that a material called QTC should be developed into an array form, because it offers benefits in terms of power consumption alongside a simple transduction principle. This chapter seeks to begin that development, first by examining the properties of a commercially available composite and identifying the beneficial and detrimental properties of its electrical characteristics. The means of developing this material into an array form will then be explored as a prelude to the development of a final version of the sensor in Chapter 4.

3.1. Commercially Available QTC

The only QTC available as an off-the-shelf commercially-available component comes in the form of small pills, measuring approximately 3.5x3.5x1.0mm. Other forms such as a thin sheet have previously been available, but now appear to be discontinued. These commercial components have been used in robotics applications, as examined in the previous chapter, and so it is worth briefly examining their behaviour to provide a basis for comparison with the sensors developed as part of this work.

In order to do this, a means of integrating the pills into a circuit where they could be deformed was developed. Such integration also required that the pill could be compressed by the application of force, in order to test the response of the pill to load. The apparatus constructed to fulfil this task is shown in Figure 3.1.

The main structure is fabricated from nylon and the entire assembly is electrically isolated from the environment using a ceramic of aluminium oxide. Polished disks of copper formed electrodes between which the pill was placed. These were soldered to wires that allowed the electrodes to be connected within a measurement circuit. The cavity had sufficient width to allow the QTC pill to spread, whilst allowing contact to be maintained. This means that the electrode contact area will change as force is applied, which may produce slightly different results from the work of Bloor *et al*. who applied metallic paint to unloaded devices and then compressed them [5]. The mechanism here, however, represents more closely the effect if the material is deployed in an array-like device, where electrodes are laid down as strips across active material. In such circumstances, the active material would be expected to spread out, but not necessarily beyond the confines of the electrode itself. Consequently, the effect of spreading should not be excluded in the assessment of the material characteristics, which justifies the approach adopted here.

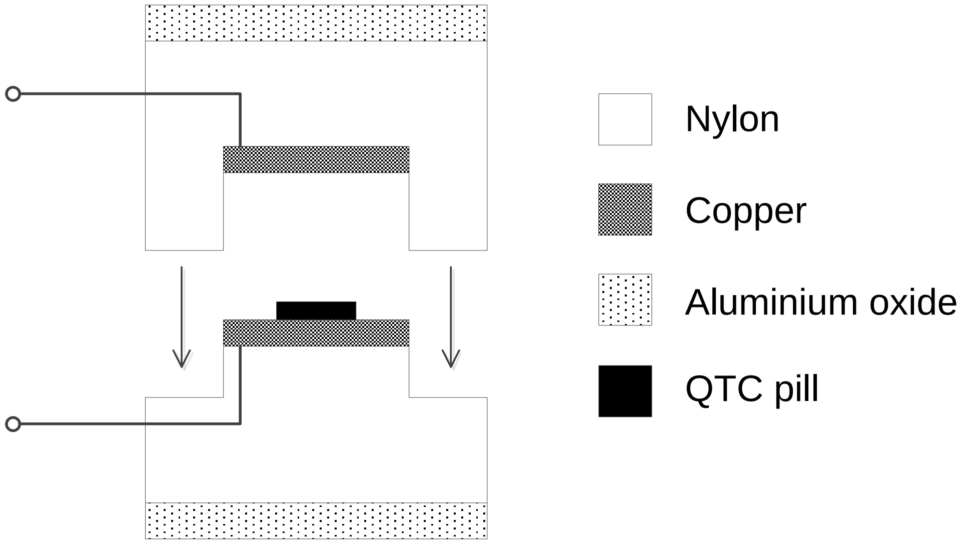


Figure 3.1: Schematic diagram of the structure supporting the pill-form QTC for electrical characterisation under load.

The measurement circuit consisted of an Agilent N6700 high-voltage power source, which allowed a current of up to 3A to be drawn by the pill. To give a precise measurement of current, an Agilent 34410A digital multimeter was integrated within the circuit. Both of these devices were controlled via a GPIB interface using a LabView™ program. All experiments were conducted at room temperature. Compressive force was applied to the pill within the electrode-containing apparatus using an INSTRON 3369 compression column that was capable of applying a compression and measuring that compression precisely. A force cell in the column was used to monitor the applied force, which was not observed to drift significantly over the course of the experiments.

Under different compressions, the resistance of several pills was measured using the resistance-measurement mode of the Agilent 34410A. This operates by sourcing a fixed current and measuring the voltage across the electrodes, converting this measurement into a resistance. The results for three typical pills under different compressions are shown in Figure 3.2.

Using the log-linear scale, it is clear that there is an exponential dependence of resistance on compression, which is congruous with the findings of Bloor *et al*. [5, 6] A limited range of compressions was used to produce the data of Figure 3.2, and this limitation was imposed by the current limits of the source and ammeter units used in the measurement circuitry for these experiments. Consequently, the limiting behaviour produced by excessive compression that is observed by Bloor *et al*. is not observed in all of these results, but the general trend confirms that the material being used here is similar to that employed in this earlier work.

In order to examine the current-voltage characteristics of these devices, voltages were applied in half-Volt increments from 0V to 40V and then reduced again to 0V, again in half-volt increments, to provide an assessment of electrical hysteresis. These measurements were performed under the same compressions used for generating the resistance curve of Figure 3.2. The current-voltage characteristics of a typical pill are shown in Figure 3.3 for the lowest compression used in these experiments.

This compression is not zero, because the assembly shown in Figure 3.1 compresses the device slightly, even without further compression from the INSTRON 3369. This is quantified by measuring the depth of the QTC pill using a micrometer, the height of the assembly of Figure 3.1 and then the height of the assembly with the pill placed between the electrodes. The differences between these values show how much the pill has been compressed by the weight of the assembly alone.

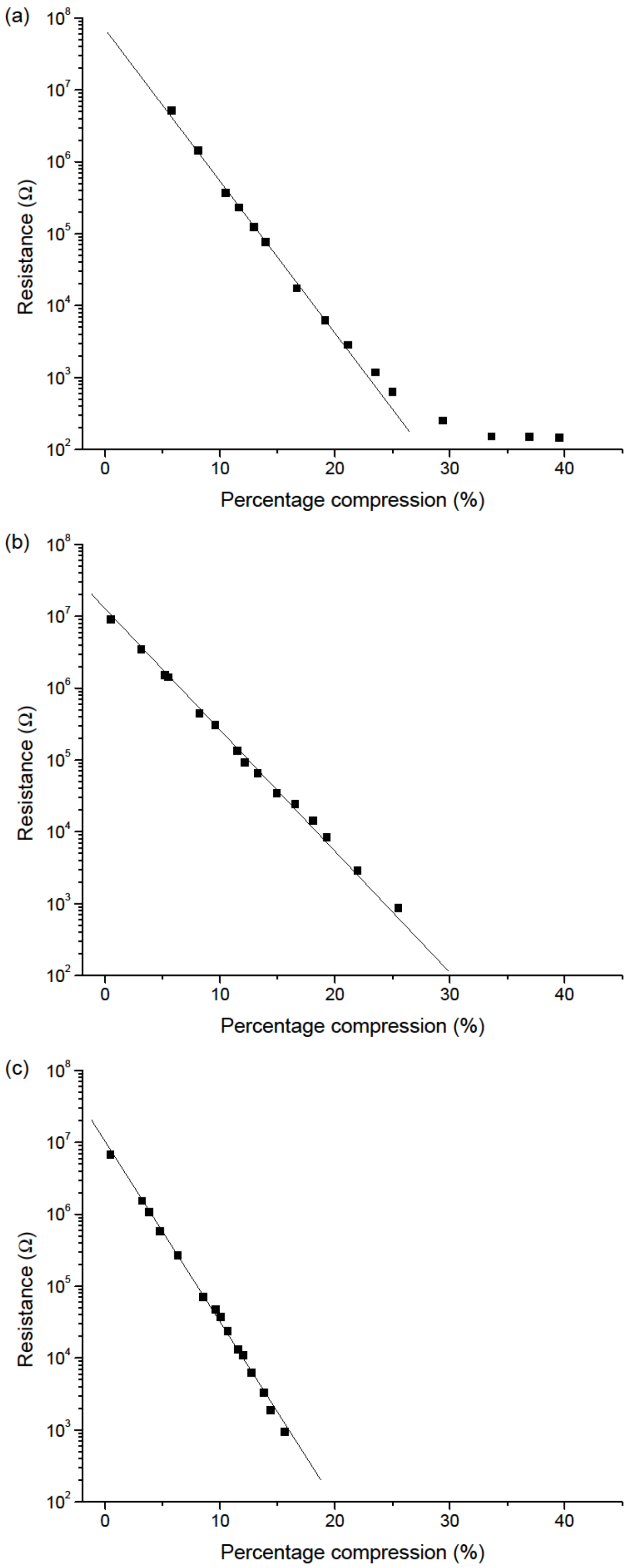


Figure 3.2: Graph showing the change of the resistance of three typical QTC pills under various compressions.

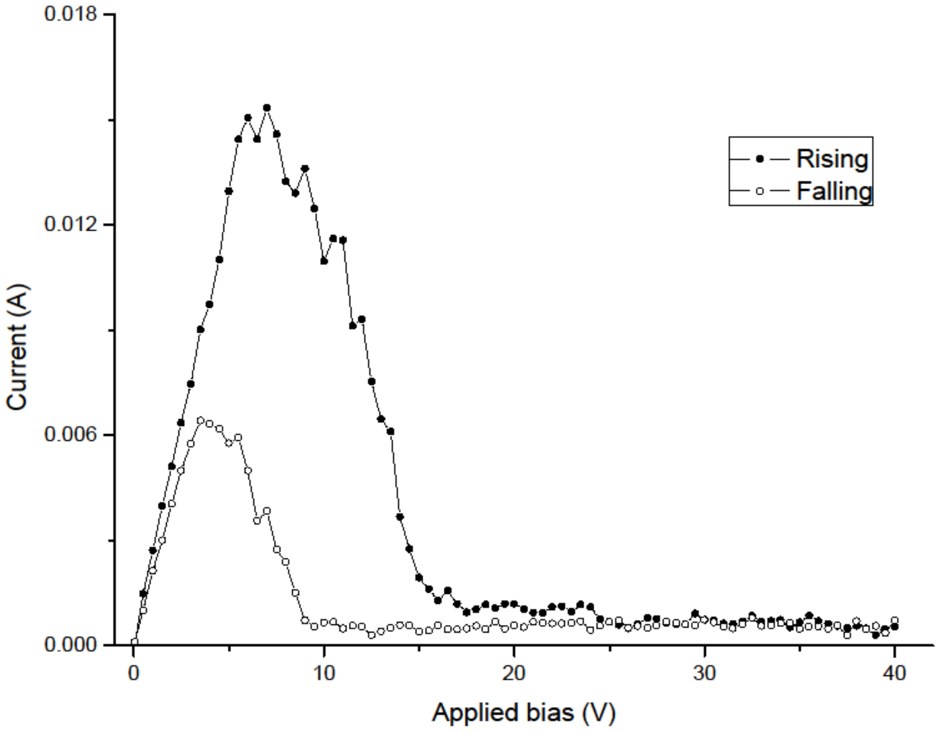


Figure 3.3: Graph of the current-voltage characteristics of a typical QTC pill under a compression of 0.17%.

These results have features similar to those of Bloor *et al*. [5]. In particular, the rising curve to a peak and the subsequent region of negative differential resistance are prominent features associated with QTC. As the voltage is subsequently reduced, the current is consistently lower than that of the increasing voltage. Subsequent measurements on the same pills show that this is not dependent on the orientation of the pill within the electrode assembly.

As the pill is compressed further, the basic shape of the characteristic remains broadly the same, with the most significant difference being the peak value which increases with compression. The characteristics for three different compressions are shown in Figure 3.4 to demonstrate this, using the data obtained from the same typical pill used to generate the data in Figure 3.3. A log-log scale is used in this figure so as to make the features of the curves easily discernible.

At higher voltages and increased compressions, the data also show an increasing positive gradient with increasing voltages. The explanation for these characteristics is presently of little importance, although these will be analysed more closely in the next chapter. It is instead important to consider the implications of these characteristics in the context of robotics.

The variation in resistance with applied force is of obvious benefit in terms of sensitivity and the high unloaded value reduces power consumption, which is the reason that QTC has been considered for further development. It is desired that producing an array with QTC as the active material does not reduce this sensitivity or other beneficial behaviour. The current-voltage characteristics will, in general, have lower relevance in operation. However, the behaviour is significant, for example, in choosing an operating voltage for the sensor. The robustness of the sensor to voltage fluctuations is important in a potentially noisy system. Since there are clearly different operating regimes discernible in the current-voltage characteristics, it may be desirable to avoid selecting operating voltages near the boundaries between them

Since there is a dearth of available off-the-shelf solutions capable of being easily adapted into the desirable low-connection count array form desired in tactile sensing, different techniques are required. These will be developed in the next section.

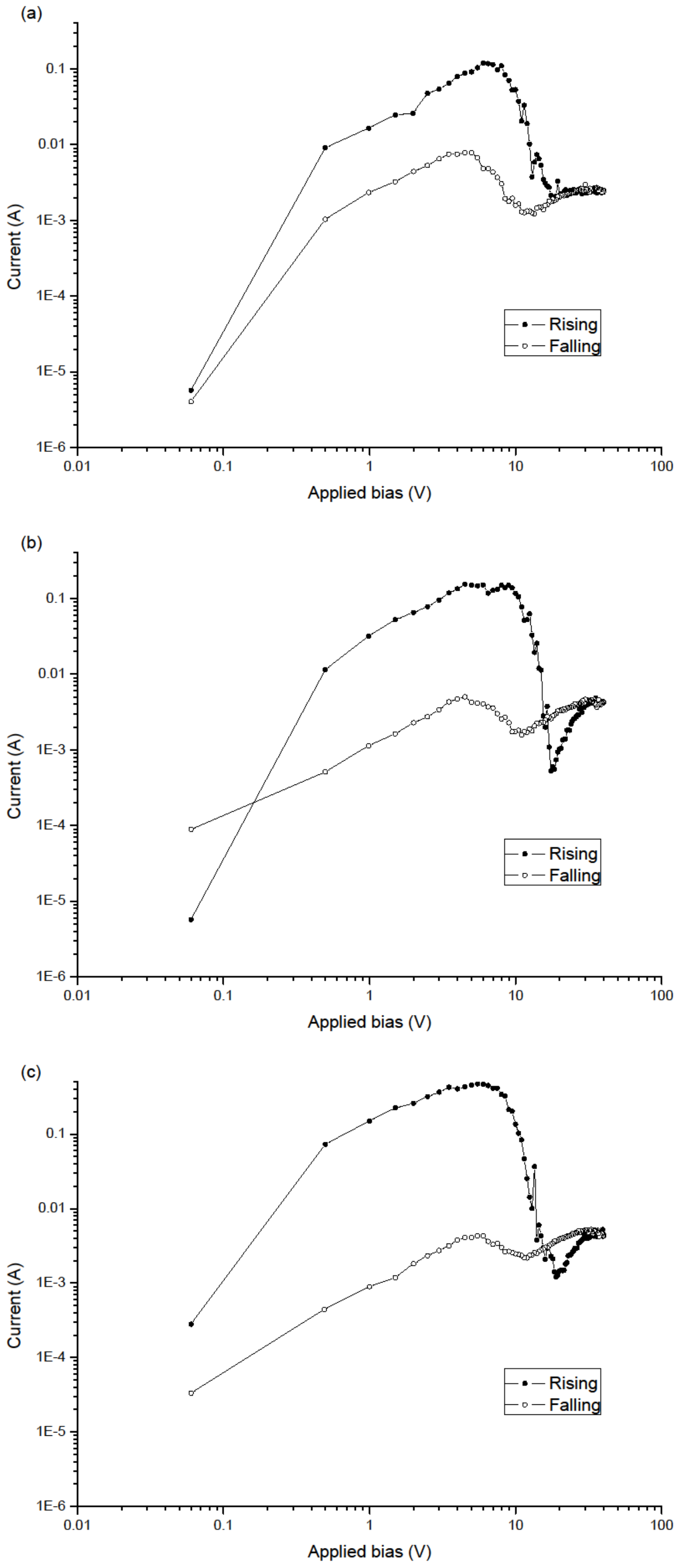


Figure 3.4: Graphs showing the current-voltage characteristics of a typical QTC pill at compressions of (a) 8.2% (b) 14.9% and (c) 25.5% on a log-log scale.

3.2. Array-based QTC Development

A key component in producing a matrix form of QTC is in the choice of electrode. To be suitable, such an electrode requires flexibility and durability, since tactile sensing entails allowing the contact forces to be transmitted as efficiently as possible to the active material and to ensure that the sensor is undamaged after repeated contacts.

One approach to this is to utilise cloth woven from conductive fabric, which has been exploited for binary switching on a larger scale in tactile sensing [188]. Rather than utilising such fabrics simply for binary sensing, in which they act alone as the force-detecting sensors, this work will consider using them solely as electrodes for the conductive material. The fabric that will be used in this design is a woven fabric called bobbinet, manufactured by Swisstulle UK Ltd, which has the structure shown schematically in Figure 3.5.

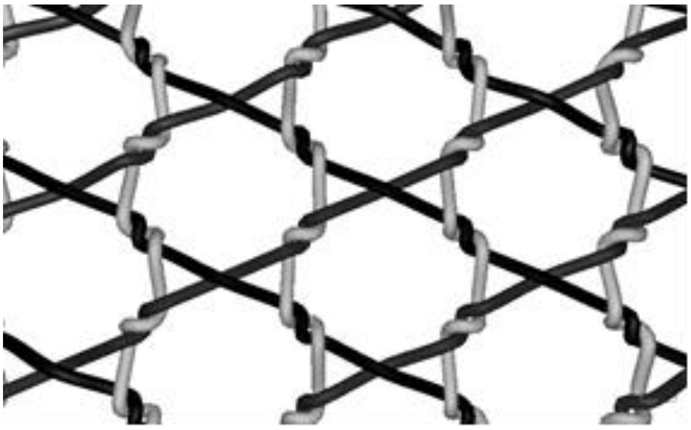


Figure 3.5: Schematic of the structure of bobbinet produced by Swisstulle UK Ltd used in this experiment.

In this fabric, the diagonal yarns are woven around the vertical ones to produce a regular hexagonal structure, which endows the material with both strength and durability [189], which have been established as necessary qualities of a tactile sensor. Typically, bobbinet is manufactured using non-conducting nylon yarns, but it is possible to replace one or more of these yarns with a silver-coated thread. These can have a resistance as low as 3.7Ω for each metre [190].

Once this material was chosen, a sensor was designed consisting of a continuous sheet of QTC sandwiched between two orthogonally-aligned pieces of bobbinet with alternating conducting and non-conducting stripes woven directly into the fabric. As a result of the proprietary manufacturing process of the QTC, the manufacturing work for producing all sensors in this work was performed by Peratech Ltd. based on a specified design. This initial design is illustrated schematically in Figure 3.6, producing an array of 16 tactile sensing cells, hereafter referred to as tactels.

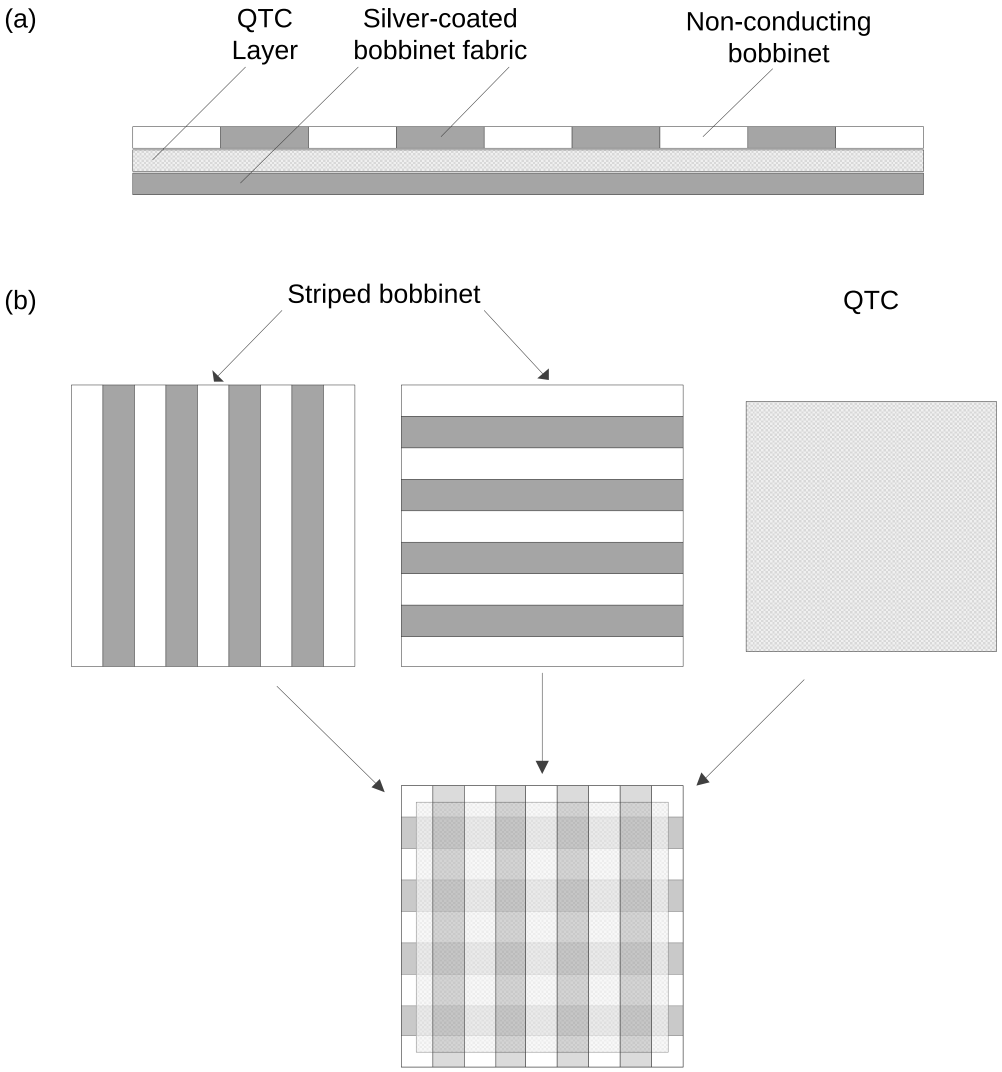


Figure 3.6: Schematic illustration of the initial design of a QTC-based tactile sensing array using a simple construction shown from (a) the side and (b) above.

In order to characterise the response of the sensor to force, a force applicator was constructed to allow various loads to be applied to specific tactels. It was necessary to do this in preference to using the INSTRON 3369 used in the previous experiment with the pill form of QTC because of spatial constraints in the loading area of the compressive column. The developed apparatus is shown in Figure 3.7 and consists of a plunger capable of freely sliding within a fixed tube held in place by a stand.

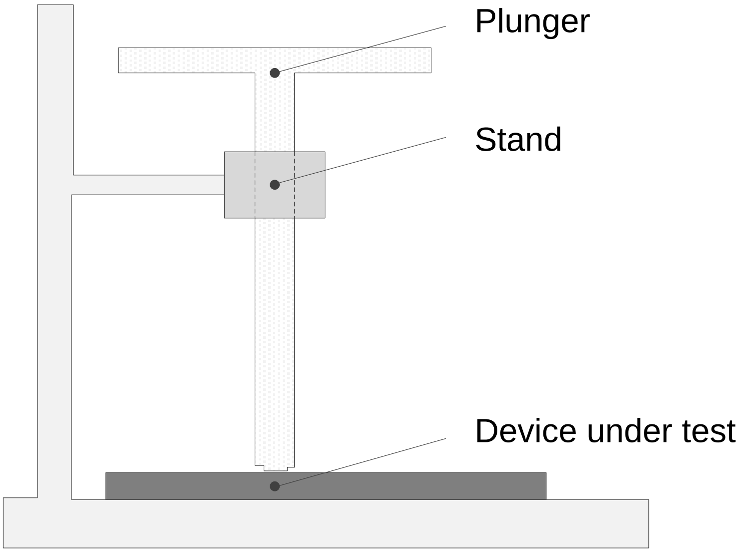


Figure 3.7: Illustration of the apparatus used to apply known loads to the sensor. Diagram is not to scale.

The end of the plunger was machined to have a fixed area of 1cm^2 so that the applied pressure could be varied simply by changing a load placed on the top. This method of applying pressure allows the quasi-static response of the tactels to be measured.

The electrical response of the tactels was measured through the change in measured current across the tactel in response to applied voltage. Voltages were applied in half-Volt increments from 0V to 40V using a Keithley 2400 SourceMeter™. Applied loads in a modest range of 0N to 65N were applied using the apparatus illustrated in Figure 3.7, which should suitably characterise the device for the majority of applications. Characterisations were made for five of the tactels and the basic form of these was found to be the same for all voltages. The force-resistance characteristics of a typical tactel are shown in Figure 3.8 for an applied voltage of 24V.

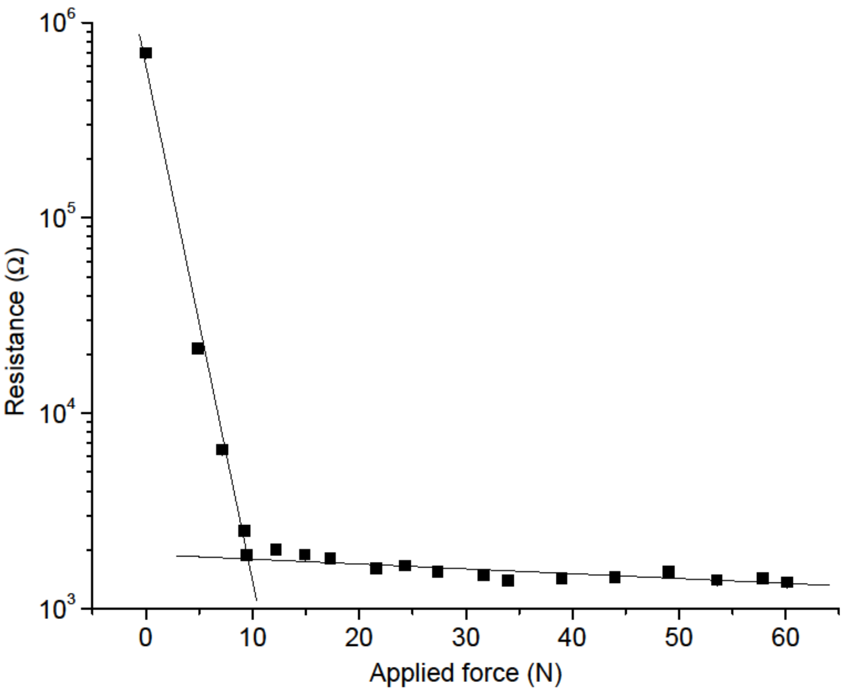


Figure 3.8: Force-resistance characteristic of a typical tactel in the initial QTC matrix prototype at an applied bias of 24V.

The results of Figure 3.8 are derived from the average of multiple measurements taken several hours apart to prevent transient effects influencing the acquired curve. One noteworthy result is that the unloaded resistance is considerably lower than that of the pill-form examined in Section 3.1. This is by design, since a wide range of force detection was desired with a relatively low initial detection threshold. This specification necessitated the use of a stiffer elastomer in the composite matrix. The precise details of this are proprietary to Peratech Ltd., but explain the apparent discrepancy in unloaded resistance.

Of more significant concern, however, is the effect that loading a tactel had on the behaviour of the neighbouring tactels. As each tactel was loaded, measurements of current in response to applied voltage were also taken around the adjacent tactels. A sample of these results for tactels adjacent to the one used to produce the data of Figure 3.8 is shown in the graphs of Figure 3.9.

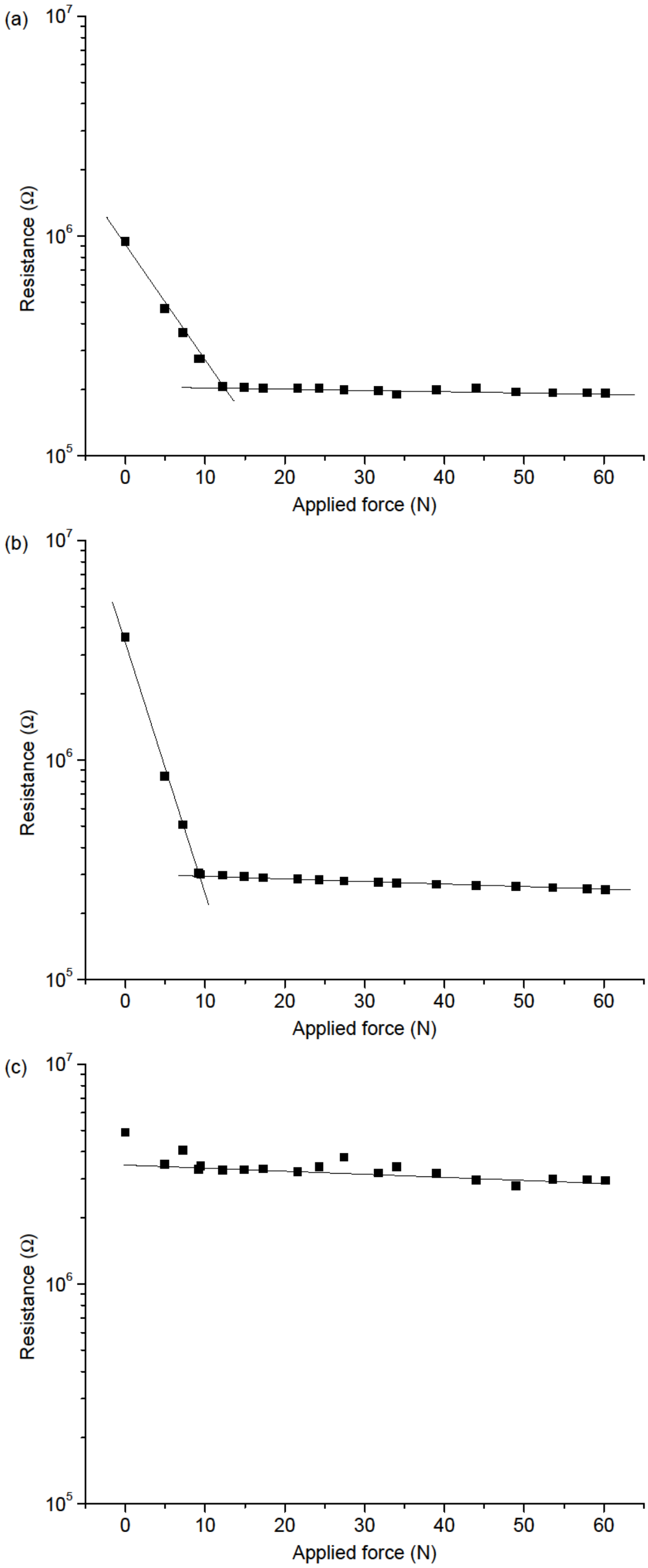


Figure 3.9: Graphs showing the effect of applied force on the measured current on tactels adjacent to one under load. These cells come from (a) the same column (b) the same row, and (c) is a cell that is on a diagonal.

These graphs show that tactels in this design are not independent of the force applied to adjacent ones. This effect is more apparent in the column- and row-adjacent tactels than in the tactels aligned diagonally to the one under load. This could result from one of two effects; electrical or mechanical. The former could arise from the continuous QTC allowing greater opportunity for the current to flow through the active material via the compressed sample. In this approach, the more muted response in the diagonally-aligned tactels arises from increased distance from the contact location. However, this could also have a mechanical cause, whereby the strain induced by contact is able to efficiently propagate through the continuous QTC. This would result in a slight change in the thickness of the active material, producing an effect similar to loading.

One other consideration arises when considering the durability of this design is the prevention of peeling and lamination of the tactile sensing components. The concern is that repeated contact will ultimately cause the electrodes to become detached from the composite, compromising the integrity of the matrix sensor. A second design was therefore developed, in which the QTC mixture was applied directly to the conductive strips of bobbinet within the fabric. This limits the possibility of the composite structure decomposing under repeated contacts. The effect of mechanical deformation is also reduced because the QTC is embedded directly into the bobbinet, which limits the lateral deformation of the material. The number of possible electrical pathways is also reduced because parts of the array are now voids, absent of any conducting material.

The experiments performed on the initial prototype were repeated on this second device. The force-resistance curve of a typical tactel from this array is shown in Figure 3.10, which shows that the new construction does not impact on the performance of an individual device.

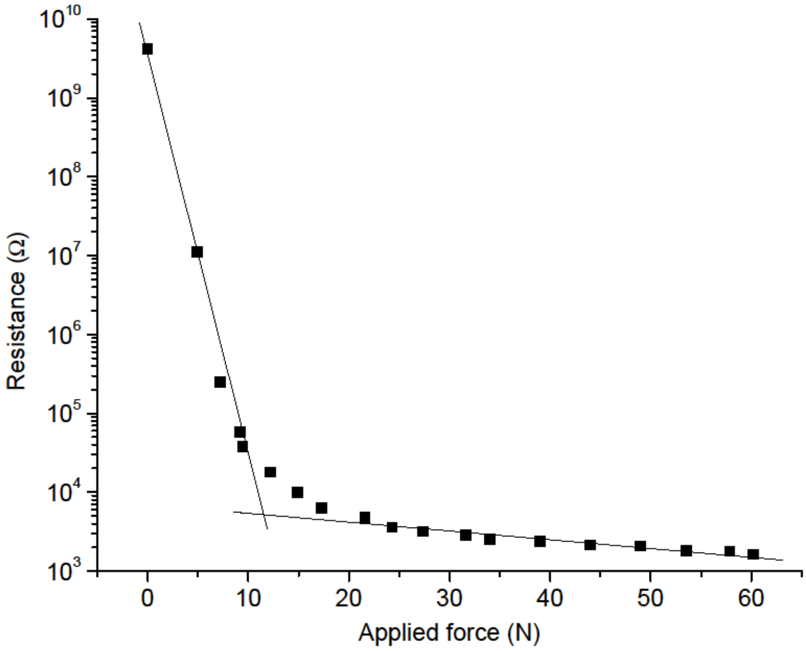


Figure 3.10: Force-resistance characteristic of a typical tactel in the second QTC matrix prototype at an applied bias of 24V.

Of greater importance are the results of Figure 3.11, showing the effect on adjacent tactels, which shows that the tactels are either completely independent or at the very least that any dependence is not discernible and can be ignored for practical purposes. There are variations in the data values with different forces applied to an adjacent tactel, but the variations appear to be random and are uncorrelated with the changed in resistance of the tactel under loading.

This suggests that isolating the active material to the conducting portion of the fabric assists in making the tactels independent. Such independence is essential to effectively localise contacts, since a lack of independence means that the extent of objects might be overestimated by the ghosting effect observed in the first prototype. These results are essential to developing the final design developed for this work.

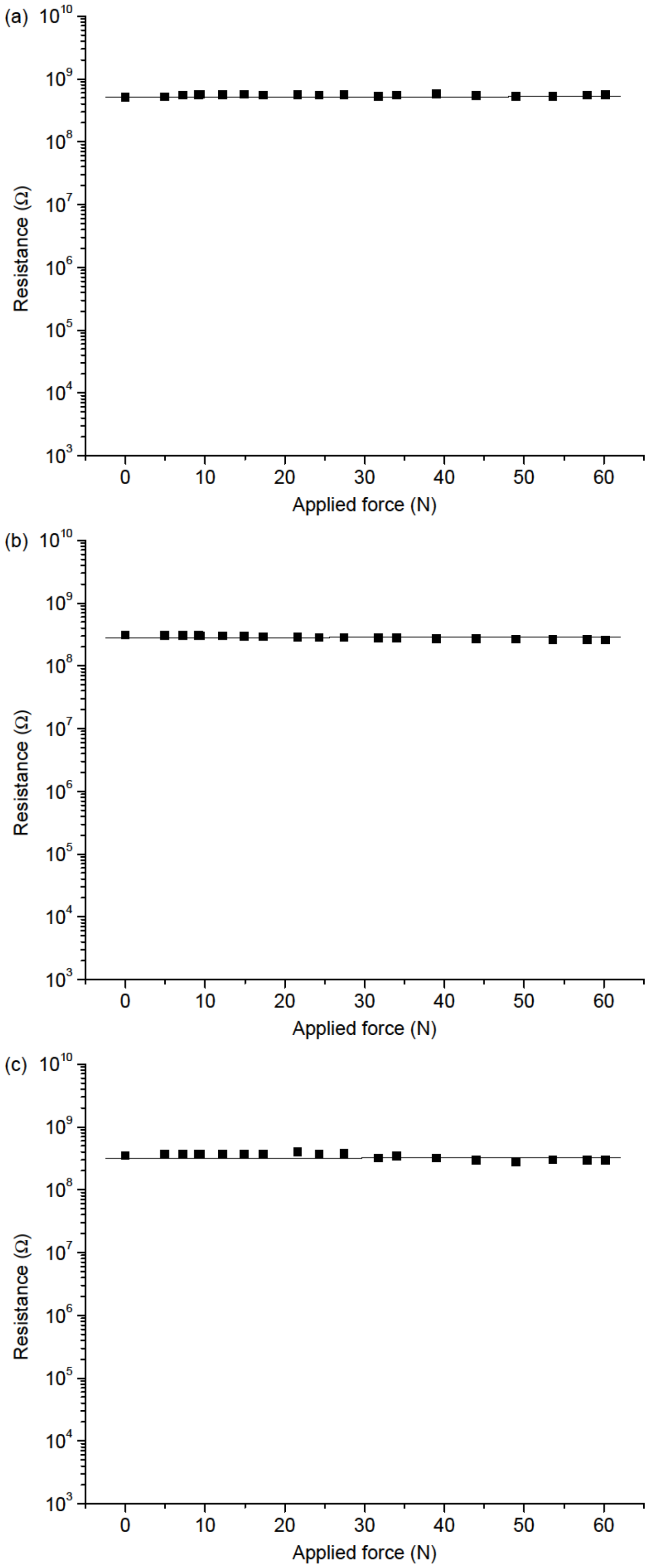


Figure 3.11: Graphs showing the effect of applied force on the measured current on tactels adjacent to one under load in the second prototype. These cells come from (a) the same column (b) the same row, and (c) a diagonal.

3.3. Discussion and Conclusions

This chapter has described the early development phases of designing a QTC-based tactile sensing array. Using a simple design such that the matrix is formed solely from the layout of the electrodes but with a bulk quantity of composite introduces problematic behaviour. This has either a mechanical or electrical explanation, or even a combination of both. By applying the active material directly to the conducting bobbinet, it appears that the tactels can be made to be independent of forces applied to their neighbours.

This is possibly effected by the reduction in the composite in the regions between the tactels. This is illustrated by Figure 3.12, which is a schematic of the cross section of two adjacent tactels through the sensor.

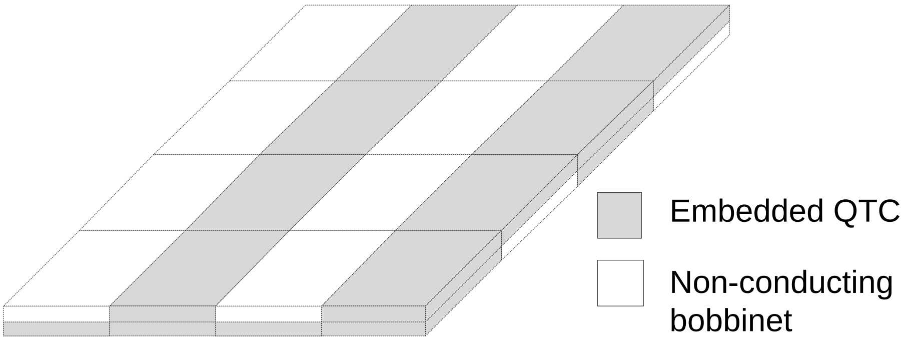


Figure 3.12: Schematic of the cross-section of the second prototype to show the differing quantities of material in a tactel compared to the interstitial regions.

Since the active material is only present on the conductive parts of the fabric, void areas are formed within the sensor array that reduce the electrical conduction pathways between tactels and also limits the distribution of the mechanical effects of applied force to adjacent tactels. The fact that the active material is embedded within the fabric also facilitates mechanical isolation because the active material is less able to stretch in response to compression. This is a positive effect from using a fabric such as bobbinet with regular gaps that can contain the material and the results indicate that there is little or no impact on the sensitivity of the device.

Nonetheless, this secondary prototype is unsatisfactory in other respects. The fabric being used constrains the alignment and deployment of the sensor array on a robot at the time that the sensor is constructed. As such, there is a risk that the sensor will have limited application or that bespoke solutions will need to be developed for each application, significantly increasing costs. From the data-centric perspective, this is not a problem, but since the active material appears to satisfy some of the requirements developed in the previous chapter, it is important that the array version of the sensor is practical and easy to deploy. One method to consider is allowing the strips to be aligned according to the surface they are applied to, rather than constraining the form of the matrix independently of the robot. The means of doing this will be considered in the next chapter with the development of the sensor to be used for the remainder of this work.

The electrical characteristics of the pill-form of QTC were also considered, but have not been explored in the prototypes. This is because the matrix sensors are merely development prototypes and the data-centric approach demands only consideration of the response of the sensor to force. In developing and characterising the final version of this sensor in the next chapter, the electrical characteristics will be considered to examine the relationship between it and the pill form of the active material. This will assist in the further development of the composite for use in array structures for tactile sensing.

Chapter 4
QTC Matrix: Design and Characterisation

4.1. Design and Experimental Methods

In terms of overall sensor design, the final QTC matrix will have the same layout as the prototype matrices described in Chapter 3. The key difference is the means by which the columns and rows of the matrix are formed. Instead of using a continuous sheet of fabric with alternating stripes of silvered and non-silvered nylon to form these, the rows and columns are instead formed from individual and isolated strips of silvered bobbinet.

In principle, this affords significant advantages for deployment of similar sensors in alternative configurations. For instance, in this work the matrix cells will be formed by the intersection of strips at right angles. This may be inappropriate, for example, when deploying the sensor on a cylindrical link where uniformity of sensing is desirable, because the connectors on one edge will lie on the cylindrical surface. The advantage of deployment using individual strips of transducer, however, is that they can be aligned in whatever configuration is appropriate to simultaneously facilitate coverage and allow simple electronic integration. The strength and durability of the bobbinet also means that it can be cut without the structure of the fabric being damaged, which also makes it appropriate for customisation.

A composite of QTC was applied to silvered bobbinet with a resistance of 3.7Ω for every metre by Peratech Ltd who produced the composite using a patented and proprietary process. Figure 4.1 shows the two halves of the final sensor, with sixteen bobbinet-backed QTC strips orientated in parallel. The active material was connected to the external connection leads by means of embroidered, silver nylon in the polyester backing.

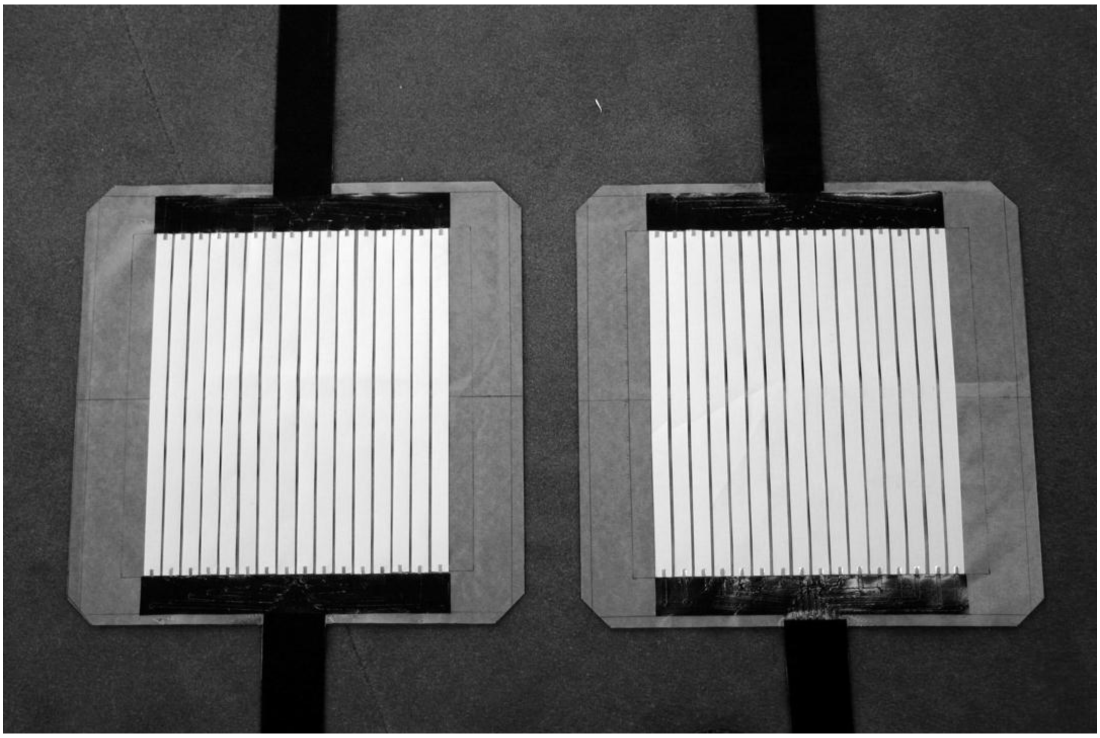


Figure 4.1: Photograph of the two layers of the final sensor prior to bonding. The light grey vertical stripes in the picture consist of QTC backed by bobbinet.

The fabric embroidering extends along two connection leads in both layers, which were terminated with a standard pin header that allowed the conductive textile to be interfaced with conventional electronic connectors. Although only two connection leads are necessary for this sensor design, a redundant connection was added to each layer to prevent damage in one lead to render the sensor unusable. It also allowed the continuity of the electrodes across each strip to be verified. It is anticipated that such redundant leads would not be necessary in a real-world deployment.

The QTC strips shown are 10mm in width and 215mm in length. The strips were separated by a gap of 3mm to facilitate electrical isolation. The separation also assists in allowing the sensor array in being mounted conformally to non-planar surfaces, since a degree of pre-stressing can be absorbed by these naturally flexible separations. This should minimise the influence of conformability on the unloaded resistance of the tactels. With these dimensions, the total sensing area of the sensor is 420.25cm^2, including the separation gaps, which should be included in the calculation of sensor resolution.

The two layers shown in Figure 4.1 were laid over each other in an orthogonal orientation. The uncovered polyester backing was then fused together to prevent lateral movement by the QTC strips, which could cause damage to the sensor. This gives a tactile array with 256 tactels, and an effective resolution of 13mm. Each tactel has the structure illustrated in Figure 4.2, with two electrodes constructed from bobbinet surrounding two layers of QTC.

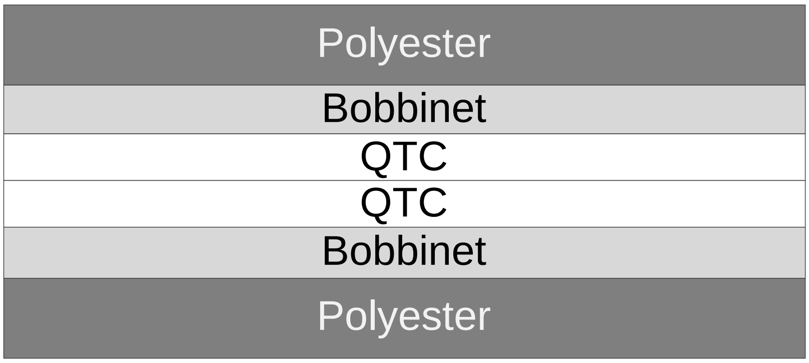


Figure 4.2: Illustration of the structure of a single tactel formed by the intersection of two strips.

As with the prototypes described in Chapter 3, individual tactels can be addressed by using switches to select the relevant row and column and so form a circuit across the desired point on the array. For convenience, the columns of the matrix will be denoted alphabetically from A to P, and the rows numerically from 1 to 16. These labels will be used to refer to individual tactels; for example, B9 refers to the tactel in the second column and ninth row of the matrix device.

The same apparatus as used for applying pressure to the prototypes in Chapter 3 was used for the experiments described in this chapter. Control of the switching necessary to address individual tactels was performed using a Keithley 7702 switching unit in a Keithley 2750 enclosure. Connections were made by closing two switches in the 7702, where all switches had been wired to a known row or column using standard pin connections. For all experiments involving quasi-static application for force, voltage was sourced to the tactel under test from a Keithley 2400 SourceMeter™ connected to the sensor through the switching unit. This unit was also used to measure current within the circuit. For measurements involving dynamic application of force, an Agilent 34410A digital multimeter was used to measure current, because it can be configured to measure current in a circuit at up to 500Hz. In this configuration, the Keithley 2400 acted as a source. A LabView™ program was used to configure the experiment and automate the collection of data, using a custom arrangement of pre-supplied LabView sub-programs.

A range of forces was typically applied to each tactel under test in order to test the responsiveness of the sensor. The range was between 0N and 65N, which should characterise the device for the majority of applications. Experiments were performed to characterise the behaviour of individual tactels under a range of forces and voltages. Measurements were also taken to evaluate the effect, if any, of the loading of a particular cell on its neighbours. In order to limit ambiguity in localising obstacles, it is important that this effect is either eliminated or insignificant.

Other experiments were also performed to characterise the behaviour of the sensor when multiple tactels were loaded, which reflects the operational likelihood that contact will not neatly strike individual tactels in isolation. Measurements of current were converted into resistance by using the standard ohmic relationship $\boldsymbol{R_i = V_i / I_i}$ for some given measurement *i*. All measurements were repeated with several hours between repeats in keeping with the quasi-static experimental methodology chosen to eliminate memory effects that can occur in the active material [5]. The experiments allowed two forms of characterisation. The first was to characterise the response of the device as a tactile sensor, and the second was to assess the electrical characteristics of the composite structure. The former will be considered in the next section, and the latter will be considered thereafter.

4.2. Results and Discussion: Application to Robotics

4.2.1. Isolated Tactels

In these experiments, the apparatus was aligned to apply force to individual tactels only. Voltages were applied in half-volt increments from 0V to 40V, but the basic shape of the relationship between force and resistance was observed to be the same for all voltages. For the purposes of this discussion, all the results shown are from measurements at an applied voltage of 24V. The experiment was performed on twenty, arbitrarily-chosen tactels and the force-resistance curves for three of these are shown in Figure 4.3.

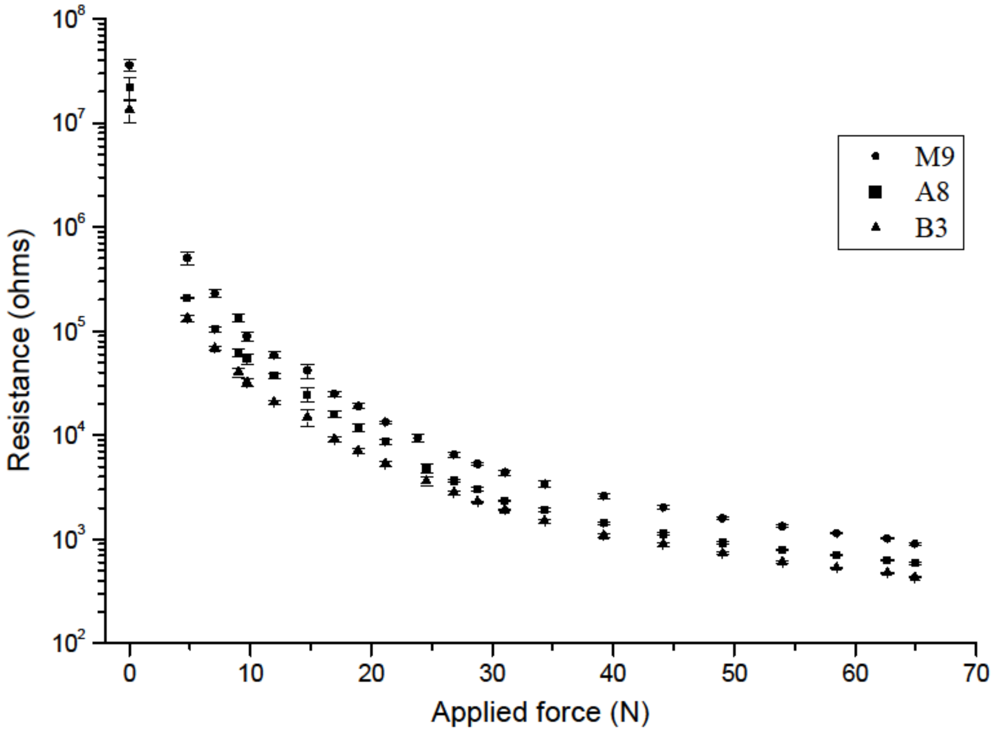


Figure 4.3: Graph showing the dependence of resistance on applied force on three typical tactels; M9, A8 and B3 at an applied voltage of 24V.

Figure 4.3 shows that there is some variation between the characteristics of the tactels, but that the characteristic relationship is nonetheless the same. These differences can be explained by differences in the structure of each tactel such as differing contact resistances with the electrodes resulting from inexact alignments in the conducting bobbinet mesh. Local variations in the filler fraction of the composite may also have an impact on these curves [5]. The unloaded resistance is of the order $10^7\Omega$, which means that the power drawn by a tactel in an unloaded state is extremely low. With a relatively small applied force of 4.7N, the resistance of the cell drops two orders of magnitude to $10^5\Omega$. This is an easily identified change, even in the presence of noise and so indicates a reasonable degree of sensitivity. The errors in the logarithmic plot are difficult to discern precisely, so Figure 4.4 shows a plot of the standard deviation of the measurements taken at each applied load of tactel B3.

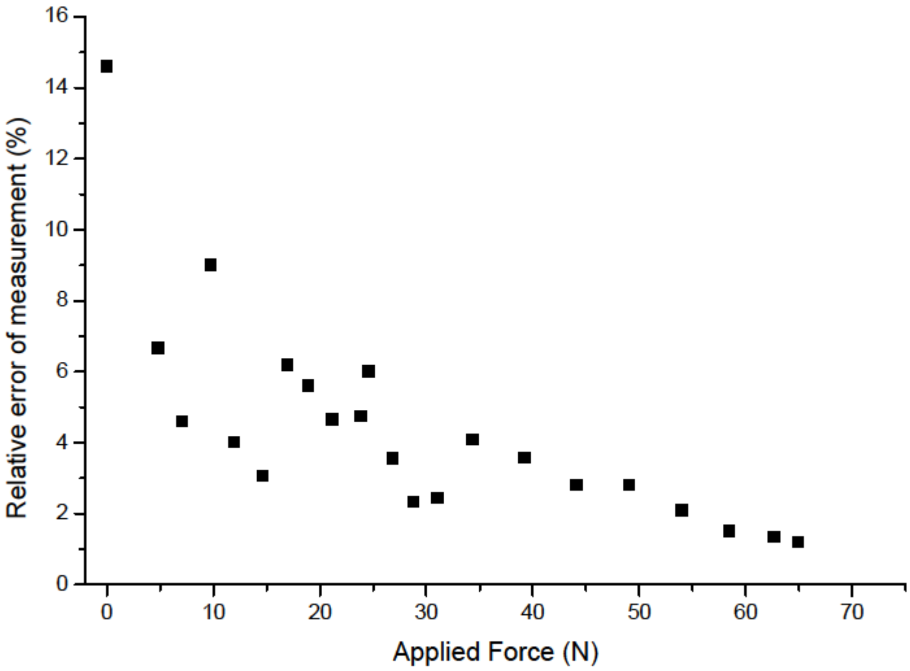


Figure 4.4: Plot of the standard deviation of measurements taken at applied loads of tactel B3 at an applied voltage of 24V.

These errors are typical of all tactel measurements, in that the unloaded state has the highest error, with negligible errors after an applied force of approximately 25N. This variation may be too high to make an accurate determination of a precise level of applied force when the force is very low, or otherwise may require that a larger number of samples be taken to increase confidence in the detected level. The alternative is simply to apply a suitable threshold either as a general rule or on a per-tactel basis that takes this large variance into account. When the resistance falls below this threshold, the system can assess that a contact has been made and apply a calibrated rule to assess the level of force.

As part of these experiments, measurements were taken from the eight tactels adjacent to the ones under test. No relationship between force applied to the central tactel and the resistance of the adjacent tactels was observed, as expected from the results of the prototyping experiments described in Chapter 3. A measurement of applied force for a given tactel can therefore be unambiguously attributed to a local contact event, independently of similar measurements in adjacent portions of the array. This independence allows for simple localisation of the features of an object or the environments, which improves the efficiency and effectiveness of consequential decision-making by a robot.

The design of this sensor was intended to allow the sensor to conform to curved surfaces without significantly impacting on the performance of the device. Experiments were therefore performed to assess the effect of making the sensor conform to curved surfaces. This required an alteration to the experimental apparatus to allow curved surfaces to be fixed in place beneath the plunger.

Several tactels on the sensor were characterised when the sensor was flat, as previously, and on two surfaces with radii of curvature of 280mm and 28mm respectively. Care was taken to align the centre of the force-applying plunger with the centre of the tactel on each curvature. The variation between cells was essentially the same as described previously and for similar reasons. Figure 4.5 shows the force-resistance characteristics of tactel J4 under the three different curvature conditions.

The data show that the characteristics of the curves appear to be almost identical, with the exception of the unloaded data point. As the curvature increases, the unloaded resistance falls, which is congruous with the findings of Bloor *et al*. who found that curving the composite induced strains that made it behave as though it was compressed [5]. It appears, however, that the rest of the force-resistance profile is unaffected by this induced strain. The loading at 5N shows the next highest difference between the curves, again caused by the internal strain induced by the curvature. As the force increases, however, the differences become negligible and two explanations present themselves for this.

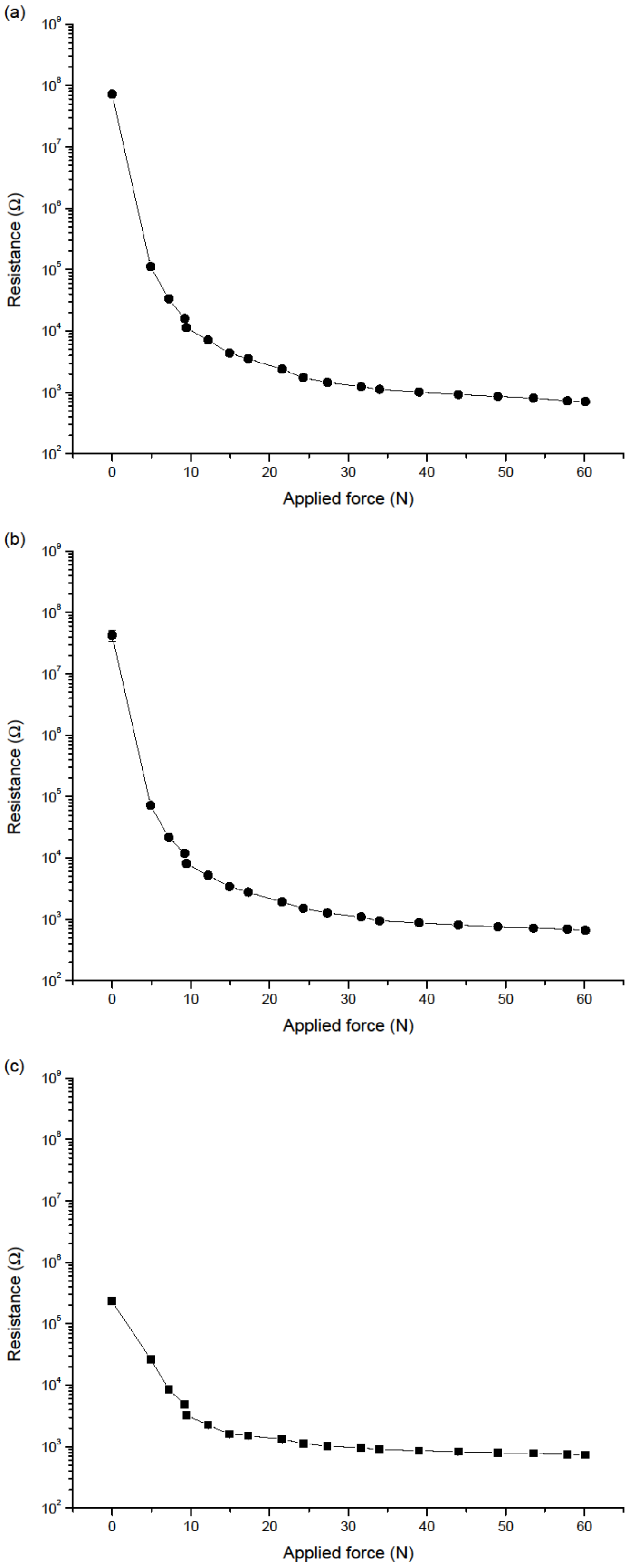


Figure 4.5: Force-resistance characteristics of tactel J4 when the sensor is applied to surfaces with radii of curvature of (a) infinity (i.e. flat) (b) 280mm and (c) 28mm.

The first is that in adjusting the apparatus, no correction was made to the curvature of the plunger head. This makes the impact area lower for increasing curvature at low force, increasing pressure and so making the resistance lower. With increasing pressure, however, the compression of the elastomer-based composite causes the impact profile to increase and so causes the force-resistance curve to more closely approximate the flat curve. The decision not to change the curvature of the plunger was based on the principle of trying to detect the same contact with the sensor in different configurations. The shape of the impact on the sensor is unlikely to change as the sensor changes, and so the additional variable change would be undesirable for comparison of sensor performance. This explanation for the difference between the effects of lower and higher applied forces is somewhat unsatisfactory, since even the highest force would seem unlikely to completely flatten the material when the radius of curvature is at its lowest.

An alternative explanation is that at the low applied force, the effects of curvature dominate the electrical characteristics of the material and this will not substantially change with applied force. As the force increases, the changes associated with applied force dominate over the impact of curvature and so the curves trend to a common characteristic. This argument is supported through examination of the results of Bloor *et al*., since they indicate that for the bending radii used in our experiments, they observed resistance dropping to $10^6\Omega$ from a non-bent resistance of around $10^9\Omega$ [5]. This is insignificant compared to the reductions produced by significant amounts of applied force that are reported, although it must be borne in mind that the precise nature of the composites may be different from those used in these experiments [5]. Although no report exists of combining lateral strain and applied force, as presented here, it is apparent that if the applied force causes the material to exhibit lower resistance than that produced by lower strain, the former will dominate the latter.

The impact of these effects on deployment is insignificant if the curvature is relatively low, as is the case for the 280mm curvature examined here. Even with significant curvature, the force-resistance characteristic tends to that of the flat form with increasing application of force. For more extreme radii of curvature, it is unlikely that a sensor of this construction would be appropriate, since the extremely low resolution will give localisation information relating to contacts that is poor to the point of being uninformative. It is possible, however, that QTC could still be used in such a system by employing a composite mixture with a stiffer elastomer than deployed here, albeit at the expense of reduced sensitivity to contact. Using narrower strips of QTC may also help, as the amount of strain required to make the material conform to the surface will be lower and this would allow the composite deployed in the sensor examined here to be used on different scales.

One final experiment relating to individual tactels is required to examine the sensor's response to dynamic impacts. This characterises both how rapidly the material reacts to a contact and how quickly it recovers after impact. The elastomeric basis of the active material would typically suggest that the recovery time would be very high, and is a common criticism of this kind of material, as discussed in Chapter 2. However, the thinness of the QTC layer relative to its strong, structured fabric electrode in this design is intended in part to reduce the deformation and so hopefully allow it to recover its shape and therefore unloaded resistance value quickly.

For these experiments, tactels were loaded with three different forces for approximately ten seconds before being unloaded. Samples were taken continuously at a frequency of 400Hz using an Agilent 34410A, which was achieved by setting it to auto-zero only at the start of the experiment and to lower its number of power line cycles (NPLCs) to its lowest value. This latter setting prevents the effective elimination of periodic noise induced by the alternating power supply, but because we are interested in the gross features of the dynamic response rather than the exact current values, this is a reasonable trade-off in performance in order to achieve a higher sampling rate. The results of this experiment when performed on a typical tactel are shown in Figure 4.6, where forces of 15N, 25N and 35N were used.

For all loads, the response to applied force was less than 0.0025 seconds to produce an order-of-magnitude change in the resistance of the tactel. The device then additionally requires approximately 0.1 seconds for the resistance value to settle. Thereafter, some gradual reduction in resistance can be observed, arising from the viscoelastic nature of the material. This drift is, however, minimal and would not be problematic in many robotics applications where a response to the force is immediate. The sensor is slower to respond to the contact being removed, with a recovery time of approximately 0.5 seconds. This time appears to be independent of the applied force, which results from the stiffness both of the QTC composite manufactured for the sensor and the bobbinet-based structure of the sensor.

These response parameters are unsuitable for fine manipulation tasks where a rapid response is required, but will find wide applicability in many other applications, particularly where a rapid response to the initial contact is desirable. The response curve of Figure 4.6 were typical for all tactels, but there was one anomaly measured that is worthy of comment. This is shown in Figure 4.7, corresponding to the dynamic response of tactel C5 with a load of 15N.

The anomaly is in the recovery where, after reaching a resistance near the original value, the resistance suddenly drops and then slowly returns to the standard value. One explanation for this is that the active portion of a tactel consists of two layers. If one of these layers expands at a different rate, it is possible that it may recompress the second layer and reduce the resistance of the overall structure slightly. Further work is necessary to confirm or otherwise identify the cause of this anomaly, and whether it might be alleviated by bonding the QTC together in a different way. This is, however, an anomaly that was only noticed in one tactel and does not detract from the performance of the device as a whole.

These experiments have characterised the force-resistance relationship when force is applied to a single tactel. A more realistic scenario in an unstructured environment is that impacts will occur across multiple tactels. The effect of such impacts is examined in the next section.

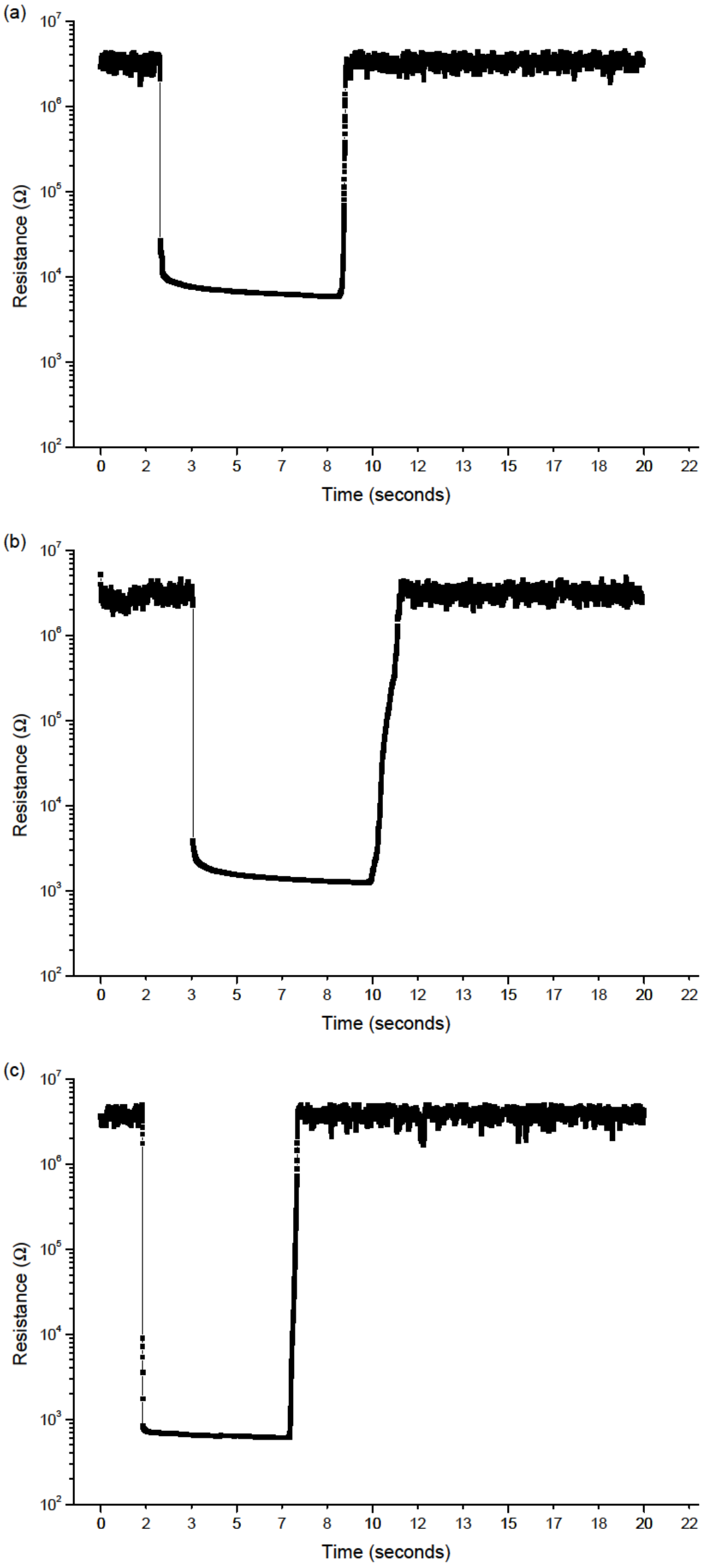


Figure 4.6: Graphs showing the results of the dynamic tactel loading experiments for tactel G7 using a load of (a) 15N (b) 25N and (c) 35N

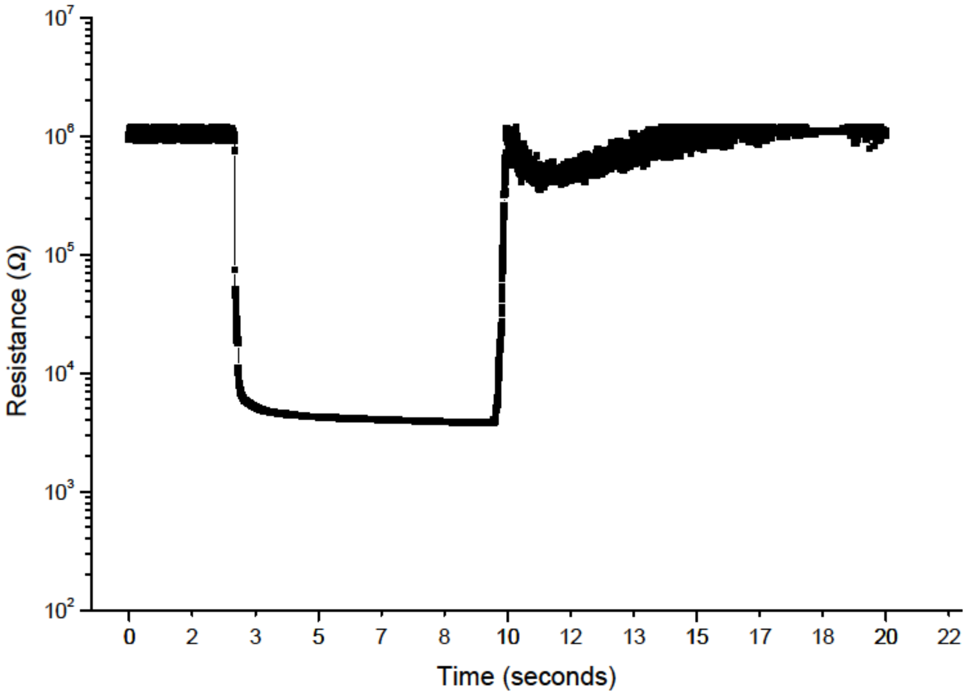


Figure 4.7: Graph of the dynamic response of tactel C5 under a load of 15N.

4.2.2. Multiple Tactels

As a contact is shared across multiple tactels, a non-uniform distribution of the impact area must be assumed in the general case. This coupled with the pressure-sensitive nature of the composite means that there may be differences in the response of multiple tactels to the same contact. An experiment to quantify this effect was performed, in which a fixed force of 43N was applied to the sensor. The position of the plunger was changed so that the applied force was moved progressively across a strip of three tactels. Following the results of the dynamic response experiments in Section 4.2.1, the rate of progress was made sufficiently slow to allow the composite to recover between applications.

Figure 4.8 illustrates how the position of the plunger was changed during this experiment, and relates these changes in position to the distance moved along the three tactel strip. This distance is the independent variable for examining the results of these experiments, which are shown in Figure 4.9.

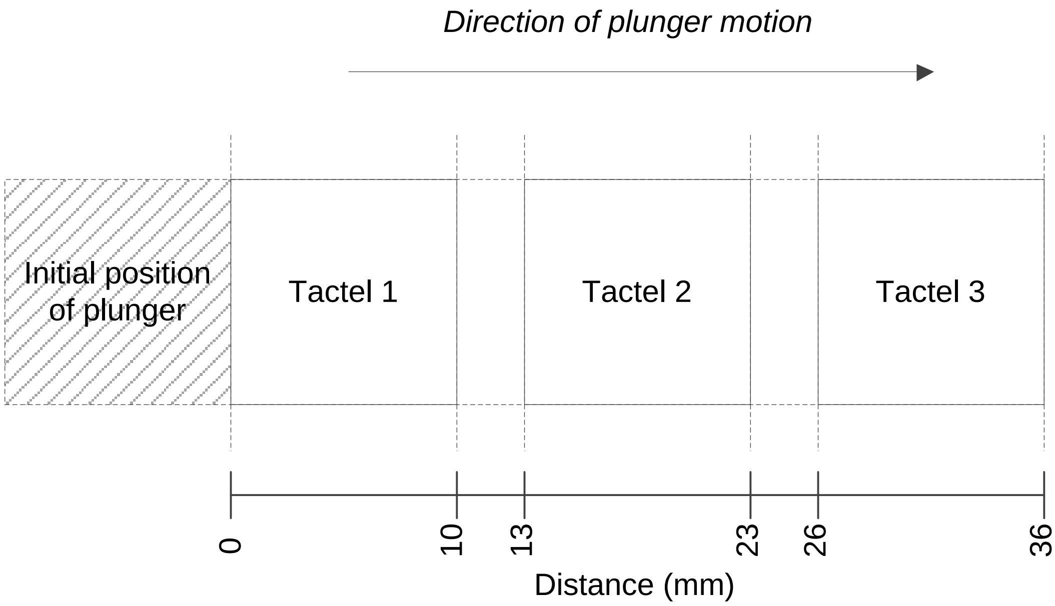


Figure 4.8: Diagram illustrating the change in the location of applied force across a strip of three tactels.

The force applicator was moved across the strip in increments of 2-3mm. Figure 4.9(a) shows a rapid drop in resistance as the plunger begins to cover the first tactel. There is a clear dependence on the coverage area, highlighted by the fact that a lower limit is not reached until the distance moved across the plunger is approximately 5mm. As the plunger begins to enter the separation between tactels, this effect is observed again, with a gradual rise in the resistance corresponding to a commensurate fall in the resistance of the adjacent tactel. Similar effects are observed with the movement between the second and third tactel. In keeping with the results indicating the independence of the resistance of a tactel to the force applied to its neighbours, Figure 4.9 also shows that the tactels return to a high resistance when they are unloaded.

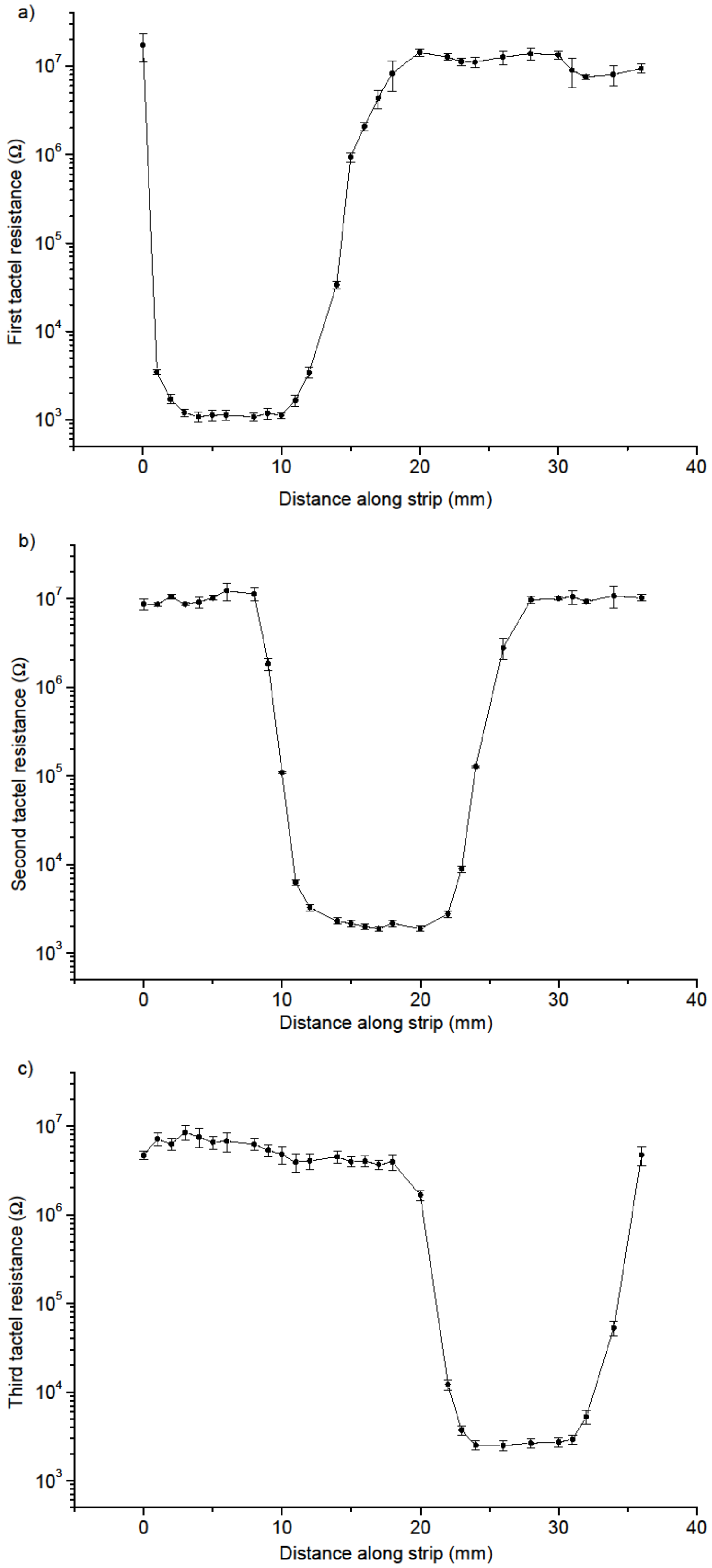


Figure 4.9: Graphs showing the effect of a fixed force of 43N being moved across a three-tactel strip on their resistance. The distance scale is based on the plunger position illustrated in Figure 4.8

Identifying the location of obstacles using tactile sensors is usually limited by their resolution and so an impact across multiple tactels would be interpreted as multiple possible locations of the obstacle. In this sense, the sensor described here behaves like any other in its class. The accompanying disadvantage is that discerning the level of force is impossible without knowledge of the distribution of the impact area, and vice versa. Again, this is common to other sensors in this category and so is not a disadvantage in of itself. There are, however, certain issues to consider in terms of the change in the force-resistance relationship of the sensor when the contact occurs over multiple cells. In order to illustrate this, a further experiment was performed to apply a range of forces to the centre of a group of four tactels.

As expected, the results of this indicate that the resistances of individual tactels are higher compared to those obtained for application of the same forces to an isolated tactel. Figure 4.10 shows the results for one such tactel, where one curve shows the force-resistance relationship for isolated impact and the other shows the characteristic when the load is shared equally between four tactels.

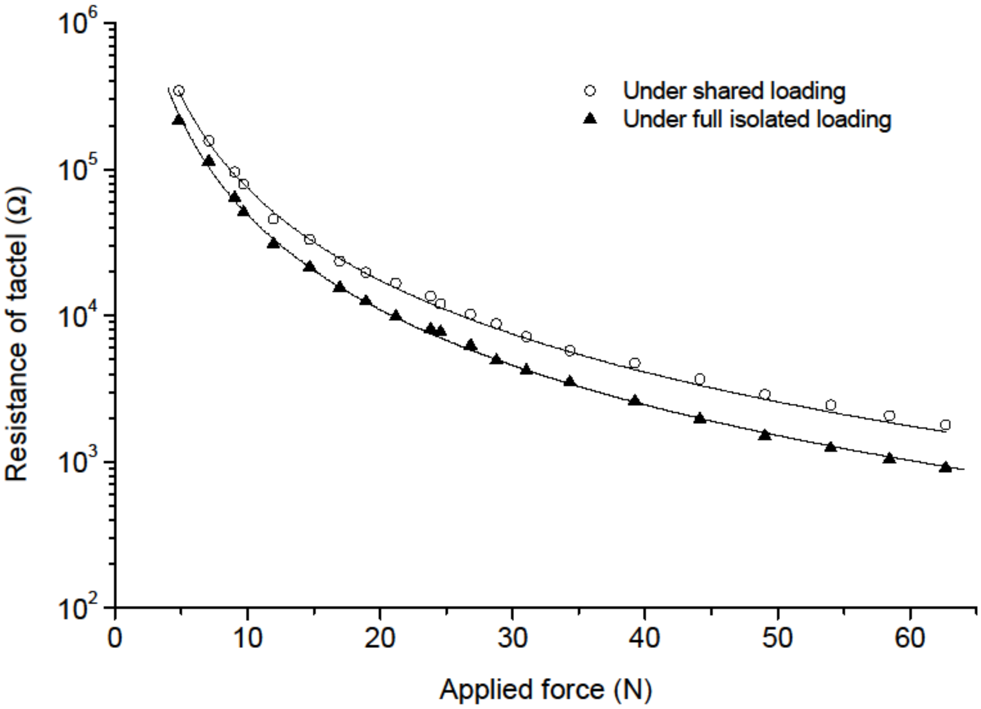


Figure 4.10: Graph comparing the force-resistance curve of tactel B15 when loaded in isolation and when loading is shared with tactels A15, A16 and B16.

Again, these results are to be expected due to the reduction in pressure. The difference between the two curves increases with applied force and this may impact on its usability for an application. If precise levels of force must be obtained on a per-tactel basis, then additional processing will be required to process the values across all adjacent cells and make an estimate of the actual impact force of the object. Other applications will, however be able to cope with a simple series of discrete thresholds to decide if a force is soft, medium, hard, etc. The allocation of these thresholds is dependent not only the characteristic response to an isolated impact, but also to the anticipated type of obstacle. For this sensor, the differences between the curves in Figure 4.10 are not very large and so still has a wide range of applicability in robotics without additional processing provided a threshold approach is used.

Overall, these results suggest that the principle of localisation can be extended from the one-dimensional case to two-dimensions over the surface of the device. Coupled with the independence of tactels from their neighbours, the QTC array can be used to locate contacts, allowing a robotic system to formulate an appropriate mechanical response to the environment.

4.2.3. Conclusions and Discussion

The experiments described in this section have shown that the design of the sensor allows for repeatable measurements of contact, whilst having a demonstrably flexible construction. The QTC has a very high unloaded resistance, which is beneficial for power consumption. Even under high curvature, which is known to affect the resistance of the composite, the effect of force dominates over that of the lateral strain induced by curvature. The independence of each tactel, as produced by the sensor structure, has also been demonstrated. This is a desirable property that allows the true location of a contact to be determined.

There are many systems in which such a sensor could be deployed, predominantly those in which the sensor provides tactile capability to a non-jointed area of the robot. The low power consumption renders the sensor particularly suitable for large-area coverage. The dynamic response of the sensor, however, whilst suitable for arms and other large-area body coverage with sensors, is not ideal for a variety of gripping applications where more rapid responses are required. The present design also lacks the ability to stretch across joints, which is a limitation of the bobbinet fabric used in its design. It is, however, possible to procure bobbinet in which a nylon warp thread is replaced by a nylon covered elastane, allowing stretch in one direction. If the other two threads in the fabric are left as silver-coated nylon, the result is an elastic fabric electrode.

Using such an electrode in conjunction with QTC would be difficult. As has been shown, the composite can stretch, but its electrical characteristics are affected and it may require combined electrical and mechanical models of the composite coupled with measurements of the robot's pose to separate the effects of contact force from those of lateral extension. However, this is a subject that could and should be explored in future work.

The use of bobbinet for the electrode does not appear to have had any detrimental effect on the performance of the sensor. This means that it is a suitable material from an electrical standpoint and permits the prospect of exploiting its mechanical properties further in pursuit of a scalable design. The flexibility and retention of strength when cut suggests that these devices may be used to achieve the goals of Lumelsky *et al*. [16] and Kawaguchi *et al*. [183] by using more conventional transducers and techniques.

4.3. Results and Discussion: Electrical Characteristics

4.3.1. Background and Comparison to QTC Pills

In Chapter 2, the response of commercially-available QTC pills to applied force was briefly examined to motivate the use of the material in a matrix. It is of interest to determine the electrical characteristics of the sensor, since it may aid future development of bespoke electronics to provide a specialised interface for deployment. The conduction mechanism proposed by Bloor *et al*. to explain the properties of QTC is dominated by Fowler-Nordheim tunnelling [5, 6]. This purportedly arises from concentrations of electrons on the sharp features of the nickel particles embedded in the elastomer, producing a sufficiently narrow potential barrier between the filler particles to allow tunnelling to occur [5]. This conclusion was drawn from observing localised discharge into air when 230-240V AC was applied across a compressed sample of the composite [5, 6]. Crucially, this conclusion was not drawn from direct analysis of the current-voltage characteristics of the tactels.

Work in the area of nickel-based pressure-sensitive adhesives supports this conclusion, where electrical characteristics of mixtures of nickel, nickel oxide and air have been explained by a tunnelling mechanism [191], but this is dependent on the distances between the nickel particles being on the order of only several nanometres [192]. This is a condition that may not occur in general. Other work has proposed tunnelling from nickel in composite materials, such as the work of Han *et al*. in which nickel was mixed with a cement-based composite [193]. This conclusion, however, was principally drawn from a comparison between the stress-resistance curves measured by these researchers and those of Bloor *et al*. The process by which the composite was formed would also, according to the measurements of Bloor *et al*., have removed the sharp features of the nickel and thus limited its tunnelling capability [5]. Despite this, the force-resistance relationship observed was the same as that measured for QTC.

These results call into question the electrical conduction mechanism of QTC, particularly as it is not clear that the average separation of the metal particles will become sufficiently small to increase the probability of tunnelling such that it becomes the dominant conduction mechanism. It is also unclear that the contribution of the polymer to the electrical conductivity of the composite can be ignored. The contribution of Fowler-Nordheim tunnelling to current density is given by the equation

$$J = C_1 V^n \exp\left(\frac{-C_2}{V}\right) \tag{4.1}$$

where $\boldsymbol{V}$ is the applied bias and $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ are terms dependent on the characteristic of the composite components , particularly the difference in their Fermi levels [194]. The integer power *n* can take a value of 1 or 2 depending on the shape of the potential barrier. A value of 2 indicates a triangular barrier characteristic of Fowler-Nordheim tunnelling. Consequently, the form of equation 4.1 suggests that by plotting $\mathbf{ln}(\boldsymbol{I}/\boldsymbol{V}^{\mathbf{2}})$ against $\boldsymbol{V}^{-\mathbf{1}}$, any conduction via Fowler-Nordheim tunnelling could be discerned from a negative linear gradient. The results of such a plot are shown in Figure 4.11, where the characteristics of a typical tactel are shown for several loads across the range applied during experiments. Increasing the applied force should reduce the separation of the conducting particles and so favour increased tunnelling. However, the data show that the opposite is happening, with the small region of negative gradient that is present when the tactel is unloaded decreasing in size as applied force increases.

In the face of such evidence contradicting an existing finding, the nature of the composite used in the experiment must be questioned. As has been observed, the material differs somewhat in its force-resistance characteristic from that observed by Bloor *et al*., as a result of requested alterations to the response of the composite. Fortunately, as discussed in Chapter 3, additional results were obtained from a commercially available QTC in a pill form. The *I-V* characteristics of a pill and one of the tactels from the sensor array are illustrated in the log-log plot of Figure 4.12.

The pill sensors display the negative resistance regime and high levels of hysteresis observed by Bloor et al. [5]. The characteristic of the sensors used in this work, however, appear to correspond to that section of the pill's characteristic prior to the switch from positive to negative differential resistance. Therefore, it must be concluded that any conclusions drawn from the *I-V* characterisation of the array sensor must be confined to this pre-switching regime. Nonetheless, plotting $\mathbf{ln}(\boldsymbol{I}/\boldsymbol{V}^{\mathbf{2}})$ against $\boldsymbol{V}^{-\mathbf{1}}$ for a typical pill-form sensor, as shown in Figure 4.13, gives similar results to those for the array sensor. Therefore, although the precise nature of the composition between pill- and array-based sensors may vary, neither displays the characteristics of Fowler-Nordheim tunnelling in their current-voltage profiles. It is concluded that Fowler-Nordheim tunnelling is not the dominant conduction mechanism in these materials, and other possible mechanisms must be considered.

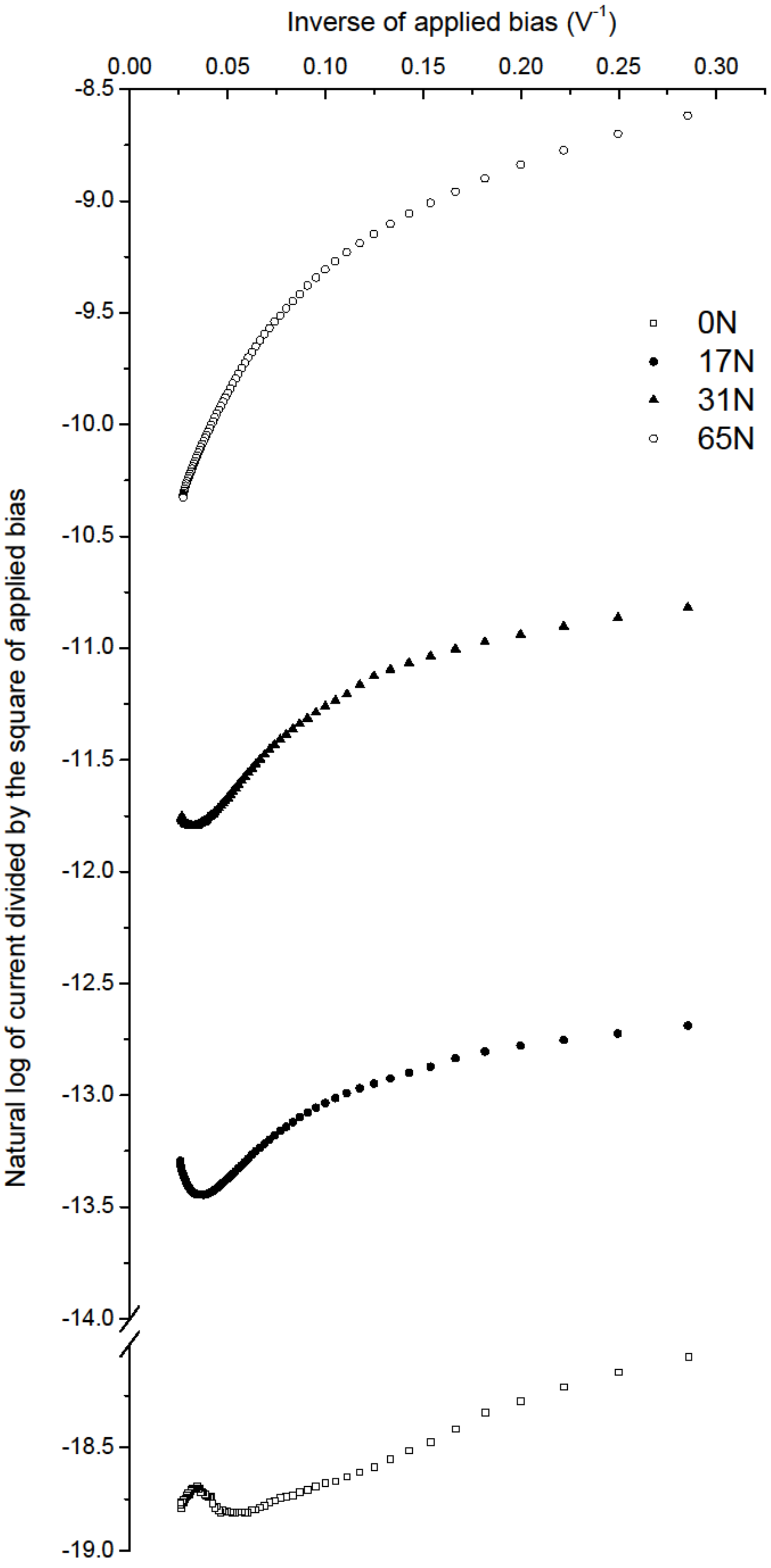


Figure 4.11: Graph of the natural logarithm of current divided by the square of voltage against the inverse of applied bias for tactel P9 at loads of 0N, 17N, 31N and 65N.

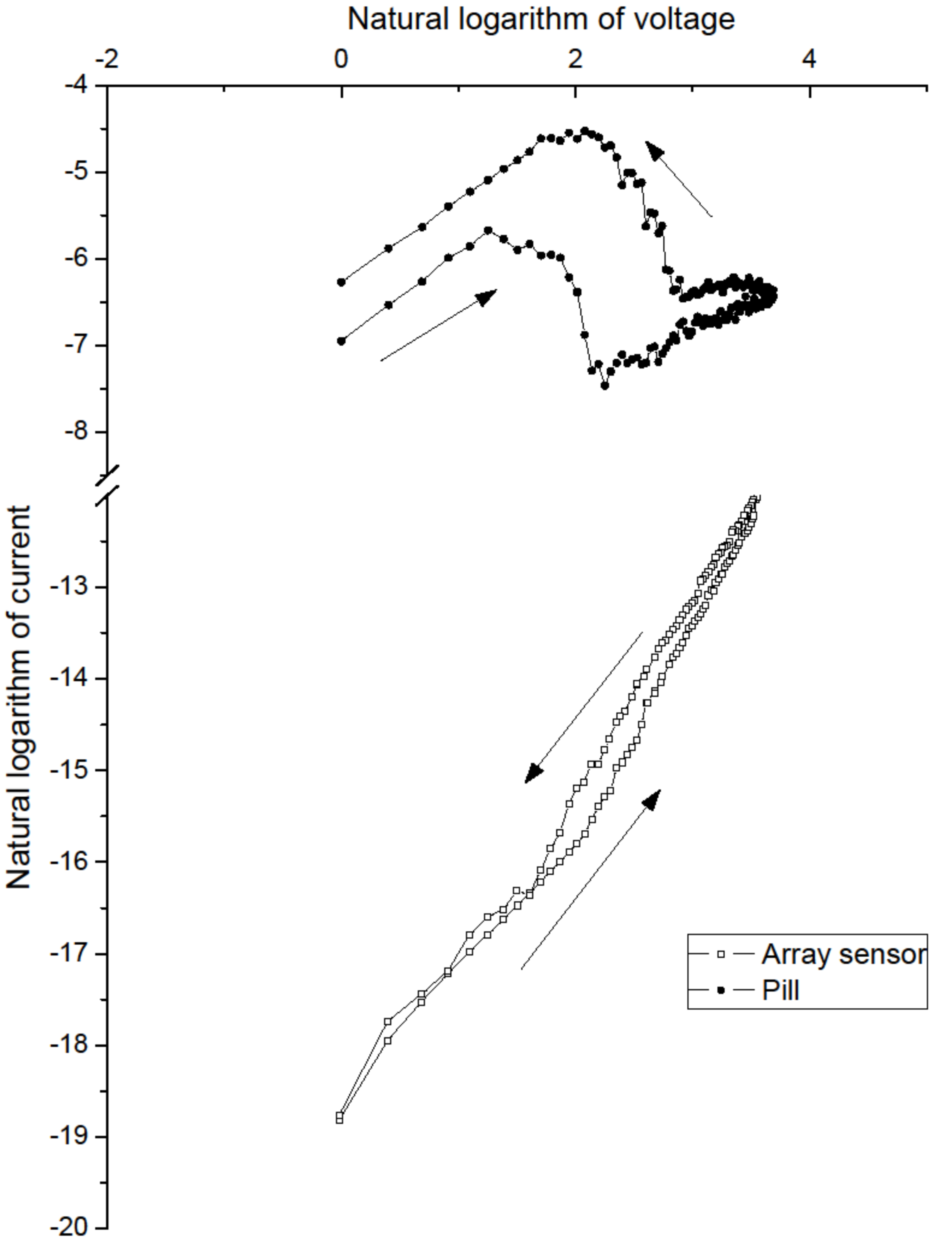


Figure 4.12: Log-log plot of the IV characteristics of a pill-form sensor and an array cell sensor in an unloaded state.

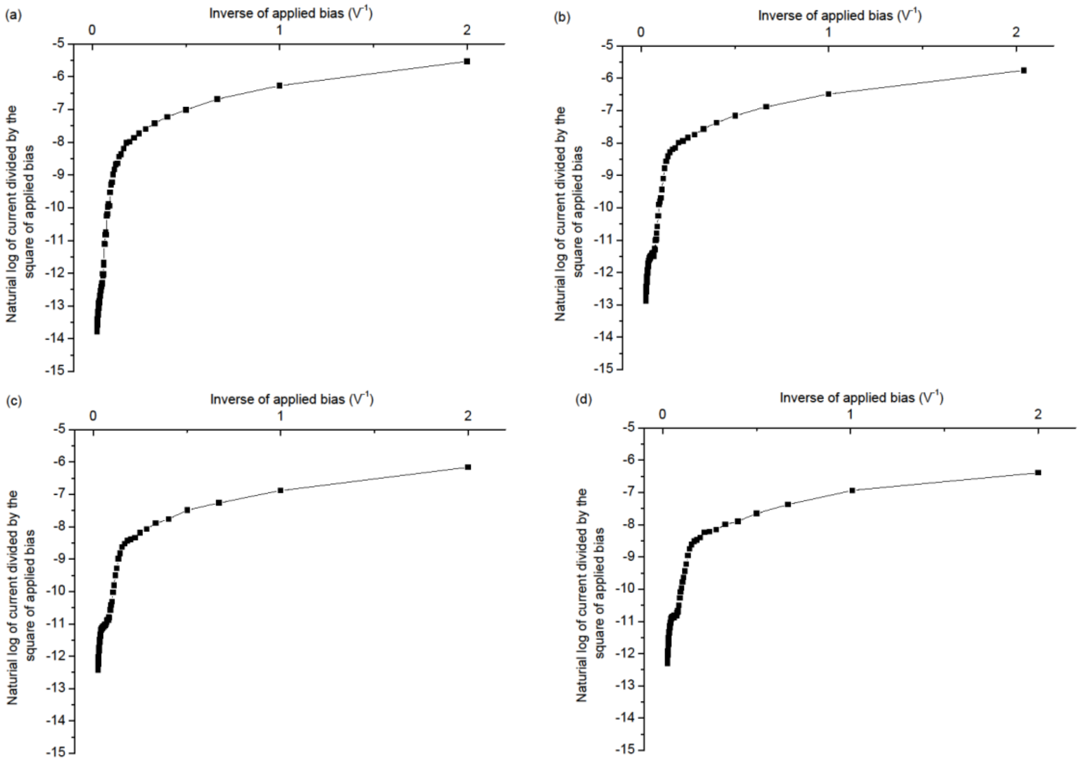


Figure 4.13: Graph of the natural logarithm of current divided by the square of voltage against the inverse of applied bias for a pill-form QTC sensor at (a) 0.17% (b) 8.2% (c) 14.9% and (d) 25.5% compression.

4.3.2. Alternatives to the Fowler-Nordheim Conduction Model

Bloor *et al*. clearly show through images obtained through scanning electron microscopy that the proprietary method of mixing the components of the composite results in complete coverage of the metal particles with few, if any, voids [5]. It is therefore incumbent on any model of conduction through this composite to consider the role of the insulating polymer, since conduction paths through the composite must pass through it. Specifically, the possibility that charge transport through localised states in the polymer facilitates conduction must be considered. This is because electrical conduction is an opportunistic process, whereby mobile charge moves under an applied field through any available conduction path. Localised states within the polymer could therefore contribute to the conductivity of this composite.

The Poole-Frenkel effect describes how an applied electric field can lower the thermal energy required for charge carriers to move from localised states in an insulator into the conduction band. Since localised states are being considered within the composite, this is a candidate as a significant mechanism for conduction. In the presence of the effect, the relationship between current *I* and voltage *V* is given by equation 4.2.

$$I \propto \exp\sqrt{V} \tag{4.2}$$

This contribution can be identified from a linear region of positive slope in a plot of the natural logarithm of the current against the square root of the voltage [194]. Figure 4.14 shows plots for three typical unloaded tactels in the sensor array and shows clear linearity for voltages above approximately 16V, although the precise onset of this behaviour varies between tactels. It is significant that above 30V, where the data of Figure 4.11 had the closest fit to tunnelling behaviour, the plots of Figure 4.14 show a better fit to the Poole-Frenkel model.

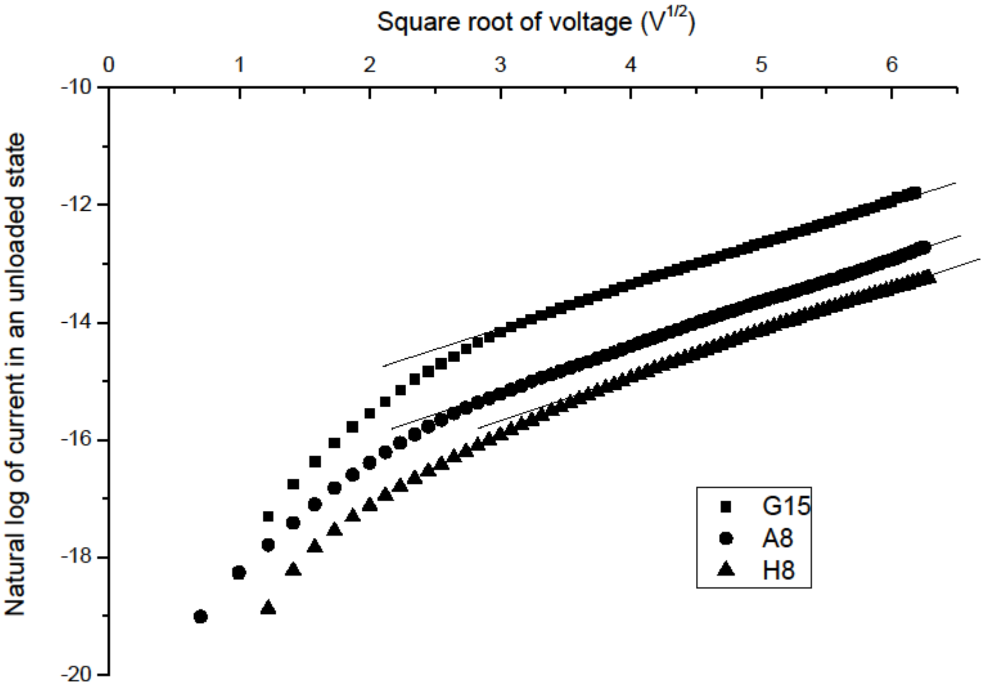


Figure 4.14: Graph of the current-voltage characteristics of three unloaded tactels with linear regions shown as appropriate.

This shows the role that localised states in the polymer have in determining the conductivity of these composite materials. In order to see how this role changes with increasing applied force, Figure 4.15 shows the same types of plot for the characteristics of a typical tactel obtained under several different loads.

Figure 4.15 shows the non-linear region becoming more prominent as applied force increases until 65N where no linear region can be discerned. For reference, similar plots are shown in Figure 4.16 for the pill form of the composite under different compressions, which shows similar features.

At lower voltages, this deviation probably arises because the applied voltage provides insufficient additional energy to enhance thermionic emission from the localised states of the elastomer. This means that the rate of transfer of charge through the polymer is low. The rate of charge injection, however, is initially little changed, allowing a space charge to develop rapidly around the metal particles, which slows the rate of injection. As the composite is loaded, the layers of polymer between the metal particles become thinner. The result of this is that the space charge regions local to the metal particles occupy an increasing fraction of the polymer volume with increasing force. Space-charge limited currents take the form

$$I \propto \frac{V^m}{t^3} \tag{4.3}$$

where *t* is the thickness of the material, and *m* is typically equal to a value of around 2, as given by the Mott and Gurney square law [195, 196]. These features can be identified from linear regions in log-log plots of the current-voltage characteristics. Figure 4.17 shows the log-log plots for a typical tactel under several different loads, with the linear region present in the characteristic of the devices highlighted.

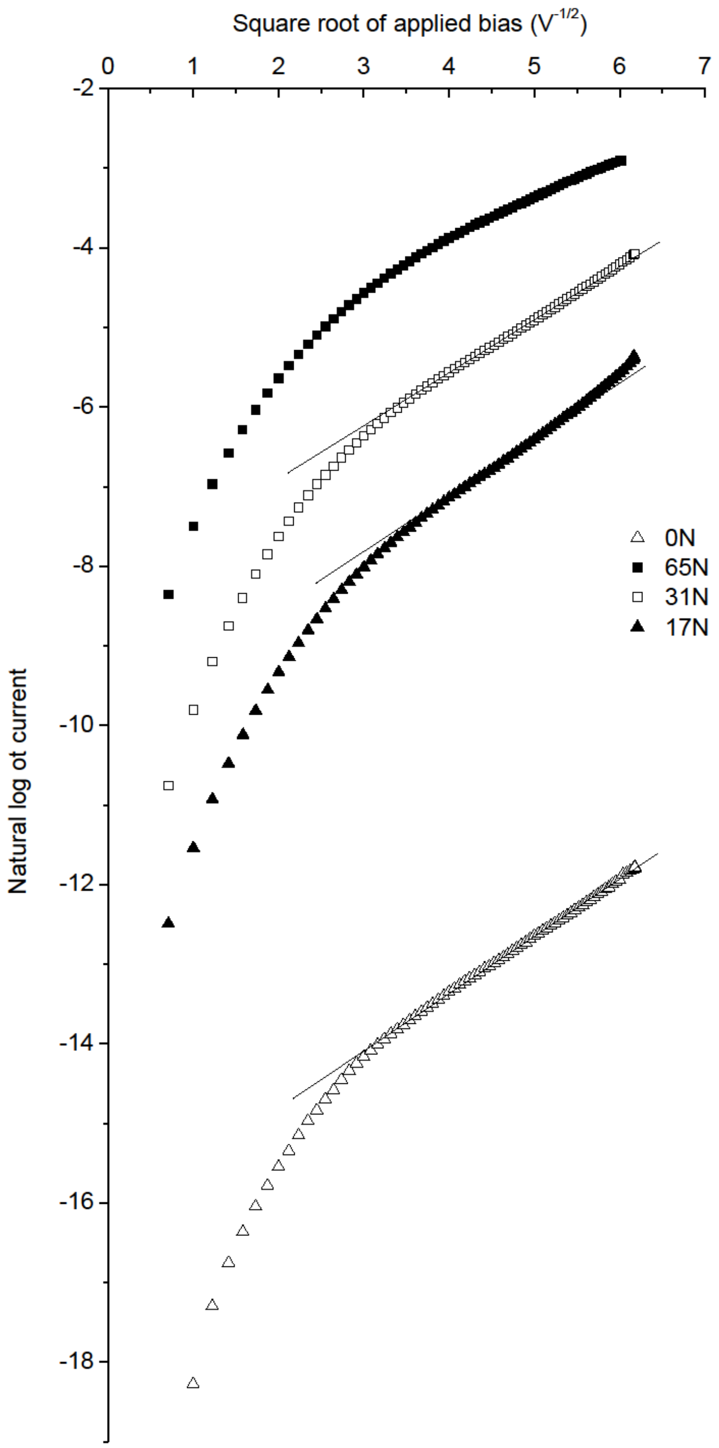


Figure 4.15: Graphs showing the current-voltage characteristics of cell G15 when loaded with 0N, 17N, 31N and 65N of force to highlight the contribution of the Poole-Frenkel effect.

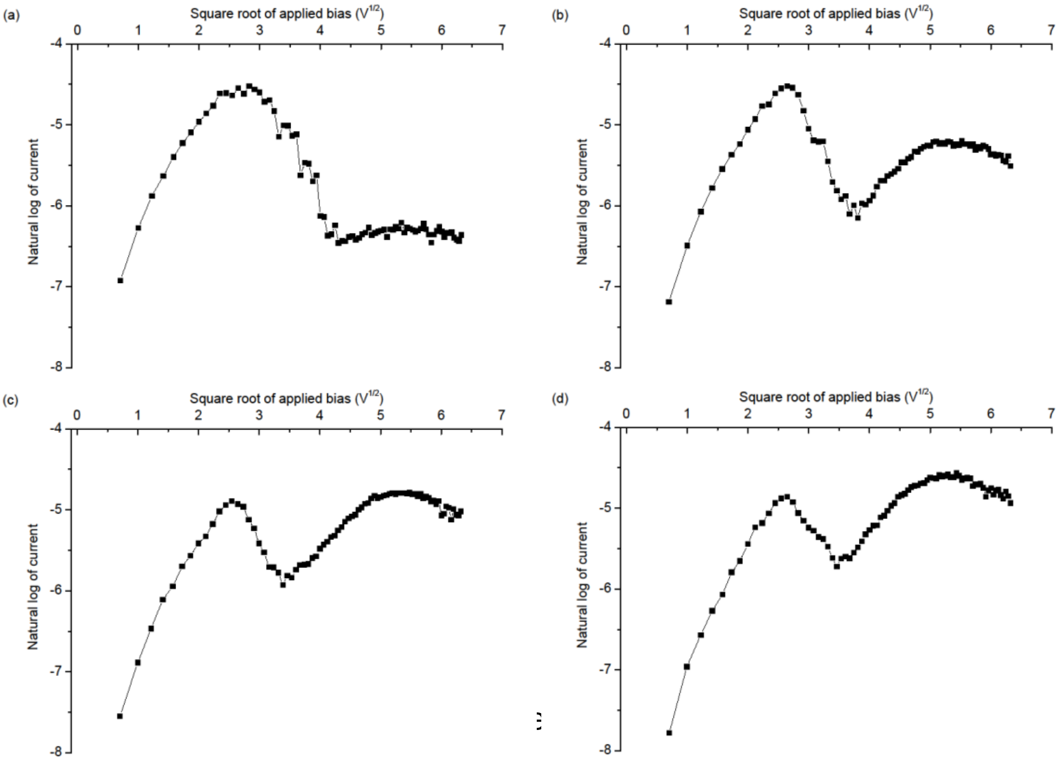


Figure 4.16: Graphs showing the current-voltage characteristics of the pill form of the composite at (a) 0.17% (b) 8.2% (c) 14.9% and (d) 25.5% compression to highlight the contribution of the Poole-Frenkel effect.

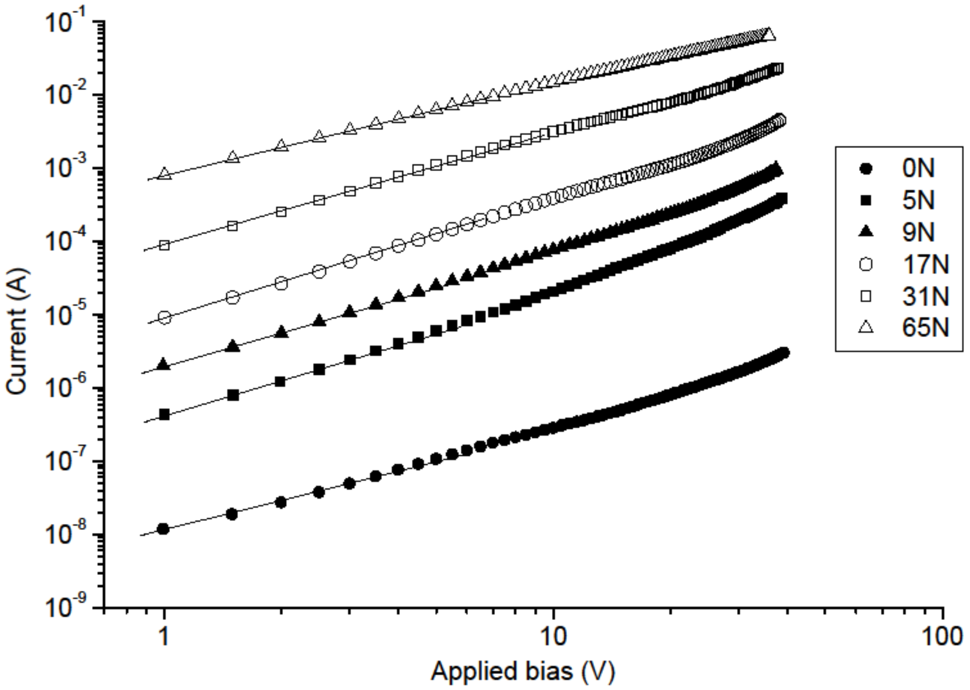


Figure 4.17: Log-log plots of the current-voltage characteristics of tactel A8 under six different loads.

The gradients of the linear regions vary with applied force, as highlighted by the plot of the magnitude of the linear gradients against applied force in Figure 4.18.

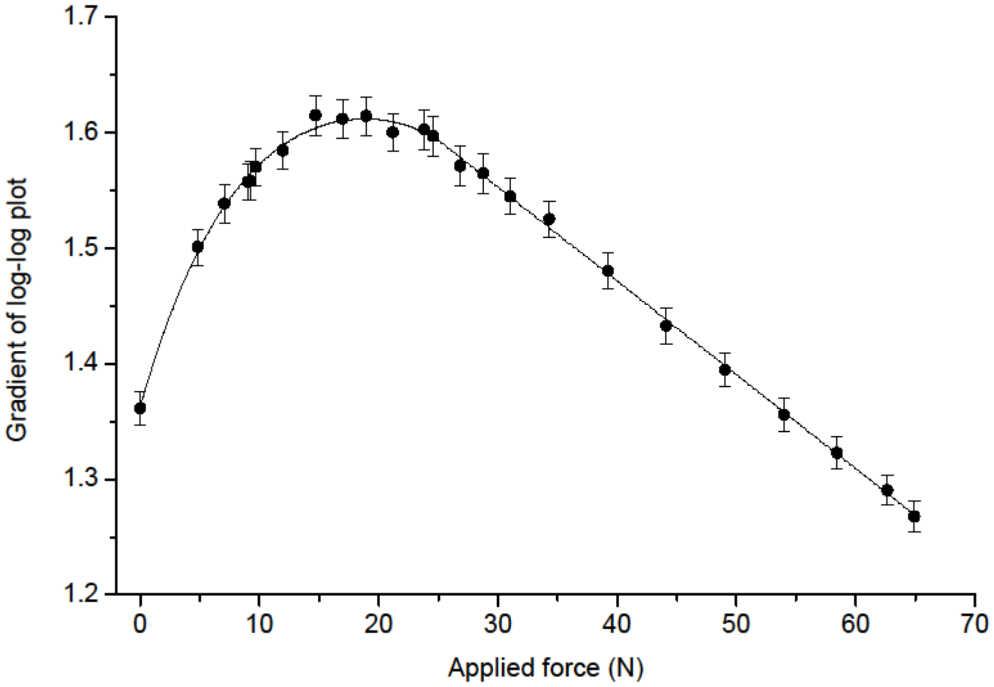


Figure 4.18: Plot to show the relationship between the gradients of the linear regions of the log-log plots of the current-voltage characteristics of tactel A8 and applied force.

The shape of this graph and the absence of any gradients equal to two can be explained by considering the conductivity of the composite as a mixture of space-charge limited conduction via the elastomer and ohmic conduction via the metal particles. This allows the conductivity, $\boldsymbol{\sigma}$, of the material at the loads and voltage ranges to be written as

$$\sigma = A\sigma_{SCL} + B\sigma_{OHM} \tag{4.4}$$

where $\boldsymbol{\sigma_{SCL}}$ is the conductivity of the elastomer, which is space-charge limited when the rate of Poole-Frenkel emission is low, and $\boldsymbol{\sigma_{OHM}}$ is the conductivity arising from ohmic conduction. *A* and *B* are constants that determine the relative contribution of these mechanisms to the gross conduction process [197, 198].

Using this principle, the curve in Figure 4.18 can be explained as resulting from changes in the factors A and B arising from different applied voltages and forces. A lower value for the log-log plot gradient indicates a higher relative contribution of ohmic conduction since $\boldsymbol{\sigma_{OHM}} \propto \boldsymbol{V}$. Similarly, a larger gradient indicates a greater contribution from space-charge limited effects since $\boldsymbol{\sigma_{SCL}} \propto \boldsymbol{V^2}$.

When unloaded, the material is at its thickest and the contribution of conduction via the elastomer will be small. At lower voltages, below the Poole-Frenkel regime, space-charge limited current will also be small because *t* is large enough to reduce the contribution based on equation 4.4. Such a proportionally higher ohmic contribution is a feature typical of space-charge limited processes in polymers [195, 199]. An improvement to equation 4.4 that incorporates this behaviour is described in [195], which gives a current-voltage relationship outside the Poole-Frenkel regime of

$$I = G\frac{V}{t} + H\frac{V^m}{t^3} \qquad (4.5)$$

where *G* and *H* are constants for a given material relating to the ohmic and space-charge limited components respectively. The ohmic term becomes more significant at low biases or at increasing thicknesses of material when the additional factor of $\boldsymbol{1/t^2}$ dominates over the additional factor of *V*. This will typically be at low applied bias and low applied load, explaining the observation of increasing ohmic conduction at low loads. This effect was also observed by Bloor *et al*. [6]. As low loads are applied, thinning of the polymer causes space-charge forming within it to have a greater influence on the injection of further carriers from the conducting particles. Consequently, space-charge limited conduction becomes increasingly dominant via the $\boldsymbol{t^3}$ term.

The density of the space charge is unlikely to be uniformly distributed throughout the elastomer. There will be a greater concentration in the regions adjacent to the metal particles as electrons are readily injected into electron energy levels in the elastomer. This density will fall with distance because of the low carrier mobility within the elastomer. Under low compression, the metal particles are more separated and so the regions of space charge are less likely to overlap and at high electric fields, the Poole-Frenkel effect enhances the conductivity of the elastomer, as indicated in the results shown above. As the load increases, the separation of the metallic particles decreases, meaning that the regions of space charge are more likely to overlap. Under these circumstances, therefore, the space charge will dominate the conduction process in the elastomer, explaining the reduction in the observed influence of the Poole-Frenkel effect. Since conduction is a process of opportunity, current will flow preferentially via the metallic particles, meaning that a pure space-charge limited regime is never achieved. As force is applied and the polymer thins further, the effective volume occupied by the metallic particles increases, with a commensurate increase in the contribution of the ohmic conduction mechanism. This is also supported by the observations of Bloor *et al*. [6].

The quasi-static conditions of the experiments used to produce this data do not permit measurements of transient effects within the material. The model proposed is nonetheless consistent with the negative resistance regime observed in these composites after rapid removal of load [5, 6]. The negative resistance derives from the finite time taken for the space-charge in the elastomer to decay under the model proposed here.

The above analysis does, however, leave open the question of how charge is injected from either the metal particles or the electrodes into the elastomer. Physically, either Schottky emission or tunnelling could perform this task and both of these would be enhanced by high electrical field concentration at the spikes reported by Bloor *et al*. The high tunnelling probability associated with these large local electric fields may enable carriers to tunnel short distances to localised states within the polymer. Crucially, this observation explains the results of Bloor *et al*. that indicated that mixing the polymer in such a way as to eliminate the spikes on the nickel produced a composite that was much less sensitive to compression. The difference is that tunnelling offers a means for injection into the polymer that is reduced by the reduction in the intensity of local electric fields, but is not the rate-limiting mechanism of the overall conductivity of the composite.

4.3.3. Conclusions and Discussion

The measurements and analysis of the current-voltage characteristics of the composite have shown that its electrical conduction is not dominated by a single process. In particular, the presumption that the material is dominated by Fowler-Nordheim tunnelling has been shown to be incorrect in the pre-switching regime. With low bias and low applied force, space-charge limited conduction dominates, whilst increasing bias allows thermal emission of charge carriers trapped in localised states to be enhanced, meaning the material is dominated by the Poole-Frenkel effect. As the force increases, the composite structure becomes thinner and so the Poole-Frenkel effect is displaced in favour of space-charge limited conduction as the space charge begins to occupy a greater fraction of the insulator volume. Further increases in compression cause the effective volume of the metallic particles to increase, giving rise to increased ohmic behaviour. Consequently, the behaviour of the composite is characterised by a mixture of mechanisms whose relative contribution is dependent on the applied load and voltage. This fits well with the model proposed by Radhakrishnan [198] for the effect of fillers on electrical conduction in polymers.

Tunnelling is not, therefore, the dominant conduction mechanism, but alongside Schottky emission is a candidate for the initial injection of charge into the localised states of the elastomer. Enhanced fields from spikes on the nickel particles facilitate this process, and this explains the observation of Bloor *et al*. with regard to spike-destroying methods of mixing the composite materials and the consequential reduction in sensitivity. The rate-limiting mechanisms, however, will be the slower process of ohmic, space-charge limited and Poole-Frenkel enhanced conduction, explaining why the signatures of these processes are found in the current-voltage characteristics of the material.

4.4. Conclusions

This chapter has described the design of a tactile sensor based on a metal-polymer composite. The sensor satisfies a number of the data-centric requirements established in Chapter 2. The transduction method is very simple, based on a changing resistance with applied force. Nonetheless, variation between the tactels means that precise determination of the value of applied force requires calibration. This is complicated by the dynamic behaviour of the tactels, where the sensor can take half a second to recover from load and 0.1 seconds to reach its loaded resistance. This means that at any given time, the resistance may not map to an exact load and so the sensor is probably unsuitable for gripping applications where rapid control is necessary. These are not significant problems for deployment over a wide body skin designed to detect the presence and location of obstacles in the environment, however, and it is possible that changes to the stiffness of the elastomer component of the composite would beneficially adapt this design to these other applications where rapid recovery is required.

Addressing the array is also performed in a common and complexity-reducing manner. The design of the sensor is such that the tactels are essentially isolated from their neighbours, allowing the position of contacts to be ascertained using only knowledge of the switching configuration and the deployment of the sensor during installation. If it is necessary for the centroid of the impact to be determined, then data from adjacently impacted cells must be input to a weighted function in the same way described by Son *et al*. [75]. Otherwise, the tactels provide a measure of object extent that is limited only by the resolution of the sensor array.

In this design, the sensor resolution is very low compared to previous work. This is deliberate, since lower sensor resolutions offer benefits in terms of the quantity of data that must be processed. One structural disadvantage to this that was noticed in these experiments is the effect of the matrix conforming to a surface with a small radius of curvature. This causes a high strain to occur in the tactels and so lowers the unloaded resistance, increasing the consumed power and so diminishing one of the key benefits of the active material from a data-centric perspective. This was only a significant problem at extremely high curvature, and it would be of interest in future work to examine the effect of increasing the resolution, inasmuch as the lateral strain induced by curvature may not be as significant. For lower curvature, however, much of the strain was absorbed by the ability of the gaps between the QTC strips to naturally flex and consequently this design, if appropriately scaled to meet specific application needs, may find significant use for a wide range of deployment scenarios.

This flexibility is also rendered possible by the fabric electrode, comprising bobbinet which is innately strong, flexible and durable. Although it has some problems, such as an inability to stretch in the deployed form, this may be mitigated by future development using elastane within the bobbinet. Crucially, however, the bobbinet affords desirable mechanical properties to the sensor whilst acting as a good electrode that appears to have no impact on the electrical performance of the sensor. The scope for exploiting the ability to cut bobbinet without affecting these mechanical properties offers the tantalising prospect of a cut-and-paste tactile skin using a simple transduction method.

In the context of robotics, the results presented in regards to the electrical characteristics of the composite may seem relatively unrelated. Certainly, in the data-centric approach, the sensor is essentially a "black box" and its internal workings are consequently irrelevant. This data-driven approach does not, however, preclude the need for further development of composites and other transducers to be used within arrays of this type. Consequently, the conclusion that the dominant conduction mechanism for these composites is not Fowler-Nordheim tunnelling, but instead a combination of ohmic, space-charge limited and Poole-Frenkel effects is significant. By using these models, further investigations and understanding of the behaviour of these composites will facilitate the development, customisation and improvement of low-power, high-sensitivity tactile sensing using a resistance-based methodology.

However, this sensor cannot satisfy the data-centric specification on its own, since it has no innate means of processing or simplifying its data. What has been observed is that in practice it is possible to use thresholds to define different but discrete levels of force rather than try to isolate a specific magnitude. This is particularly necessary at low levels of applied force where the variance of the sensor readings is relatively high and so an inverse mapping based on a calibration curve may produce misleading results. Since the properties of the sensor appear to preclude its use in high-response scenarios such as dynamic gripping applications, this is not a hindrance at all, and indeed provides scope for reducing the amount of data that must be sent back to the central controller.

Over the course of the next two chapters, a data structure will be developed to accommodate the data from an array sensor such as the one described in this chapter in the most efficient and compact way so as to satisfy the remaining principles of the data-centric specification that this work is applying to tactile systems. Following this theoretical development, the data structure will be deployed in a system that uses the sensor described here and demonstrate how maximum information can be extracted from a low-resolution system using a compact, data-efficient computer process.

Chapter 5

Bitworld: A Bit-Based Representation of Tactile Data

5.1. Introduction

As has been shown in previous chapters, tactile sensors are often best arranged as a two-dimensional grid for efficiency. Similarly, computer-based images are effectively two-dimensional data structures where each pixel contains information about the image at a given point. Robots operate within three-dimensional environments so tactile sensors may be curved around three-dimensional shapes and so be imaging a three-dimensional environment. However, a system is sought that may be generalised to any robot deploying tactile sensors and it is simpler to form two-dimensional images from the sensor outputs and transmit these to the robot's control mechanisms. Responsibility for converting these images into three-dimensions can then be delegated to the part of the robot system that has an innate understanding of the deployment of the sensors.

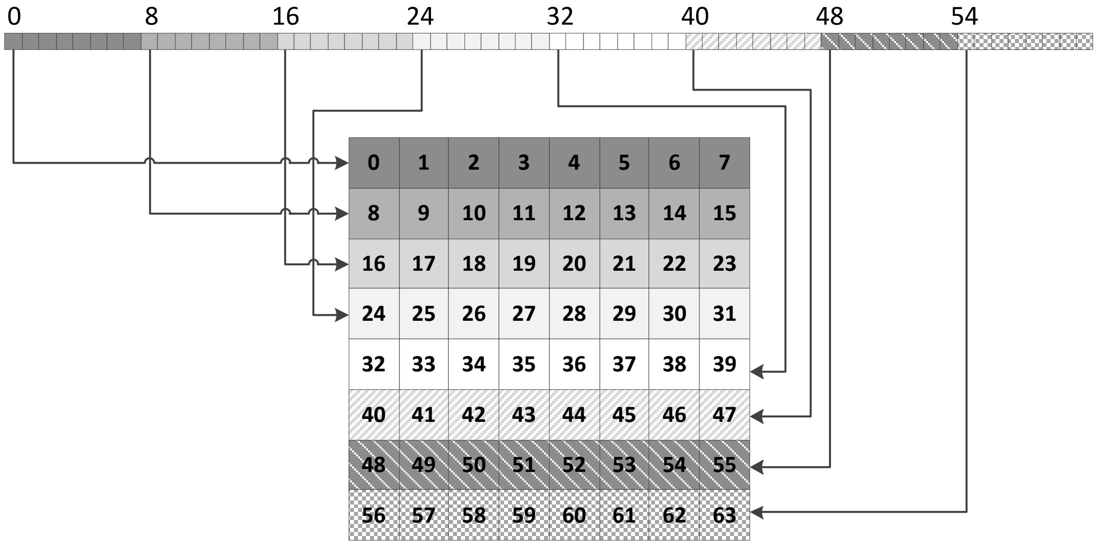
The data-centric approach to tactile sensing established in Chapter 2 was motivated in part by the high volume of tactile data that can be generated by a tactile system. High data volumes can promote higher computational costs from bandwidth and processing requirements. This cost can be mitigated by choosing a representation for the data that is compact and easily processed, reducing the burden on a system in which a tactile component is deployed.

This chapter presents a grid-based representation of tactile images that uses only one bit per cell. The principles of the approach are described, including an exploration of how such images may be manipulated. The performance benefits of this representation relative to one that is more common but less compact are then established.

5.2. Description of a Compact Representation

5.2.1 Bit-Based Spatial Representations

The representation described in this chapter utilises computer bit-strings to encode tactile images such that a 64-bit computer word is rearranged into a grid. An example of such a rearrangement is shown in Figure 5.1, where eight bit-strings of eight bits contained within a 64-bit word represent the different rows of a square grid. This form of representation has found widespread use in computerised board games, particularly chess, and have come to be known as bitboards [200]. In such games, the positions of the various types of piece are encoded as one-bits within a number of bitboards employing the same layout as shown in Figure 5.1.



0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Figure 5.1: Illustration of how a binary computer word can be used to represent a two-dimensional grid

Other bitboards can be generated that encode other game-related information, such as the valid moves of pieces on the board [200]. Figure 5.2 illustrates how two bitboards, one containing the position of the black king and one containing the possible moves by a white bishop can be combined to infer information about the game's state by applying a Boolean-AND operation on a bit-by-bit basis.

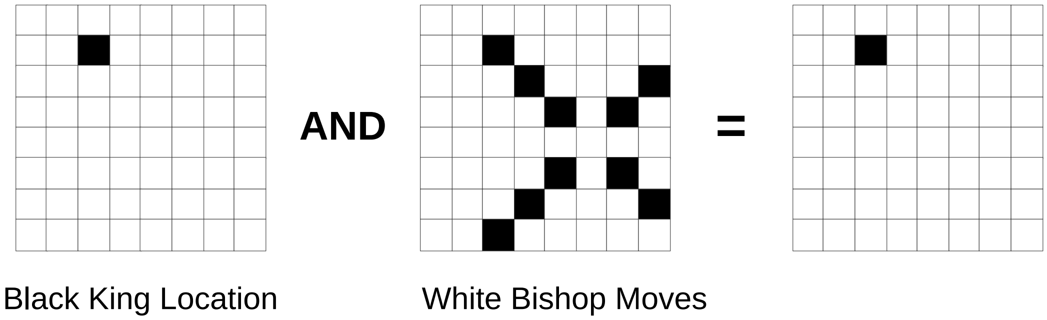


Figure 5.2: Illustration of how bitboards can be used to answer questions about the state of a board.

In this case the combination determines that the white king is in check by a white bishop, because the result of the combination is non-zero. Using this bit-based representation is efficient because combinations such as the one illustrated in Figure 5.2 can be made using bitwise operations. These are Boolean operations performed on a per-bit basis between the corresponding bits of two computer words, but the operator is applied to all bits simultaneously.

This affords an inherent parallelism that is much more efficient than if the board were encoded as a two-dimensional array of integers that had to be cycled over for each update or query [200, 201]. These techniques have also shown promise in other domains such as encoding complex problems in graph theory [201, 202]. In a more relevant application, bit-parallelism has also been exploited to benefit spatial representations for robotics, where a graph-based topological approach was employed [203]. This processing efficiency is even clearer when considered in terms of CPU clock cycles. A bitwise operation can be achieved in a single cycle and on modern processors will act on 64 bits simultaneously.

The operations can be made up of the Boolean operators AND, OR, NOT and XOR. For each of these required to update or evaluate a cell in bitworld, only an average of 1/64 CPU cycles are required per cell, compared to 1 for an integer-based operation or a minimum of between 3 and 6 for a floating-point one [204, 205]. In addition to these processing efficiencies, the bit-based grid is a compact representation, requiring one-eighth to one-sixteenth of the memory needed by equivalent integer representations. It should be noted that these efficiencies are only significant once several grid cells are being stored, and if trying to represent an environment of less than eight cells, an integer representation may be more compact. This is an unlikely edge case in the context of tactile data. In terms of transmission bandwidth and memory storage, this representation offers significant benefits in a tactile data system that produces a high volume of data.

The representation and transmission of high volumes of tactile data has been identified as an outstanding problem in the field. These problems can be mitigated by using a compact data structure similar to bitboards, reducing the size of the data transmitted for each tactel. It is proposed here in the first instance to represent each cell in a tactile image grid by a single bit.

The possibility of using multiple bits per cell whilst still affording an overall saving over conventional techniques will also be considered. A multiple bit representation can be achieved by using multiple bitboards to represent a single grid, such that each bit position has a corresponding partner in one or more other bitboards. This will not be considered in detail in this chapter.

5.2.2. Bitworld: An extension for tactile data

Naturally, a grid with only 64 cells does not afford sufficient resolution for a significant range of applications and so must be expanded. In order to achieve this, multiple bitboards must be tessellated to produce either a two- or three-dimensional image of the objects and environment that a robot wishes to interact with. Since the resulting collection of bitboards is intended as a component of a robot's wider view of the world, such a collection is hereafter referred to as a *bitworld*.

The bitworld representation imposes significant constraints on the tactile system, because only discrete levels of the stimulus can be represented. In the initial case of a single bit per cell, the tactile sensor is reduced to an on/off switch based on a pre-defined threshold. Even in the case of more than one bit, only a finite number of stimulus levels can be represented. Such thresholds will be dependent on the exact sensor being deployed, as well as the type of event that the robot is expected to recognise.

It has been shown that requiring such thresholds is not uncommon, particularly with force and pressure sensors. The extent to which this is a constraining factor is dependent on the number of bits used to represent each cell, which can be adjusted according to application. So, in a bitworld system, the data from all tactels is subject to conditioning by a function $f(\Phi_T)$ such that

$$f(\Phi_T) = \begin{cases} 00\ldots00, & \Phi_T < t_0 \\ & \vdots \\ 11\ldots10, & \Phi_T < t_{n-1} \\ 11\ldots11, & \Phi_T \geq t_n \end{cases} \tag{5.1}$$

where the t_x are system-defined thresholds. The use of bitworld thus allows the robot's controller to be independent of the implementation of the sensor. This facilitates modularity, since different tactile systems can be plugged in and out of a robotic system, which can ascribe its own meanings to the various cell values.

In some cases it may be desirable to retain the precise value of tactile stimuli as in the case of gripping applications like the NASA/DARPA Robonaut, where the force measured by the sensors is involved in a control loop [52]. In such cases, it may be difficult to replace these controllers to handle the bitworld representation, yet the reduction of bandwidth in the system is still desirable as gripper sensors are often implemented with high density arrays [48].

In this context, a single-bit bitworld can represent a map identifying those tactels being stimulated. This can be transmitted with an ordered list of the values of only the active stimuli. By using a common serialisation process, this list can be combined with the bitworld by the controller to produce a tactile image. To illustrate this, consider the schematic of a 16x16 tactel sensor array in Figure 5.3. An object is depicted impacting the sensor array in Figure 5.3(a), with the affected cells highlighted in grey in Figure 5.3(b).

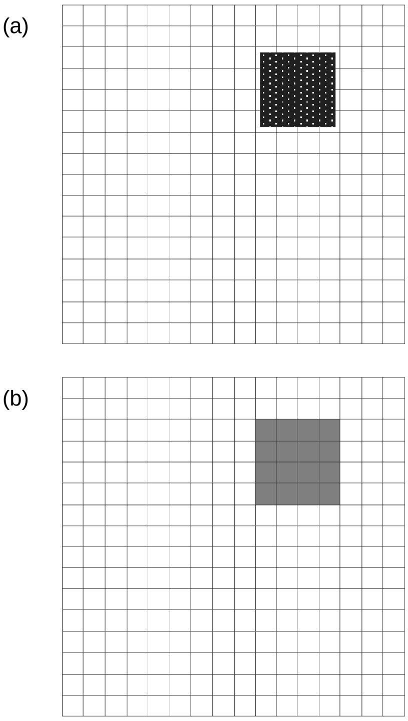


Figure 5.3: Schematics illustrating (a) an object impacting a tactile array, and (b) which tactels are stimulated as a result

In this example, sixteen tactels have been stimulated by an object. The bitworld image of this sensor requires four bitboards with a total size of 32 bytes. Here and throughout, it is assumed that a floating-point value requires 32 bits of storage space or transmission bandwidth. If the data from the stimulated tactels uses such a floating-point representation, this will require a further 64 bytes. In total the example data would require 96 bytes of space for storage or transmission using the bitworld-augmented process. This compares favourably with the 1024 bytes required to store and transmit the data from all tactels as floating-point numbers. However, in the worst-case scenario where all tactels were stimulated, the inclusion of a bitworld structure would require more storage or transmission space than simply transmitting floats.

Let us define *T* as the number of tactels in a sensor array, and *A* as the number of these tactels that are stimulated. The size of a bitworld representing this many tactels would be $8\left\lceil\frac{T}{64}\right\rceil$, where the ceiling operator and division by 64 account for the representation of bitworld using 64-bit integers.

The size of an array of *T* floating-point values would be 4*T*, sufficient to represent the values of all tactels. The size of an array containing only the stimulated tactels is 4*A*. In order for the bitworld-augmented mechanism to represent an improvement over a floating point mechanism, the following inequality must be satisfied:

$$8\left\lceil \frac{T}{64} \right\rceil + 4A < 4T \qquad (5.2)$$

This can be simplified to the following form:

$$A < T - 2\left\lceil \frac{T}{64} \right\rceil \qquad (5.3)$$

The right-hand side of this inequality is the number of tactels that have to be stimulated before the bitworld-augmented scheme becomes less efficient than simply representing the entire array using floating-point values. Figure 5.4 shows a graph plotting this value as a percentage of the total number of tactels.

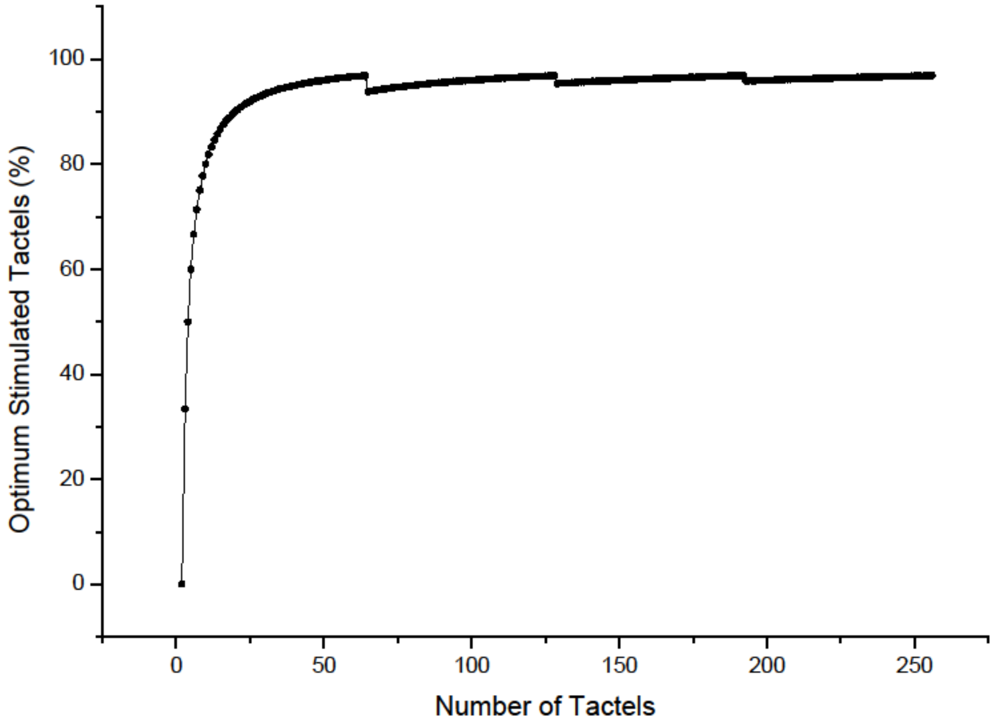


Figure 5.4: Graph showing the percentage of tactels that must be stimulated at a given time for bitworld-augmentation in a system to be less efficient than raw data transfer.

The plot shows that for more than 33 tactels, bitworld augmentation is more compact if fewer than 94% of tactels are stimulated. In most scenarios, the number of stimulated sensors is unlikely regularly to exceed this level. This is because over the duration of operation tactile sensors are most likely to be in an unloaded state as an object is sought, or briefly contacted. In the case of gripping applications, more regular contact with objects may be anticipated. It is unlikely that all tactile sensors in such an application would be sized to the exact dimensions of anticipated objects as this would be an inefficient design that would not be robust when handling larger objects. With a higher resolution for such applications bitworld augmentation may still prove beneficial, but if the condition of equation (5.3) cannot regularly be met during operation then it may not be appropriate.

5.2.3. Interim Conclusions

This section has introduced a bit-based structure called bitworld that potentially offers benefits compared to a floating-point representation in terms of compactness and speed. It should be noted that these benefits will also depend on the mechanisms by which the data can be manipulated and stored with the necessary tessellation.

The benefit to computational performance was demonstrated through reference to the relative instruction latencies of various bitwise, integer and floating-point operations on general-purpose processors. However, this is not evidence of similar relative differences in performance when these operations are combined to manipulate the images in a useful way, although it does motivate further exploration of these differences.

The subsequent sections resolve these issues by describing a real implementation of these representations. The necessary image manipulations are described and then implemented for each representation so that the performance can be compared by profiling. This will assist in exploring the potential benefits of using bitworld, as well as introducing the language used to manipulate it.

5.3. Image Manipulation: Techniques and Performance

5.3.1. Tactile Images and Manipulation

This section explores the general nature of tactile images and the operations that must be applied to them for use in a robotic system. A possible flow of tactile data in a generalised system is illustrated in Figure 5.5 as a reference for this discussion. Tactile data may come from multiple spatially-distributed sources, as highlighted by the segmentation into separate groups in Figure 5.5. Tactile data does not generally have an innate sense of location and localising the sensors is instead dependent on two pieces of knowledge; the deployment of sensors on the robot and the current pose of the robot.

The former is easily known as part of the assembly of the robot and can be programmed offline. The latter is not generally known to the robot and must be measured. Since systems in such environments are candidates for the deployment of tactile sensors, a measured pose can often be anticipated.

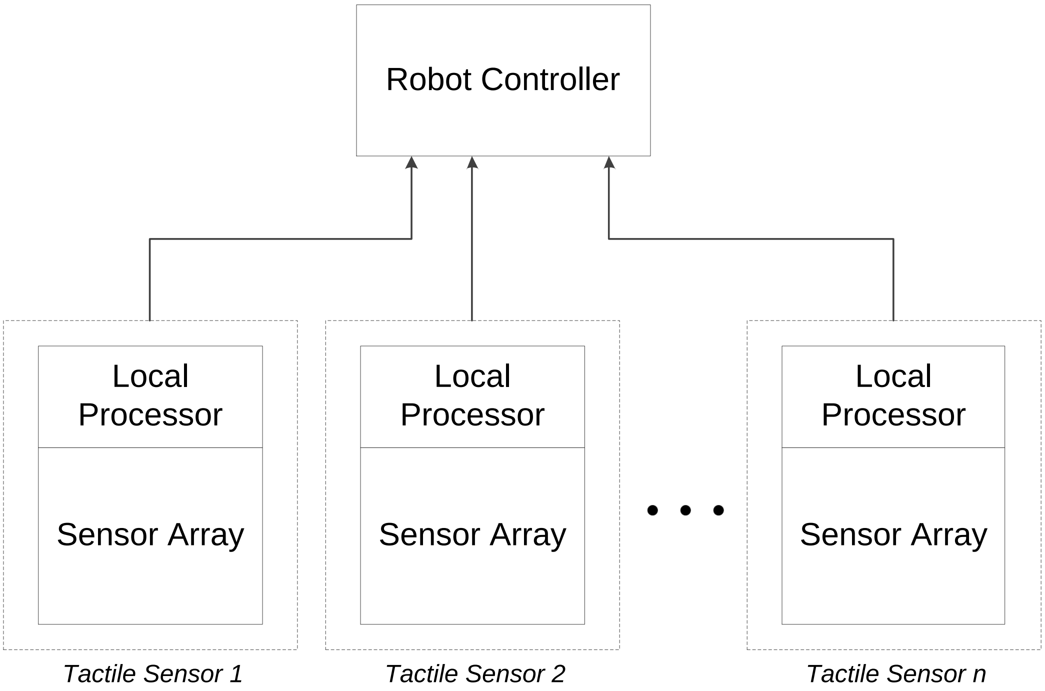


Figure 5.5: Schematic illustration of the general flow of data in a tactile system.

It is desirable to convert the image data to some common coordinate axis so that it can be placed in context with other measurements of the environment, rather than locations that are relative to the local origin of the sensors. This transformation will be dependent on the measured pose of a robot, such as the one illustrated in Figure 5.6. This illustration is of a snake-arm robot, a hyper-redundant manipulator designed specifically for automation in confined spaces [206]. As such, it is a useful example of the deployment of tactile sensors that would be extremely useful in such scenarios.

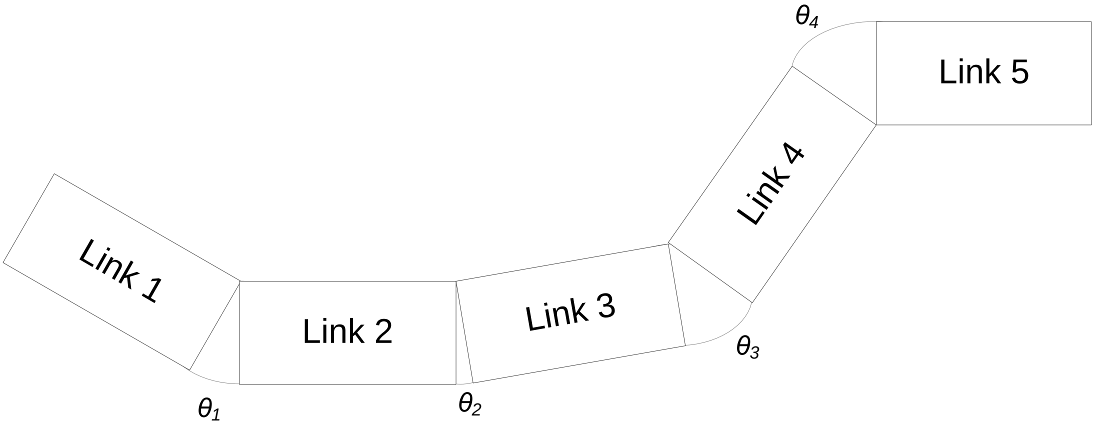


Figure 5.6: Schematic of a snake-arm robot in an arbitrary pose. Each of the links is assumed to be covered in tactile sensors.

The pose of a robot is a measurement consisting of all the quantities such as angles and translations that define the position of the links. If the data from tactile sensors is returned to the robot's controller along with a measurement of pose, detected features can be mapped into the robot's chosen frame of reference. In this two-dimensional model, this can be done by successively applying the transformation

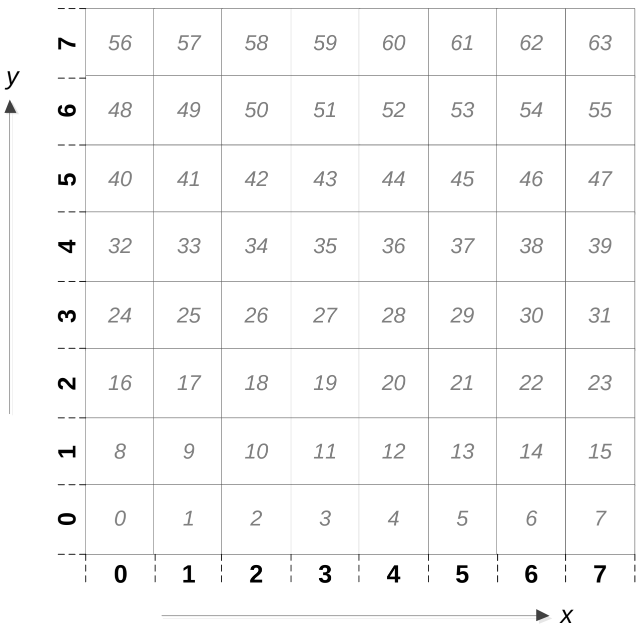
$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta_n & -\sin\theta_n & t_x \\ \sin\theta_n & \cos\theta_n & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{5.4}$$

where θ_n is the angle of link *n*, and t_x and t_y represent the translations required to align the origin of the image appropriately with the previous one. In order to function adequately on tactile data, a representation must allow a robotic system to efficiently manipulate images through rotation and translation according to measured poses of the robot.

The next sub-section describes the notation of the bitworld representation. This will facilitate understanding of the subsequent sections, which describe the means of implementing these necessary image manipulations and comparing their performance to an alternative representation based on floating-point numbers.

5.3.2. Bitworld Notation

This sub-section describes the notation of the bitworld representation. As previously described, the main mechanism by which instances of this representation can be manipulated is through bitwise operations that can act on all the bits of a computer word simultaneously. These are composed of the Boolean operations *AND*, *OR*, *NOT* and *XOR* and all of these apart from *NOT* operate on two words. Each of these operators has a particular impact on the resulting images. Determining the effect of particular operators may be easier to determine by defining bitworld in a set notation, as performed by San Segundo *et al*. in their application domain [201]. For this purpose, the fact that an individual bit in the bitworld representation is actually representative of a spatial location will be exploited.



7	56	57	58	59	60	61	62	63
6	48	49	50	51	52	53	54	55
5	40	41	42	43	44	45	46	47
4	32	33	34	35	36	37	38	39
3	24	25	26	27	28	29	30	31
2	16	17	18	19	20	21	22	23
1	8	9	10	11	12	13	14	15
0	0	1	2	3	4	5	6	7
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**

Figure 5.7: Illustration of applying a Cartesian system to a 64-bit bitboard, with the coordinates running along the orthogonal edges. The bit positions are indicated in grey for convenience.

As shown in Figure 5.7 for a board of 64 bits, a Cartesian index can be applied to a single bitboard. In order to extend this coordinate system to a bitworld, which is a tessellated collection of bitboards, the bitboards must be given a spatial index relative to a bitboard chosen to contain the origin. This index can be chosen on the basis of tessellation, so that all bitboards receive their own *X*- and *Y*-coordinates on different scales to the system shown in Figure 5.7.

As bitworld expands, new bitboards added to extend the knowledge in the image must be allocated indices to describe their relative location in the tessellation. One such indexing scheme is shown in Figure 5.8, where the central bitboard is considered the origin and the surrounding bitboards have been allocated relative indices.

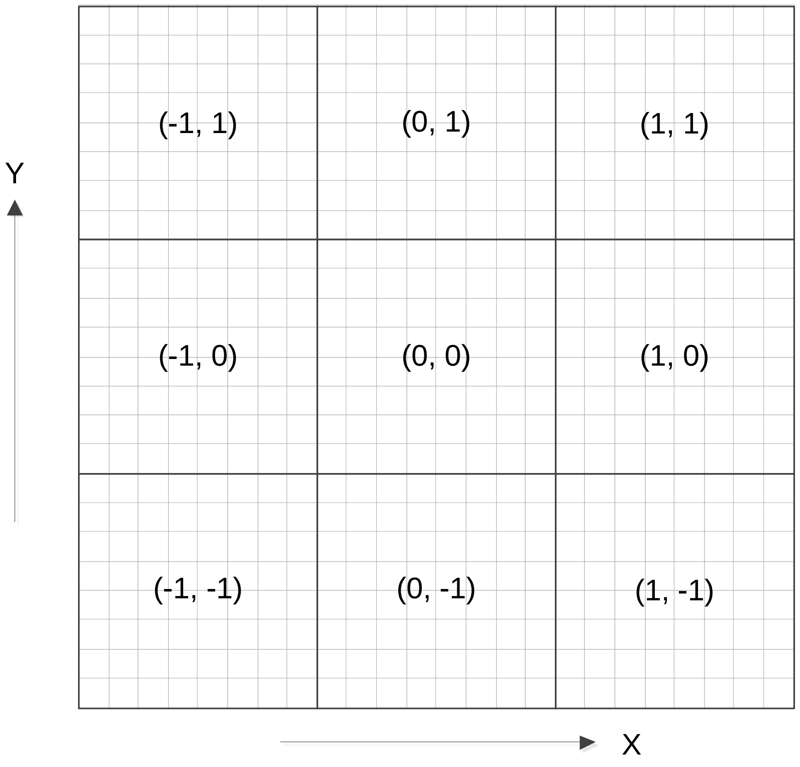


Figure 5.8: Two-dimensional bitboard indexing scheme. Each of the (x, y) indices references a single bitboard. Multiple bitboards can then be tessellated to form a bitworld.

It would be undesirable for the robot controller or some other aspect of the higher robotic system to have to concern itself with the internal indexing scheme of bitworld. Instead, it is preferable for a global coordinate system to be employed, centred on a chosen origin, but without having to be aware of the borders between bitboards. Fortunately, the schemes of Figures 5.7 and 5.8 allow this to be done. Given a global coordinate (*a*, *b*) the internal *x*- and *y*- coordinates of the correct bitboard are given by

$$x = a \bmod 8 \tag{5.5}$$

$$y = b \bmod 8 \tag{5.6}$$

respectively, where 'mod' represents division modulo 8. The *X*- and *Y*-indices of the bitboard can be identified using the relations in equations (5.7) and (5.8).

$$X = \frac{a - x}{8} \tag{5.7}$$

$$Y = \frac{b - y}{8} \tag{5.8}$$

Finally, if necessary to locate the actual bit within the word representing the bitboard, the location b_{xy} of this bit is simply found using the following formula:

$$b_{xy} = 8y + x \tag{5.9}$$

Having developed a coordinate system for bitworld, it is possible to redefine it in terms of set notation. Every bitworld simply becomes a set of the coordinates that are occupied, or set to one, in the image. For the purposes of examining the effects of the bitwise operators, it will be convenient to define two bitworlds B_1 and B_2 in the following way:

$$B_1 = \{(x, y) \in \mathbb{Z}^2 | (x, y) \text{ is occupied in image 1}\} \tag{5.10}$$

$$B_2 = \{(x, y) \in \mathbb{Z}^2 | (x, y) \text{ is occupied in image 2}\} \tag{5.11}$$

The bitwise operators can now be represented as set operations, as shown in Table 5.1.

Bitwise operator	***Pseudo-code Equivalent***	***Set equivalent***
B_1 AND B_2	$B_1 \,\&\, B_2$	$B_1 \cap B_2$
B_1 OR B_2	$B_1 \mid B_2$	$B_1 \cup B_2$
NOT B_1	$\sim B_1$	$(B_1)^{\mathrm{C}}$
B_1 XOR B_2	B_1 ^ B_2	$(B_1 \cup B_2) - (B_1 \cap B_2)$

Table 5.1: Table showing the bitwise operators and their set-wise and C-like pseudocode equivalents.

The *AND*-operator represents the intersection of two images, so that the result only has one-bits at coordinates common to both input images. In isolation, this operator would be helpful in removing erroneous bits in the global image that resulted from noise or other defects in the original image. This will be referred to as a destructive operation, as the output generally has less information than its inputs. By contrast, the *OR*-operator represents the union of two images and so will be referred to as a constructive operation. All the one-bits from both images affected by the operation appear in the result. The *NOT*-operator simply performs an inversion of its input image, whilst the *XOR* returns all of the bits that would be lost by an *AND*-operation. The utility of the *NOT* operator will be demonstrated in Chapter 7.

These operations are useful for image assignment, since in combination they allow the values of the cells in two images to be combined and the result assigned to an output image. They are not sufficient to align these images in the manner described in Section 5.3.1. This is because they only operate on bits that lie in the same position on a bitboard and are unable to directly reference the bit values in other positions. As a result, they cannot be used to move values from one part of a bitworld to another.

In order to achieve this, an additional operation is required that is commonly used to manipulate computer words. This technique is called bit-shifting, which moves the values of bits in a word along by a specified number of steps. There are two pertinent types, which are logical and circular shifts.

Both are typically denoted in code by the symbols >> or <<, depending on the direction of the shift. These have slightly different behaviours, as illustrated in Figure 5.9. The difference is that the circular shift retains all the bit values by cycling the overflowing bits onto the end of the word.

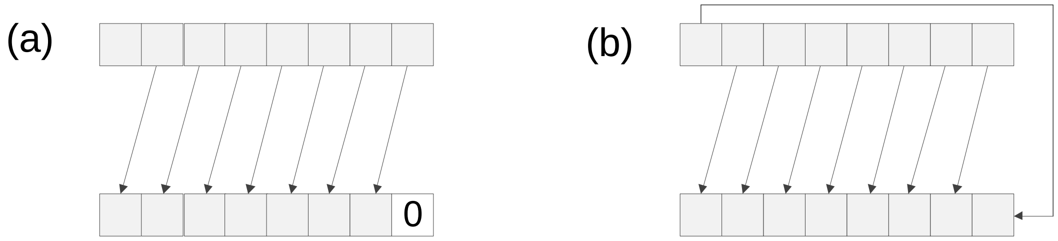


Figure 5.9: Illustration of (a) logical shifting, and (b) circular shifting applied to an 8-bit integer

In the context of bitworld, a shift can be seen as moving all the pixel values by a fixed number of steps. Correctly applied, this operation could be used in combination with others to effect translation of a bitworld and so align images to a common axis. Applying this to bitworld is not simple because the bits forming the rows and columns of bitworld are not contiguous. Figure 5.10 illustrates the effect of circular and logical shifting on a bitboard using a single left shift.

(a)

56	57	58	59	60	61	62	63
48	49	50	51	52	53	54	55
40	41	42	43	44	45	46	47
32	33	34	35	36	37	38	39
24	25	26	27	28	29	30	31
16	17	18	19	20	21	22	23
8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7

(b)

57	58	59	60	61	62	63	
49	50	51	52	53	54	55	56
41	42	43	44	45	46	47	48
33	34	35	36	37	38	39	40
25	26	27	28	29	30	31	32
17	18	19	20	21	22	23	24
9	10	11	12	13	14	15	16
1	2	3	4	5	6	7	8

(c)

57	58	59	60	61	62	63	0
49	50	51	52	53	54	55	56
41	42	43	44	45	46	47	48
33	34	35	36	37	38	39	40
25	26	27	28	29	30	31	32
17	18	19	20	21	22	23	24
9	10	11	12	13	14	15	16
1	2	3	4	5	6	7	8

Figure 5.10: Effect one-bit left shift operation on bitworld (a) using (b) logical shifting and (c) circular shifting.

The effect of both kinds of shifting is to move most of the columns one space to the left, effecting an x-axis translation that will be useful in aligning images. However, only the first seven columns in the results are in the correct location. The rightmost column contains data that was originally in the leftmost column. The correct behaviour would be for this column to become empty or to be filled with values from the leftmost column of the next bitboard to the right, if one exists. The solution to this will require this last column to be either emptied or overwritten. Since this column holds the only difference between the result of logical and circular shifting, either type of shift is acceptable in the context of bitworld so no differentiation is necessary.

Naïve application of shift operations does not produce the desired behaviour for bitworld, as highlighted by Figure 5.10. Fortunately, this can be corrected using the other bitwise operators and two bitboard masks. A mask is simply a pre-calculated bitboard with certain bits set to one in advance. These constant bitboards allow certain groups of bitboard cells to be quickly manipulated in isolation using a single bitwise operation. The two masks that will be used to implement the left- and right-shift functions are illustrated in Figure 5.11, and simply isolate the leftmost and rightmost columns in a bitboard.

1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0

M_{left}

0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1

M_{right}

Figure 5.11: Illustration of the two masks required to implement *x*-axis shifts in bitworld.

To perform a one cell left-shift using these tools can now be performed by utilising the following sequence of operations:

1. For each bitboard B_i, evaluate and store $B_i^m = B_i \,\&\, M_{left}$. These are the bits that will be translated to the rightmost side of the next bitboard.

2. Right-shift all B_i^m by 7 to move the masked bits to the right hand side. This adjusts the previously calculated boards to align their values for combining with the next bitboard across.

3. Left shift all bitboard B_i by 1. This performs the naïve shift shown in Figure 5.10 to move most of the values within each bitboard to the desired location.

4. Clear the rightmost column in each shifted bitboard B_i by setting $B_i = B_i \,\&\, {\sim}M_{right}$. This results in a bitboard in the form of Figure 5.10(c) with the rightmost column ready to receive new data.

5. For each bitboard B_i, apply the operation $B_i = B_i \mid B_{i+1}^m$ to complete the move of the leftmost column of bits from the adjacent bitboard into the rightmost column.

A similar method using different masks can be employed for shifting in any of the four possible directions. This seems cumbersome for a relatively simple alteration to a bitworld, with a complexity rising linearly with the number of bitboards. Some simplifications are possible, such as merging the fourth and fifth steps into a single update and minimising the number of individual loops. In order to shift bitworld *N* cells to the left, it would be necessary either to run the sequence *N* times or to have optimised functions specialising in performing any required number of shifts. In this work, the former approach will be adopted for simplicity although the latter might prove more efficient in some circumstances.

This implementation of a shift operator highlights a number of positive and negative features about the mechanisms used to manipulate bitworld. A benefit is the ease by which bitwise operators can be deployed to emulate more complex changes in the environment. Relatively few operators are needed to update each bitboard, and the use of masks highlights how a relatively simple operation can be used to select portions of bitworld that are of interest. The drawbacks highlighted are the inability to easily isolate and manipulate individual bits. This problem is accentuated when manipulations involve shifting data across the borders of bitboards. This provokes a general caution that operations that require individual bit manipulation need to be avoided where possible. Although this can be applied to other data representations such as occupancy maps, previous work using bitboards cautions that the performance benefits of the bit-based representation can be lost under these circumstances because there is limited scope for parallelism [201].

This section has developed the language of bitworld in terms of bitwise operations and a shift operator. It has also highlighted some of the constraints that this imposes. The necessity for bitworld to facilitate the translation of images was highlighted in the previous section and the shift operator has been shown in combination with the other bitwise operators to achieve this effect. Although the process appears quite involved, the performance of these techniques compared to their floating-point counterparts must be explored methodically for a valid comparison to be made. This is the subject of the remainder of this section, beginning with an overview of the implementation of these representations in code.

5.3.3. Methodology and Performance Metrics

In order to provide a relative measure of performance for bitworld, two C++ classes were developed. The listings for these classes and the implemented methods can be found in Appendix 5.1. One of these, called *Bitworld,* implements bit-based techniques as described in previous sections and the other, called *MatrixWorld*, implements a matrix of floating-point values. Both classes impose the same limitations on the images they contain to allow a fair comparison.

Both classes implemented the representations as having immutable dimensions. Since these images relate to tactile sensors, the size of the image can be known in advance. Transformations such as rotations can result in larger final images, but the maximum extents of these can also be known in advance. Consequently, the immutability of dimensions does not represent an onerous restriction and makes the performance comparison less dependent on the efficiency of memory allocation.

Both classes used standard libraries with no special optimisations, processor-specific instructions or third-party libraries to enhance either implementation. This was in order to provide a more generic comparison of performance, rather than one that was dependent on the characteristics of code specifically optimised for the task under test.

Both classes used containers from Standard Template Library (STL) to hold their data. The STL is an ISO/IEC standardised software library [207] that contains implementations of many data containers that are useful for writing computer programs. The STL 'map' is an associative array that associates a key with data. This container was used to implement *Bitworld*, using a key that consisted of the *XY*-indices described previously in order to reference a specific bitboard.

The *MatrixWorld* was implemented using the STL ‘deque’, which is simply a dynamic double-ended array of values. The deque was chosen in preference to the single-ended STL ‘vector’ because the double-ended nature of the deque makes shifting operations equally efficient in both directions. By storing a deque of deques, a two-dimensional representation could be formed without additional complexity. *MatrixWorld* was also implemented as a template class, which enabled an easy choice of data type for the representation without structurally altering the code. As this choice is implemented at compile-time, it has no impact on performance but allowed the use of integers and floating-point values to be compared.

Description of the precise implementation of the different manipulation techniques is deferred to subsequent sections, where this will be done in conjunction with the results. The method for quantifying the computational performance of these techniques is the same is all cases. The chosen metric is the time taken for the relevant blocks of code to run, averaged over one million runs. The averaging should mitigate the impact of unavoidable effects such as hardware interrupts that might otherwise disturb an individual run time measurement. The timing system was encapsulated in the *StopWatch* class, the listing of which is shown in Appendix 5.2 using, using standard techniques for accessing the high-performance counters available in computer hardware.

All code was written and compiled in Microsoft® Visual C++ 9.0 Professional Edition using a release build with the compilation options set to favour fast code and to maximise speed. The benchmarking was performed on a desktop computer with a 4-core Intel® Core i7 2.67GHz, 4GB of RAM and using the Windows XP x64 operating system. Although this processor is multi-core, the programs were not designed to exploit this effect and so will only have used one core at any given time.

The remaining sub-sections describe how the different image manipulation techniques are implemented in the two representations and present the results of the performance profiling.

5.3.4. Image Translation

The translation of images in the *Bitworld* implementation was performed using the technique described in Section 5.3.2 using pre-defined masks, shifting and bitwise operations. Two techniques were implemented for the *MatrixWorld* class and these were profiled separately. The first technique exploited the double-ended nature of deques to insert blank cells at one end whilst removing cells at the other. This is implemented simply using the pop- and push-functions available in the STL library and is referred to as the pop/push method.

The second technique was more rudimentary and addressed each individual cell in an appropriate order to assign an appropriate value to it. This technique is called the scanning method. This technique could exploit foreknowledge of the magnitude of the translation, using this to minimise the number of operations required for an image of constrained dimensions. This works because with fixed dimensions, an increasing magnitude of translation reduces the amount of data retained in the image and this image simply needs to be moved. There is also a requirement to zero all the remaining cells, but it is possible to do this in the same loop by setting all pixels containing data that is moved to zero before the loop continues. Profiling should indicate whether this represents an improvement over the simpler pop/push method.

These methods retain the size of the image so that data shifted beyond the image boundaries is lost. This arises from the immutability of the image dimensions specified previously, but this isolates the performance of the implementation rather than being concerned with the performance of dynamic memory allocation performed within the compiler. The performance of these techniques will be dependent on the scale of the translations since a greater number of operations are required for a greater number of shifts. An image of 256x16 cells in dimension was used for profiling, varying the scale of translation in the *x*-direction from 0 to 255 cells.

In order to properly compare the two types of representation, it is necessary to select the best implementation for the *MatrixWorld* class from the two described. The results of profiling these techniques are shown in Figure 5.12.

The data show that the time required for the two techniques follows a broadly linear trend. The pop/push method performs best for lower magnitude translations, whilst the scanning method works best for those of higher magnitude. This can easily be understood from the fact that the number of pop- and push-actions performed by the pop/push method increases with increasing magnitude of translation, whilst the scanning method must do less work.

Rather than select either method over the other, a hybrid was developed where the size of translation determines which of these two methods is used. This is simply implemented by a conditional statement that tests the magnitude of the translation and selects the appropriate code path. The critical value from the data above is about 116 cells. For values less than this, the pop/push method was used; otherwise the scanning method was employed. This provides the fairest basis for comparison with the *Bitworld* implementation and the results of this comparison are shown in Figure 5.13.

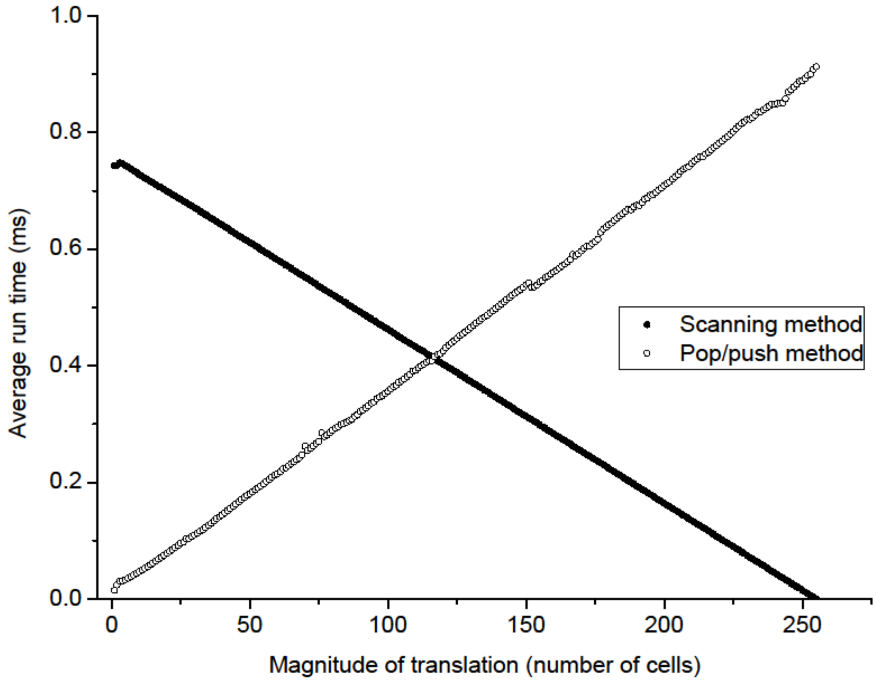


Figure 5.12: Graph comparing two possible techniques for translating the *MatrixWorld* images based on the pop/push and scanning methods.

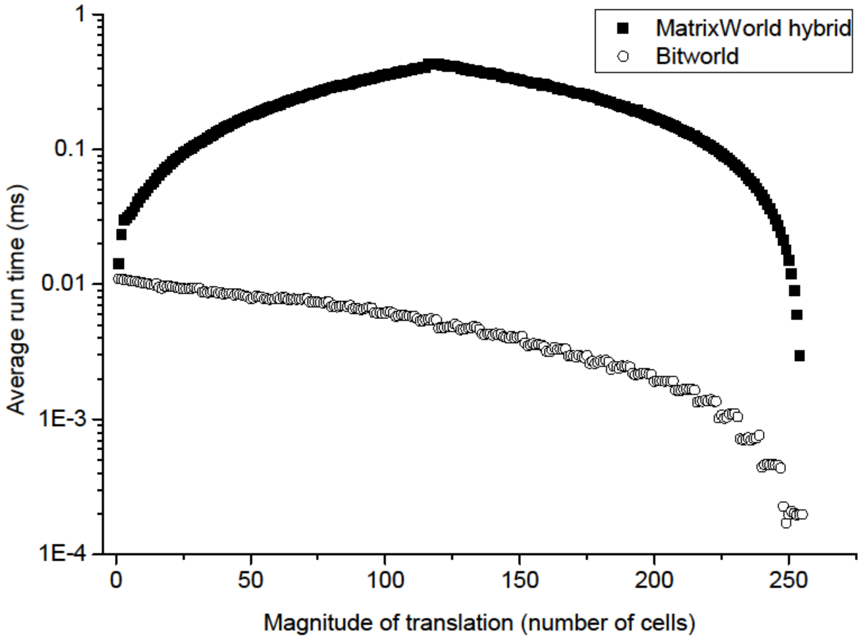


Figure 5.13: Graph comparing the performance of the *Bitworld* implementation with the *MatrixWorld* hybrid technique (lower numbers indicate better performance).

This data shows that the hybrid method does provide the benefits of both techniques, although it is worth noting that the timings for translations greater than 116bits were 2-3% higher than shown in Figure 5.13 for the relevant technique. This is attributable to poor branch prediction caused by the additional conditional statement.

Significantly, Figure 5.13 shows that the *Bitworld* implementation has superior performance to the *MatrixWorld* method over the entire range of translations. This is likely because the additional complexity associated with handling non-contiguous bits is balanced by the ability to shift cells across in a bitboard with a single processor instruction.

It is likely that variation in image sizes will change the performance profiles of these implementations considerably because more data must be translated. In such a case it may be more difficult to deploy the hybrid *MatrixWorld* implementation because the critical value would depend on the size of the image. This size might not be known at compile-time, which might further reduce performance. The increases in time in the *MatrixWorld* method will be as great, if not greater, than those of the *Bitworld* implementation since the work required for translation in a single dimension will rise linearly with that dimension.

A more important question is the performance of rotation methods in these representations, which would complete the package of required abilities for bitworld to function as a tactile image representation. In this subsection, the implementation was able to exploit significant parallelism, but as explained shortly, these benefits are not as substantial when applied to rotations.

5.3.5. Image Rotation

Rotation within bitworld may be a more complex operation than translation because it requires a coordinate change for each of the bits and so necessitates their individual manipulation. Ensuring that only bits that are set to one are transformed in this way eliminates the need to calculate transformations for every single cell on the bitworld grid. This requires serialisation of the bits, which can be done in a number of ways [208].

The efficiency of the different methods is dependent on properties of the processor, but all rely on counting the number of leading zeros in the bitboard to determine the index of the first set bit. When that index has been found and processed, the bit at this index is set to zero, and the number of leading zeros is counted again to find the position of the next bit. This is repeated until the bitboard is equal to zero. If the index of the accompanying bitboard (*i*, *j*) is known, then the global *xy*-coordinates can be calculated as

$$x = 8i + b \bmod 8 \tag{5.12}$$

$$y = 8j + \frac{b - b \bmod 8}{8} \tag{5.13}$$

where *b* is the bit position within the computer word representing the bitboard. This gives a pair of coordinates that can be transformed using the matrix in equation (5.4).

The regular need to manipulate bitworld in this fashion presents a possible performance bottleneck compared to more traditional mechanisms. Although the data transfer bandwidth is reduced by using bitworld, this benefit may be negated if the time to process the images is excessively long.

For this performance comparison, both classes implement functions to rotate their images by a desired angle and output them to a separate image. *Bitworld* used bit-scanning to access only those bits actually set, rotate the coordinates, and set those coordinates on the output image. In its basic implementation, *MatrixWorld* simply scanned over all of its cells, rotated their coordinates and allocated their values to the new coordinates. This appears to give bitworld an advantage when fewer cells are occupied because it will require fewer rotation and copy instructions to be issued. In an effort to redress this, *MatrixWorld* implements an alternative rotation routine that takes a threshold value as an additional argument. It then checks the value of each cell and only tries to move it if it exceeds that threshold.

All three of these methods were compared using the performance profiling previously described. For the avoidance of doubt, each of the one million cycles consisted of rotating the images by 360 different angles, so any advantage afforded to any algorithm by the choice of angle would be negated.

The results of the benchmark tests are shown in Figure 5.14, and show that *Bitworld* (solid black columns) outperforms the array implementations in all but one case, namely that of rotating images with 256 bits set. The simple *MatrixWorld* rotation function maintains a broadly constant time, independent of the number of pixels set in the image. This is expected behaviour, since the algorithm performs the same number of calculations without reference to the contents of the image. This simple function is outperformed by the version containing conditional checks when fewer than 128 cells within the image are above the threshold. When all the pixels in the image were over the threshold of zero, the average time to complete was nearly twice that of the simple version. This has several possible causes, the most significant of which is the fact that the function must perform the same number of rotation and memory-copying operations as the simple versions, but must perform the additional work of accessing and comparing each cell value in advance.

This additional work is also performed by the *Bitworld* methods in the form of the condition in a while-loop to serialise an unknown number of bits, which also explains the deviation when all bits in the image are set. For example, at 16 bits set within the image, *Bitworld* requires 4.49ms to perform the rotation against 21.5ms for the conditional *MatrixWorld* method. The simple *MatrixWorld* method requires a constant time of 51.2ms for all numbers of bits set within an image. This advantage for *Bitworld* decreases with an increasing number of bits. For example, when 128 bits are set, *Bitworld* requires 31.8ms compared to 51.4ms for the conditional *MatrixWorld*, and at 240 bits set *Bitworld* requires 47.5ms and the conditional *MatrixWorld* method requires 92.7ms.

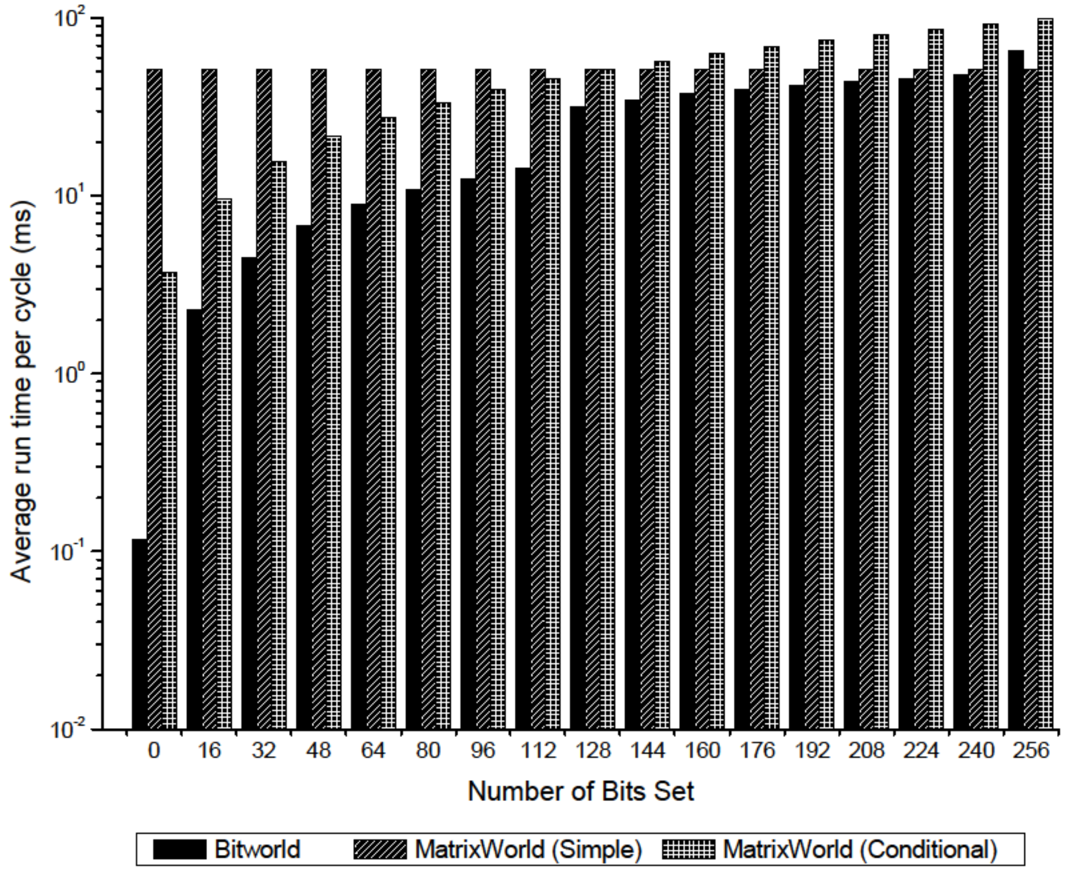


Figure 5.14: Graph showing the average running times of rotating the different image representations for different numbers of active pixels.

Despite the anomaly when all bits are set, it can be concluded that in the majority of cases the *Bitworld* methods outperform the other two implementations and that in the average case the bitworld representation offers significant performance advantages in addition to compactness.

5.3.6. Conclusions

This section has described and profiled image manipulation techniques for tactile data. In both translation and rotation, the bitworld representation was shown to have a significantly better performance than an equivalent floating-point representation. This difference arises because the bitworld cells can be manipulated in parallel to perform the necessary transformations of the images, whereas the more expansive representations must generally be processed on a cell-by-cell basis.

The ability to rotate a bitworld image coupled with the shifting operations described in the previous section provides sufficient tools to implement all required two-dimensional image transformations, which can trivially be extended to three dimensions if required.

5.4. Conclusions

This chapter has introduced a bit-based structure for tactile data called bitworld. This has been implemented and profiled, alongside an equivalent floating-point structure. The results and discussion indicate that bitworld provides a compact structure for tactile data that can be processed with high computational performance. Some of the constraints imposed on these profiling tests will have impacted on the results, such as the immutability of dimensions. These might need to be relaxed in a real deployment, but in the context of the profiling in this chapter it has allowed the fundamental differences between the representations to be compared. A real deployment will be described in Chapter 6 where bitworld will be used in combination with real sensor data to produce a tactile system entirely focussed on the data-centric paradigm.

In a real application, the measurement of pose would result from sensor readings and so might be subject to noise. This would affect the translation of images and so a means of making bitworld robust in the presence of such noise is essential. The potential effects of such noise and methods for reducing the resultant errors are explored in Chapter 7. The focus of this work will be to achieve this robustness without negating the positive attributes of the bitworld representation that have been demonstrated in this chapter.

Chapter 6
Super-resolution Bitworld Imaging

6.1. Introduction

In the previous chapter, image manipulation using the bitworld representation was shown to be more computationally efficient than floating-point representations. This chapter will combine this representation with the low-resolution sensor described in Chapter 4. Both of these aspects of a tactile sensing system were developed with the goal of lowering tactile data complexity and the present chapter will demonstrate their utility in combination.

The low sensor resolution has a distinct advantage over a higher resolution as a result of the lower volume of data that must be processed. A more subtle benefit is that the lower resolution gives the robotic system a choice about how much data it has to transmit and manipulate. Integrating a high-resolution sensor means that each time the tactile image is refreshed, the system is obliged to scan and process a potentially large number of tactels. This is acceptable if a high-resolution image is always required, but if a low resolution image would be acceptable for a significant proportion of the operating time then an unnecessary burden is imposed on the tactile system. As further motivation, consider a robot exploring a flat surface with an object lying on it as shown in Figure 6.1.

The robot can explore this surface using a tactile sensor starting from some point on the surface and following an arbitrary or systematic path across the surface with the aim of finding objects. For most of this exploration process, the tactile images will be of the featureless surface, meaning that high resolution images are unnecessary and represent a poor utilisation of computational resources for the robot. Only when the object is discovered is a high resolution image required; for the remainder, a low-resolution sensor would be sufficient.

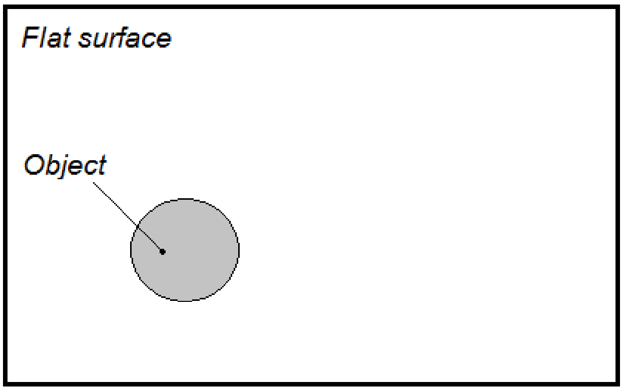


Figure 6.1: Illustration of an object lying on a flat surface that might be explored by a robot with tactile skin.

The natural riposte to this is that the low resolution sensor cannot offer the option of a high resolution image. Although this is true for a single image, work such as that of van den Heever *et al*. discussed in Chapter 2 describes how several low resolution tactile images can be combined to produce a higher resolution one [57]. An adaptation of this concept can afford a robot a choice between low- and high-resolution tactile imaging with reduced overall data complexity by using multiple images from a low resolution sensor. As a more complex motivating example, consider the snake-arm robot illustrated in Figure 6.2.

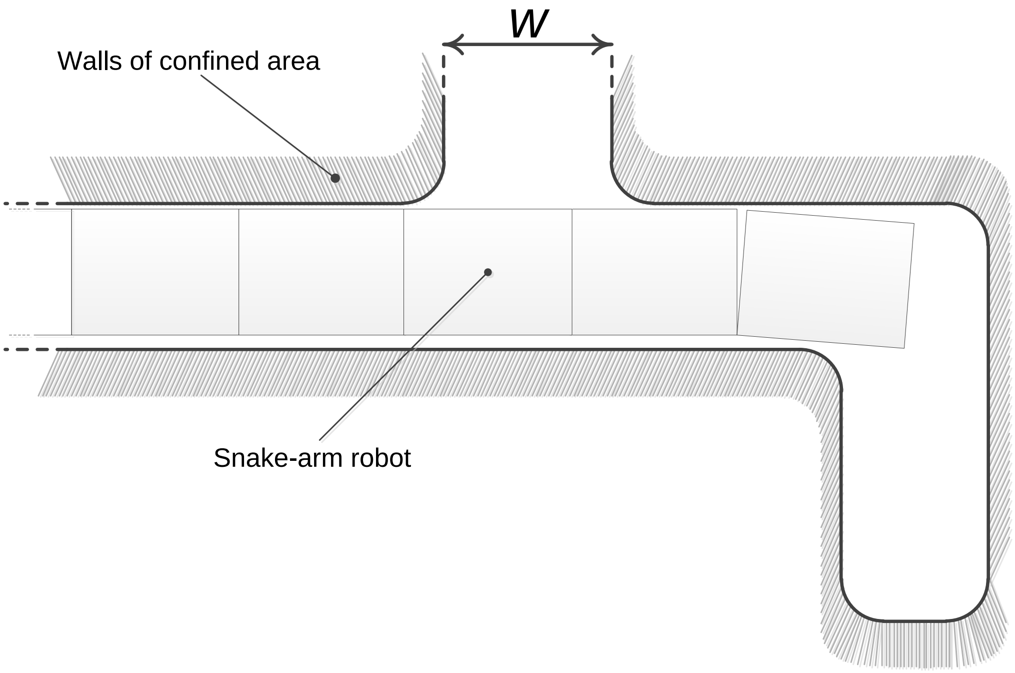


Figure 6.2: Illustration of a snake-arm robot traversing a confined space.

The snake-arm robot is traversing an unknown confined space in an effort to reach some pre-defined end goal point not shown in the diagram. Assume that the entire surface of the robot is covered with a tactile sensing array used to detect obstacles and features on the walls of the confined space. With such a high surface area requiring tactile coverage, limiting the resolution and so the number of tactile data points is essential if data complexity is to be reduced.

If the resolution of these sensors is such that the width of the side spur *w* from a single image appears smaller than the diameter of the robot, its planner might discard it as a possible route. When it rounds the corner and discovers the dead end, the planner can re-consider discarded routes using fusion techniques such as that of van den Heever *et al*. In doing this, the robot can re-determine the size of *w* to see if the opening offers an alternative but initially less obvious path.

In this chapter, it will be shown how bitworld images can be combined to produce high resolution images from the low resolution QTC sensor. This algorithm exploits knowledge of the position of the sensor in a way not achievable by van den Heever *et al*. since it can be assumed that the pose of a robot is known.

The computational efficiency of the bitworld representation was demonstrated in Chapter 5 and this chapter will provide evidence that the representation can be used to provide functional capabilities as well. As in Chapter 5, this present chapter will compare the performance of the bitworld representation with that of a floating-point one both in functional and computational efficiency terms. The next section will describe the basic computational methods and code implementation, with subsequent sections presenting and discussing the results.

6.2. Super-resolution Methods and Implementation

A tactile sensing system produces images with resolution equal to that of the sensor. The intention in this section is to develop a means of combining tactile images obtained from different orientations and translations of the sensor to produce a higher resolution image.

Multiple images will be generated, differing from each other by sub-pixel changes in position of the sensor. By aligning these so that they have a common set of axes and combining them appropriately, it will be shown that a composite image of higher resolution than those used to produce it can result. This can occur because the re-alignment of the low resolution images means that the edges of the pixels in each image will not lie in the same location. This means that the re-aligned images will form new pixel edges at a finer spacing that in the original image, so forming a higher resolution composite one. This effect is illustrated in Figure 6.3, where two images of an object are aligned.

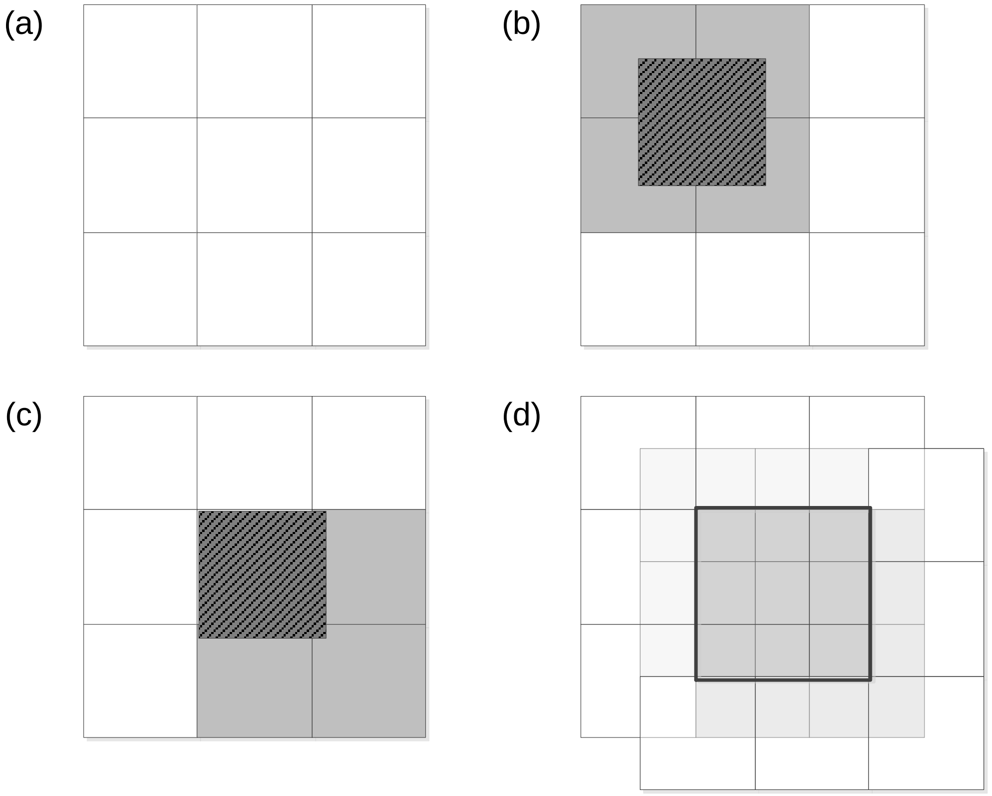


Figure 6.3: Illustration of image super-resolution. (a) shows an unloaded image for reference, (b) shows an image in light grey produced by the hatched object (c) shows the change in image after a sub-pixel shift of the object and (d) shows the effect of aligning the images.

In Figures 6.3(b) and 6.3(c), it can be seen that the individual images would image the square object as having *x*- and *y*-extents equal in size to two tactels. When combined to produce the image in Figure 6.3(d), the result is an extent estimation of 1.5 tactels. Further increases in resolution should be possible with a larger number of images taken after smaller movements of the sensor.

Effective application of these techniques requires the algorithms to know the alignment of the sensor both in terms of orientation and translation. This is often performed by image registration techniques of high complexity due to their operation on visual images. The latter are often disturbed by blurriness, optical distortion and other effects [209] that are not applicable in the tactile domain.

The most relevant work of van den Heever *et al*. aligned tactile images by rotating and translating the images through the search space of all possible translations, using a fitness function to determine the best alignment [57]. This was necessary in this instance only because the sensor was intended for deployment as a hand-held palpitation device. In a robot-based tactile system the alignment of the sensor can be known from measurements of pose and the initial deployment of the sensor.

For these experiments, the effect of robotic movement was emulated by changing the position of an object placed on the QTC matrix developed in Chapters 3 and 4. The effective translation could be measured and associated with the resulting tactile image to allow subsequent processing. Three objects of different shapes were used to produce the images in these experiments. Photographs of these are shown in Figure 6.4.

The rounded rectangle was 142mm along its longest length and 52mm along its width. The rounded triangle was 127mm along its longest length and 86mm at its largest breadth. The circle had a diameter of 80mm. In comparison, the size of each tactel was 13mm by 13mm. The shapes and sizes were chosen to test the performance of a rectilinear array of the smaller tactels in resolving the extent and shape of the three objects.

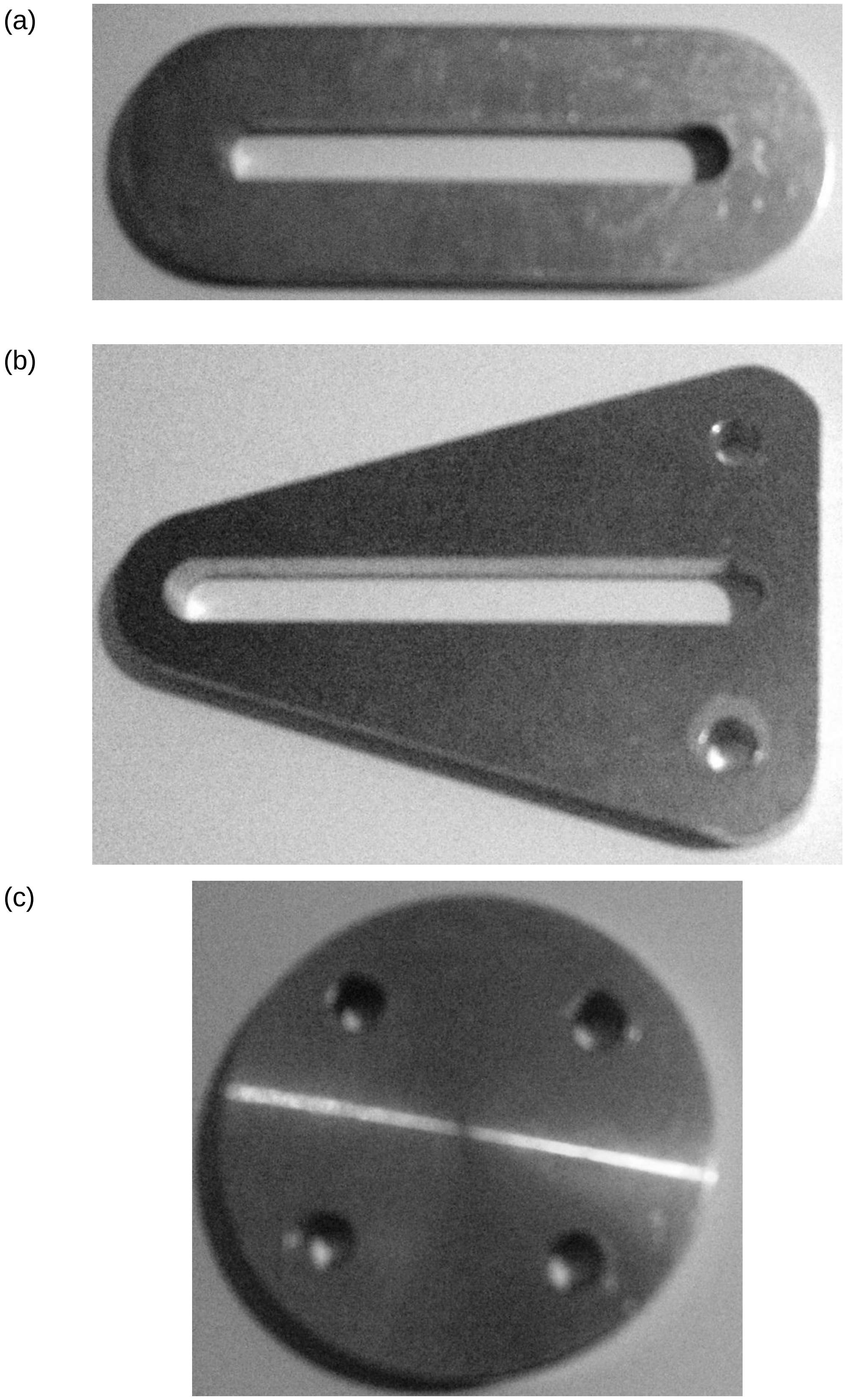


Figure 6.4: Photographs of the objects used to produce tactile images in these experiments. These were (a) a rounded rectangle (b) a rounded triangle and (c) a circle.

For this work, three different methods of combining images were explored to provide a means of comparing the relative benefits of each. The first of these uses techniques from the work developed in Chapter 5. The methods of the *Bitworld* class for shifting and rotation are used to align the images according to known translations of the images.

A Boolean operation can then be used to combine these images. In Section 6.4, this will consist simply of an AND-operation applied between the images to eliminate pixels that are not measured in all images. This represents a minimalist approach since the transmitted data could be stored simply as bits. However, there is a possibility that there will be losses in data caused by the destructive nature of the AND-operation, causing degradation in functional performance. For comparison, a second method is described using bitworld images combined onto an integer matrix that counts the number of times each cell is measured as being occupied. This approach minimises data communication but increases the amount of memory required. As an additional benchmark, a floating point matrix produced using the code of the *MatrixWorld* class from Chapter 5 is used. This combines the aligned images by averaging the values of each aligned pixel to produce the image.

The assessment of the relative efficacy of these methods will be partly qualitative in nature, as it relates to the clarity of the resulting images and the ease of identifying objects from them by eye. Two important quantitative measures also apply. For functional performance, the dimensions of the object as measured from the combined image serve as a good measure of image quality. For applications such as obstacle avoidance, over-estimation would be preferable to under-estimation, whilst other applications may have differing requirements.

For computational performance, measurements of the time taken to perform multiple repetitions of the image processing code can be used to quantify the efficiency of the process. This follows the methodology laid out in Chapter 5 and only the portions of the code that are unique to each method will be profiled in this way in order to accentuate any differences between them.

Multiple images were taken by measuring the resistance of each tactel using a Keithley 2400 SourceMeter™ interfaced through a Keithley 7702 switching unit within a Keithley 2750 enclosure to scan across the array. To produce the bitworld images necessary for two of the three methods, a threshold value was required to determine if a bit should be set to one. Based on the results of Chapter 4, this value was chosen as a reduced resistance of 30% relative to a base value determined from an unloaded image of the sensor.

The objective was to achieve an image resolution of 1mm from the low resolution QTC matrix consisting of 13mm x 13mm tactels. In order to produce this higher resolution image, the individual images from the QTC array had to be rescaled. This was done by assigning the measured value of each tactel to all of the 1mm x 1mm pixels corresponding to its 13mm x 13mm area to form a new, higher-resolution image. These images were then aligned following the approach illustrated in Figure 6.3. These methods are used to produce all the images shown in the subsequent two sections and to inform the discussion of the results.

6.3. Image Combination: Results and Discussion

This section describes the results of applying the methods described in Section 6.2. For all results shown in this section, the objects being imaged were moved diagonally across the array in twenty sub-pixel movements. The results of Chapter 4 show that a time between images on the order of a few seconds is sufficient to restore the tactels to their unloaded behaviour. In between producing images, therefore, the objects were fully removed from the array for several seconds and then placed in the next location. The *x*- and *y*-position of objects was measured using vernier callipers and then fed to the algorithm used to fuse the images.

The following sub-sections show the results of imaging the objects shown in Figure 6.4. In Section 6.3.1, the focus is on identifying the presence of straight edges in objects. Section 6.3.2 will then show that bitworld can be used to image the curved features of the objects by using the additional images. Both sections will independently report the computational performance of using bitworld, bitworld-counting and floating-point methods of image combination. This will further highlight the advantages of bitworld in this area.

6.3.1. Straight Edges

Applying the three methods described in Section 6.2 to imaging the rounded rectangle produces the three images shown in Figure 6.5. The images were aligned so as to place the object in the lower-left corner of the final, higher-resolution image.

All three methods produce an image where the rounded rectangle can be discerned in the lower-left corner. Figure 6.6 shows a zoomed-in version of Figure 6.5(a) to demonstrate that the *x*-extent of the image is 142mm and that the *y*-extent is 52mm. This matches the true dimensions of the object and is the same for both of the bitworld-based images in Figures 6.4(a) and 6.4(b).

The result of applying the averaging method shown in Figure 6.5(c) qualitatively shows the same shape in the darker tones of the image. Within this set of darker pixels, there is a significant variation in the exact colour. This is because there are an infinite number of possible pixel values compared to bitworld, which prescribes a finite set. The bitworld images thus have a better contrast than that produced by the averaging technique.

Figures 6.5(b) and 6.5(c) show additional artefacts around the image. These are caused by a relatively small number of tactels with a resistance that has fallen below the threshold and the realignment process has effectively smeared them across the image. These tactels have an abnormally low unloaded resistance, meaning that minor fluctuations in resistance represent a greater percentage change than in a cell with a normal unloaded resistance. The bitworld-AND image of Figure 6.5(a) is able to eliminate these sensor defects because of their scarcity, highlighting the strength of this bitworld method.

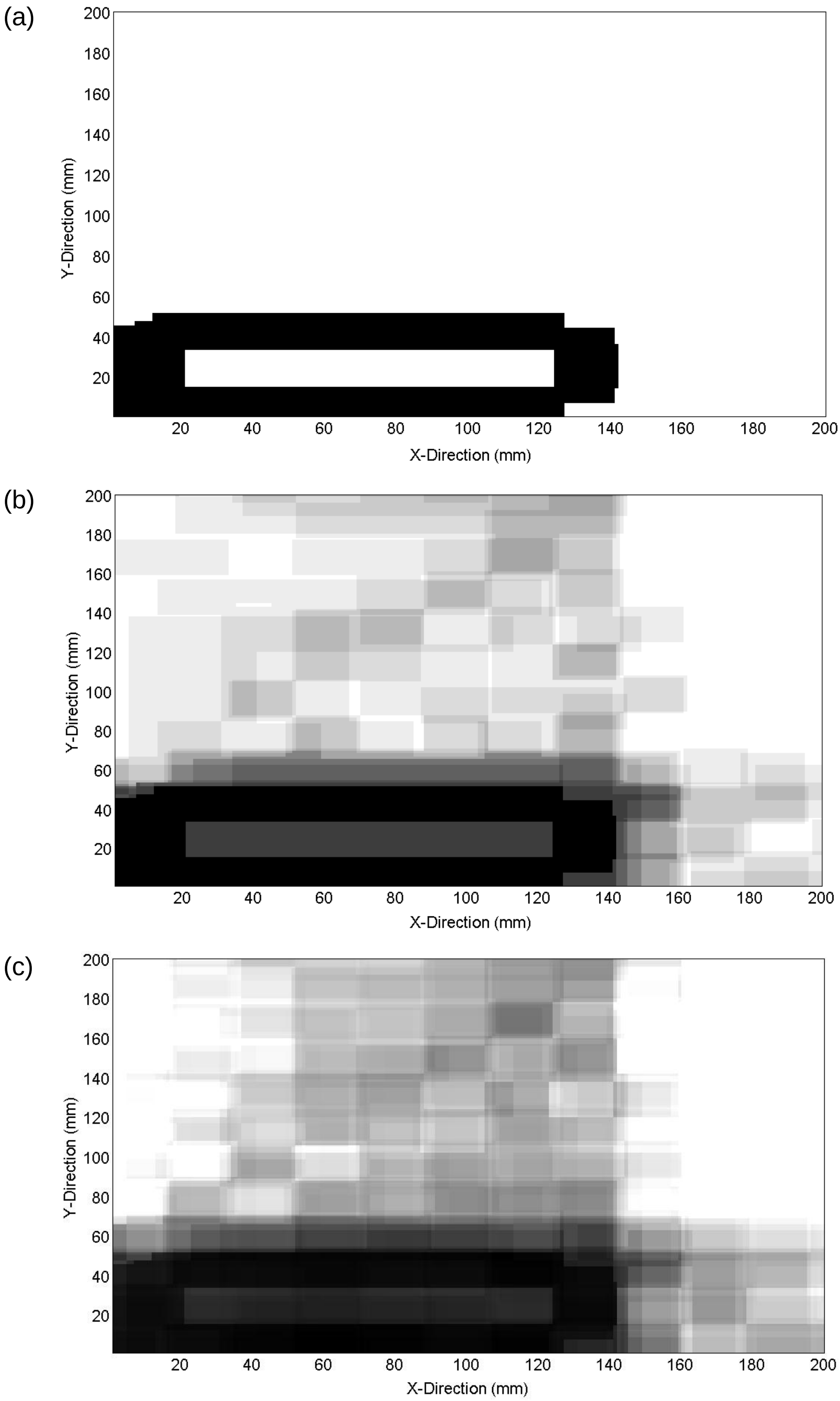


Figure 6.5: Images resulting from applying the (a) bitworld-AND (b) bitworld-counting and (c) averaging methods to multiple images of a rounded rectangle.

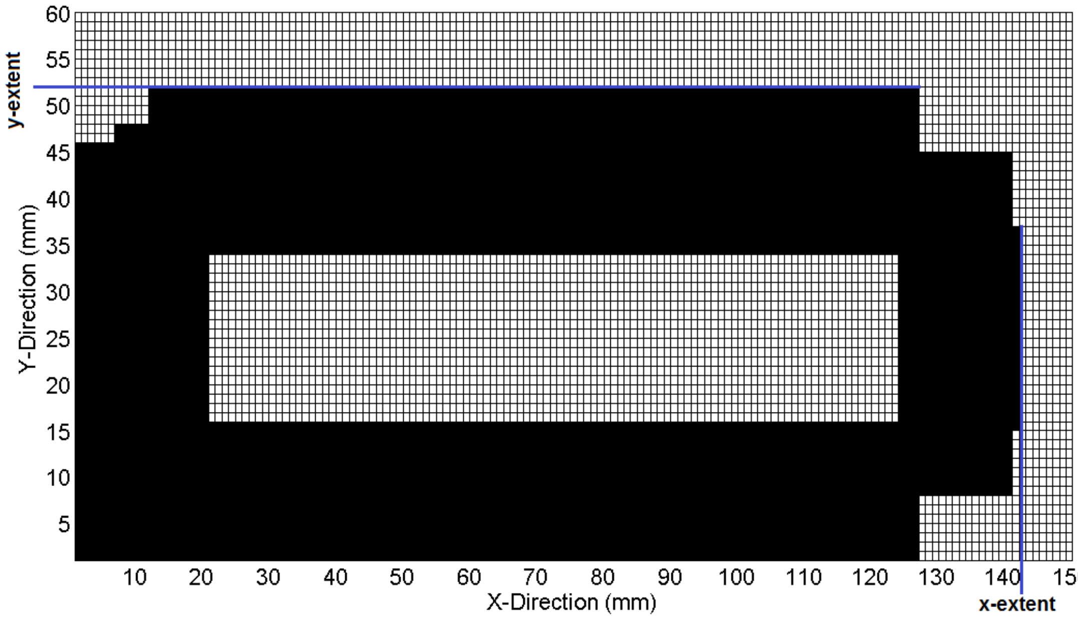


Figure 6.6: Zoomed version of Figure 6.5(a) with the *x*- and *y*-extents of the object highlighted and labelled for clarity. The same extents can be identified for Figure 6.5(b).

The clear identification of the void within the shape by all three methods is an interesting feature. This feature appears in Figures 6.5(a) and 6.5(b) because the tactels were marked as unoccupied in all but one of the twenty measurements. Interestingly, this feature develops from multiple images and not from one single image where all of these tactels were insufficiently loaded during measurement to cross the threshold resistance. Instead, the alignment of several images across the tactels was such as to lower the pressure applied to particular tactels and the central void appears as a result of image combination.

From one perspective, the presence of this artefact is a good one, since the methods are all identifying a distinctive feature of the object. However, because it only appears as the result of isolated measurements, the size of the void is overestimated as having dimensions defined by the resolution of the QTC array. Although the internal feature identification is not perfect the bitworld-based methods display considerable ability in identifying the extent of objects when fusing multiple images together to produce a higher resolution one.

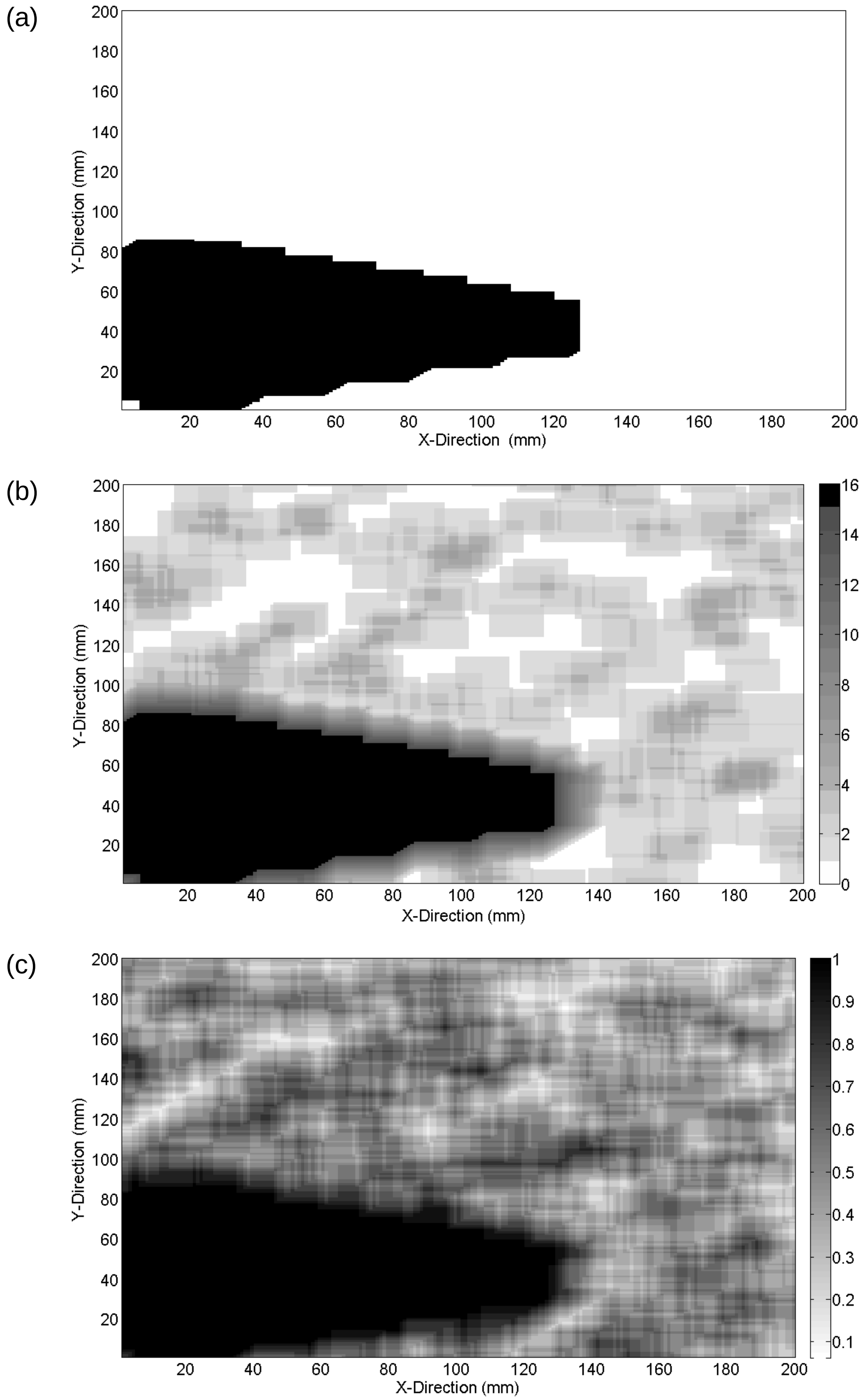


Figure 6.7: Images produced as the result of applying the (a) bitworld-AND (b) bitworld-counting and (c) averaging techniques to multiple images of a rounded triangle.

The rounded rectangle images demonstrate the capabilities of bitworld when the straight edges of the object are aligned with the edges of the rectangular tactels. To demonstrate the capability of all three methods in identifying non-aligned straight edges, Figure 6.7 shows the result of fusing multiple low resolution images of the rounded triangle, which has edges that are not orthogonally aligned to the tactel edges.

As with the rounded rectangle, the bitworld-based methods of combining multiple images to produce a higher resolution one are qualitatively very successful as shown by Figures 6.7(a) and 6.7(b). This is emphasised by Figure 6.8, which shows more clearly that the *x*-extent of the object image is 127mm and the *y*-extent is 86mm.

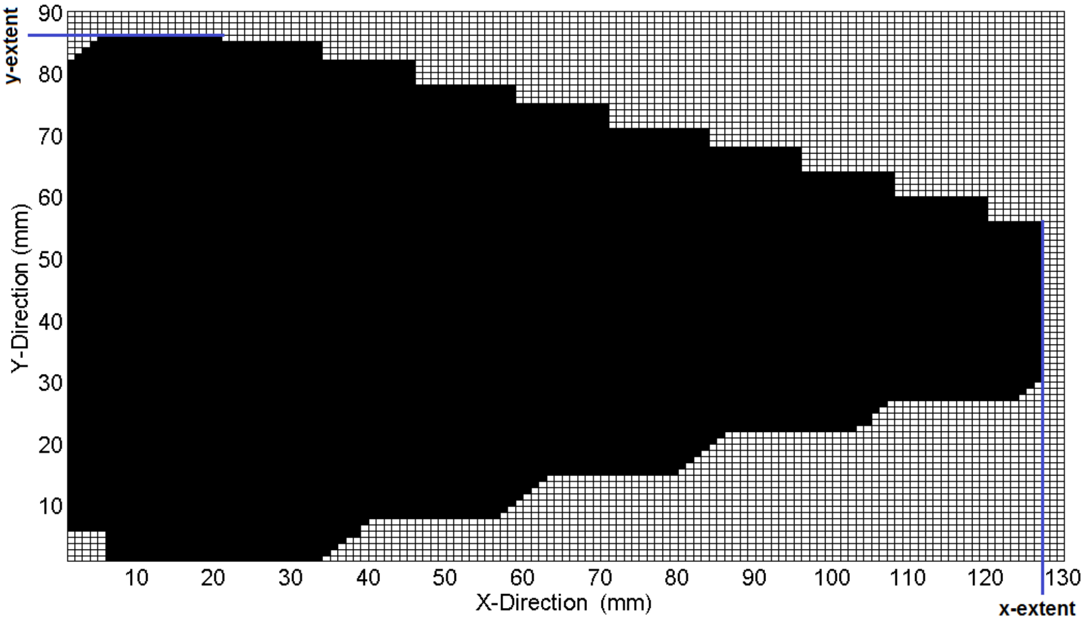


Figure 6.8: Zoomed version of Figure 6.7(a) with the *x*- and *y*-extents of the object highlighted in blue and labelled for clarity. The same extents can be identified for Figure 6.7(b).

This matches the dimensions of the actual object to within the desired resolution of the image and is the same for all three images in Figure 6.7. Although the lines defining the angled edges in the images show some stepping, this is caused by the discrete nature of the pixels and the angled edges are clearly discernible as is the triangular nature of the object.

There is an obvious asymmetry in the angled sides of the triangle in Figure 6.7(b). This occurs because the rounded triangle was moved at 45° across the sensor, but the angle of these edges and their orientation means that the effective relative change in position between points along these edges and the sensor array were different.

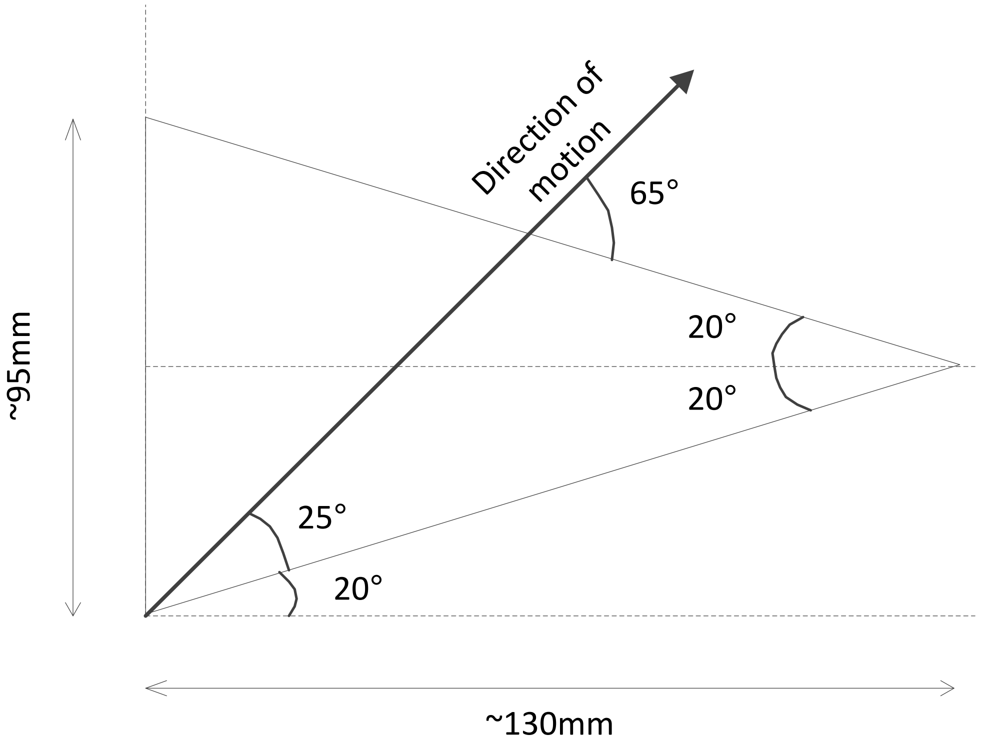


Figure 6.9: Illustration of moving a triangular object at 45° and the differing effective angles of motion by the two main edges. Illustration is not to scale and angles are approximate.

Figure 6.9 illustrates this effect using a triangle of similar dimensions to the rounded triangle used in this work. If the direction of motion was 20°, it is unlikely any improvement in identifying the position of the lower edge would be seen after image combination. This is because the way in which the edge lay across the symmetrical sensor array would be very similar in each position and it is differences between several images that can lead to improvements. Figure 6.9 shows that the difference in location for points along the lower edge between images was much smaller than that for the upper edge. This is because the lower edge was moving at 25° compared to a direction that would have little or no effect during image combination, compared to the upper edge which is moving at 65° on the same metric.

This difference causes the asymmetry observed in Figures 6.7 and 6.8 and also explains the slight asymmetries in the rounded rectangle results of Figure 6.5 and 6.6. The effect of choosing different directions of motion or orientations of the object will be explored further in Section 6.3.2.

Also of interest is the inability to discern the small internal features such as the void and the holes in the object, which are visible in Figure 6.4(b). This is because these features have dimensions smaller than the resolution of the sensor array. Consequently, the tactels over which these features are placed are never completely unloaded because there is always an overlap of the rest of the object on the tactels.

In spite of these limitations, which apply just as much to the floating-point averaging results shown in Figure 6.7(c), the bitworld-based images identify the gross features of the objects accurately and with high clarity demonstrating their functional performance in this domain. The computational performance must now also be considered.

For the purposes of producing the bitworld-based images in Figures 6.5 and 6.7, the images were aligned using the shifting procedure described in Section 5.3.2. For individual shifts, this had order-of-magnitude improvements in the run times over the use of an equivalent floating-point representation. In producing the results for the present chapter, multiple shifts were required to align the bitworld images, and this was profiled using a combination of the *Bitworld* and *StopWatch* classes of Chapter 5. This was compared to a similar arrangement using the *MatrixWorld* class in place of the *Bitworld* class.

Measuring the performance of the bitworld-counting technique required an additional class called *BitworldMatrix*, the code for which can be found in Appendix 6.1. The image manipulation techniques for this task were copied from the *Bitworld* class, with the final image stored as an integer version of the *MatrixWorld* class. In order to perform the fusion, each individual bitworld image is aligned in turn. The bits from each image are scanned and the locations of the one-bits are used to determine which cells of the output image need to be incremented. This class therefore has a higher final storage cost of one byte per pixel than the *Bitworld* class, but can make use of the high-performance shifting operations that can be applied in a bitworld-based system.

The code was compiled using version 8.0 of the Microsoft® Visual C++ compiler in release mode, with optimisations enabled to maximise speed. The program was run on a computer with an Intel® Core i7 2.6GHz processor, 4GB of RAM and Windows XP x64 Edition. Although this was a multi-core processor, the program was not developed to exploit these.

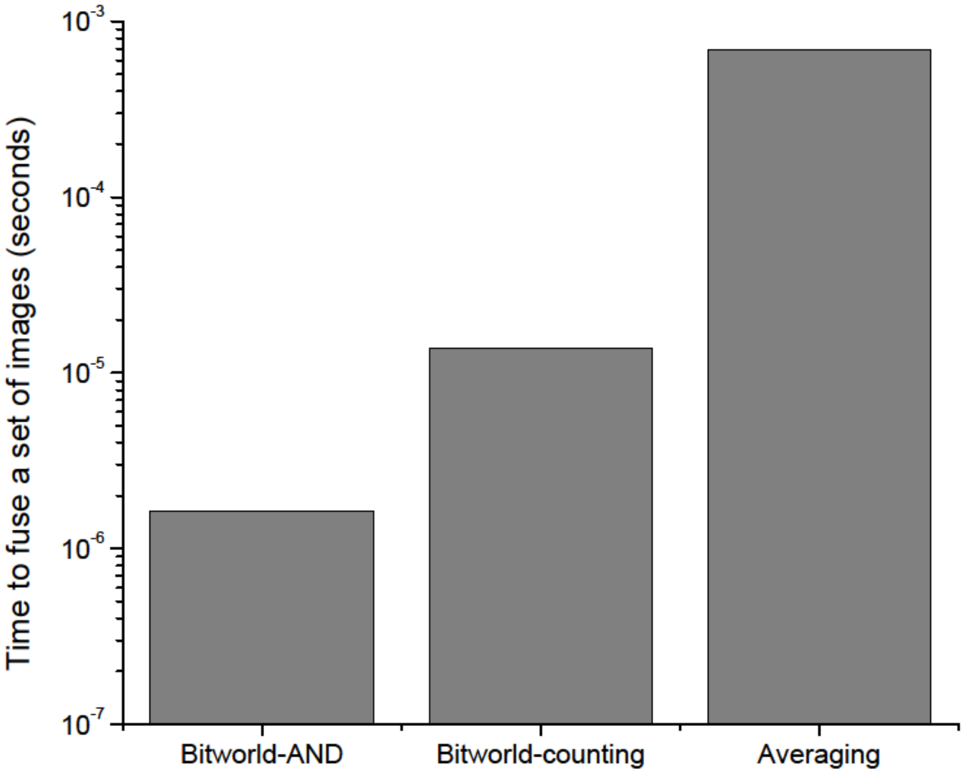


Figure 6.10: Graph showing the time to combine multiple low resolution images of a rounded rectangle using the three methods under comparison to produce a high resolution result. Lower figures indicate better performance.

All three methods were used to align and combine twenty images of the rounded rectangle and triangle and produce the images of Figures 6.5 and 6.7. These were repeated 100,000 times and the timing results used to produce an average run time for each method. As the timings for each object were very similar, only the results for the three methods of fusing images of the rounded rectangle are shown in Figure 6.10.

The bitworld-AND method required 1.63 x 10^{-6} seconds, which is 420 times faster than the floating-point averaging method at 6.87 x 10^{-4} seconds. The inherent parallelism of the bitworld methods is again shown to be superior to the floating point methodologies, which in this instance also had to use more expensive floating point addition and division operations to perform the averaging process.

Although the bitworld-counting method had additional storage and integer addition operations, it was still able to exploit bit-based parallelism to perform its operations 50 times faster than the floating point representation at 1.38 x 10^{-5} seconds. This is an order-of-magnitude greater than the bitworld-AND time, but there may be functional reasons for adopting a more expansive representation so the raw speed increase of both bitworld-based methods is encouraging.

It is worth noting that the bitworld-AND method was able to perform at a level similar to that of a single bitworld shift. The timings in Figure 5.13 show that the timings for a translation were on the order 10^{-6} to 10^{-7} seconds with a dependency of the number of translations performed. This is on a similar order of magnitude to the timing for a twenty image bitworld-AND fusion, but this may be because the data is compact and so can fit into the cache line of the processor better, making it relatively quick for multiple bitwise operations to be performed to produce a single output. The optimisations by the compiler may also have been more effective in reducing unnecessary bitwise operations, so enhancing the performance in this more complex scenario.

This speaks to a general advantage of the bitworld representation, namely that its compactness and structure make it particularly suitable for efficient and rapid processing on modern processors and within modern compilers.

It can be concluded from this that bitworld representations are at least as good as, if not better than, floating-point representations from a qualitative point of view. The performance of bitworld is superior by orders-of-magnitude, even when a compromise such as the integer counting for fusion by the bitworld-counting method is made. Consequently, low resolution data can be used in conjunction with a low resolution data type to produce a higher-resolution image.

The only qualitative feature that is not ideal in any of the three methods is the treatment of curved features. The rounded features of the rectangle and triangle in Figures 6.5 and 6.7 are not perfectly visible. This will require a small adjustment to the imaging process that will be discussed in the next sub-section.

6.3.2. Curved Features

The curved features of the rounded rectangle and triangle in the previous sub-section were difficult to discern, in contrast to the high levels of clarity achieved in identifying straight edges. This sub-section explores a method for improving this aspect of combining images. The identification of straight edges in the case of both objects was the result of correctly identifying the *x*- and *y*-extents of the objects.

Curved features represented on a rectangular image grid cannot normally be decomposed into individual straight lines in the same way as, for example, the angled but straight edges of the rounded triangle. Curved features are typically represented by pixels that are mainly not aligned orthogonally. This effect is illustrated in Figure 6.11, where arcs of various radii of curvature are illustrated as they appear when drawn on a regular rectangular grid, for instance in a computer graphics program. In fact, the images of Figure 6.11 where produced from zoomed portions of a circle in the Microsoft® Paint application that is supplied with Windows XP.

These intricate features are not detected by the methods described so far in this chapter because there are not enough low resolution images in unique translational configurations to fuse together and make these features noticeable. To image the arcs in Figure 6.11 using the super-resolution technique described so far would require an object-sensor alignment that allowed the lower-left corner of a tactel to trace the arc. The object-sensor alignment refers to the combination of the position of the object on the sensor and how features of the object align with the edges of the tactels in the array. Figure 6.12 illustrates this effect by showing three different object-sensor alignments over the course of this trace.

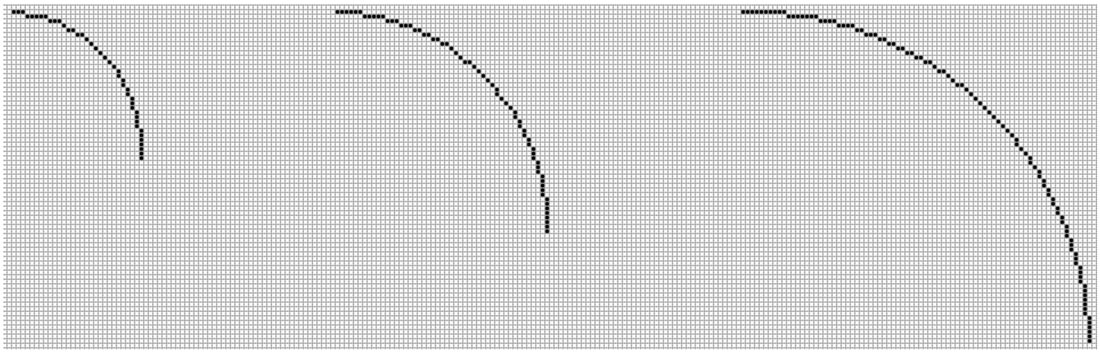


Figure 6.11: Illustration of the intricacy required to render an arc on a rectangular grid. Arcs with various radii of curvature are shown for comparison.

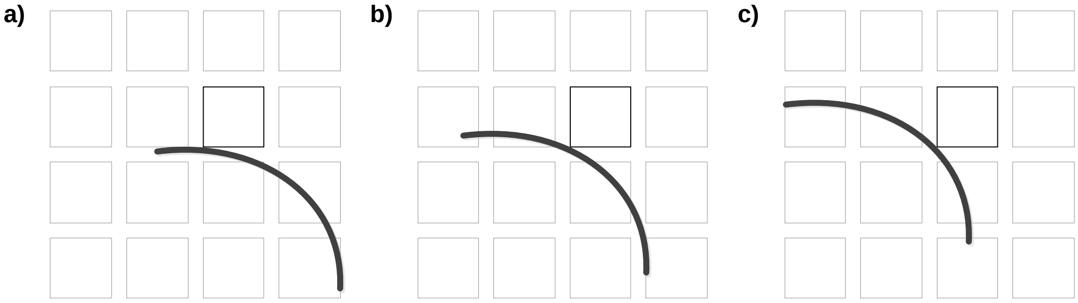


Figure 6.12: Illustration of three different object-sensor alignments required as part of a sequence to trace and so successfully identify an arc using the method of Section 6.2. The object is shown as an arc, and the sensor as a rectilinear grid

This procedure would ensure that all the correct cells in the combined image will be set to zero in the case of bitworld or of a higher value in the case of the floating-point averaging technique. Such a procedure is difficult to achieve in practice without prior knowledge of the shape because the object-sensor alignment would have to be carefully planned to follow the arc. This would starkly limit the applicability of these methods and so a more general method is required.

Since the method described in Section 6.2 can successfully determine where the *x*- and *y*-extents of an object are, the centre of the object can also be determined. Producing various low resolution images of the object that vary by rotation in addition to translation provides a similar effect to the curve tracing just described. This is because the rotation of the object or sensor makes it possible for the tactels to adopt positions relative to the curved features that are not possible in the simple rectilinear translation. Examples of these positions are shown in Figure 6.13.

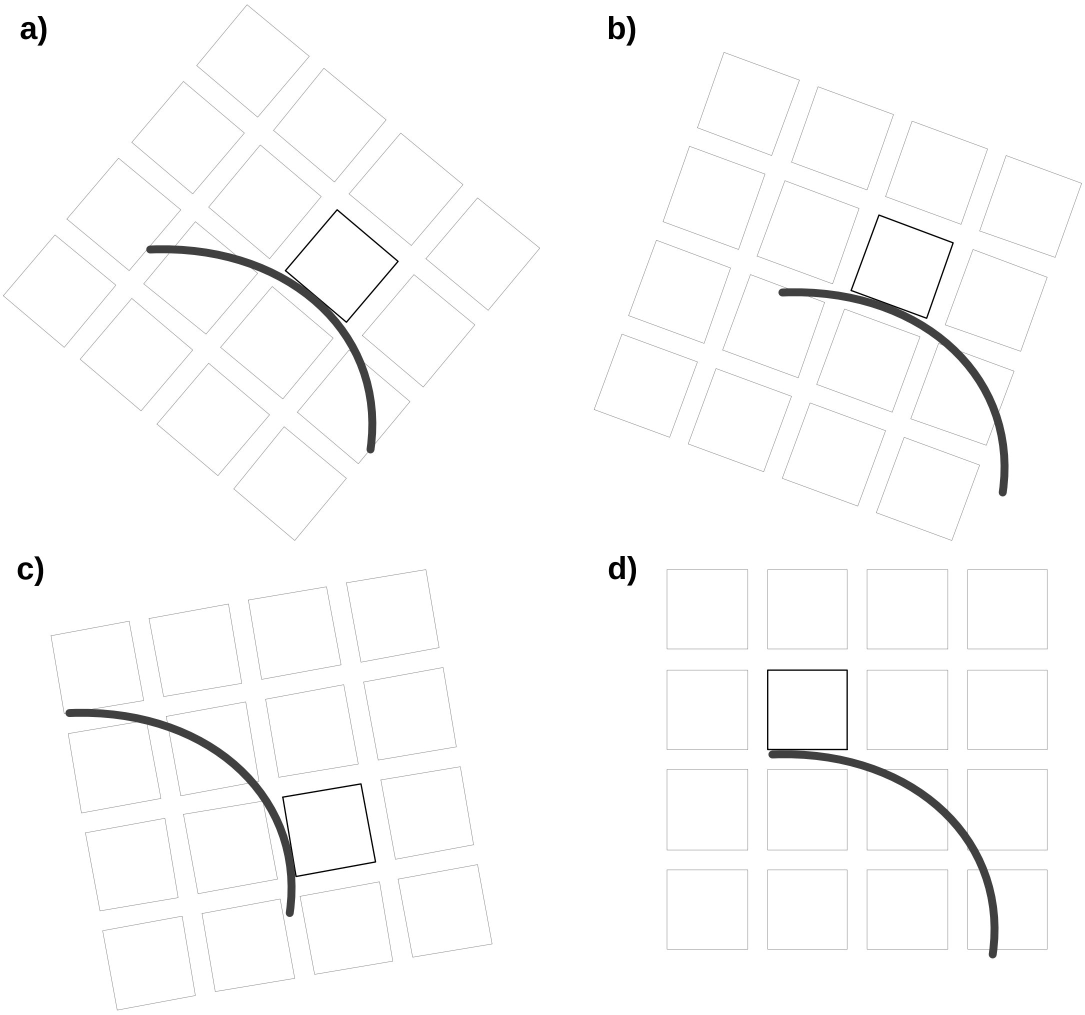


Figure 6.13: Illustration of object-sensor alignments that can be achieved using rotation – object is shown as an arc, and the sensor as a rectilinear grid. a) to c) illustrate positions that cannot be achieved using rectilinear translation, and d) shows that orthogonal alignments can still be achieved under rotation.

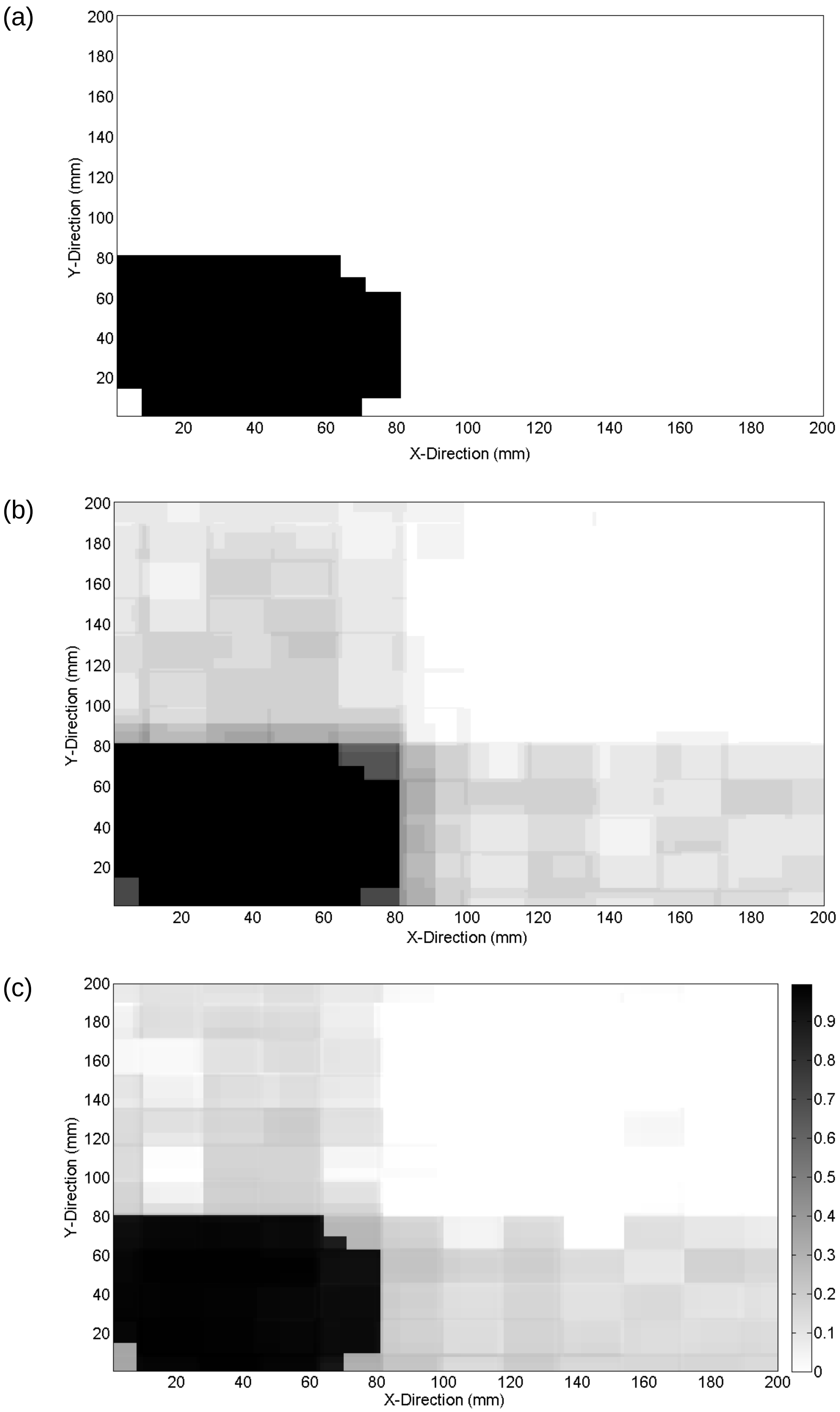


Figure 6.14: Images produced as the result of applying the (a) bitworld-AND (b) bitworld-counting and (c) averaging methods to multiple images of a circle without additional rotations

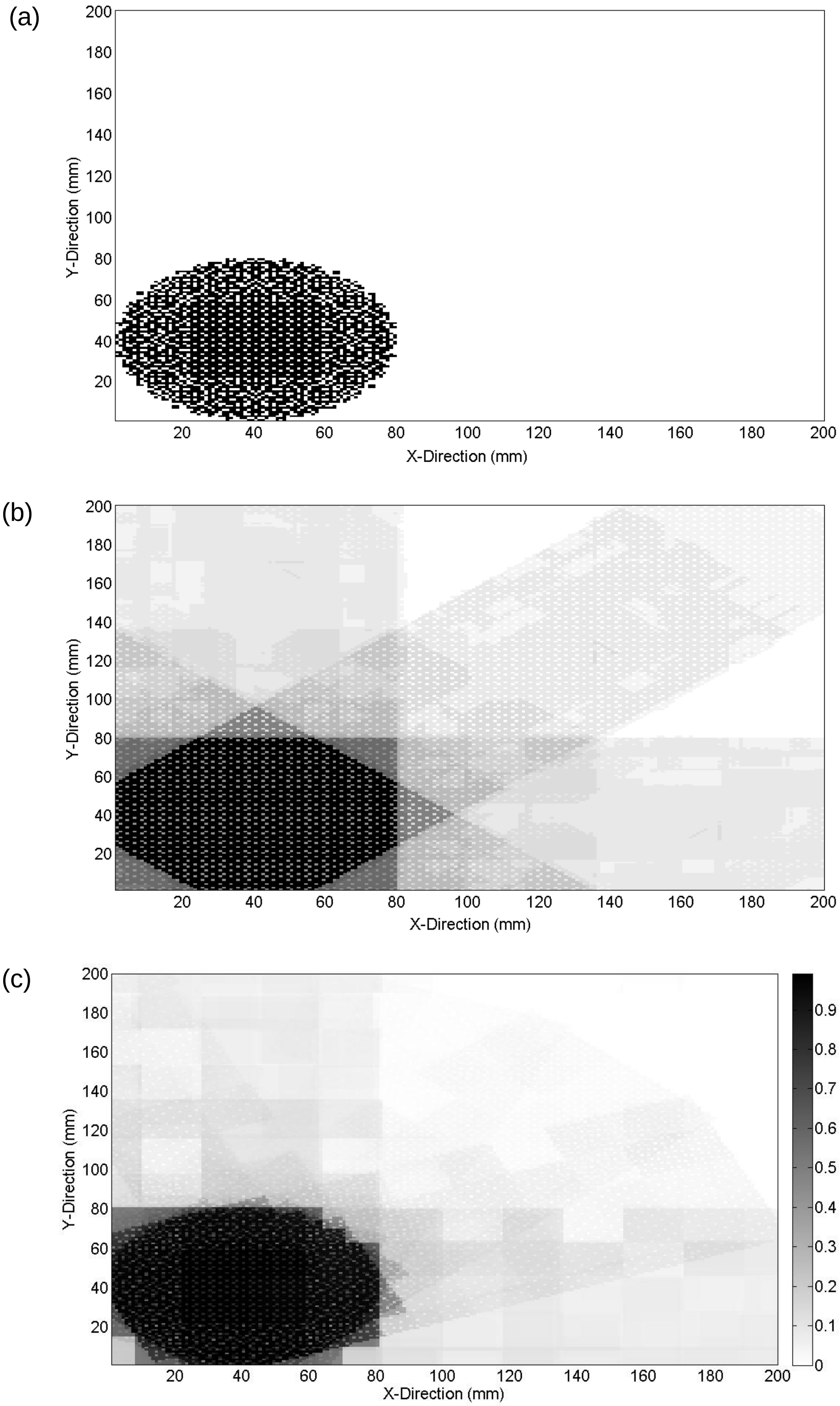


Figure 6.15: Images produced as the result of applying the (a) bitworld-AND (b) bitworld-counting and (c) averaging techniques to multiple images of a circle with rotations.

If the number and granularity of the translations is chosen carefully, a better approximation of curved features can be achieved. In the case of the experimental results in this section, the usual twenty translations were made at one millimetre intervals in both the *x*- and *y*-directions.

The procedure was adapted to add rotations of an object about its centre as determined from the results of the non-rotational linear translation method of Section 6.2. These images were then transformed back to a common axis using the same methods discussed previously, with the addition of the rotational methods of Section 5.3.3. These rotational methods are simple trigonometric functions mapping coordinates (*x, y*) in an image to new pixel coordinates (*x'*, *y'*) using the relationship

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{6.1}$$

where θ is the angle of rotation. Figure 6.14 shows the result of combining twenty low resolution images of the circular object shown in Figure 6.4(c) without making any rotations. As expected, the x- and y-extents are both 80mm as expected, but the curved features are not very clear.

Figure 6.15 shows the effect of generating an additional eight images at each of the twenty translations, applying rotations of 45°, 90°, 180°, 235°, 270° and 315°.

As in Figure 6.14 the *x*- and *y*-extents of the circle after combining the rotated images are 80mm. There is some degradation in the quality of all three images, which occurs because the rotation operation described by equation (6.1) is naïve. To understand this, Figure 6.16 illustrates the problem. Figure 6.16(a) shows a pixel prior to a rotation by 45°, and Figure 6.16(b) illustrates the fact that this pixel could lie in one of four destination pixels after rotation. In the method that produced the images of Figure 6.15, this was resolved by rounding the x- and y-coordinates to the nearest integer and this leads to pixels in all three images being unoccupied or less occupied than they should be.

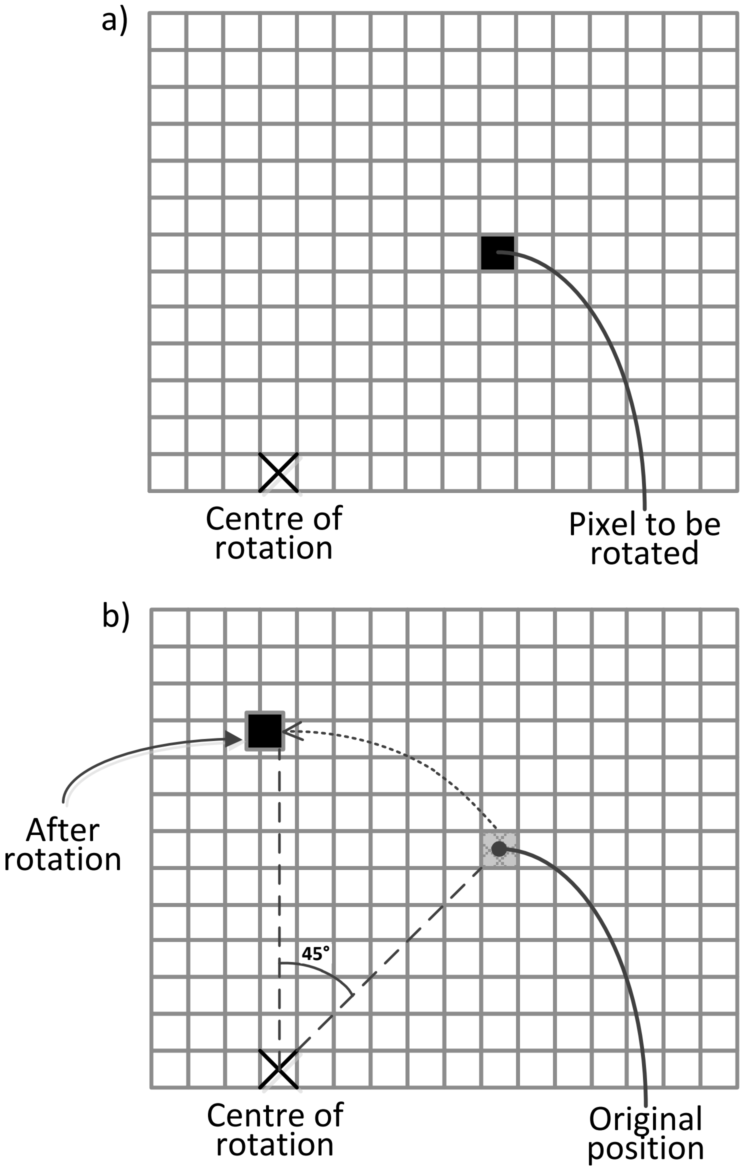


Figure 6.16: Illustrative example of rotating a pixel. The original pixel and a centre of rotation is shown in (a), followed by (b) the position of the pixel after rotation as calculated using Equation (6.1).

Despite this minor fault, the curvature of the circle is clearly visible in the pure bitworld methods indicating that the inclusion of rotations is an effective strategy. The curvature is also visible in the averaging technique, but appears to have been corrupted in the bitworld addition method, suggesting that there are limitations in the utility of this technique as regards rotations. With an improvement to the rotation method, it is likely that the minor defects in the bitworld and averaging techniques could easily be resolved.

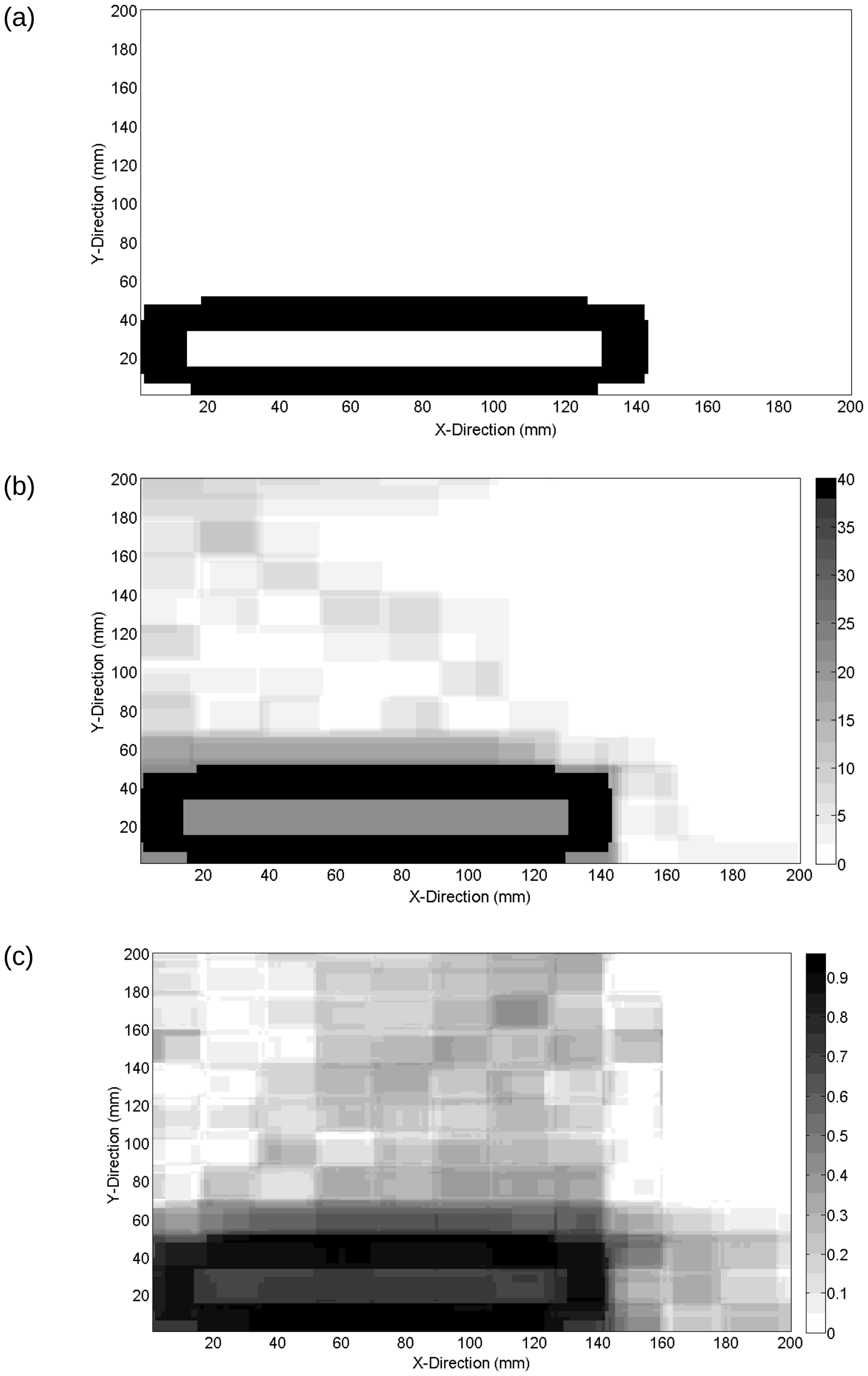


Figure 6.17: Images produced as the result of applying the (a) bitworld-AND (b) bitworld-counting and (c) averaging techniques to multiple images of a rounded rectangle including those with a single 180° rotation.

This improvement was achieved with only seven additional rotations per translation. Figure 6.17 shows how even a single rotation can improve images by re-imaging the rounded rectangle using the same twenty translations, but with a 180° rotation of the object at each translation.

These images show that additional images taken when the object is rotated can be used to improve the identification of curved features using the super-resolution technique. The two bitworld-based methods show equal or better image quality and clarity to that of the floating-point method.

In Chapter 5, the effect of applying rotations on computational performance was considered. For bitworld the performance was dependent on the number of bits set in the image, but was consistently faster than the floating-point method except in the edge case of all bits in the image being set.

In the case of producing the high resolution image of the circular object, the *Bitworld, BitworldMatrix* and *MatrixWorld* classes were used to produce Figures 6.15(a), (b) and (c) respectively. The code was timed using the *StopWatch* class using the same software configuration described in Section 6.3.1. As in Chapter 5, the *MatrixWorld* class was profiled using two rotation methods: one that simply rotates all pixels irrespective of value, and the other that relies on only rotating pixels with values in excess of an arbitrarily-selected value of 0.4. The results of the profiling are shown in Figure 6.18, including the results of non-rotational combination from Figure 6.10 for comparison.

The bitworld-based methods again show a substantial performance improvement over the floating-point representation. At a runtime of 8.54 x 10^{-6} seconds, the bitworld-AND method is 1335 times faster than the normal averaging method and 348 times faster than the averaging method including the conditional branches. The bitworld-counting method also shows significant improvements over the pure floating-point representations, albeit less impressive ones than by the pure bitworld method. It was 19 times faster than the averaging method and 5 times faster than the conditional averaging method with a run time of 6.02 x 10^{-4} seconds.

The hybrid bitworld-counting representation was 70.5 times slower than the pure bitworld-AND method, all of which demonstrates that using bitworld in this more complex situation affords enormous benefits in terms of computational performance.

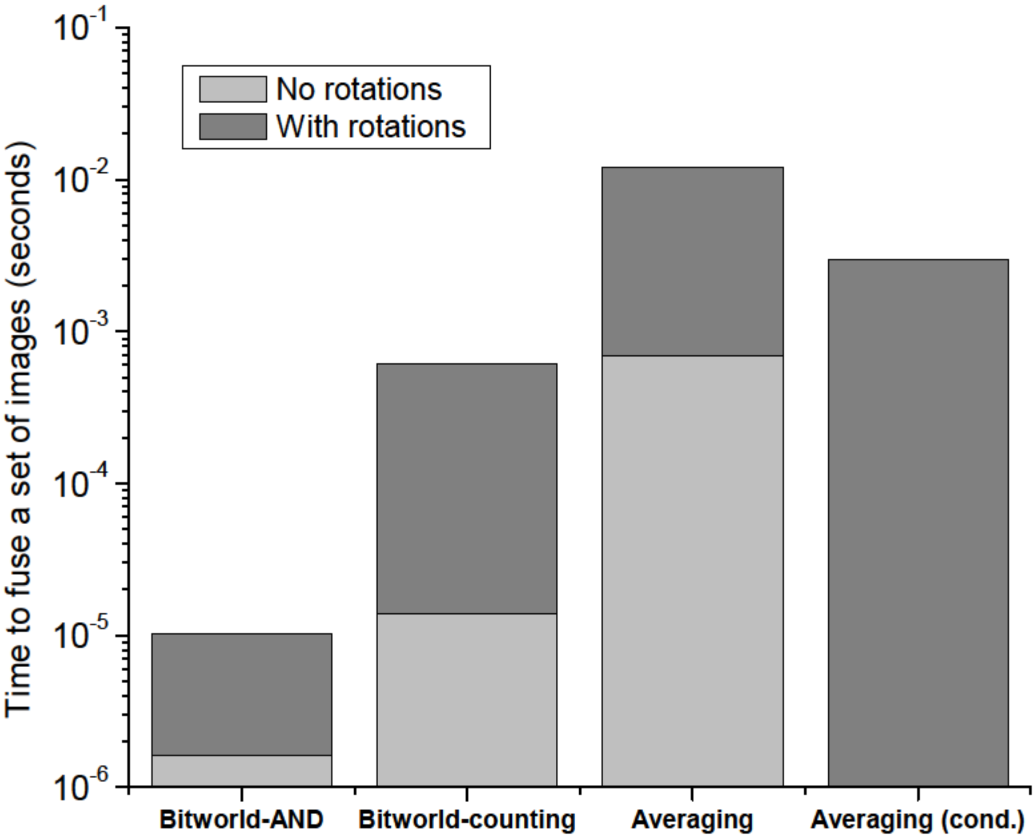


Figure 6.18: Graph showing the time to fuse the data using the three methods under comparison with additional rotated images. Relevant timings combination without rotations are shown for comparison, as well as the conditional averaging method of Chapter 5.

6.4. Conclusions

This chapter has examined the functional and computational performance characteristics of three methods of fusing low resolution images to produce ones of higher resolution. In the computational domain, the bitworld-AND has been shown to be the highest performing technique by factors of between 348 and 1335 times over a floating-point matrix representation.

The high inherent-parallelism of this representation and methodology contributes to this performance. The quality of the high resolution images produced by this method is qualitatively the same as the floating-point averaging method.

The additional computational cost of the floating-point averaging method also failed to afford significant advantages over the bitworld-counting method. This suggests that the bitworld-counting method can provide a balance of functionality and computational performance in the central processor of a robotic system. It also retains many of the data-centric advantages of bitworld, such as low bandwidth requirements, although more work will be required to assess how curvatures might be better identified.

Nonetheless, the pure bitworld-AND method was shown to be at least 70.5 times faster than the bitworld-counting method. As a result, there is still a small balance to be struck between functional and computational performance, but the performance of both methods is still a substantial improvement over a more conventional floating-point representation. Further work can examine methods of further improving the gains of the bitworld-counting method to achieve greater parity with the superior bitworld-AND method.

It has been assumed throughout this chapter that the pose of a robot is known so as to combine images effectively. In reality the pose of a robot must often be measured by sensors, which introduces the effects of noise into the fusion process. Adapting to this requires a refinement of the bitworld method which is explored in the next chapter.

Overall, this chapter has demonstrated the potential for adopting a data-centric approach to tactile sensing as advocated throughout this thesis. Low resolution sensor data can be combined using the efficient and compact bitworld representation to produce a higher resolution image. This responds directly to the problem of excessive data identified at the start of the chapter, albeit with the requirement to take more time over imaging areas of interest.

Chapter 7

Bitworld: Combining Noisy Bitworld Images

7.1. Introduction

7.1.1 Effects of Noise on Image Combination

The previous chapter showed that bitworld provides high computational performance alongside good qualitative results in the domain of tactile super-resolution. As part of these experiments, the location of the sensor was assumed to be known accurately so as to correctly align the images to a common axis.

When deployed in a robotic system, these positions will often be calculated from measurements of robot pose, which will be subject to sensor noise. In a multiple-link robot such as a snake-arm robot, multiple measurements of pose parameters like the angles between links will be necessary. The combination of multiple noisy measurements in particular can cause significant errors in determining the location of tactile sensors. If the noise is sufficient to affect the determined location of the origin of the tactile sensors, the implicit assumptions underlying the methods of Chapter 6 regarding re-alignment of the tactile images to a common axis are undermined.

The results of Chapter 6 demonstrate the high performance of bitworld in the presence of low-noise pose measurement, which covers a large proportion of robotic applications. This chapter explores the possibility of expanding the bitworld-related methods to use in noisier systems. This parallels development in robotic vision, where techniques such as occupancy grids were developed to combat error inherent to certain deployment scenarios.

The difference is that tactile sensing is highly localised, meaning that the principal source of error lies in the measurements of pose. The challenge is to do this in the compact representation of bitworld, since bitworld cannot store additional information in the same way as other representations. This chapter endeavours to minimise the effect of noise in tactile images whilst retaining the advantages of bitworld identified in Chapter 6.

7.1.2. Motivating Example

To demonstrate the effect of noise on a tactile image, the results of a simulation are informative. An object was simulated as impacting on a one-dimensional strip of tactels. In a real application, this tactile image would be converted to an image with a reference frame that is convenient to the robot. This requires translating the tactile image using the measured position of the sensor. To simulate the effect of noise, this position was simulated as being a random variable with a Gaussian distribution. The distribution had its mode at the true value of the position and a standard deviation of 0.7mm, which was chosen arbitrarily for the purposes of demonstration.

The object was simulated as impacting on the sensing strip between 3cm and 7cm on a 10cm tactile strip, with tactels 1cm in size. The simulation was run 1000 times and on each occasion a random location for the tactile sensor origin was sampled from the above distribution. The position of the object in the final image was determined from this value. Figure 7.1 shows a histogram of the believed occupied bitworld cells over all the simulations with locations placed in 1cm wide bins.

These results show that no single range of cells is always believed to be occupied under the simulated conditions. The peak at 5cm is 999 occurrences over the 1000 simulations, demonstrating that large perturbations caused by noisy pose measurements cannot guarantee that any image will be accurate.

The goal of this chapter is to minimise the tails exhibited in Figure 7.1 using the bitworld manipulation techniques described in Chapter 5. This will afford bitworld robustness in the case of noisy measurements of pose, while exploiting the high performance techniques described in Chapters 5 and 6.

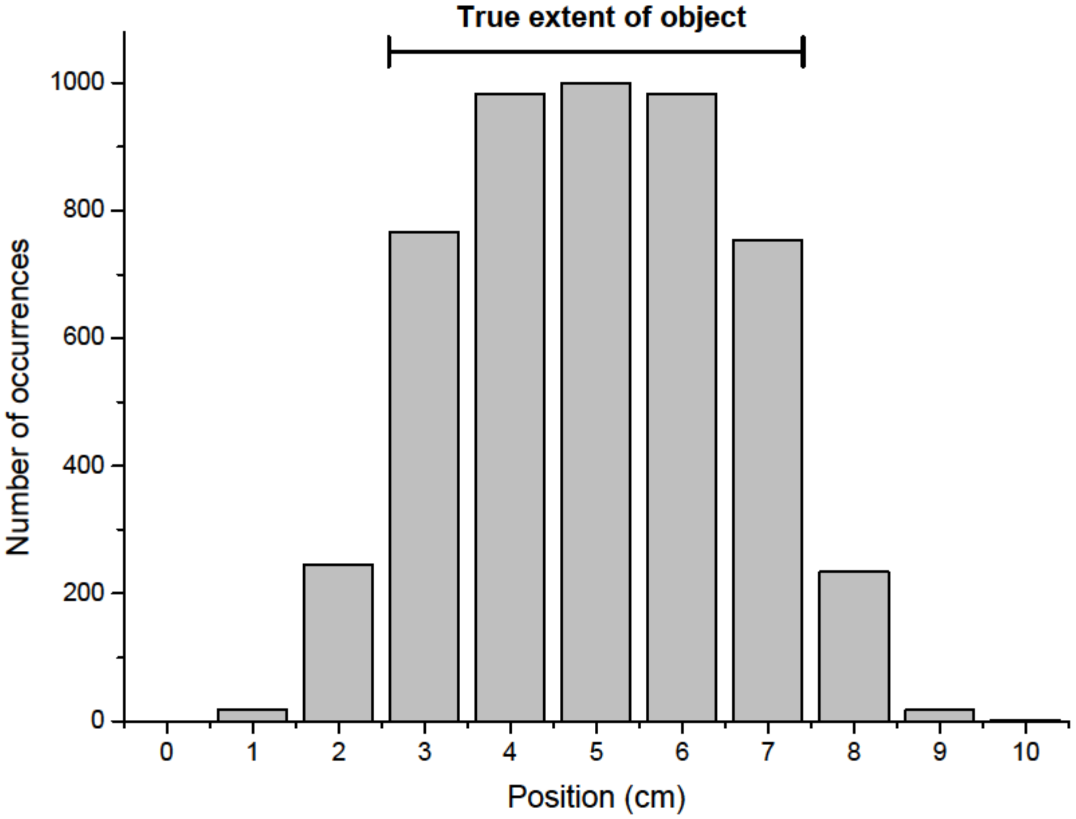


Figure 7.1: Results of simulating tactile imaging in one dimension where measurements of the position are subject to Gaussian noise with standard deviation of 0.7mm.

As further motivation, it is of interest to give a worked example of the levels of possible noise in measurements of robot pose. A worked example is given in Appendix A7.1 and suggests that standard deviations of less than 1.5mm are appropriate levels of study, particularly in the case of multi-link robots.

7.1.3. Assumptions and Definitions

In the three following sections, various techniques for minimising the effects of noise will be explored. In order to produce comparable results and in order to simplify the simulations, certain assumptions will be made.

Firstly, all results pertain to a one-dimensional tactile sensing strip. This makes interpreting the results simpler and does not affect the interpretation of the impact of each method. The goal of minimising noise will be replaced by an equivalent goal: maximising the chance of correctly identifying the position of the edge of an object on a tactile strip. Referring to the simulation in Section 7.1.2, this would mean minimising the tail results for 1cm, 2cm, 8cm, 9cm and 10cm and improving those for the range 3cm – 7cm.

The positional noise for all simulations is assumed to follow a Gaussian distribution with the mean position lying at its true value and the standard deviation being a variable parameter dependent on the specific system. This is chosen for its ubiquity, as it can become representative of other distributions for large numbers of samples. Crucially, the optimisation techniques described in subsequent sections do not rely on this specific distribution. The only restriction is the assumption that the distribution is unimodal and, in general, that the correct edge pixel has a higher probability of being correctly identified than its neighbouring cells in a single measurement. The calculation of a single measurement probability is illustrated in Figure 7.2 alongside the terminology that will be used throughout this chapter.

The neighbouring cells will have the highest probability of being erroneously selected as the correct edge cell. As a result, the goal of identifying the correct edge cell can be restated quantitatively as maximising the probability of selecting the correct edge cell while minimising the probability of erroneously deciding that a neighbouring cell is the correct edge cell.

The distribution in Figure 7.2 is shown as being centred on the correct edge cell rather than the origin of the tactile strip, which is where the random error occurs. This is valid because the edges and centres of the bitworld cells are defined as lying at fixed distances from the origin and so their positions follow the same random distribution as the origin. The probability of the correct edge cell being identified in a single measurement can be found by integrating the probability density function of the random variable between $\pm \boldsymbol{w}/\mathbf{2}$, where *w* is the width of a bitworld cell.

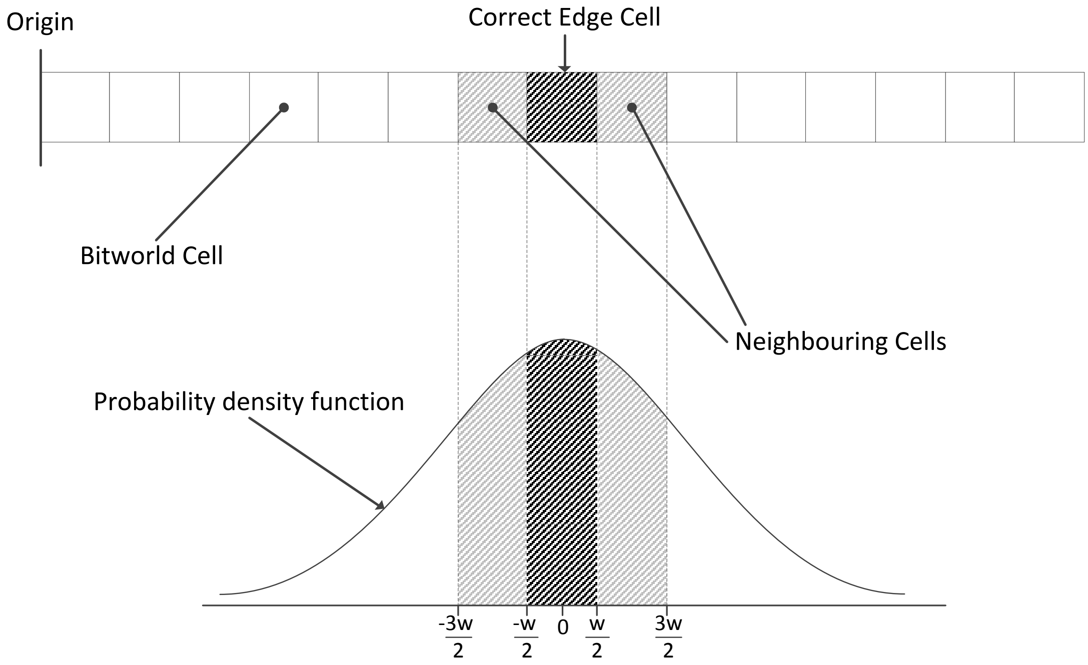


Figure 7.2: Illustration of the meaning of certain terminology used throughout this chapter. A one-dimensional bitworld strip of cells is shown, along with the relationship of the bitworld cell edges to the probability of that cell being selected.

These probabilities are used as the basis of the optimisation techniques described in the following sections. The techniques focus on taking multiple measurements, or images, and combining them onto a robot's internal representation of the environment.

7.2. Removing Noise from Bitworld

7.2.1. Bitworld fused to Bitworld

This section examines the case where a bitworld image is fused to a bitworld-based internal representation. Based on the results of Chapter 6, such a representation would not only be compact but also offer a high level of computational performance during the combination process. The purpose of this section is to try to determine the Boolean function that will combine a bitworld image to an internal representation and that gives the best probability of having a correct image in the end.

In the one-bit system of bitworld, each cell can contain only one of two possible states. This means there are only two probabilities associated with the occupancy of a particular cell.

For a bitworld cell with index i, the probability P_i^1 is the probability that the cell is measured as being occupied and P_i^0 is the probability that it is measured as unoccupied. These are related by the standard probabilistic condition for all cells that:

$$P_i^0 + P_i^1 = 1 \tag{7.1}$$

The first measured bitworld image will simply map directly to the internal representation, because the robot has no prior information on which to improve the measurement. This allows us to illustrate the problem with a probability tree such as that in Figure 7.3.

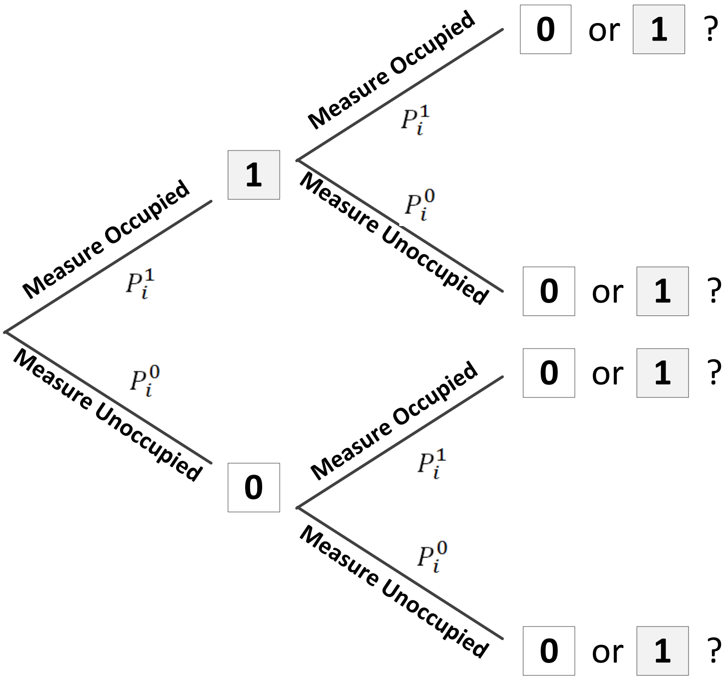


Figure 7.3: Probability tree illustrating the problem being solved in this chapter.

This figure shows the initial measurement giving an initial probability of the cell being believed to be occupied of P_i^1, as expected. The second measurement is shown leading to uncertain values, namely the question of whether the belief should be in an occupied or an unoccupied cell based on particular measurements? An examination of the probabilities in the general case will provide an answer.

After combining a second image with the internal representation using some Boolean function that we are seeking, the probabilities of a bitworld cell *i* being in one of the two states can be calculated from

$$\begin{pmatrix} P_i^{1\prime} \\ P_i^{0\prime} \end{pmatrix} = \begin{pmatrix} f_{11}(P_i^0, P_i^1) & f_{01}(P_i^0, P_i^1) \\ f_{10}(P_i^0, P_i^1) & f_{00}(P_i^0, P_i^1) \end{pmatrix} \begin{pmatrix} P_i^1 \\ P_i^0 \end{pmatrix} \qquad (7.2)$$

The values of $P_i^{1\prime}$ and $P_i^{0\prime}$ denote the probability of the same bitworld cell in the internal representation being labelled as occupied or unoccupied respectively. The values of *f_{ab}* are determined by the Boolean function used to combine the image and the internal representation. Their value will be equal to the sums of the probabilities of making measurements that will map from state *a* to state *b*. Consider an arbitrary Boolean function that leads to the probability tree in Figure 7.4.

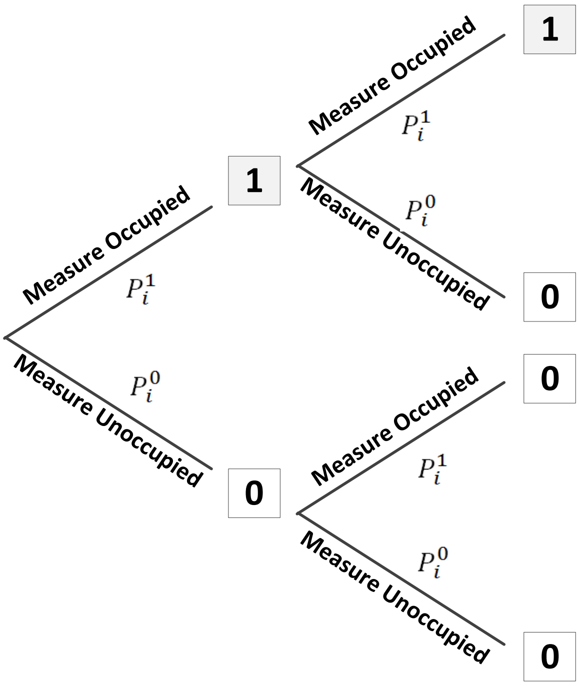


Figure 7.4: Probability tree produced by the arbitrary choice of a Boolean function that produces the final outcomes shown.

This shows that if, as in this example, a cell is in state 1 in the internal representation, the only way for it to remain in this state after combination is if the measurement also has this cell in state 1. For this particular Boolean function, therefore, $f_{11}(P_i^0, P_i^1) = P_i^1$. By similar reasoning, it can be seen that for the example function illustrated in Figure 7.4, $f_{10}(P_i^0, P_i^1) = P_i^0$, $f_{01}(P_i^0, P_i^1) = 0$ and $f_{00}(P_i^0, P_i^1) = P_i^0 + P_i^1 = 1$.

For the sake of brevity, the functions *f_{ab}* hereafter will be written without their input variables and it is assumed that each function takes all state probabilities as their inputs.

By definition for a one-bit bitworld, each f_{ab} can only have one of a discrete set of values given by $\{0,\ \ P_i^0,\ \ P_i^1,\ \ 1\}$, because of the limited choices available from the measurements. The choice of optimum outputs for the combination process will determine the precise mapping described by the f_{ab}. This mapping must be the same for all bitworld cells given that it is desirable to perform the combination using bitwise operations that necessarily apply to all cells simultaneously. The columns of the matrix in (7.2) must also add up to one for each cell to satisfy the summation to unity, so that for all cells

$$f_{00} + f_{01} = f_{10} + f_{11} = 1 \qquad (7.3)$$

These conditions restrict the number of possible transition matrices to sixteen.

This number can be reduced further by stating that allowing any f_{ab} to map to a value of one is not optimal. If values of 1 were permitted, it would imply that once a cell had attained a particular state, no further measurement could remove it from that state. Erroneous measurements would then be allowed to accumulate on all bitworld cells, which is clearly sub-optimal.

Removing the possibility of including values of 1 in the transition matrices restricts their number to four. These matrices are given in equations (7.4) - (7.7) below.

$$\begin{pmatrix} P_i^1 & P_i^1 \\ P_i^0 & P_i^0 \end{pmatrix} \qquad (7.4)$$

$$\begin{pmatrix} P_i^1 & P_i^0 \\ P_i^0 & P_i^1 \end{pmatrix} \qquad (7.5)$$

$$\begin{pmatrix} P_i^0 & P_i^1 \\ P_i^1 & P_i^0 \end{pmatrix} \qquad (7.6)$$

$$\begin{pmatrix} P_i^0 & P_i^0 \\ P_i^1 & P_i^1 \end{pmatrix} \tag{7.7}$$

In order to provide a successful transition, the transition matrix must satisfy two criteria:

1. Increase the probability of selecting the correct edge cell with index *c*, so that $P_c^{1'} > P_c^1$. This simultaneously requires that $P_c^{0'} < P_c^0$, given the relationship between the two probabilities.

2. For all other bitworld cells, their probabilities of occupancy after combination must be less than that of the correct edge cell.

The first criterion restates the general goal of improving the probabilities of having an accurate internal representation after image combination. This could, however, be satisfied simply by setting the value of every bitworld cell to 1, regardless of the measurements. The second criterion is required to prevent this undesirable case.

It can be shown that applying the matrices in equations (7.4) to (7.7) does not satisfy either of these criteria except in the case that the probability of selecting the neighbouring cell is higher than selecting the correct edge cell. This is demonstrated in Appendix A7.2. Since this type of probability distribution is unlikely to be particularly prevalent in the absence of systematic error, it is concluded that combining a one-bit bitworld measurement onto a one-bit internal representation would be insufficiently robust in the presence of noise.

7.2.2. Improvements using Counting

Although the combination of bitworld measurements to a one-bit internal representation does not appear to be very successful, it may still be possible to utilise one-bit measurements. The previous technique implicitly requires that only one measurement was processed at a time. This section will explore combining multiple bitworld images to a one-bit internal representation. Here it is assumed that the tactile system is given sufficient time to take *n* measurements from each sensor, accompanied by the same number of noisy measurements of robot pose that are used to align them.

This method essentially counts the number of times each cell in the aligned measurements equals one. The corresponding cell on the internal representation is assigned a value of one if this number exceeds a given threshold. Determining the threshold is achieved by attempting to maximise the difference between the probability of the correct edge cell and that of the neighbouring cell being selected.

This threshold, *k*, is given in equation (7.8) and the derivation can be found in Appendix A7.3.

$$k = \left\lceil \frac{n \ln\left(\dfrac{1-P_W^1}{1-P_C^1}\right)}{\ln\left(\dfrac{P_C^1}{P_W^1}\right) + \ln\left(\dfrac{1-P_W^1}{1-P_C^1}\right)} \right\rceil \tag{7.8}$$

Using appropriate values of *k* derived from this equation, Figures 7.5 and 7.6 show how the probabilities of selecting the correct edge cell and the neighbouring cells in a combination of multiple bitworld images changes for different numbers of measurements, *n.*

Different initial probabilities were generated from Gaussian probability density functions with different standard deviations. Using a bitworld resolution of 1mm, the probability for the correct edge cell was produced by integrating between -0.5mm and 0.5mm, whilst the probability of selecting the neighbouring cell comes from integrating between 0.5mm and 1.5mm.

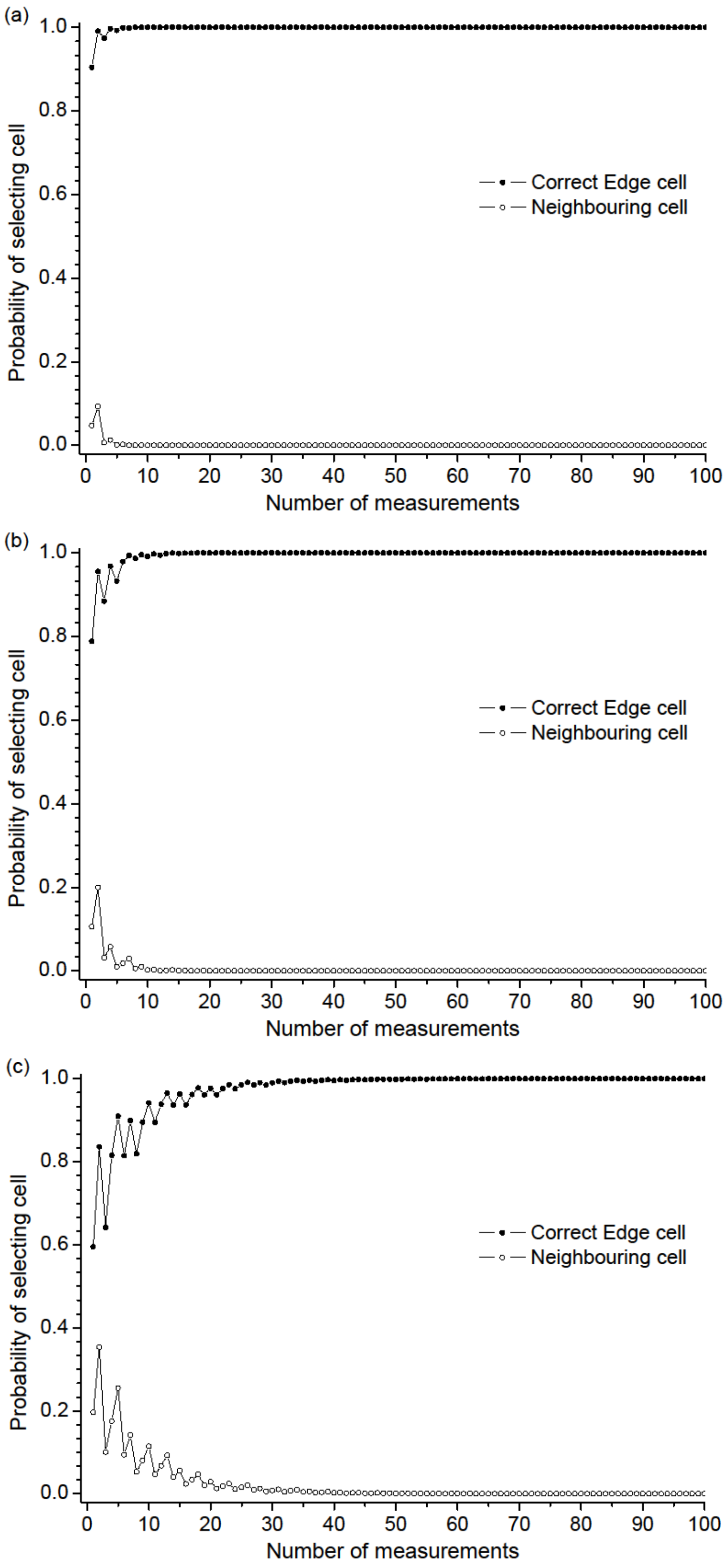


Figure 7.5: Graphs showing the change in the probability of occupancy for the correct edge cell and the neighbouring cell. Initial probabilities based on Gaussian distributions with standard deviations of (a) 0.3mm (b) 0.4mm (c) 0.6mm

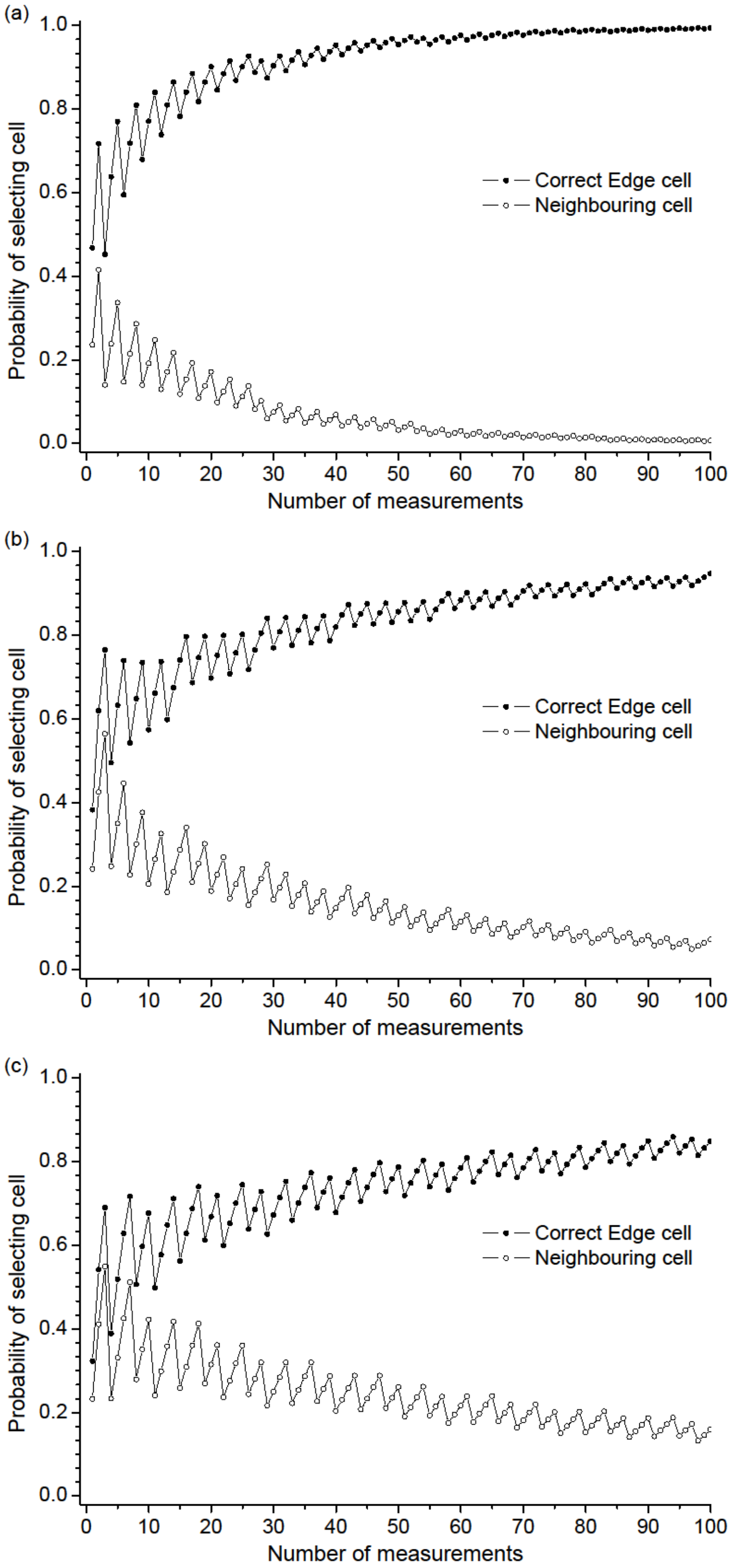


Figure 7.6: Graphs showing the change in the probability of occupancy for the correct edge cell and the neighbouring cell. Initial probabilities based on Gaussian distributions with standard deviations of (a) 0.8mm (b) 1.0mm (c) 1.2mm

Figure 7.5 shows the results for the Gaussian distribution with standard deviations of 0.3mm, 0.4mm and 0.6mm. In all three cases, the probability of selecting the correct edge cell tends very quickly to unity with increasing numbers of measurements. Similarly, the probability of selecting the neighbouring cells trends very quickly to zero The wide separation of the probabilities of selecting the correct edge and neighbouring cells for low numbers of measurements gives wide scope for making a safe choice of *k*.

These beneficial effects are still present in Figure 7.6 for larger standard deviations, but the trending effect is much slower. The non-monotonic behaviour present in both Figure 7.5 and 7.6 arises from the discrete nature of the threshold. In systems where only a fixed number of measurements are available, designers may wish to discard some measurements to avoid local minima in the probability of selecting the correct edge cell.

Combining these images using bitwise operations is simple, and the correct combination function can be found by specifying an appropriate truth table and minimising the resulting Boolean equation. For large numbers of measurements, the number of terms in the equation becomes large as it is equal to the number of ways of choosing *k* from *n*. For larger numbers of measurements, an intermediate representation such as that used for implementing the bitworld-counting method in Chapter 6 would be useful.

The Boolean functions can only represent a single threshold for each combination. If the system fails to even provide a single one of these images due to delay or some other real-world effect, then the operation will be unable to adapt. This could be mitigated by employing multiple combination functions in a routine to allow the robot controller to choose the appropriate combination mechanism.

The *k*-value could be calculated online based on the number of samples actually returned. The cells could then simply be added together to produce the required values for comparison. The results of Chapter 6 show that the bitworld-counting method provides good computational performance and this could be employed for this technique. Such a representation makes a good intermediate step for this technique, because it allows the value of *k* to be determined online and an appropriate number of samples chosen to prevent falling into a local minimum.

7.2.3. Simulation

To demonstrate the effect of applying the techniques developed in Section 7.2.2, the simulation of Section 7.1.2 was repeated. An object was simulated as impacting upon a one-dimensional 10cm tactile sensing strip with a 1cm resolution. The object was simulated as lying between 3cm and 7cm on the tactile strip.

The error in determining the position of the origin of the tactile sensing strip was simulated as a random variable with a Gaussian distribution with a standard deviation of 0.6mm. The simulation was run 1000 times and on each occasion a random location for the tactile sensor origin was sampled from this distribution. The believed position of the object was then determined from this value. Figure 7.7 shows a histogram of the believed occupied ranges over all the simulations performed with locations placed in 1cm wide bins without using any optimisation.

To produce Figure 7.8, 60 measurements were simulated in a single simulation run. Applying equation (7.8) gives a *k*-value of 23 to use as a threshold. These 60 measurements were combined using the techniques described in Section 7.2.2 so that if a bitworld cell was recorded as being occupied in 23 measurements or more, this bitworld cell was set to one in the resulting image. For comparison, this process was executed 1000 times and Figure 7.8 shows a histogram of the believed occupied ranges over all the simulations with locations placed in 1cm bins.

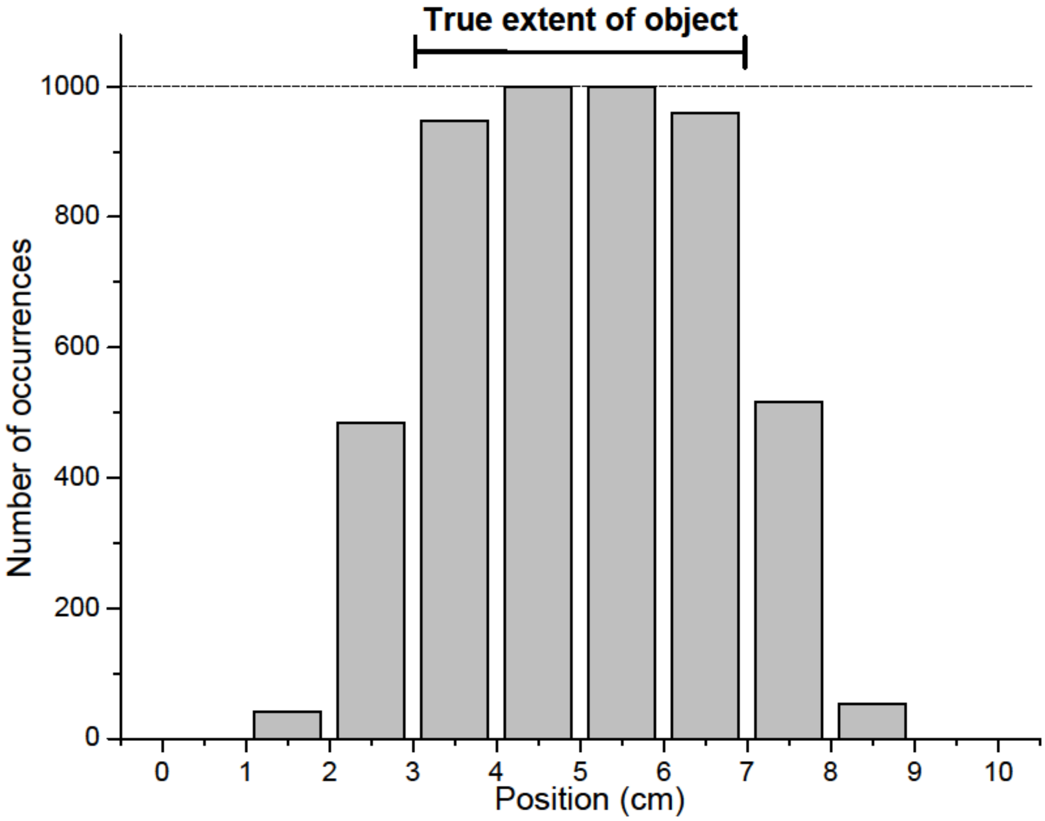


Figure 7.7: Results of simulating tactile imaging in one dimension over 1000 simulations where measurements of the position are subject to Gaussian noise with standard deviation of 0.6mm.

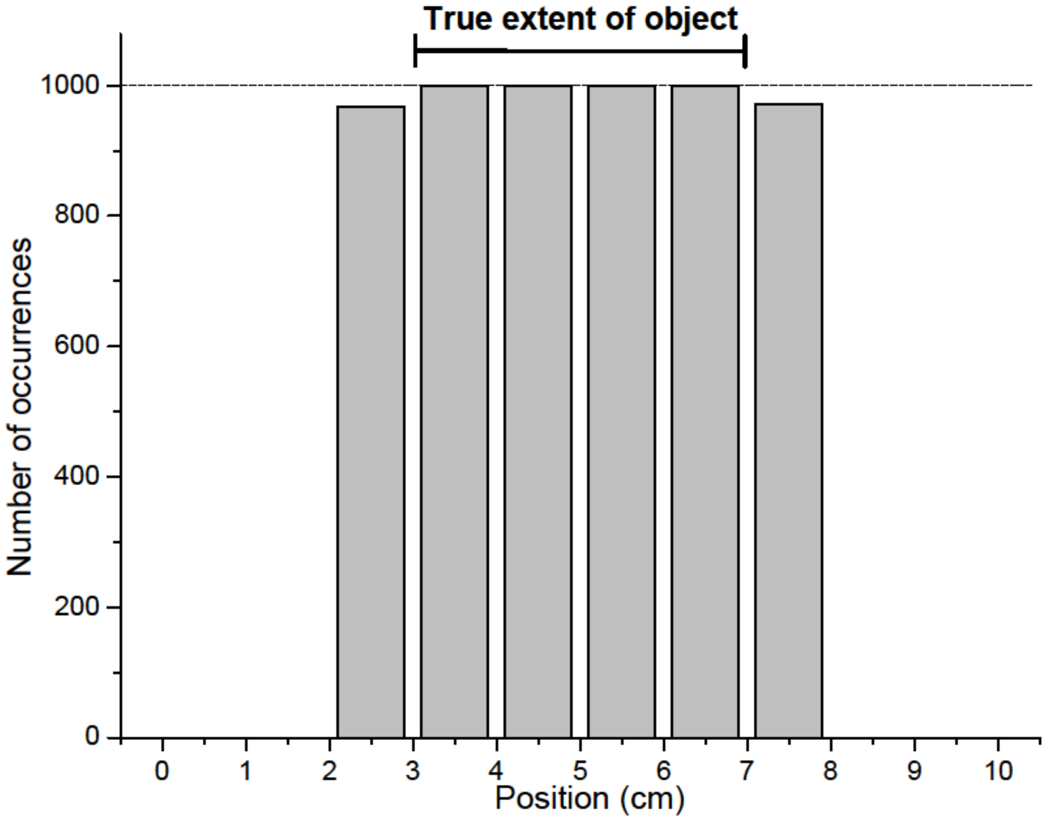


Figure 7.8: Results of applying *k*-value thresholds to tactile images in one dimension over 1000 simulations where measurements of the position are subject to Gaussian noise with standard deviation of 0.6mm.

Figure 7.8 shows that applying the appropriate *k*-value threshold to the tactile images means that the correct occupied positions are identified in all simulations, compared to the results of Figure 7.7 which suggest that this statement is only true 97% of the time. Application of the threshold has also removed the lower order noise.

It appears that bitworld cells either side of the true extent of the object also have their probabilities of appearance in the final representation amplified. This occurs because the object was simulated as lying at the 3cm mark. Consequently, the random variation in sensor location gave a high probability that the neighbouring cells would be selected in a single measurement.

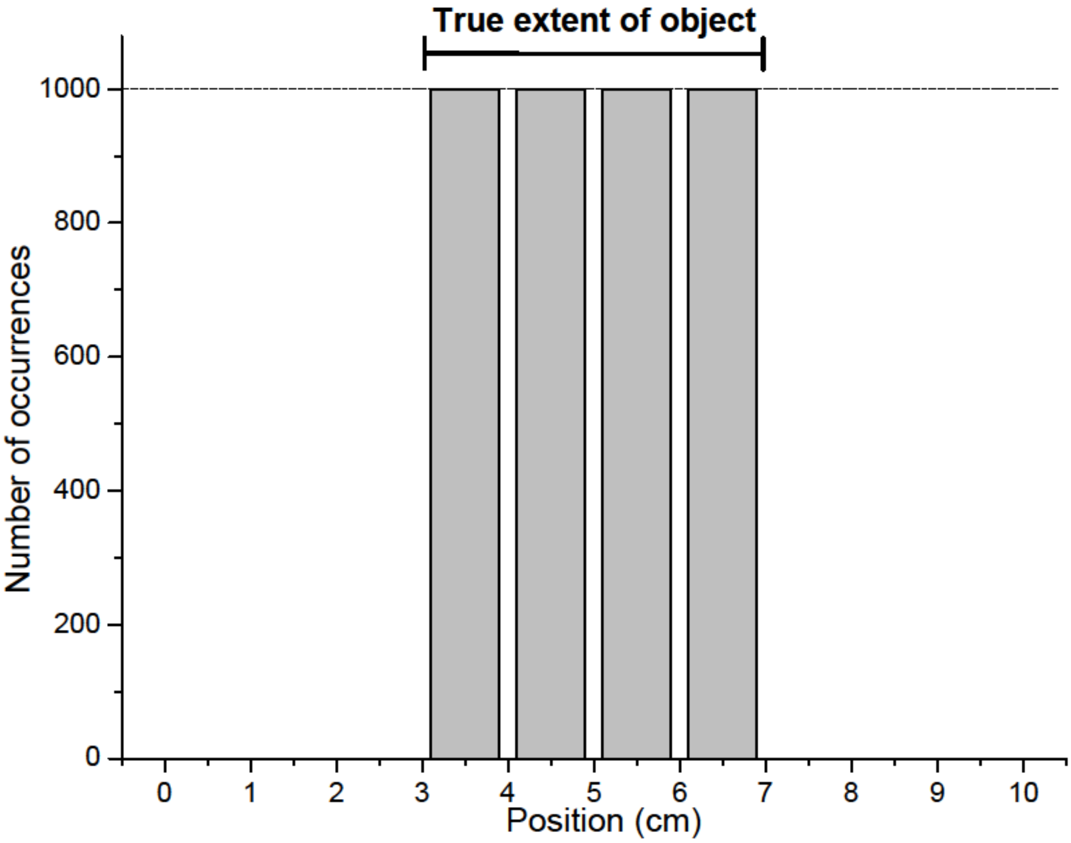


Figure 7.9: Result of applying a bitwise-AND filter between 100 combined images produced using the *k*-value threshold technique with *k*=23 on 60 tactile measurements. The simulation was repeated 1000 times.

The fact that the neighbouring cells only appeared in 96% of the simulations, suggests a further filtering step could enhance these images. By repeating the combination of the 60 images multiple times, a set of bitworld images is produced that Figure 7.8 indicates should all contain the correct values but not always contain the incorrect neighbouring cells that appear. By applying a bitwise-AND between these images, the bitworld cells that are not occupied in each image will be excluded from the final result. This will leave only the correct bitworld cells in the final image, which could be used for the internal representation.

This procedure was simulated using the same methods described previously. For each simulation, the combination of 60 images using a *k*-value threshold of 23 was repeated 100 times and a bitwise-AND applied between the images.

This simulation was repeated 1000 times and the results are shown in Figure 7.9. This confirms that in all simulations performed to generate this data, the internal representation would have been updated to include only those bitworld cells that were genuinely occupied. In effect, the combination of the *k*-value threshold with the bitwise-AND filter has completely eliminated the effects of noise observed in the baseline example of Figure 7.7.

This did require 6000 measurements for each simulation, rather than the 60 required to produce Figure 7.8. It is nonetheless clear that this has had a significant impact and could be a much lower figure if the standard deviation were lower.

7.2.4. Discussion

This section has demonstrated an effective means of reducing errors from tactile images caused by the presence of positional noise. This has been achieved using bitworld-based techniques and bit-counting as successfully utilised in Chapter 6.

The results have shown that the *k*-value threshold technique works well for Gaussian distributions with a standard deviation of 0.6 or lower. With higher standard deviations, the number of measurements required to make the probability of selecting the correct edge cell approach unity becomes substantially higher and the advantages of the *k*-value technique are diminished.

The results also showed that applying the *k*-value threshold can cause oscillation in these probabilities arising from the discrete nature of sampling. This is particularly apparent with lower numbers of measurements and high standard deviations. System designers can overcome this by restricting their sample sizes to one of the peak values in probability charts such as those in Figures 7.5 and 7.6.

This technique does not offer a way of incrementally improving on earlier results using subsequent measurements. This is acceptable if the desired level of confidence can be achieved in a single combination of multiple measurements onto the internal representation. This means that subsequent measurements must either replace the existing data or be ignored.

It can be concluded that this method is appropriate in cases where the positional noise is low, where a sufficient number of samples can be acquired in the allocated time to reach the desired confidence level in the image or where the compactness of the one-bit internal representation outweighs any of the described restrictions.

These techniques successfully extend bitworld into the domain of measurements affected by noise. The next two sections will describe efforts to improve upon this by relaxing the condition that bitworld only holds one bit per bitworld cell.

7.3. Two-bit Linear Optimisation

7.3.1. Extension of Matrix Description

This section will explore the extension of the bitworld model to utilise multiple bits per cell. The additional bits are used to allow a cell to be labelled with a level of belief regarding its occupancy. In a two bit system, for example, the four possible states can be thought of as representing beliefs that the cell is definitely, probably, probably not or definitely not occupied. In order to optimise the Boolean combination of multiple measurements under this extended representation, it is necessary to extend the mathematical description of the changing probabilities of cell occupancy.

This is achieved by extending the matrix description of the bitworld model. For a representation consisting of *N* bits, the total number of possible states is $S = 2^N$. A single update in matrix notation can therefore be written as

$$\begin{pmatrix} P_i^{1\prime} \\ \vdots \\ P_i^{S\prime} \end{pmatrix} = \begin{pmatrix} f_{1,1} & \cdots & f_{S,1} \\ \vdots & \ddots & \vdots \\ f_{1,S} & \cdots & f_{S,S} \end{pmatrix} \begin{pmatrix} P_i^{1} \\ \vdots \\ P_i^{S} \end{pmatrix} \tag{7.9}$$

where $f_{a,b}$ represents the probability of the current state transforming to *b* given that it is currently *a*. As discussed in Section 7.2.1 when explaining these values in equation (7.2), these quantities can take any linear sum of the various P_i^n as their value, including zero and one, provided that the columns of the matrix add up to one.

The desired optimisation remains as before, namely to maximise the probability of the correct edge cell being chosen whilst minimising the probability that neighbouring cells are chosen as occupied. Whilst it was previously straightforward to assess the one-bit systems analytically, the number of possible combinations for the functions $f_{a,b}$ rendered direct analysis difficult to achieve for increasing values of *N*.

The approach to finding an optimal Boolean function for image combination that is considered in this section is to re-write the update process in the form of a linear optimisation problem with appropriate constraints in order to exploit existing techniques for optimisation. In order to optimise the update process in this way, the problem will need to be restated.

Let equation (7.9) be rewritten as

$$\underline{P}' = \underline{\underline{F_P}}\,\underline{P} \tag{7.10}$$

where

$$\underline{P}' = \begin{pmatrix} P_i^{1\prime} \\ \vdots \\ P_i^{S\prime} \end{pmatrix}, \quad \underline{\underline{F_P}} = \begin{pmatrix} f_{1,1} & \cdots & f_{S,1} \\ \vdots & \ddots & \vdots \\ f_{1,S} & \cdots & f_{S,S} \end{pmatrix}, \quad \underline{P} = \begin{pmatrix} P_i^{1} \\ \vdots \\ P_i^{S} \end{pmatrix} \tag{7.11}$$

represent the different components of (7.10). Writing the probability vectors for the correct edge cell and for the neighbouring cell as

$$\underline{p} = \begin{pmatrix} P_C^1 \\ \vdots \\ P_C^S \end{pmatrix} \text{ and } \underline{q} = \begin{pmatrix} P_W^1 \\ \vdots \\ P_W^S \end{pmatrix} \tag{7.12}$$

respectively, it becomes possible to rewrite the optimisation goal as maximising a function composed from the elements of a vector $\underline{r}$, defined as

$$\underline{r} = \begin{pmatrix} r_1 \\ \vdots \\ r_S \end{pmatrix} = \underline{\underline{F_p}}\,\underline{p} - \underline{\underline{F_q}}\,\underline{q} \tag{7.13}$$

Here, $\underline{\underline{F_p}}$ and $\underline{\underline{F_q}}$ have the same form as $\underline{\underline{F_P}}$ in equation (7.11). The difference is in the values of $f_{a,b}$, which must be a linear sums of the various P_C^n for $\underline{\underline{F_p}}$, and a linear sum of the various P_W^n for $\underline{\underline{F_q}}$. Crucially the form of the sum must be the same between the two matrices, because they represent the same Boolean operation used to combine the images. For instance, if $f_{1,1} = P_C^1 + P_C^3$ in $\underline{\underline{F_p}}$, then for $\underline{\underline{F_q}}$ it is required that $f_{1,1} = P_W^1 + P_W^3$.

To achieve the optimisation goal described previously, the goal function will simply be $r_1 - r_S$. In general, the exact goal function may need to be adjusted based on the application. The form of (7.13) is still not appropriate for optimisation since the optimisation must select the correct form of the functions in $\underline{\underline{F_p}}$ and $\underline{\underline{F_q}}$. Further complicating this is the need for the forms of the functions in these matrices to be identical.

Fortunately, all of these functions are linear combinations of probabilities, allowing these matrices to be written as the product of two other matrices. For example, in a two-bit system with four possible states per cell, the matrix $\underline{\underline{F_p}}$ can be rewritten as

$$\underline{\underline{F_p}} = \underline{\underline{CP}} = \begin{pmatrix} c_{1,1} & \cdots & c_{1,16} \\ \vdots & \ddots & \vdots \\ c_{16,1} & \cdots & c_{16,16} \end{pmatrix} \begin{pmatrix} P_C^1 & 0 & 0 & 0 \\ P_C^2 & 0 & 0 & 0 \\ P_C^3 & 0 & 0 & 0 \\ P_C^4 & 0 & 0 & 0 \\ 0 & P_C^1 & 0 & 0 \\ 0 & P_C^2 & 0 & 0 \\ 0 & P_C^3 & 0 & 0 \\ 0 & P_C^4 & 0 & 0 \\ 0 & 0 & P_C^1 & 0 \\ 0 & 0 & P_C^2 & 0 \\ 0 & 0 & P_C^3 & 0 \\ 0 & 0 & P_C^4 & 0 \\ 0 & 0 & 0 & P_C^1 \\ 0 & 0 & 0 & P_C^2 \\ 0 & 0 & 0 & P_C^3 \\ 0 & 0 & 0 & P_C^4 \end{pmatrix} \tag{7.14}$$

where all $c_{a,b} \in \{0, 1\}$. This allows matrix $\underline{\underline{C}}$ to represent the form of the functions in both $\underline{\underline{F_p}}$ and $\underline{\underline{F_q}}$ so that equation (7.13) may be rewritten as:

$$\underline{r} = \underline{\underline{C}}\left(\underline{\underline{P}}\,\underline{p} - \underline{\underline{Q}}\,\underline{q}\right) \tag{7.15}$$

The bracketed term in equation (7.15) will be constant for a given system, since all the elements of the matrices $\underline{\underline{P}}$ and $\underline{\underline{Q}}$ as well as the vectors $\underline{p}$ and $\underline{q}$ are determined by the probabilities of single measurements and do not change. This leaves only the choice of the matrix elements of $\underline{\underline{C}}$ as variables to be optimised.

This has transformed the problem into a binary integer programming problem to which only a single type of constraint must be applied. This constraint specifies that the sum of each column in matrix $\underline{\underline{C}}$ must be equal to one. This prevents certain types of measurement being ignored in the optimisation process, as well as preventing the impossible outcome of a single measurement being mapped to two different states.

The binary integer optimisation routine for this work was implemented using the ‘bintprog’ function from MATLAB's Optimization Toolbox. This minimises a goal function subject to specified equality and inequality constraints where all variables are binary. The function implements a branch-and-bound algorithm to search the possible outcomes. In the worst case, this function reduces to a brute force search among all possible combinations of variables, but on average this algorithm performs the task much more quickly.

7.3.2. Implementation for one-bit measurements

In order to retain the bandwidth benefits of the bitworld representation it is desirable to retain the one-bit representation for transmission if possible, reserving the multiple-bit representation for the internal representation. Here and for the remainder of this chapter, two bits per cell will be used for the internal representation.

In a two-bit representation, the initial probability that a cell is in one of the 4 available states has to be determined. The state representing ‘definitely occupied’ will take the value of the probability that the cell is measured as occupied. The state representing ‘definitely not occupied’ will take the probability that the cell is measured as unoccupied, and the other two states will have zero probability initially.

However, applying the optimisation to the same distributions described in Section 7.3.1 suggests that improving the probability of selecting the correct edge cell typically causes the probability of the neighbouring cell to increase as well. For reference, these results are presented in Appendix A7.4. This is in keeping with the analysis of Section 7.2.1, which showed that improvements in probabilities arising from combining one-bit measurements could not be achieved under normal circumstances. The results of this optimisation suggest that this finding applies to combining one-bit measurements onto two-bit systems as well.

Rather than rely on combining single measurements, the use of a similar approach to the *k*-value threshold methodology of Section 7.2 is now considered. Multiple measurements are taken and combined together using a threshold method to produce a two-bit bitworld image that is then combined onto the internal representation. The difference between this approach and the *k*-value threshold method is that multiple thresholds are used to allocate one of the four possible states to each bitworld cell.

For this work, the states were allocated according to the following scheme:

- State 1 ('definitely occupied') was allocated in the same way as in the one-bit system, using the *k*-threshold given by equation (7.8) as a threshold.

- State 4 ('definitely not occupied') was allocated to all bitworld cells whose counts were less than or equal to a value m, where m is the value of *i* required to minimise $(P_C^1)^i(1-P_C^1)^{n-i} - (P_W^1)^i(1-P_W^1)^{n-i}$.

- The remaining $(k-m)$ probabilities are shared between States 2 ('probably occupied') and 3 ('probably not occupied') equally. If $(k-m)$ is odd, the split is unequal in favour of state 3.

The aim of this allocation scheme was to increase the difference between the probabilities of identifying the correct edge cell and incorrect neighbouring cells as occupied in the combined set of measurements. This is necessary to maximise the probability that the internal representation is correct if only a single set of measurements can be obtained.

However, applying the linear optimisation using this scheme gave results that were at best the same as simply applying the *k*-value threshold method. Figures 7.10 and 7.11 show examples of the two typical behaviours observed. For standard deviations below 0.6, the probability of selecting the neighbouring edge cell falls as previously, but then quickly ascends to unity.

This occurs because the optimisation function is no longer able effectively to minimise the difference between the probability of selecting the correct edge cell and that of selecting the neighbouring cell. Satisfying the requirement that the probability of selecting the correct edge cell must improve forces an undesirable increase in the probability that an incorrect cell is chosen. This results in the large upward movement in the probability of selecting a neighbouring cell that is observed.

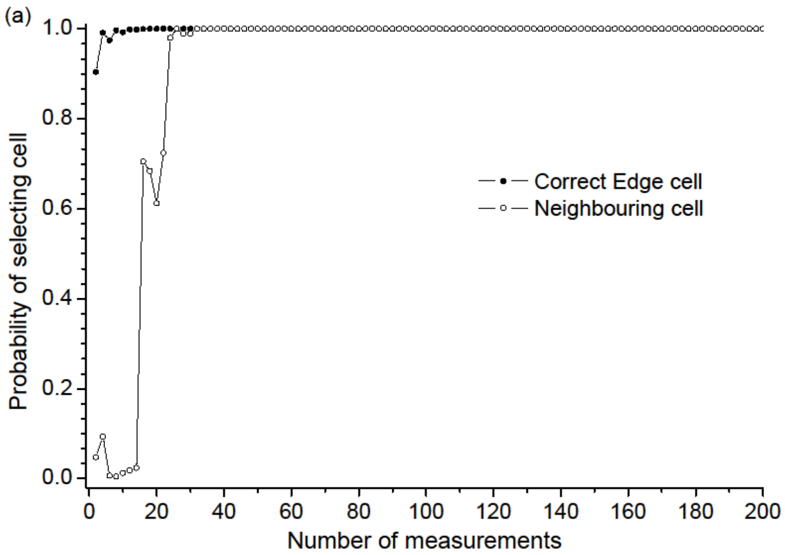


Figure 7.10: The change in the probability of occupancy for the correct edge cell and neighbouring cell versus numbers of measurements combined. Initial probabilities based on a Gaussian distribution with standard deviations of 0.2.

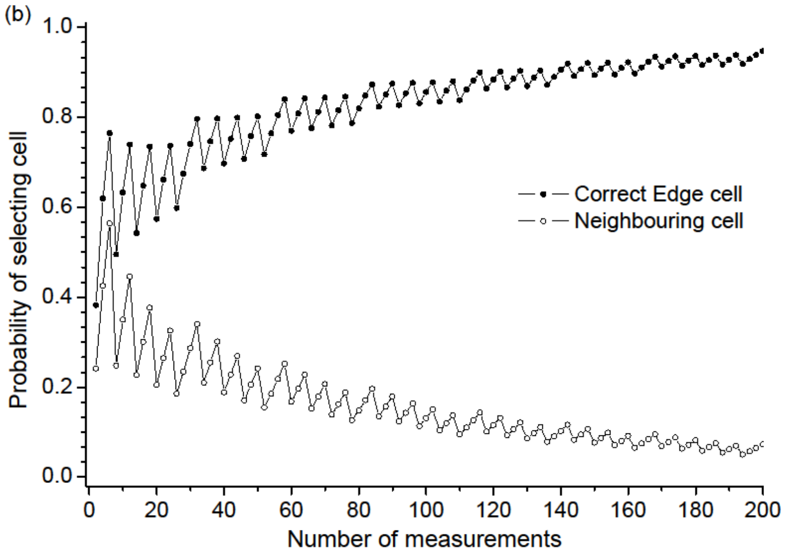


Figure 7.11: The change in the probability of occupancy for the correct edge cell and neighbouring cell versus numbers of measurements combined. Initial probabilities based on a Gaussian distribution with standard deviations of 1.0mm.

At higher standard deviations, Figure 7.11 shows that the same series of results is achieved as when using the *k*-value threshold method, except that twice as many measurements are required to produce the same changes.

These results, the full set of which is presented in Appendix A7.5, shows that improvements to the internal representation from combinations of one-bit measurements cannot exceed those achieved by the *k*-value threshold method.

These linear optimisation methods also fail to accommodate additional measurements after the second measurement is combined and the next sub-section will consider a method that attempts to partially address these failures.

7.3.3. Implementation using intermediate knowledge

To attempt improvements over the results of the previous section, single-bit measurements will be combined to form multiple intermediate bitworld images using the threshold technique described in Section 7.3.2. This process has the side-benefit of allowing the system to be robust against a failure to provide the number of measurements for which the update mechanism has been configured.

Since the optimisation is linear, only a single combination can be optimised at a time. This means that it is possible for a different Boolean function to be judged optimal at each combination step. This has the advantage of allowing the system to exploit existing knowledge to improve the accuracy of the internal representation. On the other hand, it constrains the system to optimising the internal representation on a single visit to the object, since bitworld cannot hold information on how often a cell has been visited.

In order to optimise combinations of images using this, two parameters must be specified: the total number of measurements N_S that the system must obtain for combining into the internal representation, and the number of these measurements, N_i, to be used for each incremental step. An integer number of combination steps, N_C, will be made such that $N_S = N_C \times N_i$.

The choice of values for these parameters is constrained by the system and the time available for a set of measurements to take place. There is also a trade-off involved in the selection of N_i, since a lower value will enable more incremental updates and greater robustness to system errors, but a greater value will offer a better initial probabilistic separation between the correct edge cell and the neighbouring cell. This is because a greater value of N_i means more bins over which the value of the *k*-value can be calculated from equation (7.8) and so the greater distinction between the probabilities of the correct edge cell and the neighbouring cells.

Given the constraint that N_i must be a factor of N_S it is relatively simple to explore the various possibilities and assess their merits. The optimisation of this kind of system is similar to that for the single update. However, the results of the previous optimisation recursively form the inputs for the next optimisation, replacing the values of $\underline{p}$ and $\underline{q}$ in equation (7.13), but not affecting the matrices $\underline{\underline{P}}$ and $\underline{\underline{Q}}$, which use the original probability values.

Since multiple update functions are now produced by the optimisation, it is possible to use different optimisation constraints or goals for each step. One such step arises from the observation that the goal of maximising r_1 as defined in equation (7.15) is not always compatible with the sub-goal of minimising $P_W^1{}'$, which is the probability of having selected the neighbouring cell after combination. In pre-simulation tests, it was found that the latter could be achieved by explicitly requiring the optimisation to obey the condition $P_W^1{}' < P_W^1$ at different stages of the optimisation.

The MATLAB code for performing this optimisation is shown Appendix A7.6. The application of the extra constraint is another variable, and the benefit of its use is explored by the optimisation routine.

Standard deviations of 0.6 to 1.2 are chosen for the optimisations here, since the lower standard deviations have such high initial probabilities that optimisation is probably unnecessary as the results of Figures 7.5 and 7.10 indicate. The optimisations were performed with $N_S = 50$ for these standard deviations with $N_i = 5$, 10 and 25 to assess how altering this value affects the end result.

The results for these optimisations are illustrated graphically in Appendix A7.7. The comparison with the equivalent *k*-value threshold method is also shown. A direct comparison between these results is shown in Table 7.1 for N_i= 5.

	Probability of selecting the relevant cell after applying method			
Standard deviation	Correct Edge cell (*k*-value)	Correct Edge cell (linear opt.)	Neighbouring cell (*k*-value)	Neighbouring cell (linear opt.)
0.6	0.9093	0.8269	0.2543	0.0649
0.8	0.7700	0.7406	0.3368	0.1146
1.0	0.6329	0.5677	0.3497	0.1691
1.2	0.5187	0.6224	0.3310	0.3178

Table 7.1: Numerical results presented in Appendix A7.7 for N_i= 5.

The probability of selecting the neighbouring cell improves by between 4% and 74.5%. This is broadly similar for other values of N_i, although the differences are typically larger for higher N_i. The results for identifying the correct edge cell are more mixed, ranging from a drop in confidence of 20.0% to an improvement of ~10%. However, on the metric of maximising the difference between the probabilities of selecting the correct edge cells and the neighbouring cells, the linear optimisation is generally the best with improvements of between ~16% and ~62%.

This is partly because some values of N_i do not correspond to peaks in the charts in Figure 7.5; for example, N_i=25 lies in a trough for a standard deviation of 0.6. However, the main benefit of the linear optimisation method over *k*-value thresholds is that it allows early termination of the combination process in the event that the number of samples returned from the tactile sensor is lower than expected. Figure 7.12 shows the changes in probability with increasing numbers of combination steps for standard deviations of 0.6 and 1.0 with N_i.= 5 and 10. Each data point on these graphs shows the probability of having selected the correct edge cell or the neighbouring cell if the combination process was terminated at that point.

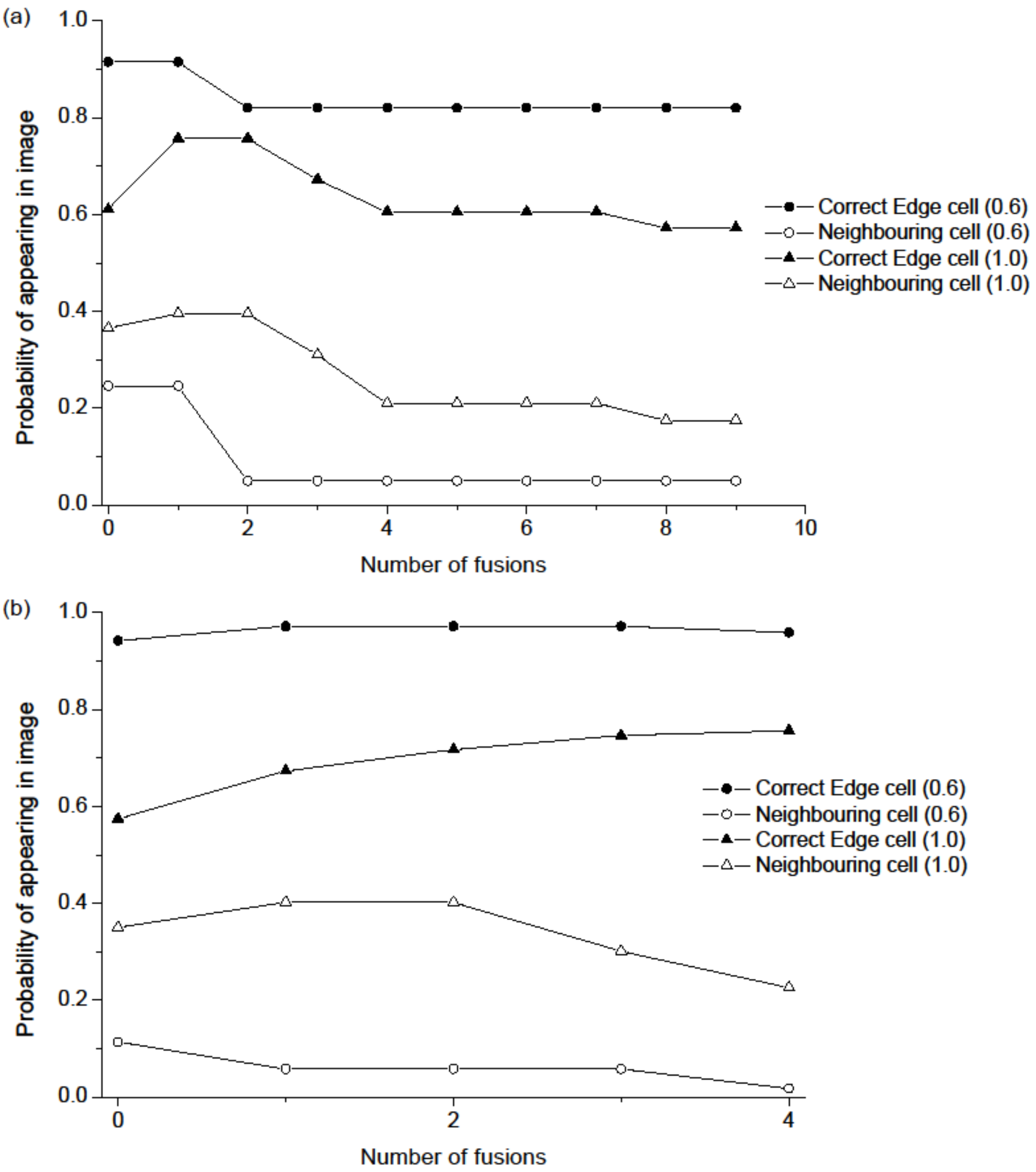


Figure 7.12: Graphs showing the intermediate probabilities of the linearly optimised combination processes for standard deviations of 0.6 and 1.0 with N_i equal to (a) 5 and (b) 10

The results of Figure 7.12 show that early termination still results in a better outcome when the standard deviation equals 1.0. Although this is less obvious when N_i=5 compared to N_i=10, the same effect is observed in absolute terms. For the standard deviation of 0.6, Figure 7.12 shows that there is limited incremental improvement when N_i=5, but there is substantial improvement at each step when N_i=10.

The benefits of the method in terms of possible early termination of the combination process depend on the standard deviation. This is probably because no constraint can be imposed in the linear optimisation method to increase the probability from every previous step, only the most recent one. Nonetheless, the early termination still appears to improve the average accuracy of the internal representation. As such, this method of optimisation is more robust in the presence of noise than the mechanisms described in Section 7.3.2.

7.3.4. Discussion

The above section has detailed a method for finding the means of allowing an internal representation of tactile information to be updated using smaller batches of tactile measurements, rather than the all-or-nothing approach employed in the *k*-value threshold method. The results show that it is possible to find ways of producing an internal representation with greater confidence levels using arbitrary numbers of measurements.

The trade-off is that the absolute confidence levels tend to be lower than the equivalent achievable by combining all the measurements together using the *k*-value threshold method. This does nonetheless make allowances for realistic failures in the system, such as an inability to produce a fixed number of measurements in a timely manner. This technique can also be useful when the available memory for intermediate storage of tactile data is low, since a large quantity of data can be processed in smaller batches.

As with the *k*-value threshold method, the linear optimisation technique does not allow updates following later exploration of the same object. In a number of applications, this limitation is still acceptable. For example, if a tactile representation is used for collision avoidance then returning to an occupied area should not occur. If it does, it is because of an error in the representation, in which case replacement might be the optimum correction

Any improvements in this area would require that the optimisation find a single Boolean method that could be used multiple times. This requires non-linear goals and constraints. Optimisation in this situation would require techniques that either offer no computational improvement over a brute-force approach or are active research areas. When the limitations of linear optimisation prevent integration into a system, other approaches must be considered.

7.4. Two-bit Brute-Force Optimisation

7.4.1. Optimisation Approach

Brute-force optimisation is the process of simply iterating through all possible transition matrices and finding the one that maximises a goal function. This requires little additional mathematics to that described in the previous sections. Generating the matrices can be simplified by using the *C*-matrix notation introduced in equation (7.14). The only constraint to be enforced is that the columns of the matrix add up to one.

Appendix A7.8 shows the helper function *ValidMatrices* written in MATLAB that assists in producing these matrices. The function produces the possible combinations of values that can legally fill four of the columns in the *C*-matrix. The optimisation routine shown in Appendix A7.8 iterates through all of these combinations and applies equation (7.16) to determine the optimum solution. Vector *r* is redefined from (7.15) to the form shown in (7.16).

$$\underline{r} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} = \left(\underline{\underline{CP}}\right)^{N_C} \underline{p} - \left(\underline{\underline{CQ}}\right)^{N_C} \underline{q} \qquad (7.16)$$

Here, the r_i represent the difference between the probabilities of the correct edge cell being in state *i* and of the neighbouring cell being in state *i*. The exponent N_C represents the number of combinations using the same Boolean function that is used to maximise the goal function. This forces the optimisation process to produce improvements after multiple combination steps.

The goal function itself is different from that used in the linear optimisation trials, because initial trials using $r_1 - r_4$ tended to show that the optimised result was not making use of the intermediate states 2 and 4. Some limited trials showed better performance when the goal function used terms involving r_2 and r_3. The formula of equation (7.17) was used for the goal function.

$$G = 3r_1 + \frac{r_2}{2} - \frac{r_3}{2} - 3r_4 \qquad (7.17)$$

The value of r_2 has a positive coefficient, since it is better that the correct edge cell is believed to be probably occupied than unoccupied. Similarly, a negative factor of r_3 was used for the empirical reason that it is better that the neighbouring cell is believed to be probably unoccupied than occupied in any way. Since a more definite statement of occupation is ultimately sought, these factors were lower than those applied to r_1 and r_4. This goal function is thus empirical rather than a strict analysis.

7.4.2. Results

Optimisations were performed for values of the total number of measurements N_S = 26, 51, 77 and 95. These were chosen because each has only two non-trivial pairs of factors which could be assigned to N_i and N_C respectively for the purposes of optimisation. These optimisations were performed against probabilities determined from a Gaussian distribution with standard deviations of 0.6, 0.8, 1.0 and 1.2.

The results of the optimisations are shown in Table 7.2. These show that setting N_i to the higher of the two values improves the probability of selecting the correct edge cell for all standard deviations except 1.2. However, the drop in the probability of selecting the correct edge cell in this case is offset the substantial reduction in the probability of erroneously selecting the neighbouring cell from 0.6617 to 0.2900. The remaining results presented in this section were produced from optimisations that used a higher value for N_i than for N_n.

Comparisons between the effects of the brute-force approach and the *k*-value threshold method were undertaken. These are shown graphically in

Appendix A7.9 and indicate that for standard deviations of 0.6 and 0.8, the two methods produce probabilities of selecting the correct edge cell that differ only by a maximum of 2%. The probability of selecting the neighbouring cell at these standard deviations also maintains a generally downward trend.

	Probability of selecting the relevant cell using optimised method after N_n combinations			
Standard deviation	Correct Edge cell (N_i=*2*, N_n=*13*)	Correct Edge cell (N_i=*13*, N_n=*2*)	Neighbouring cell (N_i=*2*, N_n=*13*)	Neighbouring cell (N_i=*13*, N_n=*2*)
0.6	0.7811	0.9870	0.0643	0.0128
0.8	0.6199	0.9188	0.1401	0.1058
1.0	0.4486	0.7873	0.1506	0.1727
1.2	0.8637	0.6999	0.6617	0.2900

Table 7.2: Effect of combining 26 total measurements in two different ways on the probabilities of selecting the correct edge cell and of selecting the neighbouring cell.

The brute-force method is particularly successful at achieving monotonic reductions in the probability of erroneously selecting a neighbouring cell. This is achieved at the expense of lower probability of selecting the correct edge cell, which for a standard deviation of 1.2 was lower by between 0.045 and 0.116.

The brute-force approach also allows incremental updates from batches of measurements combined together using the *k*-value threshold method, which is a feature not present in any other method explored so far. Figure 7.13 shows the incremental effect of applying the result of the brute-force optimisation when updating an internal representation that already contains prior knowledge. Results are shown for a standard deviation of 0.6, with full results for all four standard deviations available in Appendix A7.10.

The form of this graph is representative of all other standard deviations with N_S=95. These show that the probability of selecting the correct edge cell increases monotonically with each additional incorporation of measurements before appearing to reach a limit. In the case of a standard deviation of 0.6, this limit is 0.9996 and the limit is 0.9544 for a standard deviation of 0.8. What is difficult to discern from these figures is that, having reached this limit, the probability does begin to reduce linearly. The gradient of this line is on the order of 10^{-6} and so is not regarded as significant.

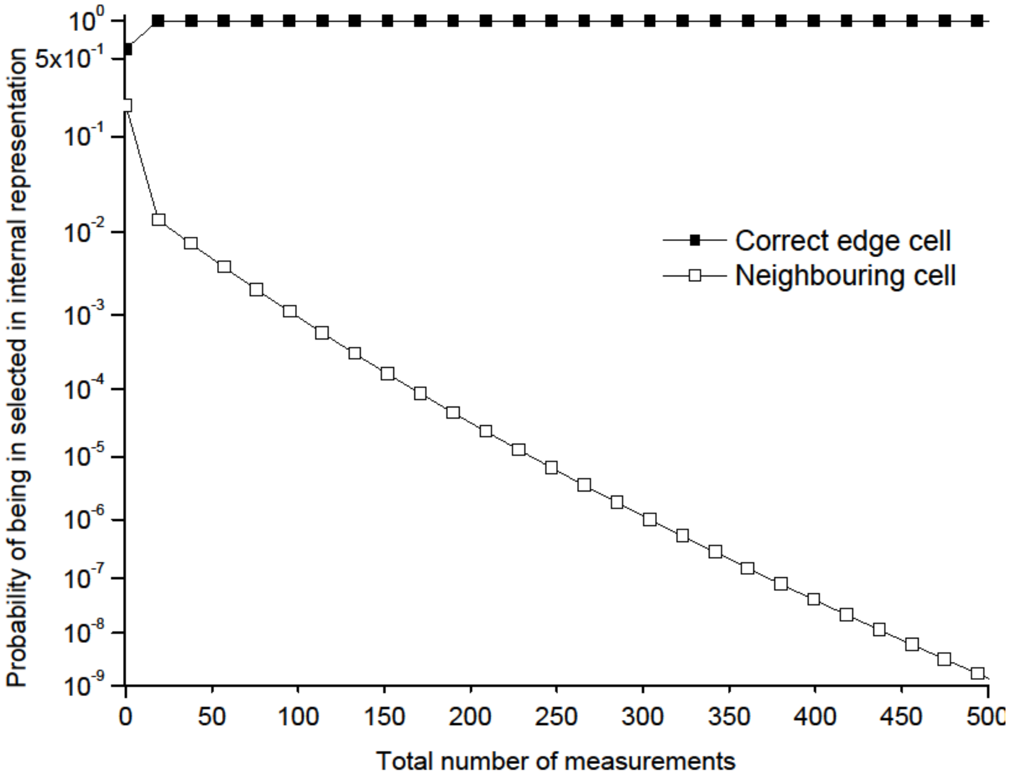


Figure 7.13: Effect of incrementally applying the results of brute-force optimisation to update an internal representation. Standard deviation of the underlying Gaussian distribution is 0.6.

This effect at lower standard deviations are not significant since the probability of erroneously selecting the neighbouring cell falls exponentially to zero with increasing numbers of measurements. These results strongly indicate the benefit of the brute-force optimisation scheme in terms of delivering the capacity for incremental updates, which are crucially independent of any knowledge of how many updates have occurred previously.

At higher standard deviations, both probability lines tend to particular values after a number of incremental updates. This indicates that after a certain point, further updates do not improve the confidence in the internal representation. When the standard deviation is 1.0 the probability of selecting the correct edge cell trends to 0.8580 and the probability of selecting the neighbouring cell trends to 0.0551. The gap between these two probabilities narrows when the standard deviation is 1.2 as they tend to 0.7923 and 0.080 respectively.

These limiting values are only reached when the total number of measurements is in excess of 300 and is preceded by continual incremental improvement. This again demonstrates the success of the brute-force method in allowing the incorporation of incrementally acquired batches of measurements. Avoiding these limiting values may be achievable using different values of N_i, N_C and N_S, but this remains untested.

The results of this section show that the brute-force optimisation method is beneficial in systems where tactile sensors may revisit locations and wish to improve prior knowledge based on new measurements. This can be done using the brute-force optimisation approach without discarding the prior knowledge in the same way as is necessary with either *k*-value thresholds or linear optimisation. The only limitation on brute-force optimisation is the empirical choice of the goal function and this needs to be explored further.

7.4.3. Simulation

The results of Section 7.4.2 demonstrate the benefits of applying the brute-force optimisation method to find the means of combining multiple images incrementally to improve the internal representation. One of the results will be applied to provide an example of applying this technique in this section.

The simulation will essentially be the same as that performed in Sections 7.1.2 and 7.2.3, using a positional Gaussian noise with a standard deviation of 0.8. The difference is that an additional set of simulations will use the results from Section 7.4.2 to combine a set of 95 images in each simulation.

The optimisation processes described in this chapter are designed to produce Boolean functions that can be used to combine tactile images in order to reduce noise. For brevity, the process of converting the optimised *C*-matrix into a Boolean function is deferred until Appendix A7.11 along with a worked example from the results of this chapter with a standard deviation of 0.8. The Boolean equation that results is given in equations (7.18) and (7.19).

$$U_1 = C_1\overline{C_2} + \overline{M_2}C_2M_1 \qquad (7.18)$$

$$U_2 = C_1\overline{M_1\ M_2} + C_1M_2 + \overline{C_1}C_2M_1 \qquad (7.19)$$

In these equations, the four inputs to the combination are the two bits representing the current state of the cell in the internal representation, denoted *C_1* and *C_2*, and the two bits representing the state of the cell in the combined measurement, denoted *M_1* and *M_2*. The outputs are the updated pair of bits to revise the internal representation, denoted *U_1* and *U_2*.

These equations were used in a simulation of an object impacting on a 10cm-long, one-dimensional tactile sensing strip with a 1cm resolution. The error in determining the position of the origin of the tactile sensing strip was simulated as a random variable with a Gaussian distribution with a standard deviation of 0.8mm. The simulation was run 1000 times and on each occasion a random location for the tactile sensor origin was sampled from this distribution. Figure 7.14 shows a histogram of the believed occupied ranges over all the simulations performed with locations placed in 1cm wide bins without using any optimisation.

To produce Figure 7.15, 5 batches of 19 measurements were simulated in a single cycle. Each batch was combined using the *k*-value threshold method and these batches were combined using equations (7.18) and (7.19). As previously, applying these equations to the tactile images means that the correct occupied positions are identified in all simulations. This is as compared to the results of Figure 7.14 which suggest that this statement is only true 88% of the time. The lower order noise has also been removed.

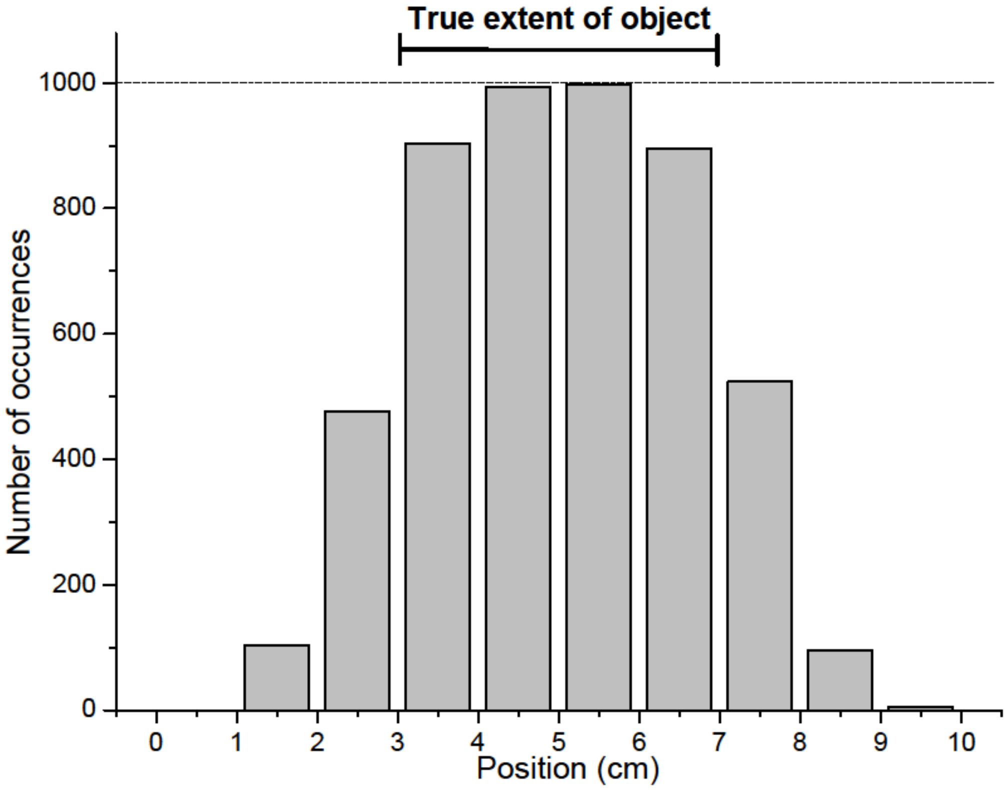


Figure 7.14: Results of simulating tactile imaging in one dimension over 1000 simulations where measurements of the position are subject to Gaussian noise with standard deviation of 0.8mm.

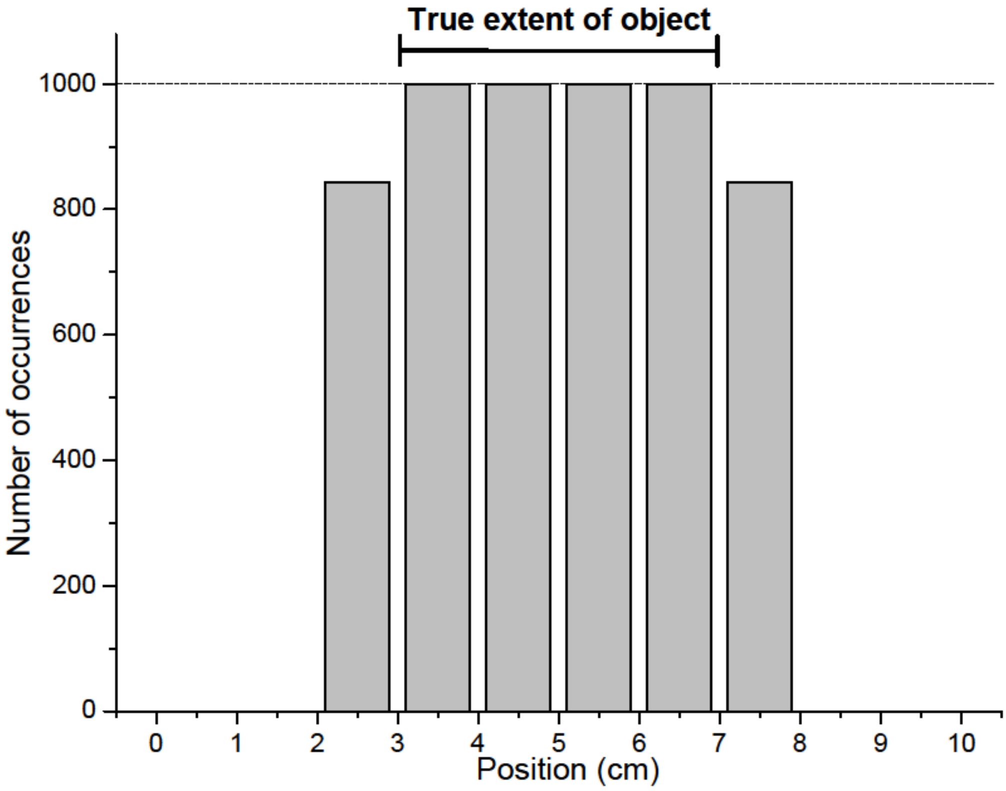


Figure 7.15: Results of applying (7.18) and (7.19) to tactile images in one dimension over 1000 simulations where measurements of the position are subject to Gaussian noise with standard deviation of 0.8mm.

The fact that all simulations contained the correct edge cells, but that the neighbouring cells only appeared in 88% of them suggested that a further filtering step to enhance these images as was done previously and this was confirmed, producing an identical histogram to Figure 7.9.

These results indicate that the application of the brute-force optimisation method can be at least as good as applying a *k*-value threshold, with the added advantage of allowing incremental updates of the internal representation.

7.4.4. Conclusions

The results of this section show that it is possible to develop a bitworld to allow incremental updates of an internal representation of tactile data. The work here has used two bits per cell, which does not represent a significant sacrifice in terms of memory usage and is still significantly better than the 32-bits used by floating-point matrix representations.

The probabilistic results of a brute-force optimisation are quantitatively as good as those for *k*-value threshold optimisation. There is generally a continuous improvement in the probabilities of selecting the correct edge cell and the neighbouring cell with increasing numbers of measurements, particularly at standard deviations below 1.0. At or above this standard deviation, the probabilities tend towards particular non-zero levels, but this might be avoided by using more measurements in a single batch.

Improvements in the goal function used might also afford greater benefits to the brute-force optimisation approach.

7.5. Discussion and Conclusions

This chapter has presented the tools for using bitworld to manipulate tactile data in the presence of positional noise.

It was shown that a one-bit representation has limited use in the presence of noise arising from the measurement of robot pose, except in the case where a measurement is less likely to select the correct edge cell than an incorrect neighbouring one.

In an effort to maintain the use of a one-bit representation, multiple samples can be taken and the frequency of measurement occurrence per cell used to reduce the noise. This process is referred to in this Chapter as the *k*-value threshold method and was extremely successful in improving the probability of selecting the correct edge cell and in minimising the probability of erroneously selected other bitworld cells. It was most successful when the standard deviation was equal to or less than 0.6, although some benefit was still observed at greater standard deviations.

Such a combination requires the storage of multiple measurements which reduces the memory benefits of using bitworld, although such storage is temporary. In order to avoid this additional cost, approaches that used multiple-bit bitworlds were explored.

The optimisation strategies that followed imposed differing requirements on the tactile system and are categorised in Table 7.3.

Method	*Divisible measurements*	*Suitable for a noisy system*	*Allows incremental updates*
One-bit bitworld combination	No	No	No
k-value threshold method	No	Yes	No
Linear optimisation	Yes	Yes	No
Brute-force optimisation	Yes	Yes	Yes

Table 7.3: Table summarising the differing capabilities of the different optimisation techniques presented in this chapter.

The brute-force method is the only method that allows incremental updates of the internal representation. The results of Section 7.4.2 show that the probabilities achieved are often close to those achieved by using the *k*-value threshold method, but it is the ability to improve prior data that would be most appealing to designers of a tactile system.

This chapter has laid the foundation for bitworld by showing how it can be made to work in theory for a number of differing scenarios. In summary, the applicability of each method is:

1. One-bit bitworld combination: Systems with no noise or where the noise is irrelevant. A simple replacement of map information upon subsequent traversal of the same spatial area is acceptable.

2. *k*-value threshold method: Noisy systems where the number of measurements returned to the system is guaranteed. A simple replacement strategy on subsequent traversal must be acceptable.

3. Linear optimisation: Noisy systems where number of measurements cannot be guaranteed. This method requires greater designer input in deciding system parameters such as the value of N_i. Simple replacement on subsequent traversal must be acceptable.

4. Two-bit system with brute-force optimisation: Noisy systems where continuous updates are preferred, including on subsequent traversal of the same spatial area. The time required to tune the combination mechanism is the highest of all methods.

In conclusion, this chapter has demonstrated that bitworld can successfully be deployed in a variety of tactile applications and can be made to be robust in the presence of positional noise. Combined with the similarly good results of Chapter 6, this work shows how bitworld can be exploited successfully as a real-world tool for compact, functional and computationally efficient representation of tactile data.

Chapter 8
Conclusions and Further Work

8.1 Conclusions

This work began by describing the need for a different focus in the design of tactile systems. The literature review of Chapter 2 outlines a different approach for assessing a tactile system based on a data-centric specification. This followed from the need identified in earlier work for tactile systems to process a large quantity of data, but extended this need to be the defining quality against which tactile systems should be assessed. This differed from earlier approaches that imposed particular requirements such as resolution that will vary according to the needs of a given application. Assessed against the data-centric requirements, existing work in the area of tactile sensing was found to concentrate principally on the specific properties of sensors and did not isolate the sensing method sufficiently from the data flow in a manner that was able to satisfy the data-centric approach.

In the literature review, a pressure-sensitive material called QTC™ was identified as being beneficial from this data-centric perspective. The operating principle was resistance-based and therefore determined to be relatively simple. Crucially, the unloaded resistance of the material is extremely high. This is beneficial in wide-area tactile sensing, because in typical applications a high percentage of the tactels will be unloaded at a given point in time. Under pressure, the resistance of the material is lowered by multiple orders-of-magnitude, making it easy to differentiate between loaded and unloaded states. The simple operating principle, potentially high sensitivity and low power consumption reduces the burden imposed on a robotics system. As a result, QTC™ emerged as the material of choice for this work.

Previously, this material had only been deployed in the form of discrete sensors. The use of QTC™ within the sensor arrays developed in Chapters

3 and 4 therefore represent a novel construction. The use of conductive bobbinet fabric was shown to provide flexible and durable electrodes, additionally endowing the sensor array with flexibility.

The prototyping work of Chapter 3 considered two means of forming a two-dimensional sensing array from the bobbinet and composite. This work showed that mechanical isolation of the tactels could be achieved if the QTC™ was formed into a series of independent, overlapping strips that form tactels in the areas where they overlap. This design was suitable for tactile sensing, as it allowed contact to be easily localised and in a manner conformant with the data-centric approach of this work. However, the implementation involved conductive stripes of bobbinet alternated with non-conductive stripes on a single sheet of fabric, which was not convenient for manufacture.

The final version of the sensor was consequently developed as a series of separate strips of conductive bobbinet with QTC™ embedded into it. This allows the stripes to be deployed in a variety of configurations to allow easy integration into a robotic system. The only deployment scenario considered was to align the strips orthogonally, meaning that any unforeseen problems with a non-orthogonal deployment were not explored. Such a deployment is not always appropriate, for example in the case of cylindrical links where one of the edges holding the electrical contacts would be difficult to access.

Measurements were taken under loading that demonstrated that this deployment allowed the beneficial properties of the QTC™ to be retained. The flexibility of the material was explored further with loading measurements performed with the sensor subjected to curvature. Although high curvature caused the unloaded resistance to fall, which diminishes one of the benefits of QTC™, the desired properties were still observed under light curvature.

Work was undertaken to characterise the electrical behaviour of the composite in order to establish the properties of the material for future development. It was demonstrated that this behaviour is not dominated by the Poole-Frenkel effect as previously asserted. Instead, the behaviour was found to be a mixture of ohmic behaviour, space-charge limited effects and Poole-Frenkel emission, with the relative contribution of each varying with applied load.

This data will assist in developing bespoke circuitry to interface with the sensor and provides information that may assist in developing the material to tune its characteristics for particular applications. However, the model of conduction proposed here does not completely explain all aspects of the electrical characteristic of the material. The outstanding issue is how the charge is injected into the polymer. Nonetheless, the main mechanisms have been identified directly from the current-voltage characteristics of the material.

This array-based sensor satisfied the data-centric requirements of low power consumption and a simple transduction principle. However, the simplicity of the design, whilst advantageous in many respects, comes at the cost of being unable to process data locally to the tactels as would be, for instance, possible in a semiconductor-based tactile sensor. As such, deployment would rely on local circuitry, which could occupy substantial space. The problems that this could present will depend on the likely applications for which this sensor will be used. The transient behaviour of the material shown in Chapter 4 showed a significant recovery time for the transition from a loaded to an unloaded state. This renders the material inappropriate for gripping applications where a rapid dynamic response is required. Grippers are also an application where limited space is available within the robot to deploy extraneous circuitry, making deployment of this material for this application similarly difficult.

However, for applications where tactile sensors are required on larger areas of a robot body such as arms, more space may be available, for example within the body of the robot. Consequently, there are application domains where the benefits of the QTC™ may be realised without imposing too onerous a burden on the overall robotic system.

The introduction and development of bitworld in Chapters 5 and 6 was intended to improve the integration of tactile sensors with a robot controller. This would be done by reducing the total amount of computational resource that a controller would have to devote to processing the large amounts of data potentially arising from a tactile system. The low resolution of this data facilitates a significant reduction in the bandwidth requirements for such a system. Monochrome imaging is not in itself a novel development, but the implementation of the system in the form of bitworlds constructed out of bitboards is, particularly given the computations advantages demonstrated by the results of profiling.

Having demonstrated the potential benefits of bitworld in the profiling exercises of Chapter 5, Chapter 6 combined the sensor of Chapter 4 with its low spatial resolution and bitworld, which has a low data resolution. The result was a process by which multiple low spatial resolution images were combined using bitworld techniques to yield images of superior resolution.

From a data-centric perspective, this is positive as it offers a robot choice; the robot does not have to have high resolution imaging and the associated costs imposed on it when such imaging may often be unnecessary. There is clearly a balance to be struck, as acquiring a higher-resolution image necessitates making sub-pixel movements across the obstacle. Each movement requires additional time and the number of these movements will depend on the difference between the sensor resolution and the desired resolution. A cost-benefit analysis will be required for different applications and it is conceivable that minimum resolutions may be established during this process of specification.

The extension of bitworld be allow robustness in the presence of noise described in Chapter 7 is also a significant achievement as is allows noisily-aligned images to be combined whilst exploiting the computational benefits of the low data resolution of bitworld. The simulations showed that the brute-force optimisation strategy was the best in terms of reducing the noise in the final images.

A significant problem with the results of Chapter 7 is that they are all based on simulations rather than experiments with an actual actuator-mounted system. As a result, it is questionable whether or not such result would be observed in a realistic robotic system where the sensors are two-dimensional. Nonetheless, Chapter 7 provides the theoretical basis for such work and the simulations justify the approaches adopted and as such represent a significant milestone in making bitworld robust in the presence of noise.

To conclude, this work has presented a novel approach to the specification of tactile systems and has shown the development of a new sensor and data representation to meet these requirements. This work has shown how the data-centric approach can solve many of the problems in tactile sensing without losing the functionality of high resolution sensors. However, the work presents various further questions that might be explored and a sample of these is presented in the next section.

8.2. Further Work

This work has focussed on a data-centric specification for tactile systems, which provided criteria that a tactile system should strive to meet in order to facilitate integration. However, the literature review of Chapter 2 followed the pattern of earlier reviews with a taxonomy based on the transduction principle of the sensors.

This fulfilled the main goal of this work, which was to assess existing sensors against the presented data-centric specification. In the process, however, the applicability of the sensors was ignored. It was therefore be of greater benefit to instead define a taxonomy based on application so that researchers can focus on developing sensors that meet particular needs.

Such an arrangement would require an analysis of the application for tactile skin, with greater differentiation in the categories of gripping, obstacle avoidance and exploration that has been explored to date. For example, gripping applications include both those where object identification is needed and those where simple feedback control would be acceptable. The differences between the sensors in these two categories alone are significant; the former requires the possibility of imaging via an array-like construct, whilst discrete sensors might suffice in the latter. Overall, defining these application domains might be a difficult task, but would facilitate development in both relevant task-based sensing applications and in the generalisation of the interfaces and communication required in a system and is a critical goal for future development in the field.

In the sensor constructed as a result of the work described in Chapters 3 and 4, the manufacture of strips of QTC™ backed with conductive fabric was shown to be very effective. However, the intention of the strip-based construction was to allow a variety of deployment configurations and this work only explored an orthogonal arrangement of the strips. Further work should explore the possible impacts of a non-orthogonal arrangement, in terms of integration with circuitry and in terms of tactel area.

This work has used general measurement systems that would be unsuitable for integrating the sensor in a robotic system. Further work should explore what circuitry is required to integrate the QTC™ sensor with a robot, such that it uses minimal space. This work could be extended to demonstrate the means of conversion from the resistance measurement to a bitworld-based representation and its transmission to a central robot control system.

In general, although this work has presented the components of a good tactile system and demonstrated their utility in experiment and simulation, there is a deficiency in that a robot was not available to test the system as a whole. A clear benefit would be derived if deployment of a system comprising the QTC™-based sensor and bitworld-based processing could be deployed on a robot. Under such circumstances, a number of experiments could be undertaken to strengthen the conclusions of this work. The work of Chapter 6 could be strengthened by allowing the robot to explore arbitrary environments using only tactile sensation, build bitworld models of the environment and attempt to make decisions based on the results. This kind of work could also form the basis of approaches to integrating bitworld-based models with the outputs from other sensing systems such as vision.

No aspect of this work would benefit more from a robotic test system than the noise-reduction approaches of Chapter 7, as both the impacts and sources of noise could be deduced experimentally and the approaches to reducing their impact verified.

In a real robotic system, it might be beneficial to allow the robot to measure the levels of noise in its pose measurement system and essentially auto-tune the tactile image recombination process accordingly. Unfortunately, the brute-force optimisation process would be an obstacle to deployment in this scenario.

The linear optimisation techniques were considerably faster, but their outputs were dependent on the choice of minimisation function and did not generally lead to monotonic increases in the probability of correctness. A better method would be to examine non-linear optimisation techniques to see if these can be made to run more efficiently that the brute-force approach. . An additional benefit to such a development would be examining the case for three or more bits per bitworld cell, which is rendered difficult to achieve in a suitable amount of time using the brute-force approach, given the large search space of possible matrices.

An increased number of bits per cell could be expected to improve the upper limit in the probability of correctness. As the number of bits gets closer to that of a floating point number, the optimum would presumably approach that of conventional techniques such as occupancy grids. An increased number of bits per bitworld cell was also proposed as a means of introducing a range of values per cell to give a better-than-monochrome image whilst not losing all the benefits of the low resolution. An obvious question is how to combine such multiple bit systems together, so that a range of values can be accommodated in noisy systems. This might be best done by devising a more appropriate probability model upon which the optimisation techniques can be based. These models would take into account the probability of measuring the different, discrete levels of force given prior knowledge. This model would likely be based on the principles of Bayesian inference and would form the basis of a generalised, robust bitworld system. It would be useful to develop this in further work.

In conclusion, the work presented in this thesis has offered novel solutions to some identified issues in the field of tactile sensing. Further work has been proposed to extend these ideas and strengthen the underlying principles.

References

1. Newspapers, K.-R., *$10 million awarded to family of plant worker killed by robot*, in *Ottawa Citizen*. 1983, Russell Mills: Ottawa. p. 14.
2. Kawamura, K., R.T. Pack, and M. Iskarous. *Design philosophy for service robots*. in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*. 1995.
3. Lumelsky, V., *Sensing, intelligence, motion : how robots and humans move in an unstructured world*. 2006, Hoboken, NJ: J. Wiley. 431 p.
4. Dahiya, R.S., et al., *Tactile Sensing - From Humans to Humanoids.* IEEE Transactions on Robotics, 2010. **26**(1): p. 1-20.
5. Bloor, D., et al., *A metal-polymer composite with unusual properties.* Journal of Physics D: Applied Physics, 2005. **38**: p. 2851-60.
6. Bloor, D., et al., *Metal-polymer composite with nanostructured filler particles and amplified physical properties.* Applied Physics Letters, 2006. **88**(10): p. 102103.
7. Lee, M.H. and H.R. Nicholls, *Tactile sensing for mechatronics - a state of the art survey.* Mechatronics, 1999. **9**(1): p. 1-31.
8. Wood, E.J. and P.T. Bladon, *The human skin*. The Institute of Biology's studies in biology, no. 164. 1985, London ; Baltimore, Md.: E. Arnold. 66 p.
9. Purves, D. and S.M. Williams, *Neuroscience*. 2nd ed. 2001, Sunderland, Mass.: Sinauer Associates. 681 p.
10. Sekuler, R. and R. Blake, *Perception*. 3rd ed. 1994, New York: McGraw-Hill. 572 p.
11. Cauna, N., *Nerve supply and nerve endings in Meissner's corpuscles.* American Journal of Anatomy, 1956. **99**(2): p. 315-50.
12. Loewenstein, W.R. and R. Skalak, *Mechanical transmission in a Pacinian corpuscle. An analysis and a theory.* Journal of Physiology, 1966. **182**(2): p. 346-378.
13. Nicholls, H.R., *Advanced tactile sensing for robotics*. World Scientific series in robotics and automated systems. 1992, Singapore ; River Edge, N.J.: World Scientific Pub. 294 p.
14. Sherrick, C.E. and R.W. Cholewiak, *Cutaneous Sensitivity*, in *Handbook of Human Perception and Performance*, L.K. K. Boff, and J. Thomas, Editor. 1986, Wiley: New York. p. 12/1 - 12/58.
15. Howe, R.D., *Tactile sensing and control of robotic manipulation.* Advanced Robotics, 1993. **8**: p. 245-261.
16. Lumelsky, V.J., M.S. Shur, and S. Wagner, *Sensitive skin.* IEEE Sensors Journal, 2001. **1**(1): p. 41-51.
17. Harmon, L.D., *Automated Tactile Sensing.* The International Journal of Robotics Research, 1982. **1**(2): p. 3-32.
18. Harmon, L.D., *Robotic Taction for Industrial Assembly.* The International Journal of Robotics Research, 1984. **3**(1): p. 72-76.
19. Lee, M.H., *Tactile Sensing: New Directions, New Challenges.* The International Journal of Robotics Research, 2000. **19**(7): p. 636-643.
20. Iggo, A. and A.R. Muir, *The structure and function of a slowly adapting touch corpuscle in hairy skin.* Journal of Physiology, 1969. **200**(3): p. 763-96.
21. Hahn, J.F., *Stimulus--response relationships in first-order sensory fibres from cat vibrissae.* Journal of Physiology, 1971. **213**(1): p. 215-226.
22. Skordos, A., et al., *A novel strain sensor based on the campaniform sensillum of insects.* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2002. **360**(1791): p. 239-253.

23. Vincent, J.F.V., S.E. Clift, and C. Menon, *Biomimetics of Campaniform Sensilla: Measuring Strain from the Deformation of Holes.* Journal of Bionic Engineering, 2007. **4**(2): p. 63-76.
24. Montagu, A., *Touching : the human significance of the skin*. 3rd ed. 1986, New York: Perennial Library. 494 p.
25. Um, D. and V. Lumelsky. *Fault tolerance via analytic redundancy for a modularized sensitive skin*. in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1999.
26. Um, D. and V. Lumelsky. *Fault tolerance via component redundancy for a modularized sensitive skin*. in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1999.
27. Um, D., et al. *A modularized sensitive skin for motion planning in uncertain environments*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1998.
28. Stillings, N.A., *Cognitive science : an introduction*. 2nd ed. 1995, Cambridge, Mass.: MIT Press. 530 p.
29. Best, J.B., *Cognitive psychology*. 2nd ed. 1989, St. Paul: West Pub. Co. 583 p.
30. Neisser, U., *Cognitive psychology*. The Century psychology series. 1967, New York,: Appleton-Century-Crofts. 351 p.
31. Baddeley, A.D., *Human memory : theory and practice*. 1990, Boston: Allyn and Bacon. 515 p.
32. Scherer, F. and M. Scherer, *How "Micro" Can Learning Be? A Neuropsychological Perspective on Microlearning*, in *Didactics of Microlearning: Concepts, Discourses and Examples*, T. Hug, Editor. 2007, Waxmann: New York. p. 110-124.
33. Sperling, G., *The information available in brief visual presentations*. Psychological monographs: general and applied,. 1960, Washington: American Psychological Association. 29 p.
34. Mandler, G. and B.J. Shebo, *Subitizing: an analysis of its component processes.* Journal of Experimental Psychology: General, 1982. **111**(1): p. 1-22.
35. Turvey, M.T., *On peripheral and central processes in vision: inferences from an information-processing analysis of masking with patterned stimuli.* Psychological Review, 1973. **80**(1): p. 1-52.
36. Parkin, A.J., *Essential cognitive psychology*. 2000, Hove, East Sussex; Philadelphia, PA: Psychology Press ; Taylor & Francis. 354 p.
37. Posner, M.I., et al., *Retention of visual and name codes of single letters.* Journal of Experimental Psychology, 1969. **79**(1): p. 1-16.
38. Phillips, W.A. and A. Baddeley, *Reaction time and short-term visual memory.* Psychonomic Science, 1971. **22**: p. 73-74.
39. Waugh, N.C. and D.A. Norman, *Primary Memory.* Psychological Review, 1965. **72**: p. 89-104.
40. Miller, G.A., *The magical number seven plus or minus two: some limits on our capacity for processing information.* Psychological Review, 1956. **63**(2): p. 81-97.
41. Nicholls, H.R. and M.H. Lee, *A Survey of Robot Tactile Sensing Technology.* The International Journal of Robotics Research, 1989. **8**(3): p. 3-30.
42. Strumpler, R. and J. Glatz-Reichenbach, *Conducting polymer composites.* Journal of Electroceramics, 1999. **3**(4): p. 329-346.
43. Xue, Q., *The influence of particle shape and size on electric conductivity of metal-polymer composites.* European Polymer Journal, 2004. **40**(2): p. 323-327.
44. Russell, R. *Compliant-skin tactile sensor*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1987.

45. Stojiljkovic, Z. and J. Clot, *Integrated Behavior of Artificial Skin.* IEEE Transactions on Biomedical Engineering, 1977. **24**(4): p. 396-399.

46. Shimojo, M., et al., *A tactile sensor sheet using pressure conductive rubber with electrical-wires stitched method.* IEEE Sensors Journal, 2004. **4**(5): p. 589-596.

47. Shimojo, M., et al. *A system for simultaneously measuring grasping posture and pressure distribution*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1995.

48. Hillis, W.D., *A High-Resolution Imaging Touch Sensor.* The International Journal of Robotics Research, 1982. **1**(2): p. 33-44.

49. Robertson, B.E. and A.J. Walkden, *Tactile Sensor System for Robotics*, in *Robot Sensors Volume 2 - Tactile and Non-Vision*, A. Pugh, Editor. 1986, Springer-Verlag: New York. p. 89-97.

50. Hussain, M., Y.H. Choa, and K. Niihara, *Conductive rubber materials for pressure sensors.* Journal of Materials Science Letters, 2001. **20**(6): p. 525-527.

51. Chang, F.-G. and et al., *Enhanced piezoresistivity in Ni–silicone rubber composites.* Chinese Physics B, 2009. **18**(2): p. 652.

52. Martin, T.B., et al. *Tactile gloves for autonomous grasping with the NASA/DARPA Robonaut*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 2004.

53. Stiehl, W.D. and C. Breaeal. *A Sensitive Skin for Robotic Companions Featuring Temperature, Force, and Electric Field Sensors*. in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2006.

54. Stiehl, W.D., L. Lalla, and C. Breazeal. *A "somatic alphabet" approach to "sensitive skin"*. in *Proceedings of IEEE/RSJ International Conference on Robotics and Systems*. 2004.

55. Stiehl, W.D. and C. Breazeal. *Applying a "somatic alphabet" approach to inferring orientation, motion, and direction in clusters of force sensing resistors*. in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2004.

56. Petriu, E.M., et al., *Robotic tactile recognition of pseudorandom encoded objects.* IEEE Transactions on Instrumentation and Measurement, 2004. **53**(5): p. 1425-1432.

57. van den Heever, D.J., K. Schreve, and C. Scheffer, *Tactile Sensing Using Force Sensing Resistors and a Super-Resolution Algorithm.* IEEE Sensors Journal, 2009. **9**(1): p. 29-35.

58. Vidal-Verdu, F., et al. *Large area smart tactile sensor for rescue robot*. in *IEEE International Workshop on Robotic and Sensors Environments*. 2009.

59. Suwanratchatamanee, K., M. Matsumoto, and S. Hashimoto, *Robotic Tactile Sensor System and Applications.* IEEE Transactions on Industrial Electronics, 2010. **57**(3): p. 1074-1087.

60. Papakostas, T.V., J. Lima, and M. Lowe. *A large area force sensor for smart skin applications*. in *Proceedings of IEEE International Conference on Sensors*. 2002.

61. Wen-Yang, C., et al., *A Large Area Flexible Array Sensors Using Screen Printing Technology.* Journal of Display Technology, 2009. **5**(6): p. 178-183.

62. Dario, P., et al. *An integrated miniature fingertip sensor*. in *Proceedings of the 7th International Symposium on Micro Machine and Human Science*. 1996.

63. Kawasaki, H., T. Komatsu, and K. Uchiyama, *Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand II.* IEEE/ASME Transactions on Mechatronics, 2002. **7**(3): p. 296-303.

64. Kang Ryeol, L., et al. *Fabrication of Polymer-based Flexible Tactile Sensing Module with Metal Strain Gauges and Interconnecter*. in *Proceedings of the 5th IEEE International Conference on Sensors*. 2006.

65. Jong-Ho, K., et al. *Development of tactile sensor with functions of contact force and thermal sensing for attachment to intelligent robot finger tip*. in *Proceedings of the 5th IEEE International Conference on Sensors*. 2006.

66. Engel, J., et al., *Polymer micromachined multimodal tactile sensors.* Sensors and Actuators A: Physical, 2005. **117**(1): p. 50-61.

67. Hong, Z. and E. So, *Hybrid resistive tactile sensing.* IEEE Transactions on Systems, Man, and Cybernetics: Part B, 2002. **32**(1): p. 57-65.

68. Shimojo, M., et al., *A High-Speed Mesh of Tactile Sensors Fitting Arbitrary Surfaces.* IEEE Sensors Journal, 2010. **10**(4): p. 822-830.

69. Kageyama, R., et al. *Development of soft and distributed tactile sensors and the application to a humanoid robot*. in *Proceedings of IEEE International Conference on System, Man and Cybernetics*. 1999.

70. Tognetti, A., et al. *Strain sensing fabric characterization*. in *Proceedings of IEEE International Conference on Sensors*. 2004.

71. *Apple Inc., iPod Touch Technology*. 2009 [cited 2010 07-09-2010]; Available from: http://www.apple.com/ipodtouch/features/technology.html.

72. Rocha, J.G., et al. *3 Axis Capacitive Tactile Sensor and Readout Electronics*. in *IEEE International Symposium on Industrial Electronics*. 2006.

73. Hoshi, T. and H. Shinoda. *A Large Area Robot Skin Based on Cell-Bridge System*. in *Proceedings of the 5th IEEE International Conference on Sensors*. 2006.

74. Hyung-Kew, L., C. Sun-Il, and Y. Euisik, *Dual-Mode Capacitive Proximity Sensor for Robot Application: Implementation of Tactile and Proximity Sensing Capability on a Single Polymer Platform Using Shared Electrodes.* IEEE Sensors Journal, 2009. **9**(12): p. 1748-1755.

75. Son, J.S., M.R. Cutkosky, and R.D. Howe. *Comparison of contact sensor localization abilities during manipulation*. in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1995.

76. Hyung-Kew, L., C. Sun-Il, and E. Yoon, *A Flexible Polymer Tactile Sensor: Fabrication and Modular Expandability for Large Area Deployment.* Journal of Microelectromechanical Systems, 2006. **15**(6): p. 1681-1686.

77. Hyung-Kew, L., et al., *Normal and Shear Force Measurement Using a Flexible Polymer Tactile Sensor With Embedded Multiple Capacitors.* Journal of Microelectromechanical Systems, 2008. **17**(4): p. 934-942.

78. Cotton, D., I.M. Graz, and S.P. Lacour, *A Multifunctional Capacitive Sensor for Stretchable Electronic Skins.* IEEE Sensors Journal, 2009. **9**(12): p. 2008-2009.

79. Sergio, M., et al. *A textile based capacitive pressure sensor*. in *Proceedings of IEEE International Conference on Sensors*. 2002.

80. Gray, B.L. and R.S. Fearing. *A surface micromachined microtactile sensor array*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1996.

81. Peer, A.S., M. Eric, and P.W. Rolf, *A sensor for dynamic tactile information with applications in human-robot interaction and object exploration.* Robotics and Autonomous Systems, 2006. **54**(12): p. 1005-1014.

82. *Pressure Profile Systems Inc., Robotouch - Overview*. 2007 [cited 2010 07-09-2010]; Available from: http://www.pressureprofile.com/products-robotouch.

83. Novak, J.L. and I.T. Feddema. *A capacitance-based proximity sensor for whole arm obstacle avoidance*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1992.

84. Dario, P., et al. *Ferroelectric polymer tactile sensors with anthropomorphic features*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1984.

85. Sawahata, K., J.P. Gong, and Y. Osada, *Soft and wet touch-sensing system made of hydrogel.* Macromolecular Rapid Communications, 1995. **16**(10): p. 713-716.

86. Howe, R.D. and M.R. Cutkosky, *Dynamic tactile sensing: perception of fine surface features with stress rate sensing.* IEEE Transactions on Robotics and Automation, 1993. **9**(2): p. 140-151.

87. Russell, R.A., *A Thermal Sensor Array to Provide Tactile Feedback for Robots.* The International Journal of Robotics Research, 1985. **4**(3): p. 35-39.

88. Yuji, J.i. and C. Sonoda. *A PVDF Tactile Sensor for Static Contact Force and Contact Temperature*. in *Proceedings of the 5th IEEE Conference on Sensors*. 2006.

89. Kolesar, E.S., Jr. and C.S. Dyson, *Object imaging with a piezoelectric robotic tactile sensor.* Journal of Microelectromechanical Systems, 1995. **4**(2): p. 87-96.

90. Cranny, A., et al., *Thick-film force and slip sensors for a prosthetic hand.* Sensors and Actuators A: Physical, 2005. **123-124**: p. 162-171.

91. Cotton, D.P.J., et al., *A Novel Thick-Film Piezoelectric Slip Sensor for a Prosthetic Hand.* IEEE Sensors Journal, 2007. **7**(5): p. 752-761.

92. Byungjune, C., C. Hyouk Ryeol, and K. Sungchul. *Development of tactile sensor for detecting contact force and slip*. in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2005.

93. Byungjune, C., et al. *Development of Anthropomorphic Robot Hand with Tactile Sensor : SKKU Hand II*. in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2006.

94. Motoo, K., F. Arai, and T. Fukuda, *Piezoelectric Vibration-Type Tactile Sensor Using Elasticity and Viscosity Change of Structure.* IEEE Sensors Journal, 2007. **7**(7): p. 1044-1051.

95. Ando, S. and H. Shinoda, *Ultrasonic emission tactile sensing.* IEEE Control Systems Magazine, 1995. **15**(1): p. 61-69.

96. Shinoda, H. and S. Ando. *Ultrasonic emission tactile sensor for contact localization and characterization*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1994.

97. Shinoda, H., K. Matsumoto, and S. Ando. *Tactile sensing based on acoustic resonance tensor cell*. in *Proceedings of the International Conference on Solid State Actuators*. 1997.

98. Krishna, G.M. and K. Rajanna, *Tactile sensor based on piezoelectric resonance.* IEEE Sensors Journal, 2004. **4**(5): p. 691-697.

99. Kimoto, A., N. Sugitani, and S. Fujisaki, *A Multifunctional Tactile Sensor Based on PVDF Films for Identification of Materials.* IEEE Sensors Journal, 2010. **10**(9): p. 1508-1513.

100. Saga, S., H. Kajimoto, and S. Tachi, *High-resolution tactile sensor using the deformation of a reflection image.* Sensor Review, 2007. **27**: p. 35-42.

101. Yamada, Y., et al. *Soft viscoelastic robot skin capable of accurately sensing contact location of object*. in *Proceedings of IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*. 1999.

102. Yamada, Y., et al., *Highly Soft Viscoelastic Robot Skin With a Contact Object-Location-Sensing Capability.* IEEE Transactions on Industrial Electronics,, 2005. **52**(4): p. 960-968.

103. Kamiyama, K., et al., *Vision-based sensor for real-time measuring of surface traction fields.* Computer Graphics and Applications, IEEE, 2005. **25**(1): p. 68-75.

104. Ferrier, N.J. and R.W. Brockett, *Reconstructing the Shape of a Deformable Membrane from Image Data.* The International Journal of Robotics Research, 2000. **19**(9): p. 795-816.

105. Maheshwari, V. and R.F. Saraf, *High-Resolution Thin-Film Device to Sense Texture by Touch.* Science, 2006. **312**(5779): p. 1501-1504.

106. Ohmura, Y., Y. Kuniyoshi, and A. Nagakubo. *Conformable and scalable tactile sensor skin for curved surfaces*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 2006.

107. Rebman, J. and K.A. Morris, *A Tactile Sensor with Electrooptical Transduction*, in *Robot Sensors Volume 2 - Tactile and Non-Vision*, A. Pugh, Editor. 1986, Springer-Verlag: New York. p. 145-155.

108. Cowie, B.M., et al., *Fibre Bragg grating sensors for distributive tactile sensing.* Measurement Science and Technology, 2007(1): p. 138.

109. Cowie, B.M., et al. *Distributive Tactile Sensing using Fibre Bragg Grating Sensors for Biomedical Applications*. in *Proceedings of IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*. 2006.

110. Yong-Lae, P., et al. *Force Sensing Robot Fingers using Embedded Fiber Bragg Grating Sensors and Shape Deposition Manufacturing*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 2007.

111. Yong-Lae, P., et al., *Exoskeletal Force-Sensing End-Effectors With Embedded Optical Fiber-Bragg-Grating Sensors.* IEEE Transactions on Robotics, 2009. **25**(6): p. 1319-1331.

112. Raibert, M.H. and J.E. Tanner, *Design and Implementation of a VLSI Tactile Sensing Computer.* The International Journal of Robotics Research, 1982. **1**(3): p. 3-18.

113. Someya, T., et al., *A large-area, flexible pressure sensor matrix with organic field-effect transistors for artificial skin applications.* Proceedings of the National Academy of Sciences of the United States of America, 2004. **101**(27): p. 9966-9970.

114. Dahiya, R.S., G. Metta, and M. Valle. *Development of fingertip tactile sensing chips for humanoid robots*. in *Proceedings of IEEE International Conference on Mechatronics*. 2009.

115. Vasarhelyi, G., et al., *Characterization of an Integrable Single-Crystalline 3-D Tactile Sensor.* IEEE Sensors Journal, 2006. **6**(4): p. 928-934.

116. Beccai, L., et al., *Design and fabrication of a hybrid silicon three-axial force sensor for biomechanical applications.* Sensors and Actuators A: Physical, 2005. **120**(2): p. 370-382.

117. Kane, B.J., M.R. Cutkosky, and G.T.A. Kovacs, *A traction stress sensor array for use in high-resolution robotic tactile imaging.* Journal of Microelectromechanical Systems, 2000. **9**(4): p. 425-434.

118. Huang, Y.-M., N.-C. Tsai, and J.-Y. Lai, *Development of tactile sensors for simultaneous, detection of normal and shear stresses.* Sensors and Actuators A: Physical, 2010. **159**(2): p. 189-195.

119. Takao, H., K. Sawada, and M. Ishida, *Monolithic silicon smart tactile image sensor with integrated strain sensor array on pneumatically swollen single-diaphragm structure.* IEEE Transactions on Electron Devices, 2006. **53**(5): p. 1250-1259.

120. Takao, H., et al. *Silicon MEMS tactile imager using flexible deformation of integrated pixel array*. in *Proceedings of the World Automation Congress*. 2008.

121. Takao, H., et al. *Multi-functional Monolithic-MEMS Tactile Imager Using Flexible Deformation of Silicon IC*. in *Proceedings of the IEEE Conference on Custom Integrated Circuits*. 2007.

122. Chu, Z., P.M. Sarro, and S. Middelhoek, *Silicon three-axial tactile sensor.* Sensors and Actuators A: Physical, 1996. **54**(1-3): p. 505-510.

123. Leineweber, M., et al., *New tactile sensor chip with silicone rubber cover.* Sensors and Actuators A: Physical, 2000. **84**(3): p. 236-245.

124. Eun-Soo, H., S. Jung-hoon, and K. Yong-Jun, *A Polymer-Based Flexible Tactile Sensor for Both Normal and Shear Load Detections and Its Application for Robotics.* Journal of Microelectromechanical Systems, 2007. **16**(3): p. 556-563.

125. Kawaguchi, H., et al., *Cut-and-paste customization of organic FET integrated circuit and its application to electronic artificial skin.* IEEE Journal of Solid-State Circuits, 2005. **40**(1): p. 177-185.

126. Mannsfeld, S.C.B., et al., *Highly sensitive flexible pressure sensors with microstructured rubber dielectric layers.* Nature Materials, 2010. **9**(10): p. 859-864.

127. Lacour, S.P., C. Tsay, and S. Wagner, *An elastically stretchable TFT circuit.* IEEE Electron Device Letters, 2004. **25**(12): p. 792-794.

128. Kim, D.-H., et al., *Stretchable and Foldable Silicon Integrated Circuits.* Science, 2008. **320**(5875): p. 507-511.

129. Takei, K., et al., *Nanowire active-matrix circuitry for low-voltage macroscale artificial skin.* Nature Materials, 2010. **9**(10): p. 821-826.

130. Salisbury, J., Jr. *Interpretation of contact geometries from force measurements*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1984.

131. Bicchi, A., J.K. Salisbury, and P. Dario. *Augmentation of grasp robustness using intrinsic tactile sensing*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1989.

132. Bicchi, A. *Intrinsic contact sensing for soft fingers*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1990.

133. Rossi, D.D., *Artificial tactile sensing and haptic perception.* Measurement Science and Technology, 1991. **2**(11): p. 1003.

134. Galvez, J.A., P. Gonzalez de Santos, and F. Pfeiffer, *Intrinsic tactile sensing for the optimization of force distribution in a pipe crawling robot.* IEEE/ASME Transactions on Mechatronics, 2001. **6**(1): p. 26-35.

135. Tsujimura, T. and T. Yabuta. *A tactile sensing method for employing force/torque information through insensitive probes*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1992.

136. Nicolson, E.J. and R.S. Fearing. *The reliability of curvature estimates from linear elastic tactile sensors*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1995.

137. Fearing, R. *Some experiments with tactile sensing during grasping*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1987.

138. Fearing, R.S. and T.O. Binford, *Using a cylindrical tactile sensor for determining curvature.* IEEE Transactions on Robotics and Automation, 1991. **7**(6): p. 806-817.

139. Ozaki, H., et al., *Pattern Recognition of a Grasped Object by Unit-Vector Distribution.* IEEE Transactions on Systems, Man and Cybernetics, 1982. **12**(3): p. 315-324.

140. Nowlin, W.C. *Experimental results on Bayesian algorithms for interpreting compliant tactile sensing data*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1991.

141. Allen, P.K. *Mapping haptic exploratory procedures to multiple shape representations*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1990.

142. Allen, P.K. and P. Michelman, *Acquisition and interpretation of 3-D sensor data from touch.* IEEE Transactions on Robotics and Automation, 1990. **6**(4): p. 397-404.

143. Petchartee, S. and G. Monkman. *Contact Classification using Tactile Arrays*. in *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*. 2007.

144. Abdullah, S.C., et al. *Object Exploration Algorithm Based on Three-Axis Tactile Data*. in *Proceedings of the 4th Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation (AMS)*. 2010.

145. Okamura, A.M. and M.R. Cutkosky, *Feature Detection for Haptic Exploration with Robotic Fingers.* The International Journal of Robotics Research, 2001. **20**(12): p. 925-938.

146. Browse, R.A., *Feature-Based Tactile Object Recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1987. **PAMI-9**(6): p. 779-786.

147. Gaston, P.C. and T. Lozano-Perez, *Tactile Recognition and Localization Using Object Models: The Case of Polyhedra on a Plane.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1984. **PAMI-6**(3): p. 257-266.

148. Luo, R. and T. Wen-Hsiang. *Object recognition using tactile image array sensors*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1986.

149. Payeur, P., et al., *Intelligent haptic sensor system for robotic manipulation.* IEEE Transactions on Instrumentation and Measurement,, 2005. **54**(4): p. 1583-1592.

150. Nakamoto, H., et al. *Shape classification based on tactile information by Universal Robot Hand*. in *Proceedings of the 35th IEEE International Conference on Industrial Electronics*. 2009.

151. Charlebois, M., K. Gupta, and S. Payandeh. *Curvature based shape estimation using tactile sensing*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1996.

152. Phung, T., et al., *Edge identification of a small object through a low-resolution tactile sensor array.* International Journal of Precision Engineering and Manufacturing, 2010. **11**(2): p. 247-254.

153. Levitt, T.S. and D.T. Lawton, *Qualitative navigation for mobile robots.* Artificial Intelligence, 1990. **44**(3): p. 305-360.

154. Salichs, M.A. and L. Moreno, *Navigation of mobile robots: open questions.* Robotica, 2000. **18**(3): p. 227-234.

155. Collins, T., J.J. Collins, and C. Ryan. *Occupancy grid mapping: An empirical evaluation*. in *Mediterranean Conference on Control & Automation*. 2007.

156. Tolman, E., *Cognitive maps in rats and men.* Psychological Review, 1948(55): p. 189--208.

157. Spoehr, K.T. and S.W. Lehmkuhle, *Visual Information Processing*. 1982, Oxford: Freeman. 298 p.

158. Rieser, J.J., D.A. Guth, and E.W. Hill, *Sensitivity to perspective structure while walking without vision.* Perception, 1986. **15**: p. 173-188.

159. Orly Lahav, D.M., *A blind person's cognitive mapping of new spaces using a haptic virtual environment.* Journal of Research in Special Educational Needs, 2003. **3**(3): p. 172-177.

160. Kitchin, R.M., M. Blades, and R.G. Golledge, *Understanding spatial concepts at the geographic scale without the use of vision.* Progress in Human Geography, 1997. **21**(2): p. 225-242.

161. Ungar, S., M. Blades, and C. Spencer, *The Construction of Cognitive Maps by Children with Visual Impairments*, in *The Construction of Cognitive Maps*. 1996. p. 247-273.

162. Harrison, Warren, and Tarr, *The geometry of "cognitive maps": Metric vs. ordinal structure.* Journal of Vision, 2001. **1**(3): p. 137-137.

163. Downs, R.M. and D. Stea, *Maps in minds: reflections on cognitive mapping*. 1977, New York: Harper and Row. 284 p.

164. Cuihong, Q., S. Hong, and X. Aiping. *Formal Properties of Cognitive Distance in Geographical Space*. in *Proceedings of 16th International Conference on Artificial Reality and Telexistence - Workshops*. 2006.

165. Foo, et al., *Do humans integrate routes into a "cognitive map"?* Journal of Vision, 2001. **1**(3): p. 186-186.

166. Byrne, R.W., *Memory for urban geography.* The Quarterly Journal of Experimental Psychology, 1979. **31**(1): p. 147 - 154.

167. O'Keefe, J. and L. Nadel, *The hippocampus as a cognitive map*. 1978, Oxford: Clarendon Press. 570 p.

168. Ekstrom, A.D., et al., *Cellular networks underlying human spatial navigation.* Nature, 2003. **425**(6954): p. 184-8.

169. Hafting, T., et al., *Microstructure of a spatial map in the entorhinal cortex.* Nature, 2005. **436**(7052): p. 801-806.

170. Vasudevan, S., et al., *Cognitive maps for mobile robots--an object based approach.* Robotics and Autonomous Systems, 2007. **55**(5): p. 359-371.

171. Messina, E. and S. Balakirsky, *Knowledge Representation and Decision Making for Mobile Robots*, in *Autonomous Mobile Robots:Sensing, Control, Decision Making and Applications*, S. Ge and F.L. Lewis, Editors. 2006, CRC: Boca Raton. p. 465-500.

172. Taylor, C.J. and D.J. Kriegman, *Vision-based motion planning and exploration algorithms for mobile robots.* IEEE Transactions on Robotics and Automation, 1998. **14**(3): p. 417-426.

173. Engelson, S.P. and D.V. McDermott. *Error correction in mobile robot map learning*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1992.

174. Kuipers, B.J. and Y.T. Byun, *A qualitative approach to robot exploration and map-learning*, in *Proceedings of the IEEE Workshop on Spatial Reasoning and Multi-Sensor Fusion*. 1987.

175. Zimmer, U.R., *Robust world-modelling and navigation in a real world.* Neurocomputing, 1996. **13**(2-4): p. 247-260.

176. Yamauchi, B. and R. Beer, *Spatial learning for navigation in dynamic environments.* IEEE Transactions on Systems, Man, and Cybernetics: Part B, 1996. **26**(3): p. 496-505.

177. Elfes, A., *Robot navigation: Integrating perception, environmental constraints and task execution within a probabilistic framework*, in *Reasoning with Uncertainty in Robotics*. 1996. p. 91-130.

178. Thrun, S., W. Burgard, and D. Fox, *Probabilistic robotics*. Intelligent robotics and autonomous agents. 2005, Cambridge, Mass.: MIT Press. 647 p.

179. Kortenkamp, D., R.P. Bonasso, and R. Murphy, eds. *Artificial intelligence and mobile robots: case studies of successful robot systems*. 1998, MIT Press: Cambridge, MA. 390.

180. Thrun, S., *Learning Occupancy Grid Maps with Forward Sensor Models.* Autonomous Robots, 2003. **15**: p. 111-127.

181. Moravec, H. and A. Elfes. *High resolution maps from wide angle sonar*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1985.

182. Matthies, L. and A. Elfes. *Integration of sonar and stereo range data using a grid-based representation*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1988.

183. Konolige, K., *Improved Occupancy Grids for Map Building.* Autonomous Robots, 1997. **4**(4): p. 351-367.

184. Thrun, S.B. *Exploration and model building in mobile robot domains*. in *Proceedings of IEEE International Conference on Neural Networks*. 1993.

185. Thrun, S., *Learning metric-topological maps for indoor mobile robot navigation.* Artificial Intelligence, 1998. **99**(1): p. 21-71.

186. Elmenreich, W., *Constructing dependable certainty grids from unreliable sensor data.* Robotics ann Autonomous Systems, 2008. **56**(12): p. 1094-1101.

187. Faria, D., R. Martins, and J. Dias, *Grasp Exploration for 3D Object Shape Representation Using Probabilistic Map*, in *Emerging Trends in Technological Innovation*, L. Camarinha-Matos, P. Pereira, and L. Ribeiro, Editors. 2010, Springer Boston. p. 215-222.

188. Inaba, M., et al. *A full-body tactile sensor suit using electrically conductive fabric and strings*. in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1996.

189. Oxley, C.H., et al., *Measurement of the reflection and transmission properties of conducting fabrics at milli-metric wave frequencies.* IET Science, Measurement & Technology, 2007. **1**(3): p. 166-169.

190. Swisstulle. *Shielding Fabrics: Technical Information*. 2005 cited 2010 02/10/2010]; Available from: http://www.swisstulle.co.uk/pdf/shielding_fabrics.pdf.

191. Sancaktar, E. and Y. Wei, *A Pressure Dependent Conduction Model for Electronically Conductive Adhesives*, in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. 1995. p. 231-+.

192. Balberg, I., *Tunneling and nonuniversal conductivity in composite materials.* Physical Review Letters, 1987. **59**(12): p. 1305.

193. Han, B.G. and et al., *Experimental study on the contribution of the quantum tunneling effect to the improvement of the conductivity and piezoresistivity of a nickel powder-filled cement-based composite.* Smart Materials and Structures, 2009. **18**(6): p. 065007.

194. Sze, S.M. and K.K. Ng, *Physics of semiconductor devices*. 3rd ed. 2007, Hoboken, N.J.: Wiley-Interscience. 815 p.

195. Dissado, L.A., J.C. Fothergill, and Institution of Electrical Engineers., *Electrical degradation and breakdown in polymers / Peregrinus on behalf of the Institution of Electrical Engineers*. 601 p.

196. Lampert and Mark, *Current injection in solids*. 1970: Academic Press.

197. Rose, R.M., L.A. Shepard, and J. Wulff, *Electronic properties*. Trans-ed. ed. Structures and properties of materials ;. 1966, New York ; London: Wiley. 306 p.

198. Radhakrishnan, S., *Effects of fillers on the electrical conduction in polymers.* Polymer Communications, 1985. **26**: p. 153-7.

199. Rose, A., *Space-Charge-Limited Currents in Solids.* Physical Review, 1955. **97**(6): p. 1538.

200. Heinz, E., *How DarkThought Plays Chess.* ICCA Journal, 1997. **20**(3): p. 166-176.

201. San Segundo, P., et al. *Efficient Search Using Bitboard Models*. in *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*. 2006.

202. San Segundo, P., et al. *A New Exact Bit-Parallel Algorithm for SAT*. in *Proceedings on the 20th IEEE International Conference on Tools with Artificial Intelligence*. 2008.

203. Pedraza, L., et al. *Building maps of large environments using splines and geometric analysis*. in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2008.

204. Intel, *Intel® 64 and IA-32 Architectures - Optimization Reference Manual*. 2010.

205. AMD, *Software Optimization Guide for AMD64 Processors*. 2005.

206. Buckingham, R. and A. Graham, *Snaking around in a nuclear jungle.* Industrial Robot: An International Journal, 2005. **32**: p. 120-127.

207. ISO/IEC, *14882:2003 (E) - Programming Languages -- C++*. 2003, ISO/IEC: Geneva.

208. Warren, H.S., *Hacker's delight*. 2003, Boston: Addison-Wesley. 306 p.

209. Sung Cheol, P., P. Min Kyu, and K. Moon Gi, *Super-resolution image reconstruction: a technical overview.* IEEE Signal Processing Magazine, 2003. **20**(3): p. 21-36.

Appendices

A5.1. Bitworld and MatrixWorld class code listings 240
A5.2. StopWatch class code listing 248
A6.1. BitworldMatrix class code listings 249
A7.1. Worked Example of Positional Noise in a Multi-Link Robot 253
A7.2. Proof of Limited Applicability of 1-bit Bitworld 255
A7.3. Derivation of k-value threshold 257
A7.4. Results of 1-bit Linear Optimisations 255
A7.5. Results of 1-bit Linear Optimisations using k-value thresholds......... 259
A7.6. Results of 2-bit Linear Optimisation... 261
A7.7. MATLAB Linear Optimisation Routines 262
A7.8. Comparison of brute-force and k-value approaches......................... 269
A7.9. Incremental updates after brute-force optimisation.......................... 271
A7.10. Converting Optimised Matrices into Boolean Functions 273

Appendix A5.1: Bitworld and MatrixWorld class code listings

This Appendix contains listings of the C++ code for the *Bitworld* and *MatrixWorld* classes.

Bitworld.h

```
#pragma once
#define _SECURE_SCL 0
#include <map>
#include <vector>

using namespace std;

class Bitworld
{
public:

	//constructors
	Bitworld(void);
	//additional c’tor allows preset size of this limited bitworld
	Bitworld(int xLower, int xUpper, int yLower, int yUpper);
	//destructor - will not specify a different one
	~Bitworld(void);

	//access functions
	void SetCoordinateOccupied( int x, int y );
	void SetCoordinateUnoccupied( int x, int y );
	int GetCoordinateValue(int x, int y);
	//function for rotating the bitworld image
	void RotateBitworld( float angle, Bitworld& outputWorld);
	/function for translating the bitworld
	void TranslateXNegative(int numberofSteps);
	void TranslateXPositive(int numberofSteps);
	void TranslateYNegative(int numberofSteps);
	void TranslateYPositive(int numberofSteps);
	//masks to help with shifting
	static __int64 maskX [];
	static __int64 maskY [];

private:
	//data stored in an std::map structure
	map<pair<int,int>, __int64> data;
	int xL, xU, yL, yU;

};
```

Bitworld.cpp

```
#include "Bitworld.h"
#include "intrin.h"
#include "math.h"

#pragma intrinsic(_BitScanForward64)
__int64 Bitworld::maskX[] =
{0x101010101010101,0x202020202020202,0x404040404040404,0x808080808080808,0x101
0101010101010,0x2020202020202020,0x4040404040404040,0x8080808080808080};
```

```
__int64 Bitworld::maskY[] = {0xFF,
0xFF00,0xFF0000,0xFF000000,0xFF00000000,0xFF0000000000,0xFF000000000000,0xFF00
000000000000};

Bitworld::Bitworld(int xLower, int xUpper, int yLower, int yUpper):
xL(xLower), xU(xUpper), yL(yLower), yU(yUpper)
{
       for( int i = xLower; i<= xUpper; ++i)
       {
              for( int j = yLower; j <= yUpper; ++j)
              {
                     data.insert(map<pair<int,int>,
__int64>::value_type(make_pair(i,j), 0));
              }
       }

}
void Bitworld::SetCoordinateOccupied(int x, int y)
{

       int xB = x>=0 ? (x & 7) : ( (x & 7) + 8) & 7;
       int yB = y>=0 ? (y & 7) : ( (y & 7) + 8) & 7;

       int XIndex = (x-xB) >> 3;
       int YIndex = (y-yB) >> 3;

       int bitShift = (yB << 3) + xB;

       map<pair<int,int>, __int64>::iterator it = data.find(pair<int,
int>(XIndex, YIndex));

       it->second |= (__int64)(1 << bitShift);

};

void Bitworld::SetCoordinateUnoccupied(int x, int y)
{
       int xB = x>=0 ? (x & 7) : ( (x & 7) + 8) & 7;
       int yB = y>=0 ? (y & 7) : ( (y & 7) + 8) & 7;
       int XIndex = (x-xB) >> 3;
       int YIndex = (y-yB) >> 3;
       int bitShift = (yB << 3) + xB;
       map<pair<int,int>, __int64>::iterator it = data.find(pair<int,
int>(XIndex, YIndex));

       it->second &= ~(1 << bitShift);
}

void Bitworld::TranslateXNegative(int numberofSteps)
{
       if(numberofSteps == 0) return;

       int n = numberofSteps & 7;
       int ns = (numberofSteps - n) >> 3;

       __int64 maskL = 0;
       for(int i=0;i<n;++i)
       {
              maskL |= maskX [i];
       }

       if( ns != 0 ) {
        for(int i = xL;i<=xU - ns;++i)
        {
            for(int j=yU;j>=yL;--j)
            {
                (*(data.find(pair<int,int>(i,j)))).second = 
(*(data.find(pair<int,int>(i + ns, j)))).second;
                (*(data.find(pair<int,int>(xU - i + 
xL,j)))).second = 0;
            }

        }
    }

    if( n != 0) {
        for(int y = yU; y<=yL;--y)
        {
            map<pair<int,int>, __int64>::iterator cIt = 
data.find(pair<int,int>(xL, y));

            //process this one here
            (*cIt).second <<= n;

            for(int x = xL + 1; x<xU;++x)
            {
              map<pair<int,int>, __int64>::iterator nIt = 
data.find(pair<int,int>(x,y));

              __int64 transfer = ((*nIt).second & maskL) >> (8-n) ;

              (*cIt).second |= transfer;
              (*nIt).second <<= n;

              cIt = nIt;
            }
        }
    }
}

void Bitworld::TranslateYNegative(int numberofSteps)
{
    int n = numberofSteps & 7;
    __int64 maskL = 0;
    for(int i=0;i<numberofSteps - n;++i)
    {
        maskL |= maskY[i];
    }

    for(int i = yL;i<=yU - (numberofSteps - n);++i)
    {
        for(int j=xU;j>=xL;--j)
        {
            (*(data.find(pair<int,int>(j,i)))).second = 
(*(data.find(pair<int,int>(i + numberofSteps - n, j)))).second;
            (*(data.find(pair<int,int>(j, yU - i + yL)))).second = 0;
        }

    }

    n <<= 3;

    for(int y = yU; y<=yL;--y)
    {
```

```
              map<pair<int,int>, __int64>::iterator cIt =
data.find(pair<int,int>(xL, y));

              //process this one here
              (*cIt).second <<= n;

              for(int x = xL + 1; x<xU;++x)
              {
                     map<pair<int,int>, __int64>::iterator nIt =
data.find(pair<int,int>(x,y));

                     __int64 transfer = (*nIt).second & maskL;
                     transfer >>= (8-n);

                     (*cIt).second |= transfer;
                     (*nIt).second <<= n;

                     cIt = nIt;

              }
       }

}

void Bitworld::RotateBitworld(float angle, Bitworld &outputWorld)
{
       for(map<pair<int,int>, __int64>::iterator it = data.begin(); it !=
data.end(); ++it)
       {
              __int64 value = it->second;
              pair<int,int> keyValue = it->first;

              __int64 XOffset = keyValue.first << 3;
              __int64 YOffset = keyValue.second << 3;

              while(value != 0)
              {
                     unsigned long index;
                     _BitScanForward64(&index, value);

                     int x = index & 7;
                     int y = (index - x) / 8 ;

                     //rotation matrix
                     float newX = x*cos(angle) - y*sin(angle);
                     float newY = x*sin(angle) + y*cos(angle);

                     int nX = (newX - floor(newX) < 0.5) ? (int)floor(newX) :
(int)floor(newX) + 1;
                     int nY = (newY - floor(newY) < 0.5) ? (int)floor(newY) :
(int)floor(newY) + 1;

                     outputWorld.SetCoordinateOccupied(nX + (int)XOffset, nY +
(int)YOffset);

                     value &= value - 1;
              }
       }

}

int Bitworld::GetCoordinateValue(int x, int y)
```

```
{
	int xB = x>=0 ? (x & 7) : ( (x & 7) + 8) & 7;
	int yB = y>=0 ? (y & 7) : ( (y & 7) + 8) & 7;

	int XIndex = (x-xB) >> 3;
	int YIndex = (y-yB) >> 3;

	int bitShift = (yB << 3) + xB;


	map<pair<int,int>, __int64>::iterator it = data.find(pair<int, 
int>(XIndex, YIndex));

	return (it->second & (1<<bitShift)) == 0 ? 0 : 1;

}
```

MatrixWorld.h

Since this is a template class, the implementation of the class is included in the class definition.

```
#pragma once
#define _SECURE_SCL 0
#include <deque>
#include "math.h"

using namespace std;

//typename T determines what data type is used to represent the class
template<typename T>
class MatrixWorld
{
public:

	int xOffset;
	int yOffset;

	MatrixWorld(void) {};
	//c’tor to set up the class to hold a particular dimension image
	MatrixWorld(int xLower, int xUpper, int yLower, int yUpper)
	{
		if(xLower < 0)
			xOffset = -xLower;
		else
			xOffset = 0;

		if(yLower < 0)
			yOffset = -yLower;
		else
			yOffset = 0;

		deque<T> dummyVector;
		for(int i = 0;i<=(yUpper-yLower);++i)
			dummyVector.push_back((T)0);

		for(int i = 0;i<=(xUpper-xLower);++i)
			data.push_back(dummyVector);

	}
```

```
        //destructor – empty/default
        ~MatrixWorld<T>(void) {};

        //set the value at a particular coordinate
        void SetCoordinateValue( T value, int x, int y )
        {
                int trueX = x + xOffset;
                int trueY = y + yOffset;

                data[x][y] = value;
        }
        //get the value at a particular coordinate
        T GetCoordinateValue( int x, int y )
        {
                int trueX = x + xOffset ;
                int trueY = y + yOffset;

                return data[trueX][trueY];
        }

        //function to rotate the image
        void RotateWorld(float angle, MatrixWorld<T> &outputWorld)
        {
                int i = -1;

                for(deque<deque<T>>::iterator it = data.begin();it !=
data.end(); ++it)
                {
                        ++i;

                        int j = -1;

                        for(deque<T>::iterator it2 = (*it).begin(); it2 !=
(*it).end(); ++it2)
                        {
                                ++j;

                                int x = i + this->xOffset;
                                int y = j + this->yOffset;

                                float newX = x*cos(angle) - y*sin(angle);
                                float newY = x*sin(angle) + y*cos(angle);

                                int nX = (newX - floor(newX) < 0.5) ?
(int)floor(newX) : (int)floor(newX) + 1;
                                int nY = (newY - floor(newY) < 0.5) ?
(int)floor(newY) : (int)floor(newY) + 1;

                                nX = nX - this->xOffset + outputWorld.xOffset;
                                nY = nY - this->yOffset + outputWorld.yOffset;

                                outputWorld.SetCoordinateValue(this-
>GetCoordinateValue(x,y), nX, nY);
                        }
                }
        };

        // branching method used for performance comparison
        void RotateWorldBranching(float angle, MatrixWorld<T> &outputWorld, T
threshold)
        {
                int i = -1;
        for(deque<deque<T>>::iterator it = data.begin();it != 
data.end(); ++it)
        {
            ++i;

            int j = -1;

            for(deque<T>::iterator it2 = (*it).begin(); it2 != 
(*it).end(); ++it2)
            {
                ++j;
                int x = i + this->xOffset;
                int y = j + this->yOffset;
                T currentValue = this->GetCoordinateValue(x,y);
                if(currentValue > threshold)
                {
                    float newX = x*cos(angle) - y*sin(angle);
                    float newY = x*sin(angle) + y*cos(angle);

                    int nX = (newX - floor(newX) < 0.5) ? 
(int)floor(newX) : (int)floor(newX) + 1;
                    int nY = (newY - floor(newY) < 0.5) ? 
(int)floor(newY) : (int)floor(newY) + 1;

                    nX = nX - this->xOffset + 
outputWorld.xOffset;
                    nY = nY - this->yOffset + 
outputWorld.yOffset;

                    outputWorld.SetCoordinateValue(currentValue, 
nX, nY);
                }
            }
        }
    };

    //series of functions used in an effort to find most efficient
    //method of translation
    void TranslateX1(int numberofSteps)
    {
        if(numberofSteps == 0) return;
        deque<T> dummyVector(data[0].size(), 0);

        for(int i=0;i<numberofSteps;++i)
        {
            data.push_back(dummyVector);
            data.pop_front();
        }
    }

    void TranslateX2(int numberofSteps)
    {
        if(numberofSteps == 0) return;
        for(int i = numberofSteps+1;i<data.size();++i)
        {
            for(int j = 0;j<data[0].size();++j)
            {
                data[i-numberofSteps-1][j] = data[i][j];
                data[i][j] = 0;
            }
        }
    }

    void TranslateX3(int numberofSteps)
    {
        if(numberofSteps == 0) return;

        if(numberofSteps < 116)
        {
            deque<T> dummyVector(data[0].size(), 0);

            for(int i=0;i<numberofSteps;++i)
            {
                data.push_back(dummyVector);
                data.pop_front();
            }
        }
        else
        {
            for(int i = numberofSteps+1;i<data.size();++i)
            {
                for(int j = 0;j<data[0].size();++j)
                {
                    data[i-numberofSteps-1] [j] = data [i][j];
                    data[i][j] = 0;
                }
            }
        }
    }
    void TranslateY(int numberofSteps)
    {
        if(numberofSteps == 0) return;

        if(numberofSteps < 116)
        {
            for(int i=0;i<numberofSteps;++i)
            {
                for(int j = 0;j<data[0].size();++j)
                {
                    data[j].push_back(0);
                    data[j].pop_front();
                }
            }
        }
        else
        {
            for(int i = numberofSteps+1;i<data.size();++i)
            {
                for(int j = 0;j<data[0].size();++j)
                {
                    data[i-numberofSteps-1][j] = data[i][j];
                    data[i][j] = 0;
                }
            }
        }
    }
private:
    //data is actually held in an std::deque of std:deque(s)
    deque<deque<T>> data;
};
```

Appendix A5.2: Stopwatch class code listing

This Appendix presents the C++ code used for the *StopWatch* class. This class is used to time sections of code.

StopWatch.h

```
#pragma once
#include <windows.h>

//normal structure used for the QueryPerformanceTimer functions
 typedef struct {
     LARGE_INTEGER start;
     LARGE_INTEGER stop;
 } stopWatchTime;


class StopWatch
{
private:
     stopWatchTime timer;
     LARGE_INTEGER frequency;
     double LIToSecs( LARGE_INTEGER & L) ;

 public:
     StopWatch() ;
     void startTimer( ) ;
     void stopTimer( ) ;
     double getElapsedTime() ;
};
```

StopWatch.cpp

```
#include "StopWatch.h"
#include <windows.h>

 double StopWatch::LIToSecs( LARGE_INTEGER & L) {
     return ((double)L.QuadPart /(double)frequency.QuadPart) ;
 }

 StopWatch:: StopWatch () {
     timer.start.QuadPart=0;
     timer.stop.QuadPart=0;
     QueryPerformanceFrequency( &frequency ) ;
 }

 void StopWatch::startTimer( ) {
     QueryPerformanceCounter(&timer.start) ;
 }

 void StopWatch::stopTimer( ) {
     QueryPerformanceCounter(&timer.stop) ;
 }

 double StopWatch::getElapsedTime() {
     LARGE_INTEGER time;
     time.QuadPart = timer.stop.QuadPart - timer.start.QuadPart;
     return LIToSecs( time ) ;
 }
```

Appendix A6.1: BitworldMatrix class code listings

This Appendix presents the C++ code of the *BitworldMatrix* class used. This class is used in Chapter 6 to consider the capabilities of the bitworld-counting method.

BitworldMatrix.h

```
#pragma once

class BitworldMatrix
{
public:
       BitworldMatrix (void);
       ~ BitworldMatrix (void);

       //access functions
       void SetCoordinateOccupied( int x, int y );
       void SetCoordinateUnoccupied( int x, int y );
       int GetCoordinateValue(int x, int y);

       void RotateBitworld( float angle, Bitworld& outputWorld);

       void TranslateXNegative(int numberofSteps);
       void TranslateXPositive(int numberofSteps);
       void TranslateYNegative(int numberofSteps);
       void TranslateYPositive(int numberofSteps);

       static __int64 maskX[];
       static __int64 maskY[];

private:

       map<pair<int,int>, __int64> data;
       int xL, xU, yL, yU;

};
```

BitworldMatrix.cpp

```
#include " BitworldMatrix.h"
#include "intrin.h"
#include "math.h"

#pragma intrinsic(_BitScanForward64)
__int64 BitworldMatrix::maskX[] =
{0x101010101010101,0x202020202020202,0x404040404040404,0x808080808080808,0x101
0101010101010,0x2020202020202020,0x4040404040404040,0x8080808080808080};
__int64 BitworldMatrix::maskY[] = {0xFF,
0xFF00,0xFF0000,0xFF000000,0xFF00000000,0xFF0000000000,0xFF000000000000,0xFF00
000000000000};

BitworldMatrix:: BitworldMatrix (int xLower, int xUpper, int yLower, int
yUpper): xL(xLower), xU(xUpper), yL(yLower), yU(yUpper)
{
       for( int i = xLower; i<= xUpper; ++i)
```

```
    {
        for( int j = yLower; j <= yUpper; ++j)
        {
            data.insert(map<pair<int,int>,
__int64>::value_type(make_pair(i,j), 0));
        }
    }

}
void BitworldMatrix::SetCoordinateOccupied(int x, int y)
{

    int xB = x>=0 ? (x & 7) : ( (x & 7) + 8) & 7;
    int yB = y>=0 ? (y & 7) : ( (y & 7) + 8) & 7;

    int XIndex = (x-xB) >> 3;
    int YIndex = (y-yB) >> 3;

    int bitShift = (yB << 3) + xB;

    map<pair<int,int>, __int64>::iterator it = data.find(pair<int,
int>(XIndex, YIndex));

    it->second |= (__int64)(1 << bitShift);

};

void BitworldMatrix::SetCoordinateUnoccupied(int x, int y)
{
    int xB = x>=0 ? (x & 7) : ( (x & 7) + 8) & 7;
    int yB = y>=0 ? (y & 7) : ( (y & 7) + 8) & 7;
    int XIndex = (x-xB) >> 3;
    int YIndex = (y-yB) >> 3;
    int bitShift = (yB << 3) + xB;

    map<pair<int,int>, __int64>::iterator it = data.find(pair<int,
int>(XIndex, YIndex));

    it->second &= ~(1 << bitShift);
}

void BitworldMatrix::TranslateXNegative(int numberofSteps)
{
    if(numberofSteps == 0) return;

    int n = numberofSteps & 7;
    int ns = (numberofSteps - n) >> 3;

    __int64 maskL = 0;
    for(int i=0;i<n;++i)
    {
        maskL |= maskX[i];
    }

    if( ns != 0 ) {
    for(int i = xL;i<=xU - ns;++i)
    {
        for(int j=yU;j>=yL;--j)
        {
            (*(data.find(pair<int,int>(i,j)))).second =
(*(data.find(pair<int,int>(i + ns, j)))).second;
            (*(data.find(pair<int,int>(xU - i + xL,j)))).second = 0;
```

```
                }
        }
        }

        if( n != 0) {
        for(int y = yU; y<=yL;--y)
        {
                map<pair<int,int>, __int64>::iterator cIt =
data.find(pair<int,int>(xL, y));

                //process this one here
                (*cIt).second <<= n;

                for(int x = xL + 1; x<xU;++x)
                {
                        map<pair<int,int>, __int64>::iterator nIt =
data.find(pair<int,int>(x,y));
                        __int64 transfer = ((*nIt).second & maskL) >> (8-n) ;

                        (*cIt).second |= transfer;
                        (*nIt).second <<= n;

                        cIt = nIt;
                }
        }
        }
}

void BitworldSum::TranslateYNegative(int numberofSteps)
{
        int n = numberofSteps & 7;
        __int64 maskL = 0;
        for(int i=0;i<numberofSteps - n;++i)
        {
                maskL |= maskY[i];
        }

        for(int i = yL;i<=yU - (numberofSteps - n);++i)
        {
                for(int j=xU;j>=xL;--j)
                {
                        (*(data.find(pair<int,int>(j,i)))).second =
(*(data.find(pair<int,int>(i + numberofSteps - n, j)))).second;
                        (*(data.find(pair<int,int>(j, yU - i + yL)))).second = 0;
                }
        }

        n <<= 3;

        for(int y = yU; y<=yL;--y)
        {
                map<pair<int,int>, __int64>::iterator cIt =
data.find(pair<int,int>(xL, y));

                //process this one here
                (*cIt).second <<= n;

                for(int x = xL + 1; x<xU;++x)
                {
                        map<pair<int,int>, __int64>::iterator nIt =
data.find(pair<int,int>(x,y));
                        __int64 transfer = (*nIt).second & maskL;
```

```
                     transfer >>= (8-n);

                     (*cIt).second |= transfer;
                     (*nIt).second <<= n;

                     cIt = nIt;
              }
       }
}

void BitworldSum::RotateBitworld(float angle, BitworldSum &outputWorld)
{
       for(map<pair<int,int>, __int64>::iterator it = data.begin(); it != 
data.end(); ++it)
       {
              __int64 value = it->second;
              pair<int,int> keyValue = it->first;

              __int64 XOffset = keyValue.first << 3;
              __int64 YOffset = keyValue.second << 3;

              while(value != 0)
              {
                     unsigned long index;
                     _BitScanForward64(&index, value);

                     int x = index & 7;
                     int y = (index - x) / 8 ;

                     //rotation matrix
                     float newX = x*cos(angle) - y*sin(angle);
                     float newY = x*sin(angle) + y*cos(angle);

                     int nX = (newX - floor(newX) < 0.5) ? (int)floor(newX) : 
(int)floor(newX) + 1;
                     int nY = (newY - floor(newY) < 0.5) ? (int)floor(newY) : 
(int)floor(newY) + 1;
                     outputWorld.SetCoordinateOccupied(nX + (int)XOffset, nY + 
(int)YOffset);

                     value &= value - 1;
              }
       }
}

int BitworldSum::GetCoordinateValue(int x, int y)
{
       int xB = x>=0 ? (x & 7) : ( (x & 7) + 8) & 7;
       int yB = y>=0 ? (y & 7) : ( (y & 7) + 8) & 7;

       int XIndex = (x-xB) >> 3;
       int YIndex = (y-yB) >> 3;

       int bitShift = (yB << 3) + xB;


       map<pair<int,int>, __int64>::iterator it = data.find(pair<int, 
int>(XIndex, YIndex));

       return (it->second & (1<<bitShift)) == 0 ? 0 : 1;

}
```

Appendix A7

This is an Appendix to Chapter 7 – Bitworld: Combining Noisy Images, containing additional data and derivations used in the text.

Appendix A7.1: Worked Example of Positional Noise in a Multi-Link Robot

Chapter 7 considers the impact of noisy pose measurements on tactile imaging. This section gives a brief worked example of a multi-link snake-arm robot to highlight how positional noise can accumulate. A schematic of a five-link snake arm robot is shown in Figure A7.1.

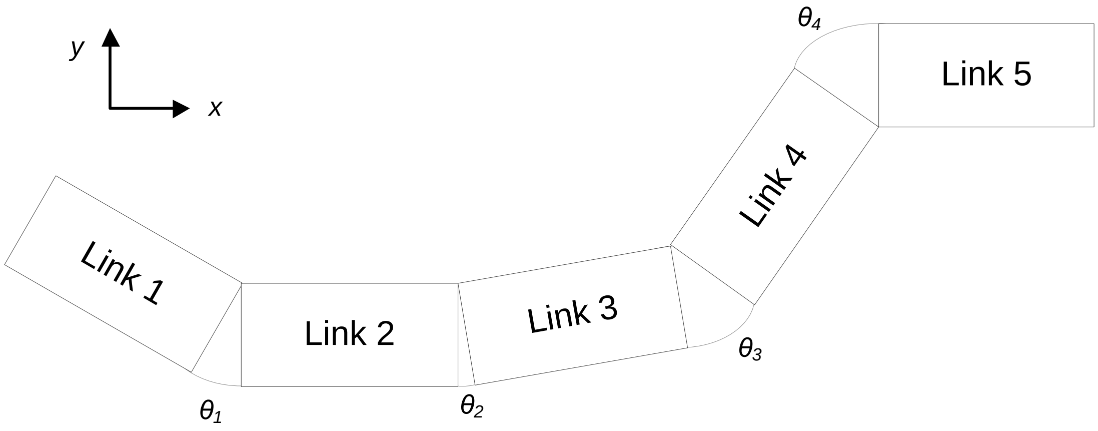


Figure A7.1: Schematic of a five-link snake-arm robot in an arbitrary pose.

In order to locate the origin of the tactile sensors, and so translate their images onto a common axis, the angles θ_n must be measured. Assuming that these measurements are independent, it can be assumed that they have the same standard deviation σ_θ. To provide an example of how large the errors can be expected to be, the error in determining the *y*-position of the end of a snake-arm robot will be calculated.

The *y*-position of the tip for a general snake-arm robot with link length *D* and *N* links can be calculated as

$$y = D \sum_{i=1}^{N-1} \sin \theta_i \qquad \text{(A7.1)}$$

The error in this value given a measured value of θ can be found from the standard formula

$$\sigma_y{}^2 = \sum_{i=1}^{N-1} (y_i - y)^2 \qquad \text{(A7.2)}$$

where $y_i$is the value of y where angle θ_i has been increased by its error σ_θ.

Let us assume a longer snake-arm robot than that shown in Figure A7.1 with *N*=15, *D*=200mm and, for simplicity, let us fix all of the angles to be equal to 10°. Figure A7.2 shows how the value of σ_y changes with changing values of σ_θ.

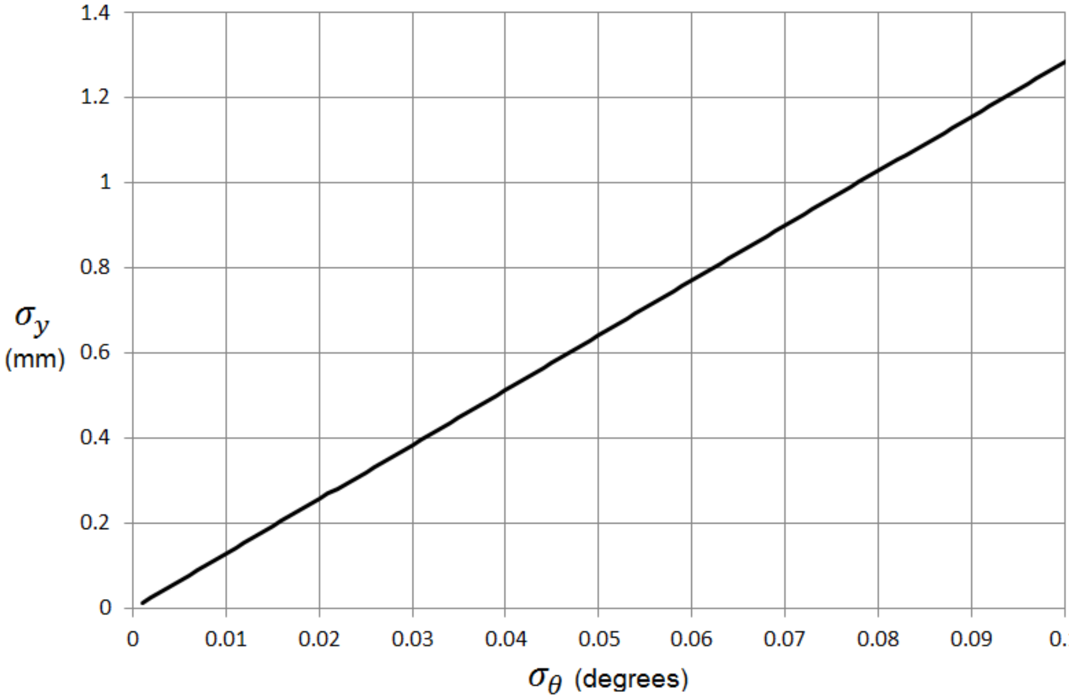


Figure A7.2: Error in the *y*-position of the tip of a snake arm robot with 15 links of length 200mm versus error in the measurements of angle.

This graph shows that the sample standard deviation of 0.7mm could have been achieved in this scenario with an error of 0.057° at each joint. For this angle, that equates to a 0.57% error. A greater number or longer links would increase the error in the *y*-position much more rapidly. This is also based on a very simple model of computing the *y*-position where, for example, rotations of joints and translation of the origin are not considered. The errors in these measurements would accumulate, giving larger errors more quickly.

This brief calculation shows a reasonable physical basis for an error in a position of a tactile sensor on a robot that is less than 1.5mm.

Appendix A7.2: Proof of Limited Applicability of 1-bit Bitworld

Section 7.2.1 of Chapter 7 stated that the application of matrices (7.4) to (7.7) did not result in improvements to the probability of selecting the correct edge cell. This Appendix applies the calculations to demonstrate the correctness of this statement.

Applying matrix (7.6) as shown in (A7.3) shows no improvement in the probability beyond that of a single measurement, since $P_c^{1\prime} = P_c^1$ for all *i*, based on the application of equation (7.3).

$$\begin{pmatrix} P_i^{1\prime} \\ P_i^{0\prime} \end{pmatrix} = \begin{pmatrix} P_i^1 & P_i^1 \\ P_i^0 & P_i^0 \end{pmatrix} \begin{pmatrix} P_i^1 \\ P_i^0 \end{pmatrix} = \begin{pmatrix} (P_i^1)^2 + P_i^1 P_i^0 \\ (P_i^0)^2 + P_i^1 P_i^0 \end{pmatrix} = \begin{pmatrix} P_i^1 \\ P_i^0 \end{pmatrix} \qquad \text{(A7.3)}$$

The application of matrices (7.5) to (7.7) leads to successful improvement only under a particular condition, which arises from the first criterion for a successful transition matrix. This is shown in (A7.4) with an application of matrix (7.7).

$$\begin{pmatrix} P_i^{1\prime} \\ P_i^{0\prime} \end{pmatrix} = \begin{pmatrix} P_i^1 & P_i^0 \\ P_i^0 & P_i^1 \end{pmatrix} \begin{pmatrix} P_i^1 \\ P_i^0 \end{pmatrix} = \begin{pmatrix} (P_i^1)^2 + (P_i^0)^2 \\ 2P_i^1 P_i^0 \end{pmatrix} \qquad \text{(A7.4)}$$

Performing this matrix multiplication leads to the requirements of (A7.5) and (A7.6) based on criterion 1.

$$P_i^1 < (P_i^1)^2 + (P_i^0)^2 \qquad \text{(A7.5)}$$

$$-P_i^0 < -2P_i^1 P_i^0 \qquad \text{(A7.6)}$$

Adding these two equations together yields:

$$P_i^1 - P_i^0 < (P_i^1)^2 + (P_i^0)^2 - 2P_i^1 P_i^0 \qquad \text{(A7.7)}$$

This can be simplified to:

$$P_i^1 - P_i^0 < (P_i^1 - P_i^0)^2 \qquad \text{(A7.8)}$$

Simplifying (A7.8) leads to two conditions, one of which must be satisfied for the transformation to meet the first criterion:

$$1 < P_i^1 \tag{A7.9}$$

$$P_i^0 > P_i^1 \tag{A7.10}$$

The condition of (A7.9) is not possible, since it implies a probability greater than unity. The second condition suggests that an improvement can only be achieved if $P_i^1 < 0.5$. This seems reasonable, but the second criterion must also be satisfied, namely that the probability of all the neighbouring cells being in the internal representation must be less than that of the correct edge cell after the transformation occurs. Figure A7.3 shows a plot of $P_i^{1\prime}$ from (A7.4) against P_i^1 for $P_i^1 < 0.5$.

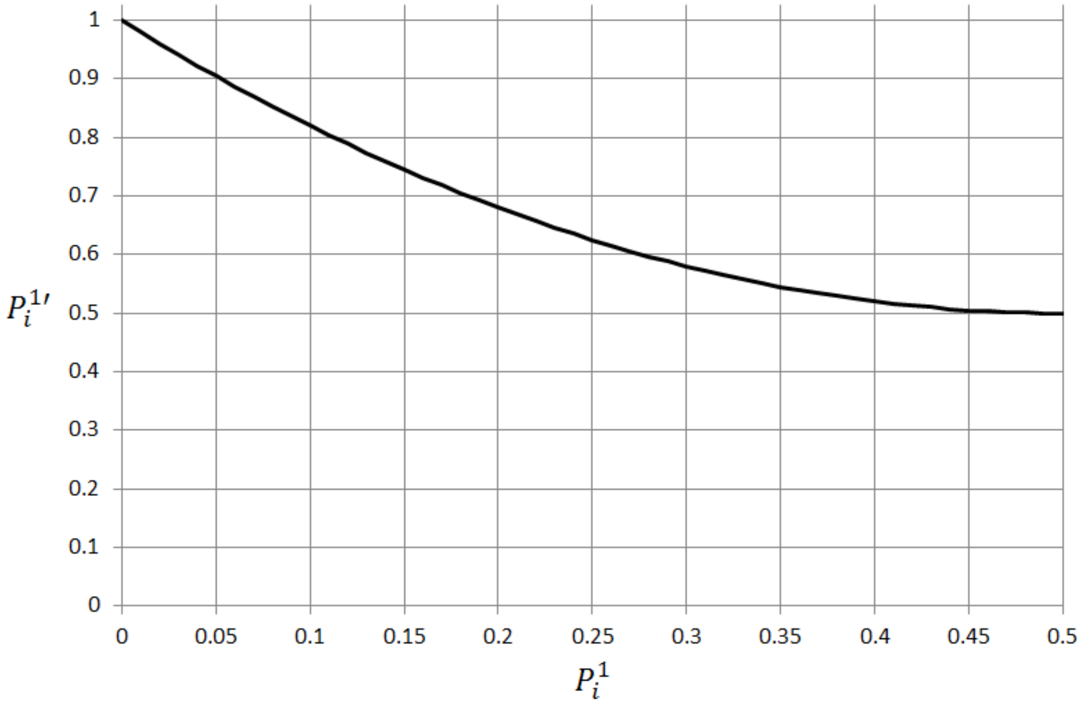


Figure A7.3: Plot of $P_i^{1\prime}$ against P_i^1 for $P_i^1 < 0.5$ that results from applying matrix (7.3)

This curve has a negative gradient in the range that has been shown to improve the probability of selecting the correct edge cell. This means that in order for the second criterion to be satisfied, the probability of the neighbouring cells being selected in the bitworld measurement must be higher than that of the correct edge cell.

This condition also arises if the matrix of (7.7) is used, as shown in (A7.11).

$$\begin{pmatrix} P_i^{1\prime} \\ P_i^{0\prime} \end{pmatrix} = \begin{pmatrix} P_i^0 & P_i^0 \\ P_i^1 & P_i^1 \end{pmatrix} \begin{pmatrix} P_i^1 \\ P_i^0 \end{pmatrix} = \begin{pmatrix} (P_i^0)^2 + P_i^1 P_i^0 \\ (P_i^1)^2 + P_i^1 P_i^0 \end{pmatrix} = \begin{pmatrix} P_i^0 \\ P_i^1 \end{pmatrix} \qquad \text{(A7.11)}$$

This type of probability distribution is unlikely to be particularly prevalent in the absence of systematic error. It is concluded that combining a one-bit bitworld measurement onto a one-bit internal representation would be insufficiently robust in the presence of noise in all except a narrow subset of distributions.

Appendix A7.3: Derivation of k-value threshold

This Appendix shows how the value of *k* for the *k*-value threshold is derived. Given that there are only two possible outcomes per cell in each individual measurement, the probabilities of bitworld cells being occupied in *n* measurements follow a binomial distribution. For the correct edge cell, the probability that it is set to one in *i* out of *n* trials is given by

$$\binom{n}{i} (P_C^1)^i (1 - P_C^1)^{n-i} \qquad \text{(A7.12)}$$

Given a threshold *k*, the post-combination probability of the correct edge cell being selected can be calculated from the binomial cumulative distribution function in equation (A7.13).

$$\sum_{i=k}^{n} \binom{n}{i} (P_C^1)^i (1 - P_C^1)^{n-i} \qquad \text{(A7.13)}$$

The goal is to find the suitable value of *k* for a given system, by maximising the difference between the final probabilities of the correct edge cell and the neighbouring cell. The neighbouring cell is given the index *W*. The goal requires maximising the function

$$\sum_{i=k}^{n} \binom{n}{i} \left[(P_C^1)^i (1 - P_C^1)^{n-i} - (P_W^1)^i (1 - P_W^1)^{n-i} \right] \qquad \text{(A7.14)}$$

through selection of a suitable *k*. This function will be maximised subject to *k* being chosen to ignore values in the sum that are negative, provided it is assumed that $P_C^1 > P_W^1$, which is the most likely case. This becomes the condition of equation (A7.15).

$$(P_C^1)^i(1-P_C^1)^{n-i} - (P_W^1)^i(1-P_W^1)^{n-i} \geq 0, \quad \forall i \geq k \tag{A7.15}$$

Solving this for the equality gives a formula for the appropriate value of *k* that maximises (A7.16).

$$k = \left\lceil \frac{n \ln\left(\frac{1-P_W^1}{1-P_C^1}\right)}{\ln\left(\frac{P_C^1}{P_W^1}\right) + \ln\left(\frac{1-P_W^1}{1-P_C^1}\right)} \right\rceil \tag{A7.16}$$

Appendix A7.4: Results of 1-bit Linear Optimisations

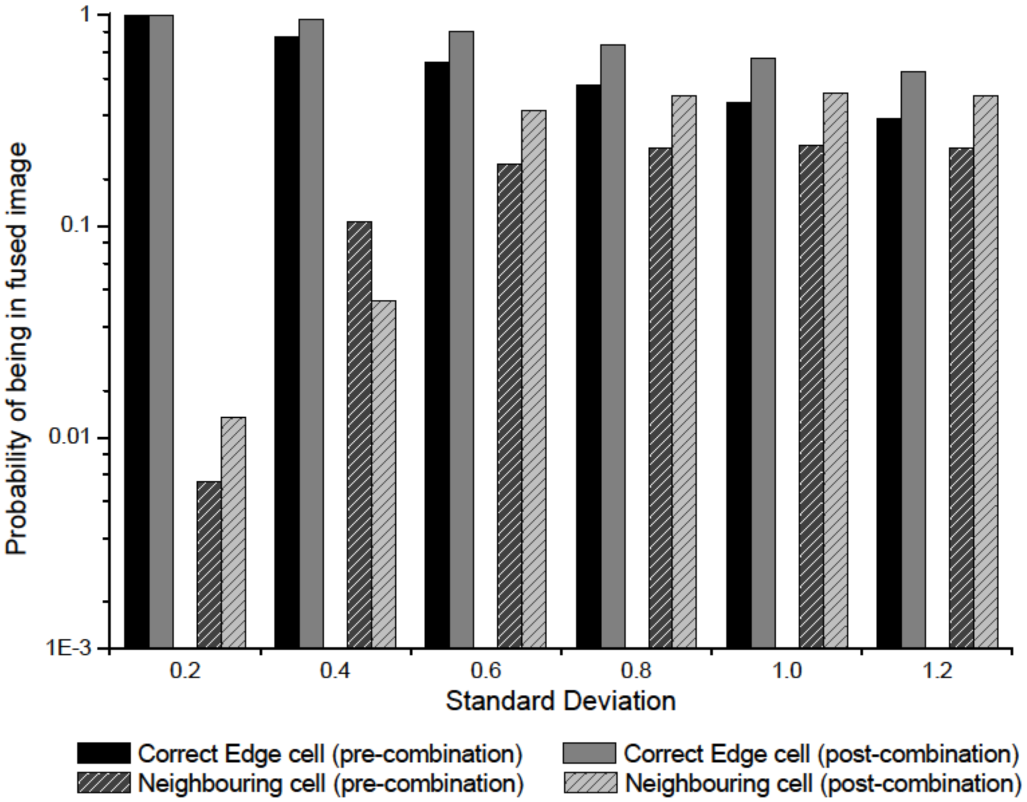


Figure A7.4: Comparison of the probabilities of the correct edge cell and the neighbouring cell before and after optimised fusion of a single-bit measurement onto a two-bit internal representation.

Appendix A7.5: Results of 1-bit Linear Optimisation using k*-value thresholds*

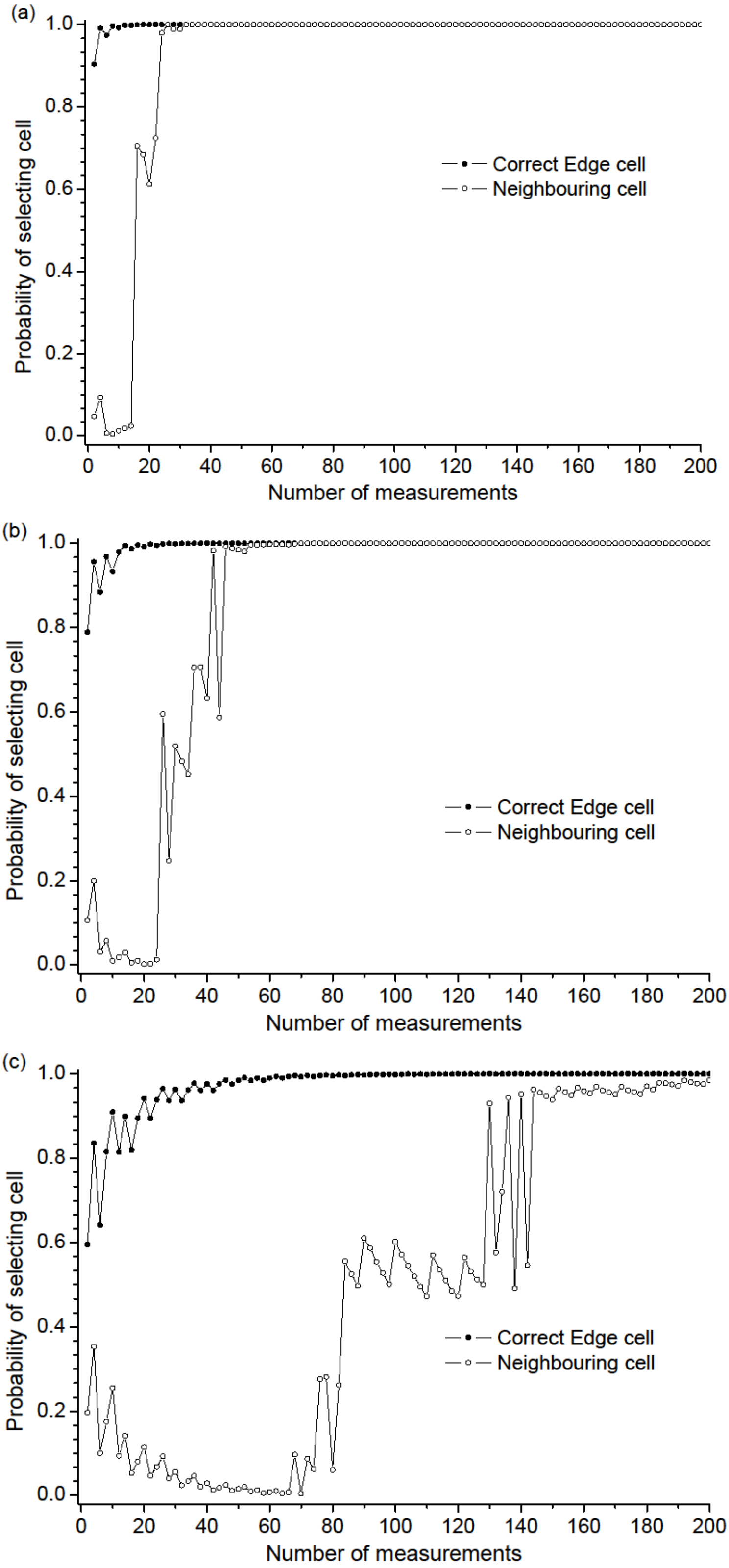


Figure A7.5: The change in the probability of occupancy for the correct edge cell and neighbouring cell versus numbers of measurements combined. Initial probabilities based on Gaussian distributions with standard deviations of (a) 0.2mm (b) 0.4mm (c) 0.6mm

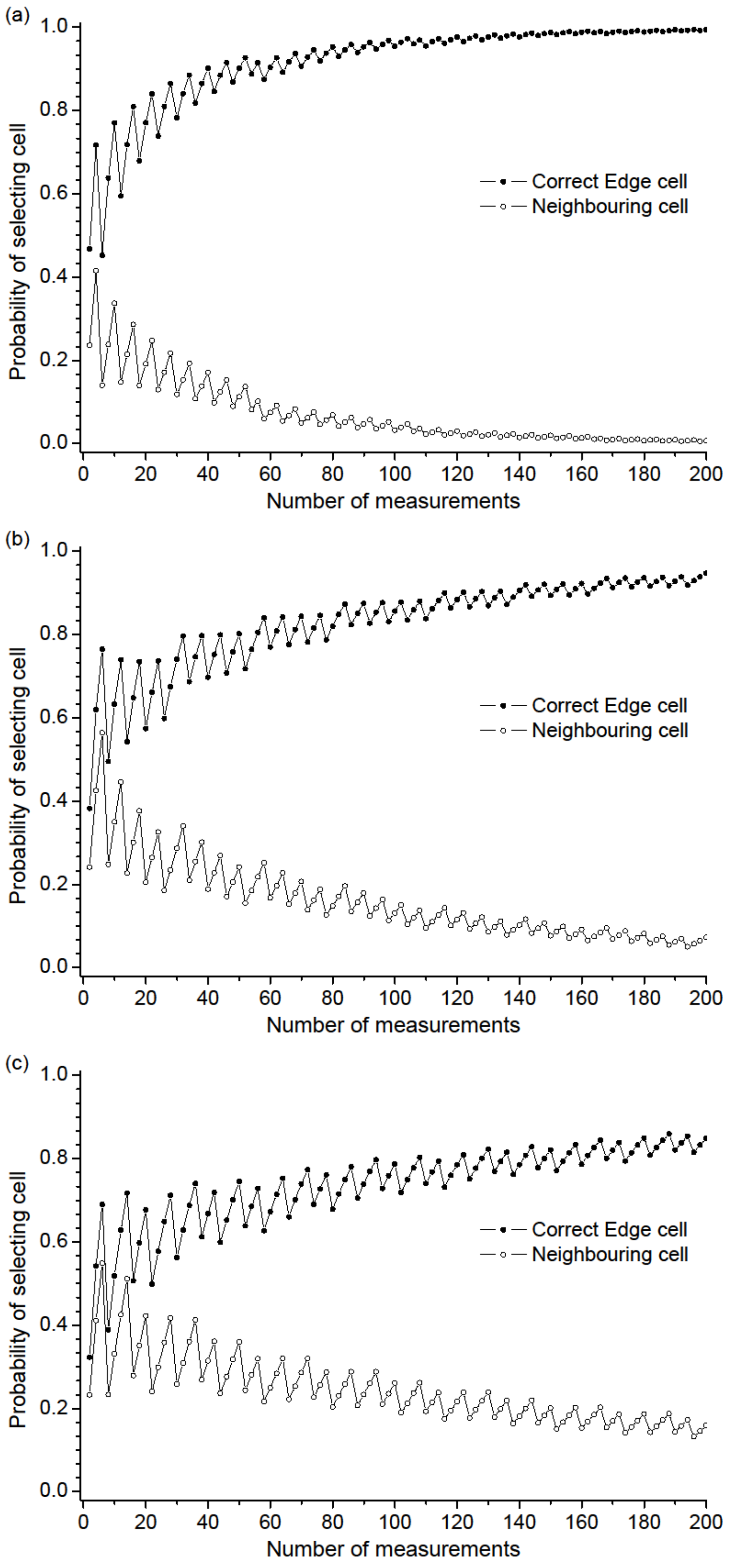


Figure A7.6: The variation in the probability of occupancy for the correct edge cell and neighbouring cell with the number of measurements combined. Initial probabilities based on Gaussian distributions with standard deviations of (a) 0.8mm (b) 1.0mm (c) 1.2mm

Appendix A7.6: Results of 2-bit Linear Optimisation

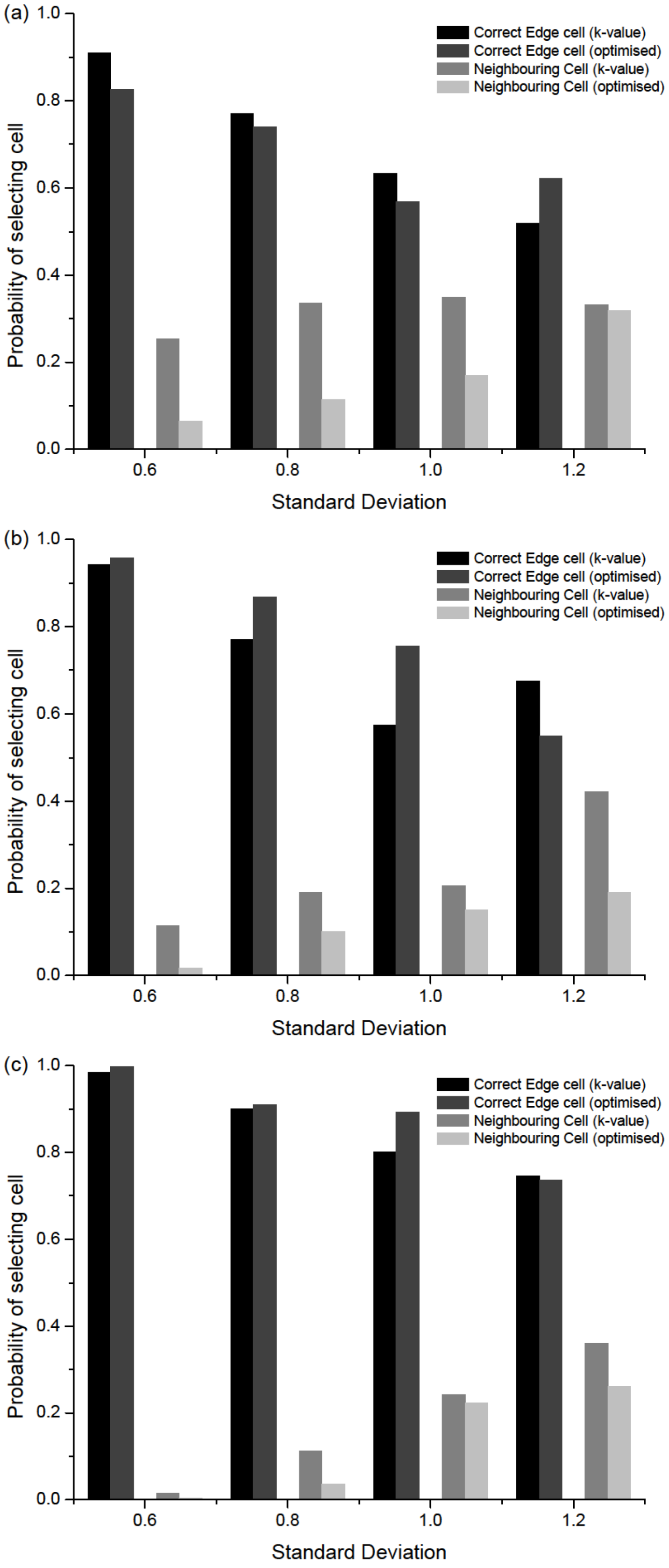


Figure A7.7: Histogram showing the results of linear optimisation for various standard deviations with N_S = 50 and N_i equal to (a) 5 (b) 10 and (c) 25.

Appendix A7.7: MATLAB Linear Optimisation Routines

This Appendix presents the MATLAB code used to configure the linear optimisations described in Chapter 7. The function *QOptimise* configures the system and calls *OptimiseSingleMeasurement* that performs the optimisation.

QOptimise.m

```
function      [ Vp, Vq,  Vdisp, pdash, qdash ] = QOptimise( stDev, nBin )
%First, calculate the single measurement vectors by dividing the counting
%probability between them
p = normcdf(0.5,0,stDev) - normcdf(-0.5,0,stDev);
q = normcdf(1.5,0,stDev) - normcdf(0.5,0,stDev);

%find distribution of probabilities for central and next best cell
pB = binopdf(0:nBin, nBin, p);
qB = binopdf(0:nBin, nBin, q);

%calculate the k-index to maximise the difference in the top two states
k = (nBin * log((1-q)/(1-p))) / (log(p/q) + log((1-q)/(1-p)));
k = 1 + ceil(k);

%set up containers for the combined probability vectors
pp = zeros(4,1);
qq = zeros(4,1);

%set the top value to the values determined by the k-index
pp(1) = sum(pB(k:(nBin+1)));
qq(1) = sum(qB(k:(nBin+1)));
% now allocate the remaining states based on the idea of generally
% distributing evenly, whilst trying where practical to maximise the
% difference in the zero-state as well.

remnantP = pB(1:(k-1));
```

```
remnantQ = qB(1:(k-1));

remnantPQ = remnantP - remnantQ;

    [C,I] = min(remnantPQ);

%set lowest value according to this minimum
pp(4) = sum(pB(1:I));
qq(4) = sum(qB(1:I));

%need to check that sufficient states remain to allocate to the two
%intermediate vector slots

remain = k - I - 1;
if(remain == 0)
   %unnecessary allocation for explanatory purposes
   pp(2) = 0;
   pp(3) = 0;
   qq(2) = 0;
   qq(3) = 0;
elseif(remain == 1)
   %odd case - must choose where the remaining probability falls.
   state = I + 1;
   if( pB(state) - qB(state) < 0)
      pp(3) = pB(state);
      qq(3) = qB(state);
   else
      pp(2) = pB(state);
      qq(2) = qB(state);
   end
elseif(remain == 2)
   %easy case - worth specialising
   pp(3) = pB(I + 1);
   pp(2) = pB(I + 2);
```

```
    qq(3) = qB(l + 1);
    qq(2) = qB(l + 2);
else
    % just split the remaining states into two, but must handle the case of
    % an odd number of states
    divided = (k - l - 1);
    if(mod(divided,2) == 0)  %even
        divided = divided/2;
        pp(3) = sum(pB((l+1):(l+divided)));
        pp(2) = sum(pB((l+divided+1:(k-1))));
        qq(3) = sum(qB((l+1):(l+divided)));
        qq(2) = sum(qB((l+divided+1:(k-1))));
    else   %odd
        pos3 = ceil(divided/2);
        pp(3) = sum(pB((l+1):(l+pos3)));
        pp(2) = sum(pB((l+pos3+1):(k-1)));
        qq(3) = sum(qB((l+1):(l+pos3)));
        qq(2) = sum(qB((l+pos3+1):(k-1)));
    end
end

%Perform two optimisations – flag changes optimisation strategy
     [ VpA, VqA,  VdispA ] = OptimizeHeuristicSingleMeasurement( pp, qq, pp, pp, 2, 0 );
     [ VpB, VqB,  VdispB ] = OptimizeHeuristicSingleMeasurement( pp, qq, pp, pp, 2, 1 );


%Calculate the effect of optimisation of the probability
pdashA = VpA*pp;
qdashA = VqA*qq;
pdashB=VpB*pp;
qdashB=VqB*qq;
```

```
%Choose the higher of the two optimisations
if (pdashA(1) - qdashA(1)) > (pdashB(1) - qdashB(1))
    Vp = VpA;
    Vq = VqA;
    Vdisp = VdispA;
    pdash = pdashA;
    qdash = qdashA;
 else
     Vp = VpB;
     Vq = VqB;
     Vdisp = VdispB;
     pdash = pdashB;
     qdash = qdashB;
 end

end
```

OptimizeHeuristicSingleMeasurement.m

```
function      [ Vp, Vq,  Vdisp ] = OptimizeHeuristicSingleMeasurement( pp, qq, pm, qm, numberOfBits, flag )
%Uses binary integer optimisation to establish the optimum selection of the
%Boolean heuristic according to summations of the probabilities of each
%branch in the tree being selected - see additional documentation for
%explanation

%sanity checking - have we presented a suitable set of inputs

Vp =      [];
Vq =      [];
Vdisp =      [];

if numberOfBits < 1
    disp('Invalid number of bits for input')
    return
end
```

```
numberOfStates = 2 ^ numberOfBits;

% need pp and qq to be of size numberOfStates x 1

if size(pp, 1) ~= numberOfStates || size(qq,1) ~= numberOfStates || size(pp,2) ~= 1 || size(qq,2) ~= 1
   disp('p- and/or q-arrays are the wrong dimension')
   return
end

if size(pm, 1) ~= numberOfStates || size(qm,1) ~= numberOfStates || size(pm,2) ~= 1 || size(qm,2) ~= 1
   disp('pm- and/or qm-arrays are the wrong dimension')
   return
end

%Checks over, now begin to formulate the problem


P = zeros(numberOfStates ^ 2, numberOfStates);
Q = zeros(numberOfStates ^ 2, numberOfStates);

for i=1:numberOfStates

   startPoint = 1 + ((i-1) * numberOfStates);
   endPoint = startPoint + (numberOfStates - 1);

   P(startPoint:endPoint, i) = pp;
   Q(startPoint:endPoint, i) = qq;
end

%calculate value of R
R = P*pm - Q*qm;
```

```
%calculate the functional representation of the optimisation function

f =     [-R; 0; 0; R];

%Establish equality constraints
Aeq = zeros(numberOfStates ^ 2, numberOfStates ^ 3);
beq = ones(numberOfStates ^ 2, 1);

for i = 1:(numberOfStates ^ 2)

    for j = 1:numberOfStates

        Aeq(i, ((j-1) * (numberOfStates^2) ) + i) = 1;

    end
end

%Establish inequality constraints
A = zeros(numberOfStates ^ 2, numberOfStates ^ 3);
b = (numberOfStates - 1) * ones(numberOfStates ^ 2, 1);

for i = 1:(numberOfStates ^ 2)

    startPoint = 1 + ((i-1) * numberOfStates);
    endPoint = startPoint + (numberOfStates - 1);

    A(i, startPoint:endPoint) = ones(1, numberOfStates);

end

%additional constraints to guarantee increase in p1' and decrease in q1'

aa = zeros(1,numberOfStates ^ 3);
```

```
if(flag == 0 || flag ==2)
   aa(1,1:(numberOfStates^2)) = -(P * pm)';
   A=cat(1,A,aa);
   b = cat(1,b, -pm(1));
end

aa = zeros(1,numberOfStates ^ 3);
if(flag == 1 || flag ==2)
   aa(1,(numberOfStates^3 - numberOfStates^2 + 1):(numberOfStates^3)) =
-(Q * qm)';
   A=cat(1,A,aa);
   b = cat(1,b, -qm(4));
end

aa = zeros(1,numberOfStates ^ 3);
if(flag == 1 || flag==2)
   aa(1,1:(numberOfStates^2)) = (Q * qm)';
   A=cat(1,A,aa);
   b = cat(1,b, qm(1));
end

%Perform optimisation using MATLAB function
    [x,fval,exitflag] = bintprog(f,A,b,Aeq,beq);

%Remaining code converts the result into a final set of probabilities
outMatp = zeros(numberOfStates, numberOfStates);
outMatq = zeros(numberOfStates, numberOfStates);
outGen =      [];

for i = 1:numberOfStates
   startPoint = 1 + ((i-1) * numberOfStates ^ 2);
   endPoint = startPoint + ((numberOfStates^2) - 1);
   r = x(startPoint:endPoint , 1);
   for j = 1:numberOfStates
```

```
        startPoint = 1 + ((j-1) * numberOfStates);
        endPoint = startPoint + (numberOfStates - 1);

        s = r(startPoint:endPoint, 1);
        outMatp(i, j) = s' * pp;
        outMatq(i,j) = s' * qq;
    end
    outGen = cat(1,outGen, r');
end

Vp = outMatp;
Vq = outMatq;
Vdisp = outGen;

End
```

Appendix A7.8: Comparison of brute-force and k*-value approaches*

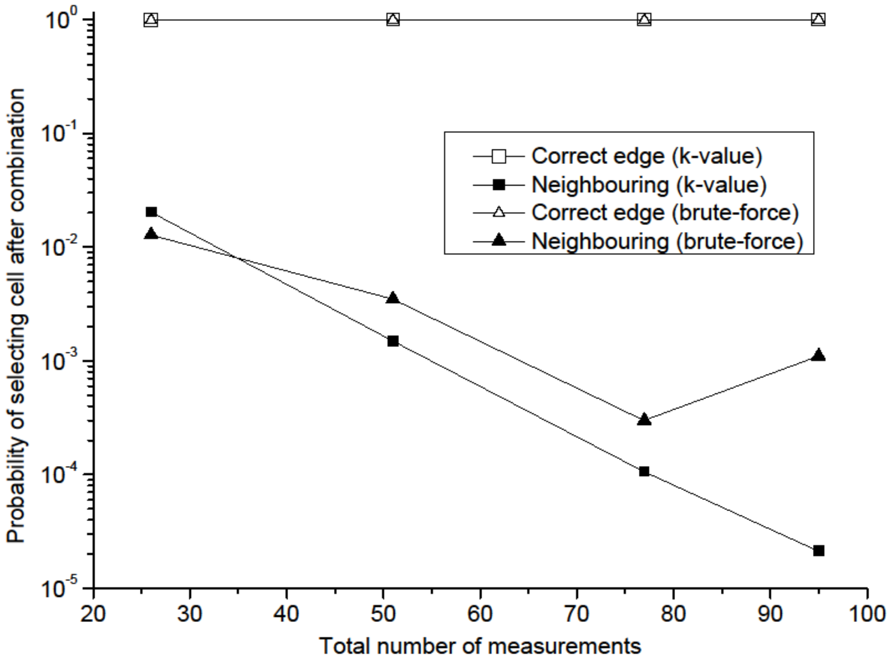


Figure A7.8: Probabilities of selecting the correct edge and neighbouring cells after applying the results of two optimisation methods with initial probabilities determined from a Gaussian distribution with a standard deviation of 0.6.

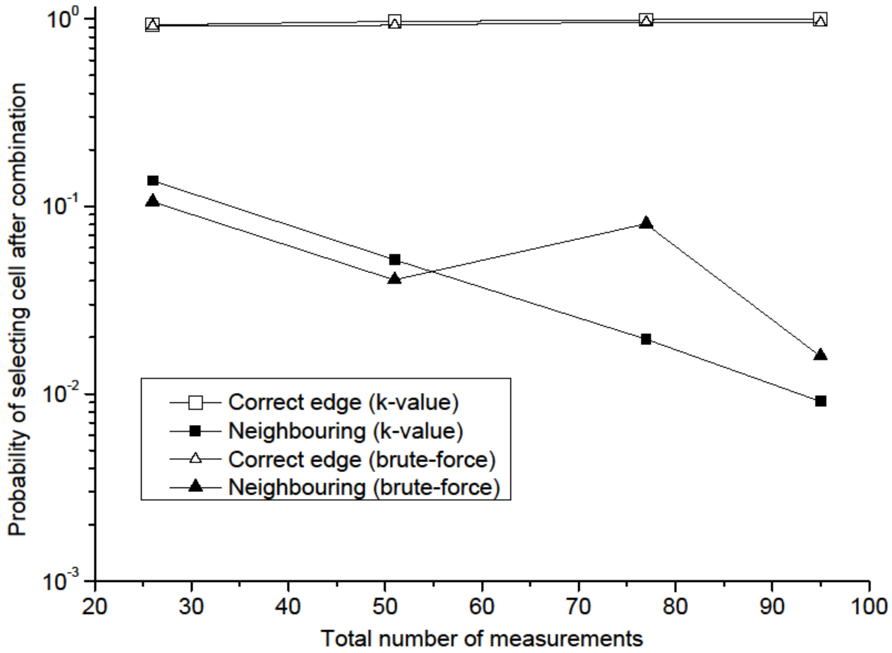


Figure A7.9: Probabilities of selecting the correct edge and neighbouring cells after applying the results of two optimisation methods with initial probabilities determined from a Gaussian distribution with a standard deviation of 0.8.

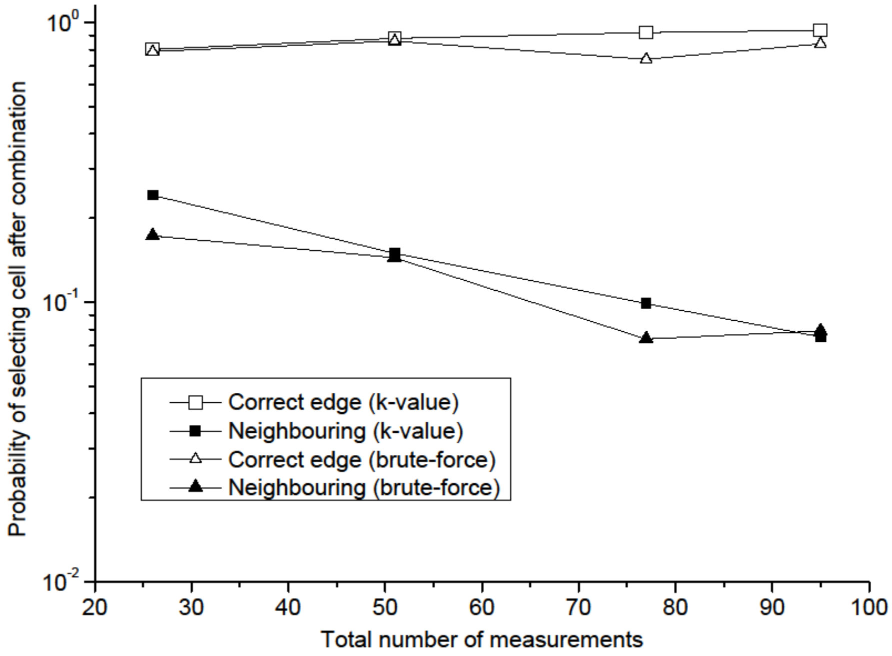


Figure A7.10: Probabilities of selecting the correct edge and neighbouring cells after applying the results of two optimisation methods with initial probabilities determined from a Gaussian distribution with a standard deviation of 1.0.

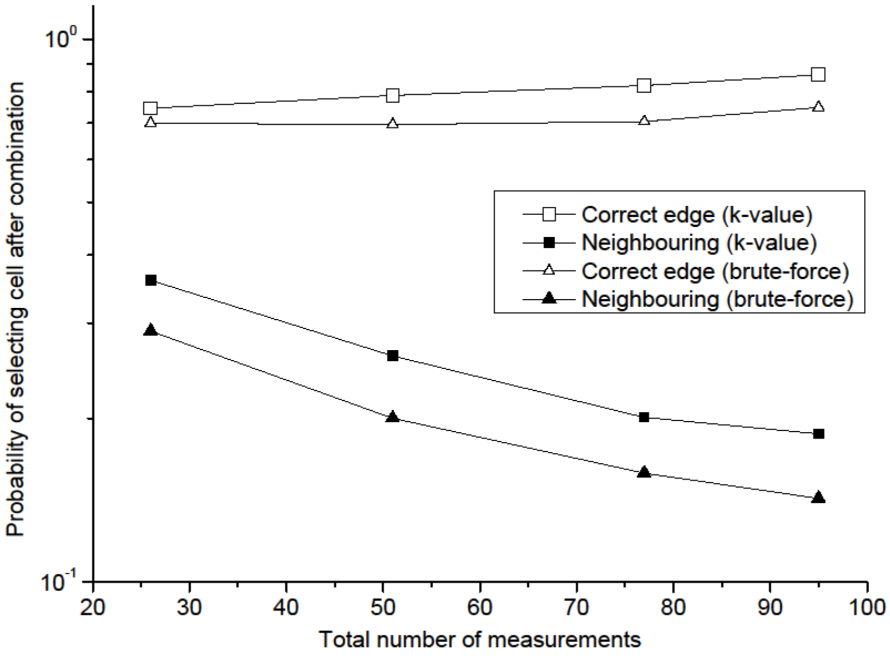


Figure A7.11: Probabilities of selecting the correct edge and neighbouring cells after applying the results of two optimisation methods with initial probabilities determined from a Gaussian distribution with a standard deviation of 1.2.

Appendix A7.9: Incremental updates after brute-force optimisation

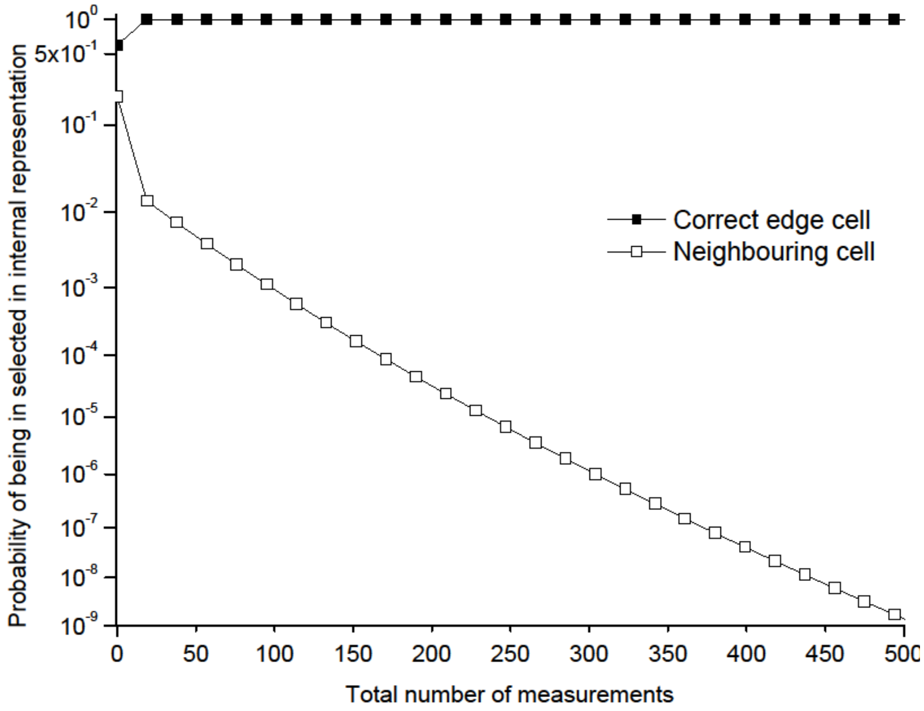


Figure A7.12: Effect of incrementally applying the results of brute-force optimisation to update an internal representation. Standard deviation of the underlying Gaussian distribution is 0.6.

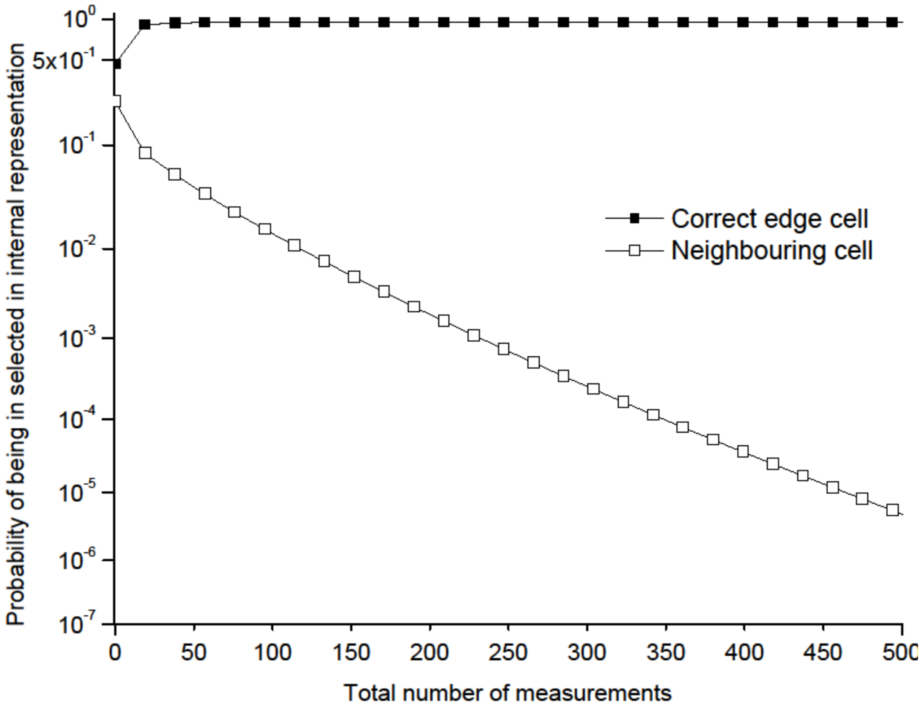


Figure A7.13: Effect of incrementally applying the results of brute-force optimisation to update an internal representation. Standard deviation of the underlying Gaussian distribution is 0.8.

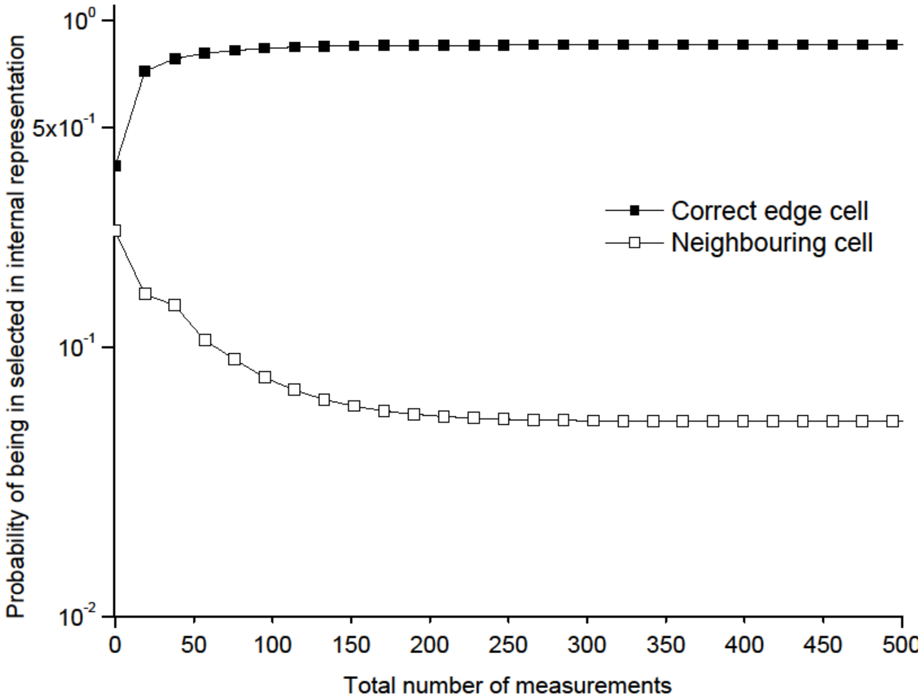


Figure A7.14: Effect of incrementally applying the results of brute-force optimisation to update an internal representation. Standard deviation of the underlying Gaussian distribution is 1.0.

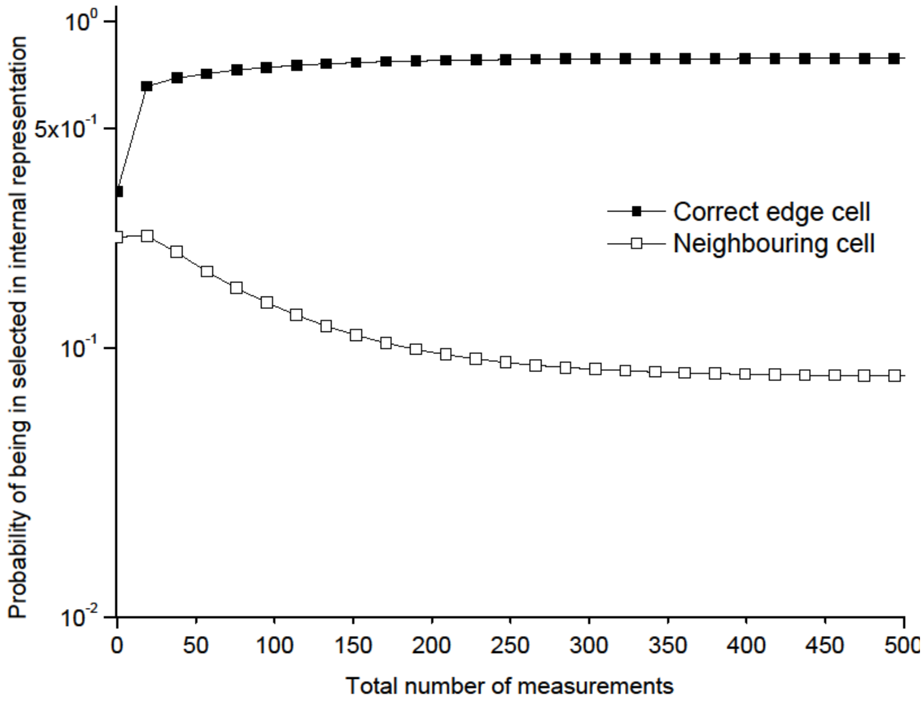


Figure A7.15: Effect of incrementally applying the results of brute-force optimisation to update an internal representation. Standard deviation of the underlying Gaussian distribution is 1.2.

Appendix A7.10: Converting Optimised Matrices into Boolean Functions

This Appendix illustrates how the optimised C-matrices are converted into Boolean expressions. When optimising the combination process for measurements affected by Gaussian noise with standard deviation of 0.8, the program gave the optimum *C*-matrix as

$$\underline{\underline{C}} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \qquad \text{(A7.17)}$$

This matrix needs to be interpreted to produce a truth table that represents the optimum Boolean function. From left-to-right, this matrix contains four 4-by-4 matrices. These define the effect of the optimisation for the current state in the internal representation, from State 1 (‘definitely occupied’) on the left to State 4 (‘definitely not occupied’) on the right.

Within these 4-by-4 matrices, the columns specify the behaviour for a given measurement. The leftmost column defines the transition if the bitworld cell in the measurement is in State 1 and the rightmost column in the 4-by-4 matrix defines the transition if the bitworld cell in the measurement is in State 4. In these columns, the row in which the number 1 appears determines the state held in that bitworld cell after the combination from State 1 at the top to State 4 at the bottom. Taking the matrix result of equation (A7.17) as an illustrative example, the first four columns are:

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{A7.18}$$

The whole matrix represents a condition that could be written in words as "if the cell in the current internal representation is in State 1…". The first column adds the statement, "…and the cell in the new measurement is in State 1…". Since the first row contains the number 1, the complete statement is, "If the cell in the current internal representation is in State 1 and the cell in the new measurement is in State 1 then update the cell in the internal representation to State 1".

By defining each state by a combination of two bits, it is possible to write the transition as a truth table with four inputs and two outputs. The four inputs are the two bits representing the current state of the cell in the internal representation, denoted C_1 and C_2, and the two bits representing the state of the cell in the combined measurement, denoted M_1 and M_2. The outputs are the updated pair of bits to revise the internal representation, denoted U_1 and U_2. The four states used in the two-bit representation were arbitrarily assigned bit values and these are shown in Table A7.1.

State	Bit 1	Bit 2
1 ('definitely occupied)	1	0
2 ('probably occupied')	1	1
3 ('probably not occupied')	0	1
4 ('definitely not occupied')	0	0

Table A7.1: Values assigned to the two bits that represent the four states used in this work.

Based on these definitions and the result in equation (A7.17), the optimum truth table for the standard deviation of 0.8 is given in Table A7.2.

Minimising this truth table using standard techniques produces the Boolean expressions of (A7.19) and (A7.20):

$$U_1 = C_1\overline{C_2} + \overline{M_2}C_2M_1 \qquad (7.18)$$

$$U_2 = C_1\overline{M_1\ M_2} + C_1M_2 + \overline{C_1}C_2M_1 \qquad (7.19)$$

C₁	***C₂***	***M₁***	***M₂***	***U₁***	***U₂***
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	1	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	0	1

Table A7.2: Truth table that represents the optimum Boolean function as determined by the brute-force optimisation method for a standard deviation of 0.8.