The Simulation Scan Comparison Process for Monitoring Manufacturing Environments

Submitted by Steven McAtee
In fulfilment of the requirements for
Masters Degree of Engineering (MEng)
Swinburne University of Technology
2014
Declaration

This thesis contains no material that has been accepted for the award of any other
degree or diploma in any university or college of advanced education and to the best of
my knowledge contains no material previously published or written by another person
except where due reference is made.

Steven McAtee
Abstract
Automated manufacturing systems have developed significantly in the last few years, but they still remain fragile and unreliable under operational conditions. The objective of this research is to develop a decision support system that can identify problems that occur in a manufacturing environment by comparing a simulation of a manufacturing process with a three dimensional (3D) scan of the physical environment.

The Simulation Scan Comparison (SSC) process was designed to compare a simulation of a manufacturing process to a 3D scan of an environment. The SSC process integrates three components: Environmental Analysis, Task Analysis and Path Planning. The Environmental Analysis is based on using a simulation for background subtraction of the 3D scan. This technique is capable of identifying known simulation objects, detecting missing simulation objects and isolating unknown objects from the scan data. The Environmental Analysis was tested by analysing the background subtraction process. For this analysis the scan was divided into areas where the scan matched to objects, areas conflicted with simulation objects and areas not matched to any simulation objects. Three scenarios were created to test the matching process: The simulation and scan environment were kept the same to test the matching of known objects; objects were added to the simulation but not the physical environment to test the missing object detection; objects were added to the physical environment, but not the simulation environment, to test the unknown object detection.

The analysis of these scenarios showed that the SSC process could successfully detect that simulation objects were correctly matched 98% of the time and that unknown objects were present 90% of the time. The use of oriented bounding boxes to measure
unknown objects was analysed to determine the reliability of the measurements produced. This shows that the measurement reliability of the objects was related to the scan resolution and that to be detected successfully objects should be at least three scan resolutions in any dimension. Two tracking algorithms based on analysing the unknown object data from unknown object were tested; a position based method and a velocity based method. The results demonstrate that the SSC object isolation technique can be utilised to identify and track unknown objects in a known workspace. This demonstrates that the unknown object detections are robust enough over time to perform object tracking.

The Task Analysis compares unknown objects to the existing path of a robotic manipulator from the simulation to determine if the objects could cause a collision. If any objects are missing from the simulation or a potential collision is detected, the physical robot can be stopped or rerouted using path planning around the object.

During the operation of the SSC process, the system was able to detect when an unknown object interfered with a movement task and was capable of planning a path around the obstacle and executing the new path on the physical robot.

The Simulation Scan Comparison (SSC) process developed for this research demonstrates the value of the combined use of simulation and scan of the physical environment to interpret scan data allowing decisions to be made about tasks. It is expected that this process can be applied in industrial and more general applications to enable these application to operate “intelligently”, to be more robust to changes in the environment, and ultimately to be more productive.
Acknowledgements

I would like to thank my primary supervisor Prof. Romesh Nagarajah for his continued guidance and constant support throughout the course of this research.

I would like to thank my co-ordinating supervisor Dr Michelle Dunn for her tireless efforts especially with all of the writing involved in this thesis and previous works for publication.

Thank you to Mr Phillip Crothers of Boeing for the assistance in examining a real manufacturing environment and some of the inspiration for this project.

Thanks also to the technical and support staff of Swinburne University Hawthorn for their assistance with the robot and the work area, especially Krys Stachowicz, Walter Chetcuti, and Sang Nguyen.

Finally thanks to all my family and friends for their support over the years especially my father, Robert McAtee, and brother, Paul McAtee, who helped with reviewing and editing this thesis.
Table of Contents

Abstract ............................................................................................................................. i
Acknowledgements ........................................................................................................ iii
Table of Contents ........................................................................................................ iv
List of Figures ................................................................................................................. viii
List of Tables .................................................................................................................... x
1 Introduction................................................................................................................ 1
  1.1 Background ........................................................................................................... 3
    1.1.1 Environmental Analysis Techniques ................................................................. 4
    1.1.2 Task Analysis in Industry ................................................................................... 7
    1.1.3 Path Planning Applications .............................................................................. 9
    1.1.4 Summary .......................................................................................................... 11
  1.2 Overview of Proposed Methodology .................................................................. 11
    1.2.1 3D Simulation................................................................................................... 12
    1.2.2 3D Scanning ...................................................................................................... 13
    1.2.3 Environmental Analysis Implementation ......................................................... 14
    1.2.4 Task Analysis for Movement tasks ................................................................. 16
    1.2.5 Path Planning Implementation ...................................................................... 17
    1.2.6 SSC process .................................................................................................... 18
  1.3 Overview of Experimental Procedures ......................................................... 18
    1.3.1 Measurement Reliability .................................................................................. 19
    1.3.2 Tracking .......................................................................................................... 20
    1.3.3 Movement Task Analysis Experiments ......................................................... 22
    1.3.4 Path Planning Experiments .......................................................................... 22
    1.3.5 SSC process testing ....................................................................................... 23
  1.4 Perceived Contributions ............................................................................... 24
    1.4.1 Simulation Background subtraction .................................................................. 25
    1.4.2 Missing Objects ............................................................................................... 26
    1.4.3 Unknown Object Detection ............................................................................ 26
    1.4.4 Movement Task Analysis ............................................................................... 27
    1.4.5 Path Planning using both Simulation and Scan Data .................................... 27
    1.4.6 SSC process .................................................................................................... 27
  1.5 Structure .............................................................................................................. 28
2 Literature Review .................................................................................................... 30
  2.1 Path Planning .................................................................................................... 31
    2.1.1 Potential Fields ............................................................................................... 33
    2.1.2 Graph Methods ............................................................................................... 38
    2.1.3 Uncertainty in Data ......................................................................................... 41
    2.1.4 Manipulator Path Planning ............................................................................. 44
  2.2 Collision Avoidance ....................................................................................... 53
    2.2.1 Collision Detection ......................................................................................... 54
    2.2.2 Safety Systems ............................................................................................... 55
  2.3 3D Scanning .................................................................................................... 59
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4</td>
<td>3D Simulation</td>
<td>61</td>
</tr>
<tr>
<td>2.5</td>
<td>Combined Scanning and Simulation</td>
<td>62</td>
</tr>
<tr>
<td>2.6</td>
<td>Summary of Literature Review</td>
<td>63</td>
</tr>
<tr>
<td>3</td>
<td>Methodology</td>
<td>65</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>65</td>
</tr>
<tr>
<td>3.2</td>
<td>3D Environment Analysis</td>
<td>68</td>
</tr>
<tr>
<td>3.2.1</td>
<td>3D Simulation Source</td>
<td>72</td>
</tr>
<tr>
<td>3.2.2</td>
<td>3D Scan Source</td>
<td>72</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Alignment</td>
<td>73</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Matching Algorithm</td>
<td>73</td>
</tr>
<tr>
<td>3.2.5</td>
<td>Simplifying Matching</td>
<td>74</td>
</tr>
<tr>
<td>3.2.6</td>
<td>Object Isolation Design</td>
<td>76</td>
</tr>
<tr>
<td>3.2.7</td>
<td>Size Measurements</td>
<td>78</td>
</tr>
<tr>
<td>3.2.8</td>
<td>Tracking</td>
<td>78</td>
</tr>
<tr>
<td>3.3</td>
<td>Task Analysis</td>
<td>78</td>
</tr>
<tr>
<td>3.4</td>
<td>Path Planning</td>
<td>79</td>
</tr>
<tr>
<td>3.5</td>
<td>Simulation Scan Comparison (SSC) Process</td>
<td>80</td>
</tr>
<tr>
<td>3.5.1</td>
<td>3D Simulation</td>
<td>83</td>
</tr>
<tr>
<td>3.5.2</td>
<td>3D Scan</td>
<td>84</td>
</tr>
<tr>
<td>3.5.3</td>
<td>Comparison Algorithm</td>
<td>87</td>
</tr>
<tr>
<td>3.5.4</td>
<td>Object Isolation</td>
<td>87</td>
</tr>
<tr>
<td>3.5.5</td>
<td>Object Measurement - Tracking</td>
<td>88</td>
</tr>
<tr>
<td>3.5.6</td>
<td>Object Tracking</td>
<td>89</td>
</tr>
<tr>
<td>3.5.7</td>
<td>Task Analysis</td>
<td>91</td>
</tr>
<tr>
<td>3.5.8</td>
<td>Path Planning</td>
<td>91</td>
</tr>
<tr>
<td>3.6</td>
<td>Methodology Summary</td>
<td>92</td>
</tr>
<tr>
<td>4</td>
<td>Experimental Design</td>
<td>94</td>
</tr>
<tr>
<td>4.1</td>
<td>Experiment Overview</td>
<td>95</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Environment Analysis Overview</td>
<td>96</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Task Analysis Overview</td>
<td>98</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Path Planning Overview</td>
<td>99</td>
</tr>
<tr>
<td>4.1.4</td>
<td>Complete System Testing</td>
<td>100</td>
</tr>
<tr>
<td>4.1.5</td>
<td>Size Measurements</td>
<td>100</td>
</tr>
<tr>
<td>4.1.6</td>
<td>Tracking</td>
<td>101</td>
</tr>
<tr>
<td>4.2</td>
<td>Software Design</td>
<td>102</td>
</tr>
<tr>
<td>4.2.1</td>
<td>SSC Implementation</td>
<td>104</td>
</tr>
<tr>
<td>4.2.2</td>
<td>User Interface</td>
<td>104</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Robot Studio API</td>
<td>104</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Robot Studio PC SDK</td>
<td>105</td>
</tr>
<tr>
<td>4.2.5</td>
<td>RAPID and Robot Controller</td>
<td>105</td>
</tr>
<tr>
<td>4.2.6</td>
<td>Kinect Scanner</td>
<td>105</td>
</tr>
<tr>
<td>4.2.7</td>
<td>Software Operation</td>
<td>106</td>
</tr>
<tr>
<td>4.2.8</td>
<td>Simulation</td>
<td>108</td>
</tr>
<tr>
<td>4.2.9</td>
<td>Scanner</td>
<td>108</td>
</tr>
<tr>
<td>4.2.10</td>
<td>Alignment</td>
<td>109</td>
</tr>
<tr>
<td>4.3</td>
<td>Environment Analysis Experiments</td>
<td>110</td>
</tr>
</tbody>
</table>
List of Figures

Figure 2.1: Diagram of the operation of potential fields ................................................. 34
Figure 2.2: 3D Occupancy Grid results [118] ................................................................. 40
Figure 3.1: Contributions ................................................................................................. 66
Figure 3.2: Matching metrics ......................................................................................... 69
Figure 3.3: Object isolation ............................................................................................ 70
Figure 3.4: Normal comparison diagram ........................................................................ 71
Figure 3.5: Object isolation region growing ..................................................................... 76
Figure 3.6: Region Growing Algorithm Flowchart ....................................................... 77
Figure 3.7: SSC process design ...................................................................................... 81
Figure 3.7: SSC algorithm pseudo code ......................................................................... 82
Figure 3.8 Nearest neighbour algorithm ....................................................................... 85
Figure 3.9 Pre-filtered model ......................................................................................... 86
Figure 3.10 Post filtered model ...................................................................................... 86
Figure 3.11: Expected tracking capabilities of position based tracking ......................... 90
Figure 4.1: A diagram of the robot work-cell with scanner used for testing ................. 94
Figure 4.2 The physical environment ............................................................................ 95
Figure 4.3: System implementation ................................................................................. 103
Figure 4.4: RAPID Pseudo-code implemented on the IRC5 Robot Controller .......... 106
Figure 4.5: A screenshot with a simulation in the left and SSC results in the right ....... 111
Figure 4.6: Camera accuracy spreadsheet ..................................................................... 118
Figure 5.1: Matching time vs cell size graph ................................................................. 126
Figure 5.2: Simulated objects matched to 3D scan ....................................................... 126
Figure 5.3: Matching analysis of 3D scan with no unknown objects present .............. 129
Figure 5.4: A missing known object highlighted by the system .................................... 134
Figure 5.5: SSC analysis of a scan with one unknown object top view ......................... 136
Figure 5.6: Top view with one unknown object textured and highlighted ..................... 137
Figure 5.7: Side view of SSC analysis with one unknown object .................................. 137
Figure 5.8: SSC analysis with unknown object textured and highlighted ..................... 138
Figure 5.9 SSC analysis of a scan with multiple objects ............................................ 139
Figure 5.10: Multiple unknown objects with bounding boxes .........................139
Figure 5.11: Unknown objects: one a hazard, the other clear of the movement ....142
Figure 5.12: Occupancy Grid with only scan data ...........................................145
Figure 5.13: Occupancy Grid with only simulation data ..................................146
Figure 5.14: Occupancy Grid with both simulation and scan data ....................146
Figure 5.15: Path Planning results .................................................................147
Figure 5.16: A graph of the object size measurement reliability .......................152
Figure 5.17: Object separation pre-test ..........................................................154
Figure 5.18: Object separation partial separation .............................................154
Figure 5.19: Object separation successful separation ......................................155
Figure 5.20: Successful tracking using position based tracking system .............156
Figure 5.21: Unsuccessful tracking using position based tracking system ..........157
Figure 5.22: Position based tracking results .....................................................158
Figure 5.23: Velocity based tracking ...............................................................159
Figure 5.24: Optimised tracking results for single objects ..............................159
Figure 5.25: Moving object failed to track using position based tracking ...........160
Figure 5.26: Successful tracking of a moving object using velocity based tracking ....161
Figure 5.27: Moving object with optimised tracking ..........................................161
Figure 5.28: Unsuccessful tracking of a bouncing object with position tracking ....162
Figure 5.29: Successful tracking of a bouncing object with optimised tracking ....162
Figure 5.30: Multiple unknown objects using the position tracking ..................163
Figure 6.1: Incorrect objects ........................................................................169
Figure 6.2: Scan clipping ...............................................................................172
Figure 6.3: Object noise ..............................................................................177
Figure 6.4: Incorrect orientation .................................................................178
Figure 6.5: Maximum bound error ..............................................................179
Figure 6.6: Limitation of Task Analysis with unknown objects due to sensor shadows 181
List of Tables

Table 4.1: A sample of object resolution tests ................................................................. 119
Table 5.1: Results of matching times with varying occupancy grid cell size ................. 125
Table 5.2: Results of SSC process matching ........................................................................ 127
Table 5.3: Errors in SSC process matching ......................................................................... 128
Table 5.4: Scan matching results ......................................................................................... 130
Table 5.5: Object analysis results ......................................................................................... 132
Table 5.6: Missing object matching results .......................................................................... 135
Table 5.7: Missing object results ......................................................................................... 135
Table 5.8: Unknown object scan matching results ............................................................... 140
Table 5.9: Unknown object detection results ........................................................................ 140
Table 5.10: Collision detection results table ....................................................................... 141
Table 5.11: Occupancy grid cells with different data sources ............................................ 143
Table 5.12: Occupancy grid cell content with multiple data sources ................................. 144
Table 5.13: Overall system testing results .......................................................................... 148
Table 5.14: Overall system testing times ............................................................................. 149
Table 5.15: Scan resolutions ................................................................................................. 150
Table 5.16: Scan resolution tests measuring the size ratio (Measured / Real) ................. 151
Table 5.17: Standard deviations of real to measured object size values ............................. 151
Table 5.18: A sample of object separation tests ................................................................... 153
Table 6.1: Results of the average overall processing times .................................................. 184
1 Introduction

This thesis documents a Masters of Engineering research program conducted at Swinburne University of Technology, Hawthorn campus, Melbourne, Australia, between the years 2009 and 2014.

In order for robotics to be used in automated production facilities, the manufacturing systems must be highly constrained. Objects located in unexpected positions or orientations, additional objects in the area or missing objects can cause a robot to perform the designated operations incorrectly leading to wastage and loss of production. Typically, robotic systems have no information about the real environment around them and operate blind. When robots operate blind, a minor difference between the designed simulation and a real world scenario can cause a robotic system to fail. In order to provide more information to the robot about its environment, information such as 3D scans can be provided to the system. As such, one objective of this research is to integrate 3D scanning into a manufacturing environment to detect potential situations that could cause a failure in the process. Failures in manufacturing systems can be due to components, processes or external events [1, 2]:

- Component failures are often due to the component being in the wrong position, missing objects, or an unknown object being unexpectedly introduced into the environment.

- An example of a process failure would be a part slipping in the grip of a robot while being moved in a pick and place operation.

- An example of an external event causing a failure would be a power outage that
1 Introduction

stops a manufacturing system - when the system is restarted it may be in an unexpected state.

These failures can often result in:

- damage to robotic systems
- damage to products
- reduced production output
- if people are involved in the system an unexpected movement of a robot may cause an injury

The typical industry approach to avoiding unknown events is to restrict access to the environment, to avoid having people working in the same area as machines and to manually examine the environment to determine if any unexpected situations exist. General maintenance practices are often helpful such as cleaning up tools and loose objects. Unfortunately manual observation by humans is often unreliable due to worker fatigue and general maintenance is often only possible when the manufacturing system is off-line.

To address these types of problems, previous literature [3, 4] has stated that improving the performance of manufacturing systems requires that they be made more intelligent. In response, the Simulation Scan Comparison (SSC) approach has been developed to allow a manufacturing process to be monitored and modified as required. The SSC process has been designed based on the scientific method of Hypothesise (Simulate), Observe (Scan) and Test (Compare) [4]. The first component of the process is to consider a hypothesis, which in the case of manufacturing systems is the design simulation. For any level of automation
1 Introduction

(manual, hybrid or fully automated), manufacturing system designers put significant effort into the design of both the products and the processes of the manufacturing system. CAD software is typically used to create a simulation of what is supposed to occur in a manufacturing process [5]. While these simulations are useful in avoiding many problems, the design tools cannot account for unknown events while in operation. As such, the second stage of the SSC process is to “Observe” the environment to determine the state of the environment. This is performed using a 3D scan of the environment, while the manufacturing process is in operation. The final stage is to “Test” the two inputs and determine any discrepancies by comparing the simulation and scan. If any discrepancies are found the manufacturing process can be stopped or modified to take into account the discrepancy.

The SSC process enables manufacturing systems to become more aware of their environment and potentially take corrective action when problems are encountered. The research outcomes will enable manufacturing processes to operate in less controlled environments.

1.1 Background

In recent years there has been a trend for the manufacturing industry to increase the number of types of products at reduced volumes, thus requiring more flexible work practices [3]. Fully autonomous manufacturing systems often have a higher production capacity than manual methods, but the environment and tasks performed must be highly constrained [3]. Also redesigning autonomous manufacturing systems for more products takes considerable design effort. This has often led to development of hybrid
1 Introduction

manufacturing systems [6–8] which operate in close proximity to humans who can perform tasks more flexibly than autonomous manufacturing systems. As such, human interaction has been a significant area of research to coordinate the operation of man and machine and also to improve the safety of these practices. The machines used in hybrid manufacturing systems must be able to take into account the potential for collisions with humans otherwise failures in the production process can occur. Failures in production systems can result in damage to a production system and in turn reduced productivity. When humans are involved with the manufacturing system, failures in safety result in injuries to operators [9 – 13]. These occurrences can be caused by unexpected events, such as power outages, or unexpected objects in a manufacturing environment, which result in unexpected conditions in the manufacturing systems environment. Unexpected conditions are those that have not been considered as part of the design simulation.

The Simulation Scan Comparison (SSC) process was developed to assess the environment, to determine if any unexpected conditions exist in the scene and make a decision to continue a movement task or take corrective action, by performing path planning around an obstruction. The SSC process combines three areas of analysis, which are based on the use of both simulation and scan data. The three components are Environmental Analysis, Task Analysis and Path Planning. In developing the SSC approach a wide variety of research topics were examined relating to these three areas.

1.1.1 Environmental Analysis Techniques

Environmental Analysis refers to subdividing or analysing scan data of a physical
1 Introduction

environment to determine what is in the environment. In previous research these concepts have been used with path planning [14 – 24] in identifying collision free paths through an unknown environment. More recently, the concept of subdividing and analysing environmental scan data has been used for object recognition purposes [9, 25 – 33] to identify specific objects in an environment or mapping, which attempts to classify large areas of scan data into different types of terrain or structures such as buildings [30, 34 – 39]. The techniques used in these applications have often been based on mesh subdivision, such as occupancy grids or oc-tree subdivision. Alternatively pose estimation have been used to identify "objects", specifically humans [9, 40 – 44], from scan data of an environment.

In recent years the field of 3D scanning has rapidly evolved, due to the availability of several types of high speed 3D scanning devices [41, 45 – 54]. These devices are commonly referred to as “Depth Cameras”. These cameras have used a variety of techniques to generate a fixed array of depth pixels including Flash LIDAR, Phase modulation and structured lighting. These devices can operate at up to 30 Hz with scan resolutions typically from 10 mm or more.

There are several alternative strategies for Environmental Analysis which have different capabilities including mesh subdivision and object recognition or pose estimation strategies. Earlier research into safety systems [19, 54, 55] is based on cameras or stereo vision, as low cost 3D scanners were not available at the time. However, with single cameras it is difficult to compare the images with a 3D simulation. Hence background subtraction was image based if used at all. Using stereo vision, or multiple cameras [19, 54, 55] it is possible to generate 3D data of an environment, but again this
1 Introduction

is a slow process that requires significant computing power, to the point that attempting to compare any scan data with a simulation was not feasible. One of the other main approaches in the field of safety systems is to consider specific objects directly, typically humans [9 – 11, 13, 56, 57]. These strategies often use pose estimation to determine the position and orientation of a human in an environment. Once the pose of a human is detected in the scene, it can be considered an obstacle and appropriate path planning can be used to avoid any potential collisions. While this method works well with humans, it neglects the manufacturing process. Many other objects may be introduced into a scene that will not be detected as human, but still have the potential to cause significant problems with a manufacturing task.

Literature on bin picking [25 – 27, 58] has demonstrated the use of Environmental Analysis techniques to detect known objects in unknown areas of an environment. A technique similar to pose estimation is often used to match a template 3D object to a point-cloud. However, this approach is limited to the use of known objects in a fixed area and cannot be used as a general purpose tool over an entire environment to detect unknown objects.

Others have attempted to subdivide meshes to identify objects [30, 34 – 39], often for mapping an environment. These types of approaches attempt to subdivide a 3D scan and attempt to classify 3D scan data into particular objects. While this approach can operate in unknown environments, it is often difficult to separate known objects using expected structure in an environment. The approaches that subdivide scans can determine what areas of the scan are supposed to be (such as roads, building or vegetation), but they cannot directly compare how well these areas match with a
1 Introduction

particular object, as there is no simulation data for matching.

The SSC approach developed in this research matches existing known objects to a scene to eliminate them from the scan. This allows unknown objects to be isolated from the remaining areas of the scan. The SSC process is capable of identifying three categories of objects: known simulation objects, missing simulation objects and unknown objects in the real environment. None of the alternative strategies reviewed in the literature have discussed these capabilities. Many of the alternative techniques for Environmental Analysis are not relevant to manufacturing and do not consider the use of simulations. Furthermore, the use of the occupancy grid technique, for both simulation scan comparison and path planning purposes, has not been discussed in previous literature. Given the alternatives for analysing environments, the SSC approach was developed based on background subtraction, using a 3D simulation from 3D scan data. This provides a more robust approach to identifying and isolating unknown objects, in particular.

Literature dealing with the idea of object isolation is rare. The main technique, considered previously, has been to use plane fitting to eliminate background objects. However, this can only be used in conjunction with situations where the background is a flat object such as a table [59].

While 3D scanning is a useful method of finding out what is happening in an environment, it cannot define what is supposed to happen in an environment.

1.1.2 Task Analysis in Industry

Part of the motivation for the use of both simulation and scan data is the context i.e.
1 Introduction

manufacturing environments, as opposed to completely unknown environments. While mobile robots often operate in completely unknown environments, [17, 60 – 65], manipulators and other industrial robots operate in manufacturing environments [25 – 28, 66] which are typically much more structured and data is available from design models. The typical task of mobile robots is to traverse and often map an environment. On occasion, mobile robots are required to make simple interactions with objects i.e. pick up an object or use a sensor on an object [23, 24, 67], but they have not yet been subject to the complexity of interactions required by manufacturing systems. In manufacturing systems, robots are often expected to perform complex assembly tasks [68], work with various tools, handle various types of objects of different sizes and weights, perform movements with high precision and interact with other robots or humans [9, 42, 44].

To define what should happen for a particular task in manufacturing systems 3D simulations are often used [5]. Simulation technology has also significantly developed over the last decade with the improvement in computing power. It is now often routine in manufacturing to develop a simulation of a product to be manufactured along with a simulation of how the product is going to be manufactured.

The SSC process has been developed to determine if a movement task can be completed. To accomplish this, the simulation of the movement is compared with scan data of the current state of the environment. If an object interferes with a movement task, it is possible to detect the interference and reroute the task.
1 Introduction

1.1.3 Path Planning Applications

In the SSC process, techniques such as occupancy grids are used in combination with the Environmental Analysis and Path Planning components. Data from both the initial simulation and scans of the environment are mapped into the occupancy grid and the cells are analysed to determine if they are obstructions or traversable free space. Previous research on path planning has used both simulation and scans to tests algorithms. Very rarely have they attempted to use both the simulation (prior knowledge of an environment) and sensor data at the same time. Early research on path planning [69 – 75], was limited by memory constraints, computing power and the sensors required to scan an environment. Literature describing the development of mobile robots has been primarily concerned with mapping unknown environments in 2D [69, 75] and often noted the increased memory requirements of 3D modelling [73, 76]. As such, most of the previous research in these areas has purely considered sensor data for mapping purposes that can then be used for path planning, but the idea of incorporating prior knowledge has been explored only in a limited sense. Research into sensor fusion, [16, 73, 76, 77] has described combining of data from different sources to assist in obtaining a detailed map of the environment, but rarely have researchers demonstrated the use of both prior knowledge (simulation) and sensor data. The SSC process essentially uses sensor fusion concepts to enable the simulation to be considered simply as another sensor input. The SSC process maps the simulation data into the grid as what is expected in the scene in a similar way to any other sensor data. The advantage of this is that when the front of an object matches with scan data it is a reasonable assumption that the back of the object will also exist. In this way parts
of objects that are not directly observed by sensors can be assumed to exist based on the fact that objects are usually solid.

Researchers have used potential field based path planners [57, 78, 79] to control manipulators (as opposed to graph methods) as sending data on the position of obstructions to control the robot is simpler than analysing data for an occupancy grid. Many of these researchers have focused on detecting and avoiding collisions with humans. In this case the potential field approach to path planning is reasonable, as there are likely only a few humans to consider. Often researchers only consider a single human when interacting with robots. Using the SSC process to identify objects may result in many more object detections than human based approaches. Also, objects smaller than humans can be detected and taken into account during the operation of a manufacturing process. Comparing any object in relation to the simulated path of the robot allows particular object detections to be identified as hazards. Using a potential field approach each of these obstructions would cause the planned path to change. With the SSC approach only those obstacles that interfere with the robot need to be dealt with.

Path planning of manipulators in static environments is considered by many researchers [80 – 85] complex and time consuming. Typically these strategies analyse an environment to find potential paths through an environment based on existing structures. This is more difficult if objects in the environment change or move [19]. It is often unrealistic to attempt to recalculate a probabilistic roadmap [19, 75, 85 – 87] through an environment in real-time. As the SSC process uses an occupancy grid as part of the matching process, scan data of an environment is also mapped into the grid
1 Introduction

which allows changes in the environment to be detected rapidly.

1.1.4 Summary

This research demonstrates that the combination of 3D scanning and 3D simulation technologies can be used to enable manufacturing systems to react to changing conditions and potentially address errors in a manufacturing system as the process is running. The Simulation Scan Comparison (SSC) process has been developed to perform Environmental Analysis, Task Analysis and Path Planning for movement tasks in a manufacturing environment.

1.2 Overview of Proposed Methodology

The focus of this research is the development of the Simulation Scan Comparison (SSC) process which is aimed at use in manufacturing environments. The SSC process has been developed using three components which perform Environmental Analysis, Task Analysis and if necessary Path Planning. The main inputs to the Environmental Analysis component are a 3D simulation of the manufacturing process and a 3D scan of a physical environment. The two datasets are compared to each other using a matching algorithm. When the scanning device and the simulation have been correctly aligned in the same coordinate system, the surfaces of both the simulation objects and the scan will match. The areas that do match can be subtracted from the scan.

The areas of the scan that are not matched to the simulation can then be grouped together using a region growing algorithm into isolated objects. The current grouping criteria uses triangles as each triangle has known neighbours and a fast algorithm can be
created to group the scan surface into unknown objects. At the same time as the unknown objects are isolated, the simulation objects are matched with areas of the scan which confirms if their position is correct. If the scan is in conflict with the simulation object it is possible to determine that the simulation object is missing from the real environment.

The current implementation utilises an occupancy grid structure to assist with reducing the number of potential matches by considering the scan triangles and simulation triangles that are allocated to each cell of the occupancy grid. The grid can also be used later in the process for path planning.

The Task Analysis uses the simulated movements of any controlled object, such as robotic arms, to determine if any unknown objects interfere with them. A “movement hull” can be created from the beginning and end points of the movement and the component objects of the simulated robot. Once the movement hulls have been created they can be compared to any unknown objects detected using standard collision detection algorithms such as the Gilbert–Johnson–Keerthi distance (GJK) algorithm [88].

The Path Planning component is required only if the Task Analysis determines that an object is interfering with a movement. In these situations a potential solution is to either: stop the movement and wait for the obstruction to be removed, or plan a path around the obstruction.

1.2.1 3D Simulation

The required components of SSC include a simulation which can be developed using
1 Introduction

Computer Aided Design software (CAD). The simulation represents what is supposed to occur in the manufacturing process. Robot Studio (ABB, Zurich, Switzerland) is an industrial standard software package for developing manufacturing simulations and can be used to operate a physical robot [5]. The SSC process has been implemented in software and integrated with Robot Studio. This approach allows a simulation to be created for a manufacturing task with a physical robot including an expected path and directly used with the SSC process. While the simulation takes into account what should happen in the environment, to detect any unexpected conditions 3D scanning is required.

1.2.2 3D Scanning

Several types of scanning have been used for testing the SSC process including stereo vision and the Microsoft Kinect™. Stereo vision has been used for developing the SSC algorithm. Stereo vision typically uses Delaunay triangulation [89, 158], which creates a complete surface over the entire image area. An effect of Delauney triangulation is that the scan will contain many triangles that connect an object to the surface behind it. These triangles do not always represent a real surface and therefore they are “unreliable” and need to be removed. There are several filtering methods that can be performed, but larger triangles, especially near the edge of the triangulation where triangles on objects connect to background objects are more likely to be incorrect [158]. As a result, the simplest and most effective method is to filter out triangles based on their size.

The Microsoft Kinect™ uses an infra-red structured light system to measure a fixed
array of points in depth. This can also be used to create a 3D triangulation of an environment that can be matched to the simulation. As the point array from the Kinect (TM) is fixed, a fixed set of triangles can be used to describe the surface without the need for Delauney triangulation, but with the same properties. Irrespective of the technology used to scan a work area the required data is the same; a set of 3D points and an associated set of triangles. This is used to compare with the simulation data to determine if any objects in the simulation are not correctly positioned. Any unknown objects will only be detected if they are significantly larger than the scan resolution, so that a reasonable estimate of the size of an unknown object can be made. For this application the accuracies and resolutions are aimed at detecting objects ranging in size from 60 mm up to 400 mm with a work area of up to 2 meters.

1.2.3 Environmental Analysis Implementation

The Environmental Analysis takes both the simulation and the scan and compares the two to determine any differences between the physical environment described by the scan and the simulation. The matching process subtracts the simulation from the scan and analyses both the simulation objects and scan to detect unexpected conditions. The implementation of the SSC developed for this research uses a triangle based surface description as opposed to a point based description. This is the preferred method as the standard method of displaying textures in three dimensions is to draw the surface using triangles. Also the direction the surface is facing is required so that it can be matched to the correct object. A brute force approach to the matching produces a $O(n^2)$ complexity
1 Introduction

algorithm. To improve the performance of the matching process an occupancy grid is used which allows the complexity of the matching process to be reduced to a linear \(O(n)\) complexity.

When a scan and model triangle are matched in position, i.e. exist in the same cell, the normal of each triangle can be compared. The surface normals are described by a unit vector orthonormal to the surface of the objects or the scan. To determine if the surface normals are pointing in the same direction, a dot product between surface normals should be close to a value of 1.0. If the dot product produces a value close to -1.0 the surface normals are facing the opposite direction. The surface matching used for the SSC was based on simple geometric projections from scan to simulation triangles. This technique was preferred as it uses the same data from simulation software and is quick to implement. The surface matching also makes use of the data structures specified previously. The use of surface normals in the matching process assists in determining if an object is missing from the environment. The use of neighbouring triangles allows the use of region growing algorithms to group unmatched triangles into unknown objects.

The surface matching allows background subtraction to be performed reliably using a 3D simulation. The advantage of using a simulation is that the context of an object can be determined from relationships to the scan and simulation. Known simulation objects match correctly with the scan. Missing objects appear in the simulation but not in the scan. Finally unknown objects appear in the scan but not in the simulation. No other literature reviewed has considered the concept of missing objects. When a simulation object is in conflict with the scan, i.e. simulation and scan have the same position but
Introduction

the surface normals are in opposite directions, it indicates that the simulation object is missing from the environment.

1.2.3.1 Object Isolation

Part of the matching process allows unknown objects to be isolated from scan data. The isolation process identifies unknown objects from the areas of a scan not matched to the simulation. A region growing algorithm is used to group areas of the scan together to form an object. Once the areas not matched are isolated from the rest of the scan, a convex hull is created of the object and an oriented bounding box is then fitted to the hull which allows the objects to be measured in three dimensions.

As objects are isolated from background clutter easily, they can be tracked by repeated observations over multiple data frames. This feature can be used to track the movement of unknown objects in a known environment. Several methods of tracking were developed using the Gilbert–Johnson–Keerthi (GJK) collision detection algorithm [88] to track objects if they move slowly.

1.2.4 Task Analysis for Movement tasks

As mentioned in the background there are many types of tasks in manufacturing. Each of these tasks such as cutting, welding, or assembly tasks have specific requirements that will be different between tasks. The methodology for Task Analysis in this research has focused on task analysis for movement tasks as movement tasks can be examined reliably with the scanner used (a Microsoft Kinect™). The Task Analysis determines if a movement task has any interference from unknown objects. A “movement hull” is created from the beginning and end points of the movement and the simulation of the
1 Introduction

Robot components using a standard convex hull generation algorithm. This technique also demonstrates the use of simulation and scan comparison as the robot movement is purely simulation which is compared to scan data of the physical environment. Once the movement hulls have been created they can be compared to any unknown objects detected using standard collision detection algorithms such as the Gilbert–Johnson–Keerthi distance (GJK) algorithm [88].

1.2.5 Path Planning Implementation

If an unknown object is detected inside a movement hull, path planning can be used to reroute the movement, without disrupting the manufacturing process. Some processes, such as welding or gluing, may require continuous contact with an object, hence they will not be able to be rerouted. Alternatively, processes such as part transfers should be able to be rerouted as they do not require a fixed path. As the occupancy grid is used for surface matching it can also be used for the Path Planning stage. There are several alternatives for path planning including potential fields and various other graph based methods. The occupancy grid was chosen as it has been used previously with the matching process and it is relatively simple to use an A* search [21]. The other methods mentioned may still work just as well but require more computational effort to achieve all of the required capabilities used with the SSC. To examine the path planning, a comparison of the occupancy grid cells populated with different datasets was made. Three datasets were created based on combinations of simulation and scan data. Specifically simulation data only, scan data only and both simulation and scan data. The occupancy grid was populated with each dataset and the resulting cells
1 Introduction

compared. This demonstrated the ability for each dataset to distinguish between cells occupied by simulation objects, the robot and unknown objects. This also demonstrated that the combination of simulation and scan data can take into account any gaps in scan data.

1.2.6 SSC process

Finally, the three components of Environmental Analysis, Task Analysis and Path Planning have been combined into a single logical process. This is referred to as the Simulation Scan Comparison (SSC) process. The SSC process is designed to examine an environment with a robot performing a task, detect potential interference and reroute a task if necessary. This approach is based on the use of an occupancy grid, which is used to map the environment, simplify the comparison process between simulation and scan and allow path planning with dynamic obstacles.

1.3 Overview of Experimental Procedures

The intention of the experiments outlined here is to firstly validate the theory and basic techniques of a system using the SSC process. The second purpose of the experiments is to validate the SSC algorithms in context with a live manufacturing system. The SSC process is based around the use of a simulation for background subtraction. An occupancy grid is used in this process to improve the performance of the matching process. To demonstrate the performance of the occupancy grid, it was measured using different cell sizes. The performance of resulting cell sizes was compared to determine what cell size was most efficient for a simulation with 7300 triangles and another
1 Introduction

containing 23261 triangles. Experiments were also performed to evaluate the resulting capabilities of the Environmental Analysis process. The three main advantages of using a simulation for background subtraction are:

1. Known object detection
2. Missing objects detection
3. Unknown object isolation

To test the known object detection feature a physical environment was created with a robot, a safety cage and boxes as test objects. A 3D simulation of this environment was also created and 3D scans of the environment were taken to compare to the simulation. Objects were added or removed from the environment as required.

To test the known object detection and validate the matching process the environment was set up with the same objects present in the simulation. To test missing object detection, objects were added to the simulation but not to the physical environment. To test the unknown object isolation, objects were added to the physical environment but not to the simulation.

Unknown object isolation also presents several other advantages. Firstly, the unknown objects can be measured to determine their physical dimensions. Experiments were performed to test the reliability of the unknown object measurements. Secondly, to test the reliability of the object measurements over time, object tracking capabilities were implemented and tested.

1.3.1 Measurement Reliability

The measurement reliability experiments focus on the measurement of isolated
1 Introduction

unknown objects using the SSC process. In previous experiments, it has been demonstrated that the SSC process allows objects unknown to the simulation to be identified, isolated from the rest of the scan and size measurements taken within a reasonable time-scale for small scans. This research demonstrates the accuracy and reliability of the unknown object isolation feature of the SSC process and defines the limitations of the measurements with respect to the “scanner's” resolution and accuracy.

The second feature is the use of triangle filtering as the system is intended to use only a single scanner. The simplest filtering method used is to compare the scan triangles with the “filter” size which is based on the scan resolution.

The results of the experiments showed that with fixed unknown object sizes compared to the filter size, the average value of measurements remains relatively stable, while the standard deviation of object sizes significantly increases. The standard deviation of the measurements appears to be closely related to the filter size. From this result it can be concluded that the standard deviation of the measurement ratio is a good estimate for the reliability of the measurements.

The second finding relates to the ability to differentiate between two objects with a small separation. The ability to make this distinction has been determined to be closely related to the comparative size of the separation and the filter size.

1.3.2 Tracking

Using the SSC algorithms, unknown objects can easily be isolated from a 3D scan of an environment. These objects can be tracked relatively easily using collision detection algorithms. The reliability of the tracking was tested using a series of experiments that
demonstrate simple situations where one or more objects can be tracked by the SSC process using a single scanner. The situations are then progressively increased in complexity to determine the limitations of different tracking techniques.

The first technique (position based tracking) is based on the idea that inanimate objects don't move of their own accord. The position algorithm operates by examining the differences between two frames of the 3D scan then determining if two unknown object detections occupy the same volume in space. The second technique (velocity based tracking) is based on estimating the velocity of the object then attempting to determine where the object should be in the next frame.

The tracking methods were tested experimentally. The first set of experiments attempts to determine how fast an object could move before the object fails to be tracked using the position based tracking technique. In theory this should be based upon the size of the object and the frame rate. If the displacement between frames is smaller than the object’s physical size the object will be successfully tracked. If the object is completely stable and not occluded it is expected that any object will be successfully tracked. The minimum size of the object that can be detected was established by the object measurement reliability experiments. The minimum size was determined to be three times the resolution of the scan (R). As a result objects smaller than three scan resolutions are not detected. Also, to be successfully tracked, the displacement between frames needs to be less than the size of the object.

More complicated experiments, where unknown objects were moving more naturally, such as swinging or bouncing, in an environment, demonstrate the capabilities of the different tracking methods in situations that would occur in a real environment. The
1 Introduction

swinging test shows the performance of the tracking algorithms under more natural conditions. While this situation may not often occur in manufacturing, it is useful to see how the tracking algorithms handle uncontrolled motion tracking. The bounce tests may occur in a manufacturing environment, for example: if an object falls off a conveyor belt. This test also shows much higher variations in position and velocity than either of the previous tests. The last test is to determine if the different tracking algorithms can cope with multiple unknown objects moving in the same area.

The tracking experiments demonstrate the value of the SSC technique for tracking unknown objects and the limitations of the SSC algorithms. These experiments were used to determine if position based tracking using collision detection was a viable method for object tracking. The experiments also show the limits of the position and velocity based tracking methods in terms of object movement.

1.3.3 Movement Task Analysis Experiments

Task Analysis experiments were performed to demonstrate the use of movement hulls and collision detection algorithms to determine if an unknown object was interfering with the movement task of a robot manipulator. The experiments were performed by placing an unknown object in the path of a simple repeating movement task and analysing the scene. When interference was detected, the Task Analysis could activate a Path Planner to find a new path around an interfering object.

1.3.4 Path Planning Experiments

The Path Planning module was tested using two sets of experiments. In the first, the
1 Introduction

general capabilities of the Path Planning module were examined using three sets of data: scan only data, simulation only data and combined simulation and scan data. The resulting cells were compared, demonstrating the ability for each of the three datasets to distinguish between cells occupied by simulation objects, the robot and unknown objects.

The second set of Path Planning experiments were used to demonstrate that when the task had interference a new path could be generated around a dynamic obstacle. The Path Planner used an A* search to create a path around an obstacle during the operation of the task.

1.3.5 SSC process testing

The SSC process experiments demonstrate the operation of the complete system integrated with a physical robot operating in a simple manufacturing cell environment. Timing data is presented on the entire SSC process and the external components required to integrate the physical robot into the control system. The simulations for this environment were developed using Robot Studio. This allowed a CAD model of a robot work-cell to be developed and compared to the live scan from a Microsoft Kinect (TM) scanner for real-time operation.

The entire system was expected to run a robot through a simple movement task while scanning the environment using the Kinect (TM) 3D scanner. If an unknown object was moved into the path of the robot, the system was expected to detect the unknown object, map it into the occupancy grid and reroute the robot around the object.
1 Introduction

1.4 Perceived Contributions

The SSC process is designed to determine if a manufacturing process can be completed successfully given a scan of the physical environment. The approach chosen requires the environment to be both simulated and scanned. The two datasets are then matched allowing the simulation to be verified and any unknown objects detected. The goal of this research is to produce a robust system that is capable of determining if a simulated manufacturing process can be completed without errors. The simulation scan comparison (SSC) algorithms were developed for this research. This research demonstrates the following concepts related to process monitoring in a manufacturing environment:

• Using a 3D simulation for background subtraction to interpret the scan
• The use of collision detection between simulation and scanning for obstacle detection
• The combined use of simulation and scan data for path planning using an occupancy grid

While these three processes are individual contributions, the concept of combining all three contributions into a single process is also considered a contribution.

As a result of these processes the combined system developed also feature the following capabilities:

• 3D scan background subtraction using a 3D simulation.
• Detecting missing objects based on conflicting normals with scan data.
• Detecting unknown objects in a controlled environment using a simulation for
1 Introduction

background subtraction.

- Experiments have been performed validating the size measurement of unknown objects and determining a reasonable lower limit for unknown objects based on the scan resolution.
- Experiments have been performed validating the position of unknown objects over time, by developing tracking algorithms for unknown objects.
- Task Analysis, by examining an existing path using a collision detection algorithm between the expected path movement hull and any unknown objects detected in the scene.
- Path Planning using an occupancy grid generated from both simulation and scan data.
- A complete system, integrating the SSC process with industrial simulation software and a physical robot.

1.4.1 Simulation Background subtraction

The Environmental Analysis of the SSC process uses a 3D simulation to remove background areas from a 3D scan. This approach allows the detection of three types of objects: known simulation objects, missing simulation objects and unknown objects. This required developing several simulations of a test environment and creating the same set-up in a physical environmental. Software was developed to match the simulation and scan datasets and isolate any unknown objects from the 3D scan data.
1.4.2 Missing Objects

Missing objects are detected by comparing the surface between a simulation object and scan data to find areas of the scan and simulation that conflict. The scan data conflicts when the surfaces of both the simulation objects and the scan data have the same position but the normals of both surfaces are in opposite directions. Objects are determined to be missing if the majority of scan data matched to an object conflicts with the simulation surface. This typically occurs when the ground is visible in the scan data, but the surface of the object is not visible.

1.4.3 Unknown Object Detection

Detecting unknown objects in a controlled environment is performed using a simulation for background subtraction. Once the background areas of the scan have been eliminated, any areas that have not matched to a simulation object can be grouped into unknown objects.

1.4.3.1 Object Measurement validation

Experiments have been performed to validate the size measurement of unknown objects. The validation of these measurements has shown that the smallest unknown objects that can be detected are a factor of the scan resolution. The scan resolution (R) is defined as the average distance between two points in a scan.

1.4.3.2 Object tracking

Two methods of performing tracking of unknown objects have been developed and compared: the position based method and the velocity based method. Both of these
methods use collision detection between frames to track unknown objects. The position based method simply looks for overlapping objects, while the velocity based method attempts to determine an estimate of where the object will be in the next frame for the collision detection test.

1.4.4 Movement Task Analysis

Task Analysis for movement tasks, compares an existing simulated path and any unknown objects detected in the scene. Collision detection algorithms are used to determine if any unknown objects interfere with the task. This demonstrates the use of the simulation being compared with physical objects from the scan data using standard collision detection algorithms.

1.4.5 Path Planning using both Simulation and Scan Data

Experiments have been performed that highlight the difference in using only simulation data, only scan data and combined simulation and scan data to generate an occupancy grid. These experiments have shown that combined simulation and scan data can distinguish between known simulation objects, cells occupied by the robot used for path planning and any unknown objects added to the scene.

1.4.6 SSC process

The complete SSC system combines the three main contributions, Environmental Analysis, Task Analysis and Path Planning, to perform the SSC process in a physical robot environment with moving objects and performs rerouting as required. The SSC process has been demonstrated to analyse an environment, analyse a movement task and
1 Introduction

determine if it is possible to reroute a planned path around an obstruction, and perform the reroute on a real manipulator.

1.5 Structure

• Chapter 1 Introduction

The introduction includes the description of the topic of the thesis and an overview of the research carried out. The concepts of the SSC are introduced and relevant areas of research are identified.

• Chapter 2 Literature Review

The Literature Review investigates the current state of the art in areas of research, especially areas related to 3D scanning and 3D simulation techniques and technologies.

• Chapter 3 Methodology

The Methodology chapter provides a detailed overview of the design and operation of the system using the SSC technique and its resulting capabilities.

• Chapter 4 Experiments

This chapter describes the testing environments and experiments used to demonstrate the techniques developed for the SSC process. These experiments include Environmental Analysis, Task Analysis and Path Planning experiments along with testing of the complete SSC process. Experiments on the object size measurement reliability and tracking experiments are also presented.
1 Introduction

- Chapter 5 Results
The results of experiments described in chapter 4 are presented here. These results include Environmental Analysis, Task Analysis and Path Planning experiment results as well as the results of overall testing. Results on the object size measurement reliability and tracking experiments are presented.

- Chapter 6 Discussion
The results of experiments are discussed highlighting the capabilities of the SSC process and comparing the SSC process and its components to alternative approaches presented in the literature review.

- Chapter 7 Conclusions
This chapter presents the conclusions that are drawn from the research in relation to the operation and performance of the SSC process and potential future developments that could improve the performance of the SSC process.
2 Literature Review

Research into automated manufacturing systems covers a wide variety of topics from process management and stock control, to fine dimensional control of products. Autonomous manufacturing has further constraints on both the products and processes used, as these processes must be able to operate often over many thousands of cycles without an error occurring. As such, these manufacturing systems need to be carefully and robustly designed. For this purpose 3D design simulations are used extensively in manufacturing, to provide a basis for what is expected, when developing both products and the manufacturing systems themselves. This has proven to be an effective tool for avoiding many problems that occur during the design of products and manufacturing systems. Despite the advances in manufacturing technologies and processes, manufacturing systems still have difficulties when dealing with anything that has not been specified in their design simulation. The problem of reacting to these unknown conditions in manufacturing is not widely addressed.

Recently low cost and effective 3D scanning technology has been developed that can be used in applications relating to manufacturing and has been used to great effect in mobile robotics. The main area of interest in manufacturing for unknown conditions has been the safe cooperative operation of humans and robots. Both safety systems for manufacturing and mobile robots use path planning and 3D sensing technologies to operate in unknown environments and react to unexpected conditions.

This chapter provides a review of the current literature relevant to path planning and collision avoidance. Of particular interest is where simulation and scanning technologies are used in combination for path planning, especially for the purpose of
2 Literature Review

reacting to unknown or unexpected conditions.

2.1 Path Planning

Path planning, often referred to as “Motion” or “Trajectory” planning, is a process that allows an object to be moved through an environment without colliding with any obstructions. Path planning allows automated robotic systems to operate in environments and interact with other objects without human assistance.

There are two main environments that have been considered in the field of path planning: mobile vehicles and manipulator robots. Mobile vehicles typically deal with fast changing and large environments whereas manipulators require more complex planning to take into account the extra degrees of freedom and changing shape of the robot.

Two main techniques can be used to determine collision free paths through an environment: potential fields and graph methods [21]. Potential fields map the obstructions in an environment and define rules to move around the objects, without colliding with them. Graph methods rely on cell decomposition of an environment, then search through the available cells for a path using a discrete planning method, as described by S. M. LaValle [21]. The discrete methods for path planning includes Dijkstra's algorithm [90] and A* (pronounced A star) [21] to determine which cells to use in the final path. Dijkstra's algorithm [90], essentially uses a best-first sorting scheme to identify the shortest or lowest cost path to a goal state. A* [21] and its variants [91 - 93] are an extension of Dijkstra's algorithm and use an estimate of the overall cost of the current path to reduce the number of possible states searched. In
many cases these types of searches can be used from both start and goal states to be combined into a bidirectional search. The more advanced methods are significantly faster as they examine fewer nodes in the search tree. This search tree is generated from analysing obstructions or free space in an environment, then linking the cells if a robot can pass between the cells without being obstructed.

In industrial environments, path planning has been applied to many different problems from vehicle or manipulator control, to assembly planning and safety systems. In industrial environments path planning is used for automated guided vehicles (AGV's) as described by Martínez-Barberá et al. [94]. Researchers dealing with industrial environments, usually have a much more controlled space to work within compared to other environments. Most of the recent research into autonomous vehicles [17, 20, 60, 62, 64, 65, 95] has been carried out in the context of large outdoors environments, requiring vehicles to autonomously navigate the environment at high speeds. But, as stated by Martínez-Barberá et al. [94], industrial environments can change unexpectedly. This requires that path planning be flexible and react to the changes in the environment and potential obstacles such as humans or other vehicles.

Path planning for manipulators is often more complicated than “navigation” as required for mobile robots, as manipulators often have more degrees of freedom and change shape as they move. Aasland [80] developed a nine degree of freedom robotic arm which requires special considerations for path planning. Configuration space is often employed for such tasks [80], but it is challenging to generate a map of more than three dimensions of configuration space. As such, this technique often takes a long time to generate a map, making it difficult to use for real-time applications and also makes it
very difficult to manage in changing environments.

Industrial applications, such as bin picking [26, 27], automated pick and place [96, 97] and safety systems have been topics of interest for path planning researchers. Path planning is used in these applications to improve the performance, increase the level of automation and improve the safety of manufacturing systems. As such, this research demonstrates a practical use of path planning to optimise the performance of a manufacturing process.

Other industrial research from Ferré et al. [68] used path planning to develop a sequence of movements to assemble a windscreen wiper assembly, using simulated environments. Park et al. [98] present a method for the optimisation of path planning to assemble printed circuit boards using optical inspections. Tan and Arai [11], and Gecks and Henrich [19, 97] have studied safety applications using multiple cameras to detect obstacles and humans near robotic systems. They have demonstrated that scan data can be used with path planning systems to reroute or stop robotic equipment. These applications demonstrate that practical path planning has been developed to the point that complex path planning can be achieved reliably in industrial environments using scan data. The techniques used in these applications include potential fields and graph based methods.

2.1.1 Potential Fields

Originally proposed by Khatib [99] and extended by others [79, 100, 101], “potential fields” is a well-developed area of research, with many researchers considering its use in navigation and path planning. As illustrated in Figure 2.1 it uses force vectors, or
2 Literature Review

“motivations”, to essentially "push" away from known obstacles and "pull" towards the final goal location. The direction of travel is determined by the sum of the “forces” away from obstacles and attracting towards the goal.

Figure 2.1: Diagram of the operation of potential fields

Koren and Borenstein [100] determined that there are four inherent limitations in the potential field approach. They are:

1. Trap situations due to local minima
2. No passage between closely spaced obstacles
3. Oscillations in the presence of obstacles
4. Oscillations in narrow passages
2 Literature Review

They proposed an alternate method called Vector Field Histogram (VFH) [72] specifically to address the limitations they discovered. Even with the modifications, it was noted that the VFH could still become trapped in cyclic behaviour.

There have been many other attempts to improve the performance and reliability of the potential field approach to take into account the limitations discovered. Ge and Cui [102] determined that by modifying the potential function of the obstacles depending on the distance to the goal, any Local minima could be eliminated. Barraquand et al. [103] also attempted to avoid being limited by local minima in the potential field. The four techniques they presented were:

1. best-first motion
2. random motion
3. valley-guided motion
4. constrained motion

Barraquand et al. [103] tested both mobile robot and manipulators in simulated environments. Yegenoglu and Stephanou [104] proposed methods that allow local minima to be avoided using escape vectors parallel to the object rather than perpendicular to them.

Cheng et al. [101] developed the virtual obstacle potential field (VOPF), which was aimed at overcoming the limitations of the potential field approach. The VOPF allowed a simulated robot to follow a linear or circular path. Minguez et al. [105] demonstrated a real mobile robot using several path planning strategies. They compared the performance of the potential field approach to a “nearness diagram” approach. Their
2 Literature Review

work demonstrated that the nearness diagram approach produced a more stable path than the potential field [105].

Point motivations, as described by Cheng et al. [101] are a fast method for dealing with large numbers of obstacles, but are not suitable for complex shapes.

The potential field approaches often use, but not exclusively, idealised data such as CAD models to simulate the environment, as opposed to real scans of an environment. As manufacturing systems require parts to be designed, CAD data is often readily available and it works well with the potential field approach. In past decades potential field approaches have often used two dimensional datasets to describe environments. Until recently, computing power has not been available to model potential functions of environments rapidly. Fagerjord [106], demonstrated a method of evaluating a potential function of complex shapes in three dimensions using Graphic Processor Unit (GPU) based calculations, compared to Central Processing Unit (CPU) based calculations.

Most researchers have used simulation data to test theories [99-104, 106]; a few have demonstrated the potential field with real robotic systems [79, 105, 107]. Demonstrations with real robots such as those by Minguez [105] and Park et al. [107] have been with mobile vehicles, but a few have used manipulators such as Park et al. [79]. Barraquand [103] demonstrated the ability to deal with large degrees of freedom and demonstrated in simulation using a manipulator with 31 degrees of freedom. Park et al. [79] investigated the navigation of a manipulator using the potential fields approach and showed that the adjustment of a potential field due to an obstacle was possible. The experiments were limited to a single fixed obstacle and did not use sensors to detect the object, but did show the variation of the path calculated in real time with and without
the object. One advantage of the potential field approach is that with a well-known environment it can be very fast to determine appropriate force vectors and hence can often be run in “real-time” [79, 105].

Potential fields can be used for off-line path planning using the gradient descent method [21, 80]. This can be calculated from the direction the sum of forces leads to as it approaches its goal. Planning a known repeatable path using this method is difficult, as the gradients change with every new object, especially in configuration space, and they take a significant time to calculate. Any new object added to the list of obstructions will completely change the nature of the field and hence will change the entire path. Using this method researchers [21, 80] have noted that the path generated is completely different even when minor changes are made to the number or position of obstacles.

Weaknesses with the potential field approach include difficulty determining simple and stable paths, difficulty determining repeatable paths and difficulty integrating real-time data from sensors.

It is difficult to integrate real-time data from sensors as objects need to be complete and separate. Sensor data typically only shows the closest faces perpendicular to the sensor which results in the potential field simply pushing back towards the sensor. It is generally difficult to separate a single object from the bulk of the sensor data without significant analysis. Conversely, to take into account unknown objects, scanning of the environment needs to occur, but few researchers have used scan data with the potential fields approach. To be more useful a sensor based path planning system requires the use of graph methods.
2.1.2 **Graph Methods**

Graph methods involve subdividing environmental data into cells, so that discrete path planning techniques can be applied to generate a minimum cost path through the cells. For these algorithms to operate they require a search tree to be generated. The search tree is typically generated synthetically from “simulated” data or collected from the real world using scanning devices. The data is then subdivided into cells that form a searchable tree structure.

The occupancy grid method [69, 76, 109, 110] subdivides the environment into fixed sized cells. Each cell is examined with respect to the environmental data to determine if it is occupied or traversable. Once the traversable cells have been identified they can be collected into a tree structure by linking cells. Other graph methods use varied sizes of the cells either by using known objects and splitting space around them as with Voronoi space [14] or combining cells that can be treated collectively into larger cells [135].

Voronoi and Delauney cell decomposition [14, 111 – 114] are useful in a continuous area with a large number of small obstructions as demonstrated by Ok et al. [14]. Voronoi decomposition determines cells by breaking down the environment either into cells with obstructions then finding paths by linking the mid-points between the edges of the cells, or by breaking down the free space into cells which can be linked if they are traversable. Delauney triangulation can be used in a similar method as the triangles created by the technique can represent either the free space or obstacles in an environment. As such, the obstruction free triangles can be considered cells and be used as the basis of the cell search.

One difficulty found with occupancy grids was the fact that the grids use a large amount
of memory. However, over the past twenty years computing power and memory have increased to the point that this is no longer a significant limitation. Researchers such as Kraetzschmar et al. [115] used quad-trees and Hornung et al. [116] used oc-trees to map large environments with minimal memory requirements. Quad-tree and oc-tree structures have also been utilised to define cells for a search tree. These structures subdivide the space hierarchically into areas of either free space or obstructions. The use of quad-trees and oc-trees is an ongoing area of research. While this technique has some benefits with large environments the difficulty with quad-trees and oc-trees is that they are more difficult to update than occupancy grids for new or rapidly changing environments[19].

Applications of these approaches have been limited by the available computing power and sensors. Only recently have sensors become available that allow 3D scanning and computing power has increased sufficiently to be able to process the volume of data the new sensors provide.

2.1.2.1 Occupancy Grids

Occupancy grids subdivide a space into regular sized cells as demonstrated in Figure 2.2. These cells are populated with data often from sensors and used to map areas that are occupied by objects. Occupancy grids are commonly used with mobile robotics as they can be easily integrated with one or more sensors. Real world measurements provide up to date information about a scene including “unknown objects”.

39
Elfes [117] developed an occupancy grid method and demonstrated the method using two real robots equipped with sonar sensors. The sonar data is mapped into the grid which is represented using a bitmap image. Elfes demonstrated that the occupancy grid could be used for several purposes, including mapping an environment, determining motion from map readings and path planning. Tsai [70] extended the work of Elfes to improve the integration of sonar data with the occupancy grid approach and extended the technique to handle dynamic objects as well as static ones.

Many mobile robotics applications [20, 23, 69, 94, 118] use 3D scanning systems to examine an unknown environment, for navigation purposes. These applications often use the occupancy grid approach to perform path planning, as this makes it easy to manage the incoming data and can easily combine data from multiple sensors, a technique often referred to as “sensor fusion”. With the occupancy grid approach, any data about the environment is mapped into a grid of regularly sized cells. Other graph
approaches may consider different sized cells to attempt to simplify the final network. The areas of the grid that do not contain data can then be considered obstruction free and used for the rest of the planning process. From the set of nodes generated, a tree searching algorithm such as Dijkstra's algorithm [119, 80], the A* search [120] or variants [22, 92, 121] can be used to determine an appropriate path. There are several variations of the Dijkstra's algorithm, but the general approach is to assign a value to each link and search for the lowest accumulative value to approach the goal. Another version is the D* algorithm [22, 92, 121] which is aimed at finding an optimum path through a real environment, taking into account any unknown areas of a searchable map.

Comparing the path generated using potential fields [79, 100 – 103, 107] with those generated by occupancy grids [69, 76, 110], it is notable that the graph methods produce more stable and repeatable paths than the potential field approaches.

The other methods of cell decomposition (such as Voronoi and Delauney) while more efficient at the discrete path planning stage, as they have fewer nodes to search, take significantly more processing to generate the cells. As such, many researchers still implement occupancy grid methods, rather than the more complicated but efficient cell decomposition methods. The occupancy grid technique has also been applied to many practical problems involving sensor fusion and mobile robotics.

2.1.3 Uncertainty in Data

One of the difficulties of dealing with real world data is that sensor data is not always 100% certain. All sensors have limitations in range and resolution which limits the area
2 Literature Review

and detail that can be observed with them. Typically the further away an object of
interest, the larger the area a sensor reading will cover, as a consequence the resulting
data is less reliable. The other difficulty with most sensors is noise; multiple sensor
readings may not produce exactly the same result. Researchers have developed
methods, referred to as “sensor fusion”, to combine these readings, often using graph
based methods, such as occupancy grids [73 76, 117] or oc-tree structures [129] that
allow sensor readings to be compared and combined. Occupancy grids can be used to
combine data from multiple sensors and appropriately weighted and combined in each
cell. Different weights can be applied to each sensor depending on the certainty of the
data.

From early research into occupancy grids, Elfes [117] demonstrated that the occupancy
grid method could be used to compensate for inaccurate sensor data using 2D image
techniques. These techniques include convolution with the sensor’s response
waveform, blurring sensor readings across cells and thresholding combined
measurements. Many researchers have been able to integrate different types of sensors,
for example Kohara et al. [76] integrated stereo vision and laser scanners to perform
collision avoidance while driving.

Martin and Moravec [73] used sensor fusion with stereo vision and ultrasound, in
combination with the occupancy grid approach, in a simulated 2D environment to map
an environment and perform path planning. This was demonstrated using a robot
equipped with stereo vision. As stereo vision produces a full 3D scan of an area the use
of 3D grids to form an environment map was considered.

One issue identified in this research was the case of a robot that repeatedly examines an
area of a map. This requires the pose of the robot to be accurate for the sensor data to line up correctly. The consideration of this problem has led researchers to develop Simultaneous Localisation And Mapping (SLAM) algorithms [122 – 127] to map environments and at the same time determine the position and orientation of the robot. SLAM is a technique that allows a robot’s path to be extracted from scan data of an environment while a global map of the environment is developed.

Mobasseri [128], discussed path planning and the use of occupancy grids with sensor data. A path planning algorithm was developed based on the A* algorithm using soft linked quad-trees to take into account unknown areas of a map. The technique was demonstrated using a simulated environment. This problem was used as the basis for the development of the D* algorithm [22, 92, 121] which was specifically aimed at path planning in partially unknown environments.

More recently, Kaelbling and Lozano-Perez [129] developed a robot that was capable of analysing an environment, identifying objects and interacting with them. The robot’s operation was developed in the context of belief space, where regions in an environment are mapped into an oc-tree structure, which is progressively refined to determine the shape of obstacles as they become more “certain”. The path planning uses a conditional plan to take into account the uncertainty in object positions and is demonstrated in both simulation and using a physical robot.

This research demonstrates that the occupancy grid and other cell decomposition strategies can be used to compensate for uncertainty in environmental data. The use of these techniques to successfully perform sensor fusion, demonstrates that it is possible to reliably perform path planning, even in unknown or partially unknown environments.
2 Literature Review

using sensor data.

2.1.3.1 Mobile robots

Path planning and sensor fusion have been successfully applied to enable mobile robots to autonomously navigate unknown environments [17, 60 – 65]. The robots are capable of scanning an unknown environment with multiple scanners including laser, stereo vision and radar. The scans are used to analyse the terrain surrounding a vehicle while it is moving at speeds of up to 120 km per hour. Path planning research has also been applied to the NASA Mars Rovers [23, 24, 67] which, while slower, operate autonomously in extremely remote and unknown environments. Researchers have also begun to have significant positive results when considering mobile robots that have integrated manipulators [25, 129, 130], but have yet to deal with the same dynamic unknown environments as manufacturing systems.

In many of these approaches the terrain is typically analysed using occupancy grid techniques. Each cell of the occupancy grid is analysed for “traverse-ability” and obstacles. Thereafter the resulting grid is examined using discrete path planning methods. The operation of mobile autonomous robots demonstrates the ability for the occupancy grid method to identify obstacles at high speed from sensor data.

2.1.4 Manipulator Path Planning

While mobile robots often need to be able to handle unknown environments at high speeds, manipulators are normally only used in areas that are well known and time is not a critical issue. Far from being easier, manipulator path planning requires the
2 Literature Review

consideration of more than just the area they can move in. Manipulators change
geometry as they move and often operate in areas more cluttered than mobile robots. As
manipulators are used in many industrial operations there has been significant research
into techniques to control and plan paths using them.

The difficulties researchers have addressed include creating complex control systems
[131 – 133] that are required to manage all of the degrees of freedom while still
maintaining the correct direction of movement. Seraji and Colbaugh [132] developed a
kinematic model of a robot using a Jacobian based control system. The modelling was
used to develop a control system for the manipulator to reduce the error in the
manipulator’s trajectory when moving along a path. These concepts were also
discussed in detail by Dullerud and Paganini [133] and Sciavicco and Siciliano [131].

One of the largest areas of research is mapping an environment for use with
manipulators. This becomes more complex for manipulators, especially for those with
more than three degrees of freedom. One of the techniques considered is Minkowski
sums [134], which can be used in terms of path planning, to calculate the free space that
a movable object may occupy considering the geometry of the obstacles. The inverse of
the Minkowski sum represents the free space. The advantage of this technique is that it
allows the use of complex (non convex) shapes in close proximity to obstacles. The
difficulty with this approach is that it requires the exact geometry of any obstructions to
be known beforehand and requires significant processing. Only after these calculations
are completed can it be used for path planning of static environments. Much of the
erlier work into control systems [135 , 136] for robots used Minkowski sums to show
the free space of an environment. The Minkowski sum is useful when considering two

45
dimensions in space, but has been used less with three dimensional space or configuration space. The Minkowski sum has been utilised more recently for collision detection as discussed in section 2.2.1.

There have been many techniques used in the control of manipulators for path planning. Lozano-Perez [120, 137], considered the requirements for automatic planning of manipulator transfer movements, including analysing an environment using configuration space, path planning, collision detection and object grasping. Faverjon and Tournassoud [81] used a potential field approach to control multiple cooperative robots using both simulated 3D environments and real robots. A cross section of configuration space was used to determine cells for a discrete A* based path planner to use. Lingelbach [135] applied probabilistic cell decomposition to determine appropriate cells for path planning.

Most of this earlier research into path planning has been refined to the point that applications of path planning are common in many areas, including mobile robotics and industrial applications. Padula and Perdereau [78] demonstrated the use of a manipulator using potential fields to determine a path through a cluttered environment, while avoiding collisions. The solution proposed was tested in both simulation and a real world example. The robot used was a Motoman IA20, which has seven degrees of freedom. The oscillations produced by the potential field approach in narrow passages were reduced using a “pseudo inverse Jacobian” method. Dearden and Burbridge [138] considered the use of geometric and symbolic planning to identify objects and plan paths to interact with objects. Geometric methods search for geometric similarities in an environment. Symbolic planning uses names to identify objects allowing a more
natural language interface to create tasks for robots. This is demonstrated using 3D simulations of a robot in a simple environment.

These concepts have proven useful in many applications of manipulator technology including, automated packing, and bin picking. For example Wilson [139] demonstrated the use of path planning to create a path for assembling products, using simulations. Bozma and Kalaloglu [96] considered the problem of multi-robot coordination in a fully autonomous packaging system. Game theory, related to the use of multiple agents, in both cooperative and non-cooperative strategies, was used to improve the overall performance of all robots in the system, where the paths of two robots were adjusted based on incoming objects on a conveyor. The goal of the strategies was to either individually (non-cooperatively) or collectively (cooperatively) reduce the energy required to perform a pick-and-place task on a moving conveyor. The research demonstrated the differences resulting from the two different strategies using a 3D simulation of the environment in the context of a manufacturing process.

Many researchers [25 – 28, 66] have attempted to combine vision systems and manipulator path planning. An example of this is the field of bin picking [25 – 27]. An extension of this work has been completed by Nieuwenhuisen et al. [25], who developed a mobile robot aimed at general home maintenance that was capable of bin picking.

Other research into the use of manipulator path planning includes safety systems [19] and assembly planning [68]. The manipulators themselves can have as few as three degrees of freedom and more recently seven or more. Aasland [80] focused research on a manipulator with nine degrees of freedom. Many industrial applications typically use
2 Literature Review

robots with at least six degrees of freedom, as this is the minimum required to change both the position and orientation of a component in three dimensional space. The complexity of a control system for robotic arms increases with the number of degrees of freedom of the robot. It is noted that the more complicated the manipulator the more difficult it is to control especially with standard techniques, such as configuration space. Aasland [80] noted that reducing the ability to access areas of the workspace reduced complexity, and hence improved the reliability and robustness of the control system. The control systems of these robots are required to consider the environment in which they operate and often the use of configuration space.

2.1.4.1 Configuration space

The basis of configuration space is to consider each degree of freedom of a different dimension and map objects into the dimensional space of the manipulator [21]. The difficulty with this approach is that datasets become extremely large and unmanageable very quickly. As stated by Aasland [80] the more degrees of freedom the robot has, the quicker configuration space becomes impractical to use. For example, given a robot with six degrees of freedom and ten samples per degree of freedom the resulting configuration space will require one million samples.

Many researchers have considered configuration space and its application with path planning. Abrams and Ghrist [140] provided a general discussion of configuration space and its applications in industrial environments. Lozano-Pérez [141] introduced the concept of configuration space and applied the technique to consider path planning for both fixed shape objects and manipulators. Faverjon [142] considered the use of
configuration space with simple planar translating robots. Researchers such as Brooks and Lozano-Pérez [143], and Chibisov et al. [136] have discussed the use of configuration space and the use of potential fields to perform path planning. Kavraki [144] used an approach with planar translating robots by mapping objects into a bitmap from configuration space and using Fast Fourier Transforms and convolution techniques to produce a map for use with potential field path planning.

While many of these researchers made progress with the control of manipulators and path planning, there remains significant difficulty with the use of configuration space. Part of the difficulty with this approach is that configuration space can be massively redundant; many configurations produce the same final position and orientation of the robot end-effector. When considering three dimensional space, a manipulator can have its end effector in a particular position and orientation. There may be many configurations of the robot joints that produce the required output. Much of the research into manipulator path planning deals with simplifying the search structures of the configuration space. A commonly used technique in this regard is referred to as Probabilistic Road-Maps.

2.1.4.2 Probabilistic Road-Maps

As configuration space produces such large and complex search environments, they are often impractical for path planning [21, 75, 80, 82]. To reduce this complexity Probabilistic Road-Maps (PRMs) use sampling of configuration space, which produces a much smaller network of configurations that can still be used for searching. The more sparse the network the faster the search will be, but the more likely that a solution will
be missed. Conversely, the denser the network, the slower the search will be.

Hsu [145] demonstrated the use of a sampled configuration space to reduce the complexity of any path generated. This was achieved by sampling the configuration space relevant to the search area and demonstrated significant improvements in the speed of configuration space path planning.

There have been many attempts to improve the performance of the path planning process using sampling methods. LaValle et al. [75] and Geraerts and Overmars [84] compared the performance of several different approaches including several quasi-random sampling schemes; Halton Points, Hammersly Points, Lattice Points and Sukharev Grids. These sampling schemes are applied to problems with configuration space dimensions varying from two to ten. Geraerts and Overmars [84] conclude that none of the techniques provide an optimal solution in all cases and recommend further research into different techniques, to both analyse the current sampling techniques and develop new sampling techniques. Specifically, adaptive sampling techniques were recommended for further research. Aarno et al. [83] recommended a biased sampling scheme to improve path planning near narrow passages.

Kavraki et al. [82] developed a PRM for use with a seven degree of freedom robot in a static work-cell environment. The approach developed used a two stage process, the first phase or learning stage examined the environment and created a searchable network. The second stage is the query stage where a path is generated based on the required destination.

More current research from Geleri et al. [146] demonstrated the use of parallel algorithms implemented on GPU processors, to sample configuration spaces for path
planning purposes. This demonstrated that GPU implementations can be as much as 100 times faster than single CPU implementations, but still using sampling techniques.

One of the limitations of the PRM technique is that it produces longer paths than necessary. Nieuwenhuisen and Overmars [85] demonstrated smoothing techniques that significantly improved the speed of execution and shortened the final path.

Notably the vast majority of the authors considering PRM that were reviewed, only used simulated environments, as opposed to real environments through scanning. As the sampling process can take a significant amount of time to generate the data required to perform path planning, these types of planners are more suitable for static environments where the entire environment can be examined. In dynamic environments path planners need to be able to adapt to the situation.

2.1.4.3 Object Grasping

An application of manipulator path planning that has involved dynamic environments is object grasping. Object grasping involves analysing an object, often in-situ, to determine the best configuration to interact with the object given its orientation. Aleotti [29] investigated gripping of various 3D objects using Reeb graphs. The object is divided into sections to assist in identifying areas to grasp and pick up the object. Aleotti's interest in this technique is to be able to analyse 3D objects, so that robots can effectively interact with the objects by identifying suitable gripping positions. Cipolla [147] used stereo vision to assist a robotic manipulator detect an object and plan a path to interact with the object. The research was aimed at indirect user control of the operation by determining which object was being targeted in the images. This strategy
is only used in a controlled environment where the scan is limited to a known and well
defined area that objects could be easily isolated from. This is opposed to a complex
scene with moving objects that would require a full 3D simulation to track. Cipolla
[147] used un-calibrated stereo cameras to scan an unstructured scene and allow a
human to direct a robot to interact with objects in the scene. The robot was capable of
identifying the human operator and an object the operator was pointing at, then plan a
path to pick up the object.

Schraft [27] demonstrated several concepts related to bin picking. A 3D laser scanner
was used to scan a box of randomly piled known objects. A CAD model of the object
was matched to the scan data to determine the position and orientation of the object
using a convolution method. Once the position and orientation of the object was
identified, an appropriate gripping location was determined. The path planning system
then used this information to find a collision free path to pick up the object. This work
was continued by other researchers including Rastgar [26] who presented a Hopfield
neural network that used stereo vision to identify simple cube objects and to perform
bin picking in an unstructured environment. Rastgar discussed the use of the neural
networks for both calibrating the camera and data extraction from the stereo images.
The process of updating the neural network and using it to identify objects was
demonstrated.

Mobile robotics researchers are also interested in interacting with human based
environments. Klingbeil et al. [148] presented a concept of detecting and being able to
detect and open doors based on 3D scans. The objective was to detect the existence and
position of door handles with a mobile robot. This system was able to recognise
2 Literature Review

different types of door handles from a live scan and create appropriate strategies to turn the door handle correctly. Klingbeil et al. [148] demonstrated the ability to isolate known objects from a part of a scan, the ability to determine the position and orientation of a specific object and the determination of an appropriate “path” to grip the door-handle and open the door.

Stueckler et al. [59] developed a robot that could automatically examine an object and identify an appropriate pose to pick up the object. The robot was designed as a general purpose service robot capable of everyday tasks, such as picking up a spoon and pouring cereal. A Microsoft Kinect (TM) was used for real-time 3D scanning and object recognition.

These researchers have demonstrated that the integration of manipulator based tasks, path planning, sensors and object gripping analysis can be used to determine appropriate poses based on existing object models and from sensor data to interact with the physical world. Part of the objective of an interaction task, such as object grasping, is to avoid unintentionally colliding with any object in the environment.

2.2 Collision Avoidance

There are two possibilities to consider when mapping an environment. Mapping the free space or mapping the occupied space. Path planning is mainly concerned with operation in the free space, whereas collision avoidance is primarily concerned with occupied space.

Collision avoidance can be considered a component of path planning especially with manipulators, as a given configuration of a manipulator may intersect with an obstacle.
at any point along its kinematic chain. One of the components of manipulator path
planning is determining if and when a collision will occur. To accomplish this requires
detecting if shapes overlap in space.

### 2.2.1 Collision Detection

Collision detection is a major part of any manufacturing design [5] to ensure that a
manufacturing process can be accomplished without causing collisions between
components of the manufacturing system or parts being manufactured. Any overlap
represents a collision in the real world. Detecting these events is typically done using
CAD data of objects in a simulated 3D environment.

There are many software application programming interfaces (APIs) specifically for
collision detection, including PhysX [149]. These are all well developed software
systems that allow efficient collision detection in 3D simulations and many of these are
used in both industrial application and computer games. These software systems have
been developed for high speed operation and typically use well known and well
optimised techniques for minimising interactions between objects and performing
collision detection. These techniques are often propriety code and are not always
available for research purposes, but they include: Nearest Neighbour Analysis, Spatial
Hashing, K-nearest Trees, Oriented Bounding Boxes and Convex Hulls. These
approaches can only deal with known objects, both in terms of their location in space
and types of objects, any unknown objects are not dealt with.

Reichenbach [150] used a purely virtual environment to determine appropriate path
information. The appropriate path is determined by identifying collision locations from
overlapping oriented bounding boxes then progressively re-orienting the kinematic chain. As this is a purely virtual approach it does not attempt to take into account unknown objects or unknown environments, but it demonstrates the use of several useful tools such as oriented bounding boxes and a hierarchy of reference frames for positioning of objects and relative object position calculations.

To detect if an unknown object can cause a problem requires isolating it from the environment and then using techniques such as convex hulls to determine any overlap with other objects. A convex hull is a subset of a group of points where all of the points are inside the surface generated from the convex hull points. The convex hull can be visualised as a rubber band (in 2D) or balloon (in 3D) that surrounds the complete set of points. There are several methods for generating the convex hull including the quick hull [151] and Chan’s method [152]. A convex hull can be used to measure the smallest set of dimensions of an object. The Rotating Callipers method [153] uses each face of the convex hull to determine minimum dimensions of the hull. These measurements can then be formed into a minimum sized oriented bounding box. Convex hulls can be used to detect overlapping shapes using standard collision detection algorithms such as the Gilbert–Johnson–Keerthi distance (GJK) algorithm [88]. This algorithm determines if shapes overlap using properties of the Minkowski sum.

2.2.2 Safety Systems

An application of collision avoidance is Safety Systems. Safety Systems are used to detect objects in unsafe areas and to allow humans to interact with robots. This can be accomplished using both physical methods [13, 154, 155] and external sensors [12, 54,
2 Literature Review

These approaches are mainly concerned with detecting humans in a workspace. The typical strategy with current industrial safety systems is to completely stop any automated equipment when a human is detected.

Oleksa [13] reviewed many of the safety systems currently used to allow industrial robots to be able to interact or operate in the same environment as humans. These techniques include simple sensing devices, detecting errors in the machine operation, physical barriers, emergency braking and load limiting. Most of these approaches deal only with unknown event detection, but not recognition or tracking of objects. For example a light barrier may be used to detect if an object has passed through a restricted area, but the type of object will not be determined and the exact location of the object will not be known. The environments, in which these techniques are implemented, while known, are not modelled using 3D simulations. The performance of these systems are typically real-time as they are hardware implementations.

Zinn et al. [154] examined the effects of impact, specifically in terms of injuries caused by a manipulator impacting with a human body part, in particular the head. They proposed a manipulator design based on a double actuator that allows the inherent inertia of the manipulator to be reduced and still maintain high torque characteristics of the system. This research focused on making manipulator arms safe for humans to interact with by redesigning the manipulators themselves, but not by “avoiding” collisions. This approach is not always viable in manufacturing environments as many types of manufacturing require robots to operate with far greater loads than humans can safely interact with.

Haddadin [155] focused on human robot interaction and cooperation, specifically
2 Literature Review

examining physical impact and minimising the effect of collisions via Trajectory Scaling, but not on avoiding the collision. The main technique examined is to use the inputs from the motor encoders to detect impulses from external impacts which are outside of the expected operation of a programmed movement. The control system is then designed to minimise the energy of the impact by using the control system itself to reverse or brake appropriate motors quickly to reduce reaction forces.

When performing real-time tasks through tele-operation, human operators performing collision avoidance do not always have the best perception of a work area. As such, operators may request an action that causes a collision. A significant amount of work has been completed involving tele-operated robotic arms. Cheung & Lumelsky [156] describe tele-operation of robotic arms involving a robotic arm controlled by a human operator and using visual monitoring. The monitoring can be via video camera or in some cases stereo vision using a stereo shutter screen system to allow operators to perceive depth. Cheung & Lumelsky [156] also examined both the movement of the arm via tele-operation using stereo vision, haptic feedback and contact feedback from the “skin” of a robot. The concept was to allow a human operator to feel obstacles while controlling the robot, but this technique is not able to avoid collisions.

Physical methods for safety applications typically use sensors to limit or avoid damage, but while effective do not avoid the cause of the damage. Unsafe events can only be avoided entirely if they are identified beforehand. Vision based systems such as described by Yamamoto et al. [10] have been used to track the position of operator’s hands and ensure that they are a particular distance from any hazardous area. The position detection system was based on identifying markings on the operator and used
2 Literature Review

Kullback-Leibler divergence $D$ and Levy-distance $L$ to determine if the operator was in a safe area.

While individual cameras are less costly and effective at times, they have significant limitations with lighting and occlusion. Stereo vision based monitoring systems as discussed by Tan and Arai [11] improve the reliability of safety systems, as 3D data can be used to examine an environment. In their work, Tan and Arai [11] used a stereo vision system for safety monitoring in a human-robot collaboration production cell. Three cameras were used to improve the robustness of the system to avoid losing tracking and improve occlusion tolerance. A hand position tracking experiment was conducted to evaluate the performance of the 3D position estimation.

Ghobadi et al. [12] Henrich et al. [54] and Ebert et al. [55] all considered the use of cameras or 3D scanning systems to attempt to identify unknown objects that may cause collisions. They typically use a background subtraction process for eliminating irrelevant information from background clutter in an attempt to identify moving objects. Anton et al. [42] discussed a method of tracking and controlling robots that interact with humans to provide assistance in manufacturing tasks. Using depth sensors, the robots are able to analyse the environment to detect the human operator and to avoid collisions. They demonstrated that pose estimation of the human operator can be used to send basic commands to the robot.

Wang [157] presented a web-based solution for real-time collaboration in adaptive manufacturing, including active collision avoidance for human-robot collaborations. It combines virtual 3D models and real sensor data to detect the presence of human operators. Wang's objective was to improve the human safety for humans and robots.
co-existing in an environment. The system was implemented as a remote system operating over the internet, as opposed to performing the methods locally.

Puls and Graf [9] focused on the identification and tracking of human kinematics based on 3D imaging sensors. The aim of this work was to improve human-robot cooperation using human pose reconstruction, situation and activity recognition. A kinematic model of a human was developed for the purpose of tracking and fused with knowledge about robot kinematics and any surrounding objects into an environmental model. This allows for efficient risk estimation and subsequent risk minimisation through adaptive robot motion. The components were merged into a single framework for human-robot cooperation. Several simulated examples of interactive and cooperative scenarios with human-robot collaboration were presented.

Many of these approaches demonstrate the use of simulations to control some aspect of the operation of a robot and several demonstrate the use of scanning physical environments. While these approaches are used to detect unknown objects, they typically assume that the object is a human. In some cases [42, 157] the approach was to simply assume the object is a human and perform pose estimation and check later if the object matches the data correctly. To be able to detect any type of unknown or unexpected objects 3D scanning is required as 3D scanning allows objects in the real environment to be detected.

2.3 3D Scanning

There are many technologies that are currently available for performing 3D scanning. Ultrasound detectors and lasers can continuously measure in a single direction from the
sensor and provide a distance measurement to any object in line with the direction. These can be used as area scanners by rapidly rotating them, but with this technique it still takes time to scan an entire area, meaning that any moving objects captured by the scan will be distorted.

Bi and Wang [158] have reviewed many types of 3D scanning systems, their capabilities and applications. The full table of their review is presented in the Appendix with several more recent scanners added. There are several technologies that can be used to scan 3D environments to produce a 3D model including stereo vision [11, 26, 47], Laser scanners [27, 34, 38, 39], Flash LIDAR [31, 45, 46], PMD scanners [12, 47, 48] and Microsoft's Kinect (TM) [51, 59]. While many of the devices presented in the Appendix are capable of performing specific tasks [158], such as scanning a particular object, only a few are useful for live scans of an environment. These devices use many different techniques to achieve their goals and many are used for specific manufacturing processes.

Stereo vision [11, 26], Flash LIDAR [31, 45, 46], PMD [12, 47,48] and the Kinect (TM) [51, 59] all capture an area at the same time (as opposed to laser scanning which measures a single point at a time rapidly) although, especially with stereo vision, there may be significant post processing. The resolution of 3D scanners also significantly affects the size of objects that can be examined. Laser scanners and stereo vision are the most flexible as they can achieve accuracies and resolution of less than 1 mm and are often employed for tasks such as reverse engineering and 3D modelling in industry. The difficulty with laser and stereo vision is that the rate at which they can scan an entire scene is limited. Stereo vision will often only run at 3-5 Hz and takes significant
2 Literature Review

processing or custom hardware to achieve faster rates. Laser scanning may take minutes to completely scan a scene [37, 38]. In comparison Flash LIDAR, PMD and the Kinect (TM) can scan an area at a higher rate of 30 Hz [12, 31, 45 – 48, 51, 59]. The limitation of these sensors is that they are not as accurate and have less resolution than laser or stereo vision scanners.

2.4 3D Simulation

While 3D scanning is useful for examining an environment, it cannot be used to directly specify a sequence of events. 3D simulation is often used to design objects and processes before they are constructed. Autonomous manufacturing systems often rely heavily on 3D simulation for the design [5], operation and to analyse its performance [96]. 3D simulation research has been used for optimisation and control of manufacturing systems to improve their productivity.

Connolly [5], provides an overview of how Robot Studio (ABB, Zurich, Switzerland), a simulation software package, is used in industry and the improvements that can be made to production systems via simulation. Robot Studio has been developed to allow manufacturing processes to be developed “off line”. This enables operators to begin development of processes for new products while old products are still being manufactured. Connolly [5] demonstrates that designers of the manufacturing processes are able to model both the static physical components of the process and the dynamics of the machines involved. This allows collision detection, part mating, clearance checks and overall productivity to be calculated before the system has been built. This was noted as one of the features that reduce risk in developing and implementing new
2 Literature Review

manufacturing systems.

Steopan et al. [159] discussed the use of CAM software to improve factory floor productivity in manual production systems. The simulations of the factory floor allow the relationships between the required manufacturing processes to be determined. From this data the layout of machinery can be optimised to minimise the complexity of the processes and reduce the time taken to move parts between each process. Steopan et al. [159] concluded that the production from a layout, optimised using simulation, can be improved by as much as 14%. In one case, the simulation indicated that using conveyor belts between processes would improve productivity by as much as 35%.

Miklic et al. [160] discussed the use of Automated Guided Vehicles (AGV's) in automated warehouses. The system presented handles all aspects of warehouse operation, from individual vehicle control to high level mission planning. To validate the algorithms developed, both “WarehouseSim” and “USARSim” simulation software packages were used [160]. This research showed that AGVs could be automatically routed while safely avoiding collisions and deadlocks and ensuring that all deliveries were correctly made throughout the warehouse environment. While research with off line systems shows that problems can be avoided at the design stage, problems can still occur in the real system if it does not operate as designed.

2.5 Combined Scanning and Simulation

In an extensive literature review very few researchers were found to have considered the combination of 3D simulation and 3D scanning technologies. Fischer and Henrich [53] developed a concept which used both 3D scanning and a simulation of a robot work
cell. They incorporated the concept of using a simulation of the cell and a real-time 3D scanner for comparison. The applications mentioned include safety systems and monitoring of museum objects. Mixed reality [161] has been used with mobile robots to allow a real robot to interact with virtual objects. This concept requires a 3D simulation to be created and position synchronised with sensor data from the robot. This allows appropriate navigation in the real world while still taking into account virtual objects. This shows that a 3D simulation can be used as a basis to interact with the real world.

2.6 Summary of Literature Review

From the review of literature on path planning, techniques such as potential fields and occupancy grids are found to be common methods of analysing 3D data, especially with mobile robotics, to determine collision free paths through environments. Occupancy grids also allow sensor fusion techniques to be used to combine data from multiple sensors into a single coherent dataset for analysis. SLAM combines sensor fusion, navigation, path planning and surface fitting techniques to determine the path a robot or vehicle has taken and at the same time produce a map of the environment.

In comparison with path planning for mobile robots, manipulator path planning has to be considered in the context of robots with more degrees of freedom resulting in larger search trees that are more difficult to update. There has been a paucity of research dealing with manipulators in dynamic environments or involving unknown objects. Most published research on manipulator path planning describes static environments from simulated data. Very few have considered information on dynamic environments,
2 Literature Review

especially collected from sensor data.

Given the differences between the two applications, mobile robots and manipulators, it is not surprising that there is a significant disparity in research. The main feature to note is that while the discipline of mobile robotics has achieved significant success in dealing with unknown environments, using 3D scanning and appropriate analysis, research on manipulator path planning has not yet considered these approaches.
3 Methodology

3.1 Introduction

Path planning is a well-developed area of research that has found considerable success in many applications including mobile robotics and industrial manipulators in manufacturing environments. Collision avoidance is closely related to path planning, as where path planning uses unoccupied areas, collision detection examines the occupied areas. Collision detection is especially important for manipulators as they have a complex and changing shape when moving through environments.

The aim of this research is to develop a system that allows both real-time collision avoidance and path planning of a manipulator to be performed in a dynamic manufacturing environment. This research is based on the scientific method aimed at industrial environments, as introduced by Reigeluth [4]. The scientific method is to Hypothesise, Observe, and Test. The process developed for robotic or industrial operations defines these as Simulate (Hypothesise), Scan (Observe) and Compare (Test). Considering these concepts, the review of current literature identified three main areas of research that have not yet been thoroughly developed as shown in Figure 3.1.

The contributions are related to Environment Analysis, Task Analysis and Path Planning to update a simulation and modify a task. In previous research these concepts have not been separately investigated. For example, mobile robot path planning has considered environment analysis by subdividing scans of a local environment to generate a searchable node network for path planning [23]. In the new design, as developed in this research, path planning becomes the feedback of a movement task. None of the
3 Methodology

literature reviewed, especially on path planning, considered the concept of task analysis, or specifically: can a task be completed given a set of current circumstances?

The Environment Analysis method was designed to perform background subtraction of

![Diagram](image)

**Figure 3.1: Contributions**

a 3D scan using a 3D simulation. The use of simulation for background subtraction has been designed to address limitations in previous 2D techniques for identifying and isolating objects in an environment. This process required the development of several algorithms that could be used to match triangles from a 3D scan to a 3D simulation. The result of this process in matching objects of the simulation to the scan allows any remaining areas of the scan to be isolated, grouped and considered as unknown objects. Finally, it is possible to identify objects that were specified in the simulation, but missing from the scan by considering where the scan matched objects but the surface
3 Methodology

normals were in opposite directions.

The Task Analysis module was developed based on the capabilities generated by the simulation background subtraction process and using them to determine if a task could be successfully completed or if it required modification. The main purpose of Task Analysis is to determine if it is possible to complete a specified movement task based on both the simulation data and the current state of the environment. This was completed by considering the simulated movement of a robot in a work-cell and generating a convex hull of the movement. The movement hull can then be compared to any unknown object in the vicinity using standard collision detection algorithms such as the Gilbert–Johnson–Keerthi (GJK) algorithm [88]. This required the development of software to analyse the operation of a real robot in a test environment. It is expected that this technique can be used to determine if a real robot's planned actions will be interfered with by any unknown objects in an environment and allow path planning to be performed.

The SSC process is based around an occupancy grid. This design populates the grid cells with both simulated and scan data for matching purposes, but this also allows a path planner to use the same data. Using this technique it was expected that the path planner using the occupancy grid would be able to take into account both unknown objects and any simulated objects that have been matched to simulation data. It was expected that the use of simulation data would allow the path planner to take into account gaps in the sensor data.

Finally, these three modules were combined into a single process which would utilise all
3 Methodology

three components in a real test environment. This required the integration of the software from the Environmental Analysis, Task Analysis and Path Planning components into a single component, which was integrated into an off the shelf commercial product (Robot Studio) for developing 3D simulations of manufacturing processes.

It is expected that the final SSC process would be able to:

- Run the robot through the current task, while monitoring the environment
- Identify objects that were not specified in the 3D simulation for the manufacturing process
- Determine if any of these unknown objects presented a problem with a current movement task
- Make a decision to either stop the task from continuing or plan a reroute path around an obstacle, if an obstacle was determined to be interfering.

3.2 3D Environment Analysis

The Environment Analysis method was designed using a 3D simulation to perform background subtraction. The question this method addresses is: “is the simulation the same as the scan?” In terms of the scientific methodology this becomes “is the observation the same as predicted by the hypothesis?” This question is tested by comparing the two inputs i.e. the subtraction process. The “subtraction” process developed for 3D objects is more like matching between surfaces. When a simulation triangle matches a scan triangle it can be “eliminated”. Two metrics were used for the
3 Methodology

matching process:

- Average point projection distance
- Relative surface normal

The average projection distance is calculated from the projected distance onto the surface of the simulation objects from the apexes of each triangle of the scan ($d_0$, $d_1$ and $d_2$) as shown in Figure 3.2.

![Figure 3.2: Matching metrics](image)

The angle between the surfaces is calculated from the dot product between the simulation object and scan surface normals. The surface normals are calculated from a triangle from the scan and simulation objects. It is expected that this matching process will result in more stable distance measurement than individual point measurement as an average of point distances is used. The normal comparison allows the detection of surfaces that are facing opposite directions.

The use of the simulation for background subtraction creates several capabilities that are not present with other methods. Firstly, using a simulation for background subtraction allows any type of object to be subtracted from the scan as opposed to attempting to detect planes or subdivide the scan. Additionally, the concept of using the simulation
3 Methodology

for background subtraction creates several capabilities:

1. The ability to match simulated (known) objects with scan data.
2. Detecting missing simulation objects.
3. Isolating and identifying unknown objects (as opposed to known simulation objects).

In the process developed, if a simulation object successfully matches with the scan in both position and surface normal it is assumed that the whole object exists. The successful matching between simulation object and scan is assumed to demonstrate that the object is in both the correct position and orientation.

Areas of the scan that are not matched to a simulation object are grouped and each group is assumed to be an unknown object as shown in Figure 3.3.

![Figure 3.3: Object isolation](image)

Triangles in green have been successfully matched to a simulation surface. Triangles in black are filtered as they connect an object to the background and do not represent a real surface. Triangles in red have not been matched to a surface. Using a region growing algorithm, and taking into account that triangles have a fixed number of neighbours
3 Methodology

(three) the unmatched triangles can be grouped and measured with an oriented bounding box algorithm, shown in blue.

If any simulated object matches with a scan, but with a surface normal in the opposite direction, as shown in Figure 3.4, it can be assumed that the simulated object does not exist in the real scene. Most likely the ground will be visible below an object to demonstrate that the object does not exist in a scene. Notably none of the literature reviewed has considered the idea of detecting missing objects.

![Figure 3.4: Normal comparison diagram](image)

Using the simulation it is possible to tell what the sensor data should be, not necessarily what it is. The simulation itself could be wrong (for example a component missing from the environment) which gives the potential for a corrective action to be taken. However, in a well-controlled manufacturing system that has significant restrictions on the performance of tasks, if the simulation is not correct there is probably an error in the production process, for example a jam in a feed mechanism. Hence if errors in the simulation can be detected, including either missing objects or unknown objects (within a fraction of a second) either the task can be stopped or a correction can be determined. The aim is for this to occur fast enough to avoid damaging either the product or the
3 Methodology

manufacturing equipment and avoid injuring any human in the environment.

3.2.1 3D Simulation Source

When products are developed for manufacturing, substantial information about the object will be generated. As such, it makes sense to utilise this information for comparing with the scans of the real environment. For many types of products this includes 3D models and in many cases full 3D simulations of the manufacturing process. To develop simulations, the software developed for testing was integrated with an industry standard software package (Robot Studio). Simulations can use several formats to describe 3D objects in an environment including polygon based models and solid modelling. Either of these formats is acceptable as long as the surfaces can be interrogated and compared to the scan data.

The simulation models need to be created to cover all areas of the scan including active objects, such as robots, conveyor belts and any parts required for the manufacturing process. Also any incidental inactive objects, such as safety barriers need to be modelled just in case any of these items enter the scan area. In addition, the scanner should be oriented to avoid detecting areas that are not relevant to the test operations.

3.2.2 3D Scan Source

There are several technologies that can be used to scan an environment including stereo vision, laser scanners and a variety of “depth” cameras. Stereo vision takes significant processing and comparing the simulation and scan is also a computationally intensive process. While laser scanners are accurate they typically cannot scan an entire
3 Methodology

environment in a fraction of a second. Depth cameras were found to be the most suitable technology for acquiring a 3D scan of an environment rapidly enough to detect dynamic objects in a scene.

3.2.3 Alignment

The first part of simulation background subtraction is aligning the simulation and scan to the same coordinate system. This can be achieved using either model fitting between simulation and scan, or control markers (control points) in the scene. In the current design a control point based system for aligning the simulation and scan data was used. It was preferred to establish the coordinate system using control points as this simplifies the processing requirements. Alternatively, model fitting can be used adaptively during operation to correct any misalignment in the scan orientation. The difficulty with this approach is that it may mismatch with objects that have moved in the scene and throw the entire model out of alignment. Using control points in a scene is a more robust method of aligning the environment, but it is often difficult to run during the matching process as identifying the control points automatically takes significant processing to complete. It is more efficient to align the scanner and assume it does not move during scanning.

3.2.4 Matching Algorithm

Once the scanner orientation is accurately matched to the simulation coordinate system there are several options for matching the surfaces together including: point to point, point to plane, and plane to plane. All of these options require geometric calculations
3 Methodology

such as point projections and normal comparisons. In theory any of these options will work as long as the technique used identifies sets of points close to each object and isolates any points not associated with any object.

With the software developed, a plane to plane technique was used as shown in Figure 3.2. This uses an average of three points from the scan data at once to determine an appropriate distance to the simulated plane, as such, it should be less susceptible to surface noise. In addition, the surface normal of the scan plane can be compared directly to the surface normal of the simulation data and a second metric based on the dot product can be used as matching criteria.

Several alternative techniques have been developed for matching between scans especially with SLAM [122–124, 127] algorithms. These techniques usually use point to point based matching strategies. The difficulty with many of these processes is that they require similar point densities to match between surfaces. Simulation data can require significantly fewer points to model an object. As such, it is often difficult to determine useful point pairs between simulations and scan data making these matching alternatives difficult to implement successfully.

3.2.5 Simplifying Matching

A brute force method of matching the simulation and scan triangles would simply match each scan triangle to each simulation triangle. In terms of algorithmic complexity a brute force matching algorithm is $O(n^2)$ or quadratic complexity. As such, there has been a significant effort to simplify the matching complexity. Occupancy grids have often been used for sensor fusion, which allows several sets of data to be combined into
3 Methodology

a single coherent picture of the environment. The occupancy grid approach can be extended to accept both scan data of the real environment and the simulation environment. If the alignment process is accurate, both scan and simulation data should occupy the same cells. In the worst case, scan and simulation data should occupy neighbouring cells.

By using an occupancy grid the complexity of the matching can be reduced as shown below. Using $m$ model triangles, $n$ scan triangles, $k$ cells and $p$ triangles per cell the simulation space can be split into $k$ cells where the complexity of each cell will be:

$$O\left( \frac{m \times n}{k^2} \right)$$

The total complexity of $k$ cells will be:

$$O\left( \left( \frac{m \times n}{k^2} \right) \times k \right) = O\left( \frac{m \times n}{k^2} \right) \approx O\left( \frac{n^2}{k} \right)$$

If we allow $p$ triangles in each cell:

$$k = \frac{n}{p}$$

The complexity simplifies to $O(pn)$, linear time but flexible based on the size of the cells used. With a typical model of 10,000 triangles and a $p$ value of 100, the number of comparisons is reduced from $10^8$ comparisons to $10^6$ comparisons, i.e. 100 times faster. The actual performance will be slightly slower as a triangle may appear in multiple cells. This improvement is limited by the size of the triangles. If the cell size is smaller than the triangle size, each triangle will appear in multiple cells.
3 Methodology

3.2.6 Object Isolation Design

By examining which triangles have not been matched to the simulation, the unmatched triangles can be grouped together to form unknown objects. Using the triangle neighbours, a region growing algorithm [34, 37, 162] is used to expand groups locally on the surface, as shown in Figure 3.5 and flowchart in Figure 3.6. The region growing algorithm scans the triangles to find a seed triangle where the match value is less than a matching tolerance and not filtered. From this seed triangle the neighbouring triangles can be examined using the same criteria to add to a new group. The growing process is iterated until no neighbouring areas can be found.

These groups once isolated are considered unknown objects and other measurements can be performed on them such as size measurement or collision detection for tracking and task analysis.
3 Methodology

Figure 3.6: Region Growing Algorithm Flowchart
3 Methodology

3.2.7 Size Measurements

Once the matching is complete and objects have been isolated from the remaining triangles, the size of unknown objects is measured using a minimum oriented bounding box algorithm [163]. Essentially a convex hull is generated [151] around the isolated surface to create a completely closed object. Then the rotating callipers algorithm [164] is used to create measurements of the surface and the smallest set of measurement is used.

3.2.8 Tracking

While size measurements determine how accurate objects are in space, tracking can be used to determine the position of objects over time. Two different tracking algorithms were developed including position and velocity based methods. The position based method simply uses the GJK collision detection algorithm [88] between sequential scans to determine if an unknown object overlaps. The velocity method extends this concept by considering where the object is moving and updating the previous position of the object with the expected movement given the time between frames. The same collision detection algorithm is then used on the updated object.

3.3 Task Analysis

The Task Analysis chosen was to examine the pre-programmed path of a manipulator, as this was appropriate for the capabilities of the depth camera used. To analyse a movement task, the path of the robot is broken up into linear section between points. The initial and final positions of the robot are used to create a convex hull of the
3 Methodology

movement using a quick-hull algorithm [151]. The movement hull is then compared to any unknown objects isolated from the real environment using the GJK algorithm [88] for collision detection. It is expected that this technique can be used to determine if a physical robot's planned actions will be interfered with by any unknown objects in an environment and allow path planning to be performed.

3.4 Path Planning

The feedback given for a task is dependent on the type of task. As the task demonstrated was based on a robot movement (a movement task), the appropriate feedback is to re-plan the path, if it is obstructed. In the current design, feedback into the simulation will only occur after both Environmental Analysis and Task Analysis have been performed. The Task Analysis for path planning determines if a path has interference by performing a collision detection check between a simulated path and any unknown objects detected in the scene. When the Task Analysis has determined that an operation has an object interfering with a planned path there are two options:

1. Stop performing the task
2. Alter the task to avoid the interference

Stopping the task is simple, rerouting a task is more complicated. Rerouting a manipulator to avoid collisions with dynamic objects requires taking into account the complex shape of the manipulator.

As an occupancy grid is used as a central component of the matching process to isolate objects, the grid can also be used for path planning. During the two previous stages
3 Methodology

(Environment and Task Analyses) an occupancy grid is populated with both simulation and scan data. This allows the Path Planner to take into account real objects detected using the scanner and can cover any gaps in the sensor data using the simulation data. When scan data matches to simulation objects, both the visible front surfaces of objects and the back surfaces that are not visible can be used to flag the relevant cells in the occupancy grid as occupied. This still leaves some areas of the 3D environment unchecked, but it is expected that this approach is more flexible than just assuming the blank spaces are either empty or occupied.

### 3.5 Simulation Scan Comparison (SSC) Process

The three methods of Environment Analysis, Task Analysis and Path Planning were combined into a single process which is referred to as the Simulation Scan Comparison (SSC) process. The SSC process is designed to scan and map an environment, so that path planning of a manipulator can take into account dynamic unknown objects if they interfere with a specified task. Typically, the Environment Analysis will be performed continuously to determine if any unknown objects exist or any objects required by the task are missing. Task Analysis will be performed if unknown objects or missing objects are detected. Path planning will only occur if the Task Analysis determines that the current task has interference. This design is based around the use of an occupancy grid, which is used to map the environment, simplify the comparison process between simulation and scan, and allow path planning with dynamic obstacles.

The SSC process has been designed with two main inputs, these being the 3D simulation of the manufacturing process and the 3D scan from the scanning device. An
3 Methodology

An overview of the process is shown in Figure 3.7 and an outline in pseudo code is shown in Figure 3.8.

Figure 3.7: SSC process design

The environment analysis consists of taking the two inputs, scan and simulation, comparing them while the manufacturing process is running, then analysing the results to isolate objects. The scan provides a real-time overview of what is occurring in the manufacturing area while the simulation shows what is supposed to occur. The comparison of these two datasets allows any unknown objects to be isolated from the scan. At the same time the scan is matched to the objects. If the objects match to the scan correctly the objects position is effectively confirmed. If the scan and objects are completely in conflict it can be assumed that the simulation object is missing.
3 Methodology

**Initialisation**
- Off-line initialisation before scan
- Create 3D Simulation
  - for all object triangles
    - map fixed and static objects into occupancy grid
- Upload robot path
- Upload configuration cell data

**Environment Analysis**
- During scanning
  - for all moving objects triangles
    - Map Robot triangles into occupancy grid cells, flag cells as occupied
    - Create Movement hull from current robot position and current path point
  - for all Movement hull triangles
    - map known Movements into occupancy grid cells
  - for all scan triangles
    - if triangle size > threshold
      - flag as invisible
    - else
      - compare to triangles already mapped into occupancy grid
      - record best match value for each triangle
      - update simulation objects status based on triangle matches
  - for all scan triangles
    - if scan triangles not used and match value < threshold
      - add to new group
      - using a region growing method to expand the group
      - based on neighbouring scan triangles
  - for each detected group
    - Create Convex hulls of new objects

**Task Analysis**
- Compare detected groups to movement hulls
- if (Detected a group interferes with a Movement hull )
- Run Feedback

**Feedback**
- Run path planning A* algorithm

**Outputs:**
- Unknown objects (Location and Size)
- Unknown object interference of movements
- Simulation object matching status

**Simulation object status outputs:**
- Matched correctly to scan ie object exists and is in the correct position.
- Simulation object outside scan area
- Simulation object not matched, Could be occluded or missing
- Simulation object missing

Figure 3.8: SSC algorithm pseudo code
3 Methodology

Task Analysis in this design takes the data from object isolation and tracking and performs collision detection with the movement hulls generated from the planned robot movement. The Path Planning is only activated if the Task Analysis detects a collision between the planned path and an unknown object.

3.5.1 3D Simulation

The simulation is developed manually from the design of products and from the layout of machinery in the manufacturing area. This simulation needs to be an accurate representation of the process including robot movements. The accuracy is determined by the resolution of the scanner used. The simulation data represents what should be happening if the manufacturing process is operating correctly. Each component of an assembly should be predefined and presented to the manufacturing system in a predictable manner. Each object in the simulation is represented by a set of points and a set of triangles. Surface texture of known objects is optional, but can be used for object recognition purposes.

A reference frame hierarchy is used to represent objects forming the robotic arms so that they can be positioned appropriately using joint angles. This allows the joints of a robot to be updated quickly as only a single position matrix needs to be updated for each mobile object. The movement hulls are generated from the initial and final positions of the components of the robotic arm and creating a convex hull for each component of the robot. All of these objects are inserted into the occupancy grid as the basis for comparison with the 3D scan. Any cell occupied by an object is flagged as occupied so that the path planning algorithm can avoid using the cell.
3 Methodology

3.5.2 3D Scan

The 3D scan is acquired from the scanning device. The device used could be a set of stereo cameras or a depth camera. Each of these devices has different capabilities which allow them to be used in different situations. Depth cameras have high refresh rates which allow them to be used with fast moving objects. Stereo vision potentially has higher resolution, but requires a textured scene, which is not common in manufacturing environments [158].

The scanner needs to be positioned in an optimum location to be able to see the process so that all of the objects critical to the process are visible. A set of control points is marked in the scene so that the scanner position can be calibrated. The scan data is stored as a set of points, a set of triangles and several other arrays based upon the points and triangles. A set of normals is generated from the points and triangles each representing the normal of a triangle. The triangle normal can be calculated using the cross product of the two sides of each triangle. This normal calculation is used as it is very simple and quick to calculate. An array of indices is also generated representing the neighbours of each triangle.

As each triangle has three sides it also has three neighbours and creates a structure that allows the triangles to be searched through based on the position of the “nearest neighbouring triangle” as shown in Figure 3.9. The 3D scan shows the current state of the scene including unknown objects and the current position of any known objects.
Before the scan can be matched it needs to be filtered to remove any large triangles that will never match. Triangles are often created connecting a foreground object to a background object resulting in large “stretched” triangles as shown in Figure 3.10. The method currently used for filtering triangles is based on size. Alternate methods such as comparing the triangle normal direction to the scanner direction often remove valid triangles. The filter size approach allows large triangles that need to be eliminated and assists in the isolation process, as illustrated in Figure 3.11. The large stretched triangles on the side of the cubes have been removed allowing connections to be made during grouping of the front and top of the cubes but not to the background. The filtering removes background triangles that do not match with the simulation and can potentially lead to false positive detections of objects.
3 Methodology

Figure 3.10 Pre-filtered model

Figure 3.11 Post filtered model
3 Methodology

3.5.3 Comparison Algorithm
The comparison algorithm is the main component of the SSC process, which takes both simulation and scan data and compares the two. As explained in Sections 3.2, and 3.4, the two sets of data are mapped into an occupancy grid and matched together. The matching technique uses a plane to plane matching technique, where triangles from the scan data are matched against triangles from the simulation data. This allows both the distance between points and the surface normals of the triangles to be considered as comparison metrics. The use of the occupancy grid for this matching process has a dual capability in that it is used for both path planning and simplifying the matching process.

3.5.4 Object Isolation
By examining which triangles have not been matched to the simulation, unmatched triangles are grouped together to form unknown objects. Using the triangle neighbours, a region growing algorithm [34, 37, 162] is used to expand groups locally on the surface. Once detected, a convex hull is generated to create a closed surface. This is also used for path checks as the collision detection algorithms work quickly with convex hulls.

Once objects are isolated they are stored as separate sets of triangles along with several other structures including a convex hull and an oriented bounding box. A convex hull is created from the isolated points as most likely, the back face of the object will not be detected. This allows a rough estimate of the volume to be determined based on the surface of the 3D scan data.

The isolated objects represent unknown objects that have been introduced into the
3 Methodology

manufacturing area. These objects can be compared to the movement hulls generated from any mobile components of the simulation or tracked over time to determine if they will become a problem.

3.5.4.1 Missing Object Detection

One of the advantages of the SSC process is that it can detect if objects are missing from the simulation. Conflicting data can be found by comparing the surfaces described by the scan triangles to the simulation. If the simulation and scan surfaces are in conflict the surfaces have the same position, but the direction of the vector orthonormal to the surface of the scan and simulation, data will be in opposite directions. This will produce a dot product between normals of close to -1.0. In this system, the surface normal always points away from the centre of the object. If the surface normal of the simulation is pointing in the opposite direction to the surface normal of the scan then either the object is missing, or is incorrectly located (Figure 3.4 in section 3.2). For an object to be detected as existing in the correct location, both the simulated object and the scan must have surface normals in the same direction. If a missing object is a critical part of the manufacturing process the process can be paused until the object is available or, stopped completely.

3.5.5 Object Measurement - Tracking

Once the unknown objects have been isolated they can be measured and tracked as described in sections 3.2.7 and 3.2.8. As the unknown objects are generated from 3D data it is possible to estimate their size. The two components of the size measurement are the convex hull and an oriented bounding box [163]. A convex hull is created
3 Methodology

around an unknown object, which can then be used to measure the smallest set of
dimensions of the object. The convex hull is used as this is the simplest method for
estimating the “back” of the object that was not detected by the scan. The convex hulls
are calculated using a method similar to the quick-hull algorithm [151].
The process to create the oriented bounding box currently uses a method similar to the
rotating callipers method [164]. This technique uses each face of the convex hull to
determine a point at a maximum distance from the face, which creates the first axis.
The remaining two axes are measured by considering a projection of the hull in the
direction of the first axis. Then a 2D version of the rotating callipers method can be
used to measure the hull. The smallest of these measurements can then be formed into a
minimum sized oriented bounding box.

The size measurements of the object is stable enough to allow an object to be identified
as the same object between frames. Object identification in this sense is distinguished
from object recognition, as there is insufficient information to determine exactly what
an object is.

3.5.6 Object Tracking

As the unknown objects isolated using the SSC process are detected in three dimensions
they should be able to be tracked easily. The main advantage of using 3D scanning
compared to 2D scanning is that the object will be spatially separated in depth from any
others; two objects can never occupy the same volume.
The first technique is based on the idea that inanimate objects don't move of their own
accord. By comparing between two frames of the 3D scan it is possible to determine if
two unknown object detections occupy the same volume in space. Essentially a collision detection algorithm such as the GJK algorithm [88] can be used for this purpose. This is referred to as position based tracking. To be successfully tracked the displacement between frames needs to be less than the size of the object. This is demonstrated in Figure 3.12.

![Figure 3.12: Expected tracking capabilities of position based tracking](image)

If an unknown object does move, velocity based tracking can be used to better estimate where the object will be. By using previous positions of an object its velocity and acceleration can be estimated and hence, it can be determined where the object will be in the next frame. The main difficulty with this approach is that the first two positions of the unknown object need to be identified before the velocity can be estimated. Essentially the position tracking needs to work before the velocity tracking can be used.
3 Methodology

With either technique (position or velocity) a successful identification of objects allows the position of the object in the next frame to be estimated and fed back into the simulation.

3.5.7 Task Analysis

To perform Task Analysis the path of the robot was broken into line segments. The initial and final positions of each segment, combined with each component of the robot, were used to create convex hulls of the movement for each segment. The “movement hull” can be created using a standard convex hull generation algorithm [151] for visualising the hull. Once the movement hulls have been created they can be compared to any unknown objects detected using standard collision detection algorithms such as the GJK algorithm [88]. The GJK implementation was chosen for the SSC process as it is simple to implement with the data generated by the movement hulls and unknown objects. The GJK algorithm does not require a convex hull, as a simple implementation will inherently use the convex hull of a set of points to detect overlapping shapes. Hence the convex hull generation is not specifically required except for visualisation purposes.

3.5.8 Path Planning

The occupancy grid used for the matching process can also be used for path planning. The path is generated from the occupancy grid using both the scan and the simulation data. As the occupancy grid is updated with data from a live scan, it contains information on any unknown objects. If interference is detected by the Task Analysis
3 Methodology

module, Path Planning can be performed.

To examine the Path Planning a comparison of the occupancy grid cells populated with different datasets was made. There are three useful combinations of simulation and scan data; simulation data only, scan data only and both simulation and scan data. The resulting cells were compared between the datasets demonstrating the ability for each dataset to distinguish between cells occupied by simulation objects, the robot and unknown objects. This also demonstrates that the combination of simulation and scan data can take into account gaps in scan data.

Path Planning for the final process was used when the task had interference where a new path was generated around a dynamic obstacle. The Path Planning module used an A* [21] search to create a path around an obstacle during the operation of the task.

3.6 Methodology Summary

Several contributions were identified from an examination of current literature. These contributions are related to areas of Environment Analysis, Task Analysis and Path Planning.

A simulation was used for background subtraction to interpret scan data for Environment Analysis. The ability to match simulated objects with scan data validates the position and orientation of the known (simulated) objects. The use of an occupancy grid for the matching process simplifies and improves the processing speed. Any areas of a scan that are left over after the simulation is subtracted can be isolated and identified as unknown objects (as opposed to the known simulation objects). Any scan data that conflicts with the simulation demonstrates that the simulation itself is
3 Methodology

incorrect, such as where simulated objects are missing from the real environment. The surface description used for the scan and simulation has a significant effect on the capabilities of the system including the growth algorithm to detect unknown objects and the detection of objects missing from the environment.

For Task Analysis the use of collision detection between movement hulls and isolated objects has been developed. This analysis determines if tasks have interference and need to be modified so that they can be correctly completed.

The occupancy grid approach was evaluated using three different datasets including scan only data, simulation only data and combined simulation and scan data. Comparing the resulting occupancy grids it is possible to distinguish between areas of the scan that relate to known simulation objects, the robot being controlled or unknown objects. This also demonstrates that the combination of simulation and scan data can take into account gaps in scan data.

These contributions have led to the development of the simulation scan comparison (SSC) process. This process can be tested for the combined capabilities using experimental methods that combine both simulation and real environments.
4 Experimental Design

The contributions of the research described in this thesis are 3D analysis algorithms that are designed to analyse a manufacturing task while in operation. This analysis is performed in three stages, Environment Analysis, Task Analysis and Path Planning. These stages have been combined to produce a single process referred to as the Simulation Scan Comparison (SSC) process. This process is intended to reduce or eliminate damage caused by interference with unknown or unexpected conditions in a manufacturing environment, such as extra unknown objects not present in a design simulation or missing objects.

To test the SSC process, a physical environment was used. A diagram of the environment is shown in Figure 4.1 and the physical environment is shown in Figure 4.2. The experiment environment included a manipulator robot and a safety cage. Other objects were added to the work area as required. The scanner was mounted on the top of the safety cage with a full view of the main work area of the robot.

![Diagram of the robot work-cell with scanner](image)

*Figure 4.1: A diagram of the robot work-cell with scanner used for testing*
4 Experimental Design

The manipulator used for these experiments was an IRB 140 made by ABB. The IRB 140 was connected to an IRC 5 controller which handles the path computations and control of the robot. The robot used a gripper from Robotiq as an end effector.

![Figure 4.2 The physical environment](image)

4.1 Experiment Overview

The environment shown in Figure 4.1 was used to test the capabilities of a software implementation of the Simulation Scan Comparison (SSC) process. The SSC process used a 3D simulation to compare with a 3D scan to perform Environmental Analysis, Task Analysis and Path Planning. These experiments are intended to determine if the SSC process is capable of detecting unknown or unexpected conditions in a
manufacturing cell and perform a reroute of a designed path if interference with the current path is detected.

### 4.1.1 Environment Analysis Overview

The Environmental Analysis is performed by subtracting the simulation input from the scan data. As described in section 3.2 the background subtraction allows objects to be identified in three different categories:

- **Known objects**
- **Missing objects**
- **Unknown objects**

Known objects either exist in the correct position and match correctly to scan data or they can't be confirmed as no scan data matches. Missing objects can either be confirmed that they are not in the scene as only conflicting scan data is found or they can't be confirmed as there is no matching scan data. Unknown objects will only be detected after the subtraction process if they have not been occluded by other objects and small objects may not be detected.

The Environmental Analysis allows the SSC process to confirm that all of the objects required for the simulation are present. It will also determine if any unexpected conditions exist in the environment, such as missing objects or detecting the existence of any unknown objects.
4 Experimental Design

4.1.1 Matching Complexity

As described in sections 3.2.4 and 3.2.5 the matching process of the simulation is heavily influenced by the size of the occupancy grid. To determine the effect of the occupancy grid size on matching performance, two sets of models were matched with several scans using varying sizes of the occupancy grid. These sizes were compared with the time that the matching algorithm took to complete.

4.1.1.2 Known object testing

Known objects are the objects specified in the simulation. If simulation objects are matched correctly to scan data, their positions are confirmed. To test the validity of known object matching, objects were added to the simulation and correctly positioned in the physical environment. The matching process was then run on scans of the physical environment to determine the robustness and speed of the matching process.

4.1.1.3 Missing object testing

The scan data can conflict with the simulation. Conflicting, in this case, means that the scan and simulation data share the same position (i.e. close to an average projected distance of 0.0) and at the same time the scan surface has a normal in the opposite direction of the simulation surface. If scan data is in conflict with simulation data it suggests that the simulation object is missing from the environment. To test the validity of missing object detections, objects were added to the simulation but not in the real world. The matching process was expected to find scan data in conflict with the simulation object and determine that the object was missing.
4.1.1.4 Unknown Object testing

Unknown objects are isolated by grouping together areas of a scan having no matching or conflicting data. To test the validity of unknown object detections, objects were added to the real world, but not to the simulation. By subtracting known objects from the scan data, the remaining areas of the scan can be grouped using a region growing algorithm and each separate area classified as an unknown object. The size of unknown objects can be measured using convex hulls and the rotating callipers algorithm, which produced an oriented bounding box for each unknown object. The position of the unknown objects can also be approximated based on the central point of the grouped scan data.

4.1.2 Task Analysis Overview

The main purpose of Task Analysis is to determine if it is possible to complete a specified task based on both the simulation data and the current state of the environment. There were several possibilities to demonstrate Task Analysis. In general terms these tasks could have been either gross motion, such as path planning, or fine motion, such as tool positioning. The task demonstration chosen was a gross motion task, to examine the pre-programmed path of a manipulator, as this was more appropriate for the capabilities of the depth sensor used.

A collision detection algorithm compares the movement hull of the robot as it moves along its path and the convex hull of any unknown object nearby. This required the development of software to create and compare the path of a physical robot. A movement hull was created from the initial and final segments of the movement and a
convex hull created for each of the robot's components. Unknown objects detected by the environment analysis could then be compared to the movement hull using a collision detection algorithm. If the collision detection algorithm showed that an object interfered with the movement, path planning could be used to reroute the movement.

4.1.3 Path Planning Overview

There were also several options for the type of feedback to provide to the system including: assembly tasks, cutting tasks, updating object positions, selecting alternate tasks, part damage inspection and rerouting path planning. The feedback provided for a task is dependent on the type of task. Hence, path planning was chosen as a demonstration of the feedback capabilities for movement tasks as generating paths is widely used (for example pick and place operations) and could be completed using the occupancy grid structure of the matching process. The other tasks require more specific analysis to determine if they can be completed, such as fine alignment measurement which was beyond the capabilities of the depth scanner.

The occupancy grid was analysed using several sets of data: scan data only, simulation data only and combined simulation and scan data. By using these datasets to populate an occupancy grid it is possible to show the different areas of coverage and the limitations of each dataset. From these results it is possible to show which combinations of data can be used to discriminate between cells occupied by fixed objects and which contain the robot, while still maintaining the capability of avoiding unknown objects.

To test the path planning the test environment with the robot (as shown in Figure 4.1)
Experimental Design

was used. A simple simulation of the robot moving along a simple path was created. An unknown object was deliberately placed in the path of the robot. The simulation and scan were analysed using both the Environment and Task Analysis stages. When the Task Analysis detected the interference, the path planner generated a path around the obstacle.

4.1.4 Complete System Testing

The Environmental Analysis, Task Analysis and Path Planning components of the Simulation Scan Comparison process (SSC) were combined in a software system. The final software system was expected to be able to perform all of the tasks of the previous experiments in combination. To test the entire system it was used with a physical robot in a test environment similar to a manufacturing system. As described in the Section 3.1 it is expected that the final process would be able to:

- Run the robot through a task while monitoring the environment.
- Identify objects that were not specified in the 3D simulation for the manufacturing process.
- Determine if any of these unknown objects presented a problem with a current task.
- Either stop the task from continuing or plan a reroute path around an obstacle if an obstacle was determined to be interfering.

4.1.5 Size Measurements

As there was very little research found in the literature considering the concept of object
Experimental Design

isolation, it was felt that this concept required rigorous testing. As such, several sets of experiments were performed to test the robustness of the new capabilities. These tests were used to determine how reliable the unknown object detection algorithms from the SSC process were. The two types of tests were aimed at determining the size of unknown objects compared to real measurements and object tracking.

The size measurements of objects were based on the oriented bounding box created from areas of triangles isolated from the scan. The experiments were aimed at determining how accurately the isolation process could estimate the overall size of objects from scan data. These measurements provide a basis to consider how unknown objects could interact with a real system. The size measurement testing required creating 3D scans of an environment at different resolutions. Several objects of known size were added to the environment for scanning. The resulting detections of these objects and their measured sizes were compared to the real size of the objects at each resolution. The comparisons were used to determine how accurately the environment analysis could determine each objects size. Also experiments were carried out to determine the minimum distance between objects that would allow two objects to be detected as separate objects rather than a single larger object.

4.1.6 Tracking

The tracking experiments were performed to determine if it was possible to use the isolated objects and their associated measurements for tracking. It was expected that the use of a simulation for background subtraction and unknown object isolation could be used to measure and track unknown objects moving in a real environment.
4 Experimental Design

While size measurements determine how accurately objects are described in space, tracking can be used to determine the position of objects over time. To test the tracking capabilities of the Environment Analysis module, one or more unknown objects were added to a scene and moved around to determine how well tracking could be performed. Several different tracking algorithms were developed and the performance of each was compared. The tracking testing demonstrated the robustness of the Environment Analysis and assists in improving the reliability of the Task Analysis module.

4.2 Software Design

The SSC process required two internal components and four external components to be integrated. Experiments were designed to test both the individual components and the overall operation and interaction between the components.

The internal components were:

- The implementation of the SSC process written in C++
- A C# User Interface integrated into Robot Studio

The external components were:

- Robot Studio API
- ABB's PCSDK API for communicating with the robot directly
- RAPID Code specialised for handling interrupts and rerouting on the robot controller
- The Kinect (TM) scanner

The overall process is described in section 3.5 which provides a design diagram of the SSC process. As shown in Figure 4.3, the implementation of this design required
4 Experimental Design

several other external components to be considered including a physical robot (IRB 140), the robot controller (IRC5) which executed RAPID code, the scanner (a Microsoft Kinect™) and Robot Studio including the PCSDK for communication with the robot controller. Testing of the algorithms required the various software components to be developed, integrated and executed with a real environment.

Figure 4.3: System implementation
4 Experimental Design

4.2.1 SSC Implementation

The internal components were the SSC process implementation and the interface to Robot Studio and the PCSDK. The main SSC process was written in C++ using various multi-threading capabilities and optimised for performance.

4.2.2 User Interface

The second internal component was the user interface, which was embedded into the Robot Studio software, where a user could activate and run the software. This was written in C#. The API for Robot Studio allows extensions to be developed and integrated into the main program, as such, an “Add-on” was developed that utilised the features of Robot Studio to download the current simulation loaded in Robot Studio and at the same time communicate with the physical robot. The user interface handled the interface between the simulation environment and the physical robot. This included activating interrupts on the physical robot and uploading the reroute path.

4.2.3 Robot Studio API

The Robot Studio API system allows developers to create “Add on” programs that can access the internal workings of the Robot Studio software including any simulation developed on the software. The Robot Studio API was integrated into the user interface and used for downloading the simulation of the environment including the path into the SSC implementation.

The use of the Robot Studio API allowed simulations to be developed for these experiments with Robot Studio as described in section 3.2.1 and downloaded directly
into the SSC software.

### 4.2.4 Robot Studio PC SDK

ABB's PCSDK API was also integrated into the user interface and used for communicating with the robot via the controller to determine its current position in the scene and to modify the code for rerouting the robot on the robot controller.

### 4.2.5 RAPID and Robot Controller

RAPID is an ABB proprietary language used for programming its robots. A RAPID code module was developed for handling the rerouting and several signals were added to the controller to allow interrupts to be sent between the controller and the SSC implementation. To be able to interrupt the robot required the integration of a digital input that could be used for an interrupt and specialised RAPID code to handle updating a reroute point. Pseudo-code for the RAPID implementation is shown in Figure 4.4.

### 4.2.6 Kinect Scanner

The Microsoft Kinect™ was used to scan the environment. The Kinect™ uses an infra-red structured light projection system, along with embedded image analysis software to produce a depth map of an area at up to 30 Hz. This was integrated directly into the internal implementation of the SSC as this produced the lowest latency and highest processing throughput.
4.2.7 Software Operation

The sequence of events run by the software would be as follows:

Initialisation:

1. Robot Studio activated and a simulation is created or loaded, including a planned path.
2. Settings for the sensor position and path planning configuration cells are loaded.
3. The SSC user interface is activated and the simulation is uploaded.
4. A first pass scan to check the initial state of the environment is performed.

```plaintext
MODULE RerouteModule
    Robot Target declaration ReroutePoint
TRAP Path_ReRoute
    MoveL to RerouteTarget
ENDTRAP
PROC SetupReroute()
    Create interrupts for Path_ReRoute
ENDPROC
END MODULE

MODULE Main Module
Robot Target declaration for Path_10 points (Target_10, Target_20)

PROC Path_10()
    MoveL to each point (Target_10, Target_20)
ENDPROC

PROC Main()
    SetupReroute;
    Path_10;
ENDPROC
END MODULE
```

Figure 4.4: RAPID Pseudo-code implemented on the IRC5 Robot Controller
4 Experimental Design

5. Once the initial state of the environment is found to be safe the system can be started.

Once the initialisation is complete, continuous scanning of the environment can begin.

1. For every scan the simulation is updated with the current position of the robot.

2. A scan is collected and both the current simulation and the scan are mapped into an occupancy grid and the simulation subtracted from (matched with) the scan.

3. Any remaining areas are grouped together to form unknown objects.

4. The unknown objects are checked against the planned movement of the robot for interference.

5. If an object is found to be interfering, a reroute path is generated.

6. If the reroute path is successfully generated the robot task is stopped to allow the RAPID code to be edited and the reroute point is updated to the first point of the reroute path.

7. The robot task is then reactivated and a signal sent to the controller to activate the reroute interrupt “Path_ReRoute”.

8. The robot then moves to the reroute point and the system rescans the environment and the procedure begins again.

Only the first reroute point is used as the obstacles may be dynamic. As such, a subsequent frame may show that the obstacle has moved. If the object moves slightly or not sufficiently to get out of the path of the robot a completely new reroute path may be required. If the obstruction has not moved, the same reroute path (or at least one very close) will be calculated and the robot progressively moves around the object and will continue on the main path (Path_10 from Figure 4.4). If the object has completely
moved out of the way, the reroute is no longer required. In this case the robot would simply continue to move to the next point in the pre-planned simulation path.

### 4.2.8 Simulation

The simulations developed for the experiments included the robot, the safety cage and the path the robot followed. A few other small objects, such as boxes, were created in simulation and added to either the real environment or simulated scene as required. The task created was a simple path where the robot moved from one location to another located 400 mm away, then returned to the initial position.

The distance and position of the path points were limited due to the control constraints of the real ABB robot. These control constraints are that the maximum rotation between any two joint positions in a path must be less than 45 degrees. The initial and final points could not be changed by the rerouting process as the path sequence could not be modified in RAPID, only the position of the points. Once the simulations were created they could be implemented in the real environment.

### 4.2.9 Scanner

To scan the real environment a Depth camera (Microsoft Kinect\textsuperscript{TM}) was installed at the top of the safety cage as shown in Figure 4.1. Depth cameras were found to be the most suitable technology for acquiring a 3D scan of an environment rapidly enough to detect dynamic objects in a scene. Depth cameras in general have many of the specifications required for scanning a dynamic environment, including speed, resolution and simple data structures. The limitation with depth cameras is that they have limited
4 Experimental Design

depth resolution compared to other technologies. The main useful feature of depth cameras is that many are capable of scanning a scene at a rate of 30 Hz. The Kinect™ was used, as it is low cost, readily available and provides the required depth data at sufficient resolution to detect objects sized at 60 mm or larger and can run at up to 30 Hz. Irrespective of how fast the scan is, to match objects, the simulation and scan need to be aligned correctly.

4.2.10 Alignment

All three components (simulation, scan and robot) need to share the same coordinate system so that when unknown objects are detected they can be described with the same coordinates. To accomplish this, a set of control points (small black dots) were placed in the scene and five of these points were measured with the robot by positioning the end effector above and as close to the centre of each point. This allowed the robot to be used as the origin for the rest of the points in the scene. Other points were placed around the scene outside the range of robot arm movement. The extra points were measured using the 3DM Analyst Photogrammetry Software (ADAM Technology) using multiple images of the scene from different directions. Each point was defined from the 3D coordinates and were accurate to within 1 mm. To align the Kinect™ scanner, it was fixed to the safety cage in its final position oriented to include the control points. Several images were taken from the scanner and the camera position was determined using the 3DM Analyst photogrammetry suite which provides the camera position and orientation for each image used to determine the extra control points. Finally, the simulations developed in ABB Robot Studio utilised the robot as the
4 Experimental Design

main coordinate system of the environment i.e. the robot was positioned at the origin. With all three components aligned to the same coordinate system the final software was able to display the simulation and scan overlaid in three dimensions. The simulation was able to be updated from receiving inputs from the physical robot via Robot Studio. Thus when the physical robot moved, the simulated robot would also be updated to correctly match the configuration of the physical robot.

4.3 Environment Analysis Experiments

To test the effectiveness of the SSC process, it was run on a series of scenarios in a robotic work-cell. The environment included an ABB IRB140 robot in the layout shown in Figure 4.1. The simulation data was developed in Robot Studio using the standard models created by ABB. Several simple geometric objects, including the safety cage, were developed manually. An add-in to Robot Studio was created to allow the program to access the Robot Studio simulation. The scan data was provided by a Kinect\textsuperscript{TM} mounted on top of the safety cage. With a 1.2 m distance from the Kinect\textsuperscript{TM} to the scene the average scan resolution (R) was found to be approximately 20 mm. The objects in the scene included the robot, several boxes that were included in the simulation and several unknown objects that were moved around the scene for various tests.

The matching process was analysed by comparing two sets of scans with associated simulations and varying the cell size to determine the difference in the matching times for the different cell sizes. The first simulations had 7300 triangles and the other 23261 triangles. The simulations were matched with sets of scans (39 for the first simulation
4 Experimental Design

and 41 for the second) and the average time for performing the matching, in seconds, was measured against the size and total number of cells.

The overall performance of the matching process was tested with three scenarios:

- A working system with no faults, but a known object in multiple positions.
- An example of a working system with an object missing from the scenario.
- An example of a working system with an “unknown” object in the scene.

Each of these experiments used the same set of scans with simulation objects added or removed as required for the appropriate test.

An overview of one of the scenarios and results are shown in Figure 4.5. This scenario included the robot, its safety cage and three objects positioned around the scene. For this application, the accuracies and resolutions are aimed at detecting objects ranging in size from 60 mm up to 400 mm with a work area of up to two by two meters. The simulation objects were inserted into an occupancy grid of 80 mm cells and compared to scans of the area.

Figure 4.5: A screenshot with a simulation in the left and SSC results in the right.
4 Experimental Design

4.3.1 Known Object Testing

The robot was used as the main set of known objects, as this could be moved around the scene with a known position. A total of 1473 scan frames were captured with the robot in view in different positions. Several sequences of scans were recorded with between 10 and 40 frames captured for each run. These scans were matched with the simulation of the robot and often with extra objects in the environment. This allows the SSC process to determine how effectively the simulation could be matched with scan data to determine object status. The states of the objects could be termed correct, missing, not-matched or outside the environment.

Each scan was analysed to determine what each triangle matched to. The triangles could be filtered out before matching, correctly matched to a simulation object, not matched to anything or conflicting with a simulation object. The objects were correct if they matched correctly (green wire-frame) to scan data in their current position to within a matching tolerance of 50 mm. The objects were considered missing (red wire-frame) if scan data conflicted with the simulation data, i.e. the scan data was in the same position, but with the opposite normal. If no scan data matched to a simulated object due to occlusion or being just outside the scan volume, the not-matched state was used (magenta wire-frame). Finally it was possible for simulation objects to be added outside the scan area (blue wire-frame) in which case this could be detected and the object state set to outside environment.

4.3.2 Missing Object Testing

Simulation objects could be orientated incorrectly by as much as allowed by the
matching tolerance. If the surfaces of the simulation objects match with the scan, but had conflicting normals, they were identified as missing from the environment. Any areas of a scan that were not matched to an object and not in the simulation were isolated and identified as unknown objects. Missing objects could be tested simply by adding an object to the simulation, without adding it to the real environment. As such, all of the previous 1473 scan could be reused in an off-line mode to detect the missing objects. The missing object testing verified that missing objects are not confused with known objects that are correctly matched or with any unknown objects in the scene.

4.3.3 Unknown Object Testing

A total of 1473 frames were collected with unknown objects in various positions and of various sizes. Several sequences of scans were recorded with between 10 and 40 frames captured for each scenario. Approximately 10 frames were captured where objects were moving quickly and approximately 40 frames were captured if objects were moving slowly. Each frame was analysed individually to determine if there were any unknown objects in the environment, where the objects were and generate an oriented bounding box around the object to determine a measurement of its size in three dimensions. These were designated as the maximum, intermediate and minimum axes.

4.4 Task Analysis Experiments

To test the Task Analysis module, a short repeating path was developed for the physical robot and a 3D simulation of the task created. From this task a convex hull of the volume enclosed by the robot during the movement between two path points is created.
4 Experimental Design

A convex hull of each component of the robot is created by considering the robot in both the initial and final configurations of the movement. Using the GJK algorithm [88], the movement hull of the path was compared to any unknown object detected nearby in the scene. If any unknown object was determined to be colliding with the movement hull, the path planning for the task could be activated. Once one object is identified as interfering with a movement any remaining components of the robot no longer need to be examined. Any further obstructions will be accounted for in the Path Planning stage. It is expected that this technique can be used to determine if a physical robot’s planned actions will be interfered with by any unknown objects in an environment and allow Path Planning to be performed.

The Task Analysis could be tested with individual frames where the robot path existed in the scene and in the complete system during continuous online operation. An unknown object could be moved into and out of the path of the robot and the system was expected be able to determine which frames had an obstruction as they were scanned.

A total of 387 frames were used in the Task Analysis testing. These were captured in sets of 10 to 65 frames. During each run the robot was moved along the simulated path and an unknown object was introduced into the environment to determine if the SSC process could detect if the unknown object interfered with the simulation path. The same frames that were captured live could be re-run in an off-line mode where the simulation could be modified and the scan rechecked to determine if any objects were missing or still interfered with the altered simulation path.
4 Experimental Design

4.5 Path Planning Experiments

The Path Planning experiments were aimed at demonstrating the SSC process in a simple situation that can occur in real manufacturing environments. As described in section 3.4 the SSC process utilises both simulation and scan data. As such, the path planning experiments included a comparison between the cells occupied by scan data only, simulation data only and combined simulation and scan data. The combinations of data sources results in different occupied cells in the final grid. This shows that the simulation data in the occupancy grid can be used to distinguish between fixed objects and the robot. In combination with scan data the comparison is aimed at demonstrating that the simulation data fills many gaps in the sensor data.

In the complete testing, the manipulator is required to perform a simple movement operation simulating a pick and place operation where there is an obstruction between the pick-up and drop off locations. It is expected that the system will automatically attempt to create a reroute path for the manipulator to avoid the unknown obstruction if the Task Analysis detects an unknown object interfering with the original simulation path. If either of the end points were obstructed, the Path Planning will fail and the robot is expected to stop. If and when the obstruction is removed, the task will be expected to continue. The frames used with Task Analysis were re-used with off-line testing to determine if the system could determine an appropriate path with a modified simulation. The Path Planning was based on an A* algorithm, which is not usually used with manipulators. To allow the manipulator to be used with the A* algorithm each of the occupancy grid cells was assigned a set of robot configurations (joint angles), which were created in Robot Studio. The A* algorithm would select the appropriate set of
4 Experimental Design

cells to find a path around an obstruction and look up the appropriate configuration from those available in the cell. In this way, once a cell is occupied the A* algorithm will not have access to any of the configurations associated with a cell.

4.6 SSC Process Experiments

The complete system experiments were aimed at examining the entire process from Environment Analysis to Task Analysis and Path Planning. As such, the physical robot was expected to perform a simple task. An object not included in the simulation was moved into the path of the robot while the robot was moving. When the SSC process detected interference, it was expected to either stop or reroute the robot. When the unknown object is removed from the scene, the robot should continue the task. These experiments were performed in the live mode to determine if the entire system would operate as expected.

4.7 Object Size Measurements

The SSC process is capable of identifying unknown objects from scan data by subtracting known simulation objects from the scan. The resulting areas of the scan are grouped into unknown objects which can be measured using an oriented bounding box method. The oriented bounding box will produce three measurements, which can be sorted into the maximum, intermediate and minimum axes. The objective of these experiments is to determine how close the bounding box measurements are to the size of the real objects.

It was expected that if objects were smaller than a certain size they would not be
detected. Given a scanner with a known resolution $R$, where $R$ is the average distance between points of a scan. It was expected that unknown objects of less than three resolutions (3$R$) are often not detected at all. At a size of 2$R$ or less, an object will often appear to be a spike in the 3D scan or sensor noise. At 3$R$ or more, objects can be successfully isolated. With the scenarios tested, the resolution was 20 mm and hence objects of size 60 mm or less will not be expected to be detected.

The size of unknown objects is important for collision avoidance as the size measurement is used to flag areas that are occupied and hence need to be avoided by path planning algorithms. The experimentation completed here provides a validation of the size measurement of isolated objects. Given that this system can use a single camera system, the scan will only show the faces on the scanner side, the back of the object will, obviously, not be detected. Without 100% coverage of an object it is difficult to be sure that the measurements made will be reliable, hence the need to determine how close the hull measurements are to real object sizes. These experiments were performed using ADAM technology's 3DM Analyst photogrammetry software package. The advantage of using this software is that it has the capability of producing 3D models with progressively smaller resolutions from the same set of images.

As shown in Figure 4.7, using a Ge 1250 Camera at 4000 x 3000 pixels image resolution and at a range of 1 m, an expected ground resolution of 0.24 mm per pixel is achieved. Scaled images could then be used to simulate the other resolutions. This experiment is intended to determine the resolution value at which an object fails to be detected correctly, thus establishing a guideline for the resolution required for any project using 3D scanning.
Using a cubic object of 30 mm sides at 0.24 mm resolution produces an object in the images of approximately 125 pixels. Based on the ADAM Technology proprietary algorithms the triangle density is 1 triangle per 4 pixels, meaning the object appeared to be at most 32 triangles wide.

Sets of images with the sample resolutions from Table 4.1 were generated accordingly. The columns show the resolution of the scan in both pixels (Pixel size) and average triangle size (Scan resolution, R) and the expected object size again in both pixels and triangles. This shows that progressively larger pixel sizes produce larger filter sizes, which will approach the size of any objects. It is expected that near the point where the filter size is the same as any object, the detection algorithm will completely fail. As such, by model 6 the objects are not expected to be detected. The experiments here used small textured blocks with dimensions of 15, 30 or 60 mm. Blocks would be

---

**Figure 4.6: Camera accuracy spreadsheet**

Using a cubic object of 30 mm sides at 0.24 mm resolution produces an object in the images of approximately 125 pixels. Based on the ADAM Technology proprietary algorithms the triangle density is 1 triangle per 4 pixels, meaning the object appeared to be at most 32 triangles wide.

Sets of images with the sample resolutions from Table 4.1 were generated accordingly. The columns show the resolution of the scan in both pixels (Pixel size) and average triangle size (Scan resolution, R) and the expected object size again in both pixels and triangles. This shows that progressively larger pixel sizes produce larger filter sizes, which will approach the size of any objects. It is expected that near the point where the filter size is the same as any object, the detection algorithm will completely fail. As such, by model 6 the objects are not expected to be detected. The experiments here used small textured blocks with dimensions of 15, 30 or 60 mm. Blocks would be
4 Experimental Design

cubic at 30mm, i.e., all sides 30mm, or rectangular with dimensions either 30mm by 30mm by 60mm or 15mm by 30mm by 60mm.

Table 4.1: A sample of object resolution tests

<table>
<thead>
<tr>
<th>Model No</th>
<th>Pixel Size (mm)</th>
<th>Scan Resolution (mm)</th>
<th>Object size (Pixels)</th>
<th>Object Size (Triangles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.24</td>
<td>0.96</td>
<td>125</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>0.33</td>
<td>1.31</td>
<td>91</td>
<td>23</td>
</tr>
<tr>
<td>3</td>
<td>0.49</td>
<td>1.92</td>
<td>62</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>0.97</td>
<td>3.84</td>
<td>31</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>1.94</td>
<td>7.68</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>3.84</td>
<td>15.36</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>

4.7.1 Object Separation Detection

If two “unknown” objects of similar dimensions are positioned right next to each other, it is always difficult to be able to separate them. An experiment was developed to test how far apart two unknown objects needed to be separated before they were successfully detected as separate objects. This required setting up two objects of similar size next to each other. As models were captured, the two objects were separated by small increments to determine when the two objects would be detected separately.

4.8 Object Tracking

Object tracking experiments were performed using the data output from the SSC process. These were intended to determine the robustness of the unknown object detection components of the SSC over time. The simplest form of tracking developed considered that an object would not move far between frames. The collision detection algorithm was used to detect overlapping shapes between frames. There were two main
4 Experimental Design

techniques considered for tracking:

1. Position based tracking - objects were considered static

2. Velocity based tracking - objects were considered to have a constant velocity

Both of these techniques were tested using the same environment as previous tests (shown in Figure 4.1). Several unknown objects were added to the scene and moved while continuously scanning the environment at an average rate of 4 Hz. The unknown objects were detected using the SSC process and the resulting detections were tracked with each algorithm and the performance of the different methods compared.

4.8.1 Position Based Tracking

Position based tracking is run on the assumption that the object has minimal velocity and will remain visible in the scene. The ability of the position based tracking algorithm to track individual objects can be expressed as shown below:

\[ V_{\text{object}} < \frac{S_{xyz}}{T_{\text{Scan}}} \]

Where \( V_{\text{object}} \) is the velocity of the object, \( S_{xyz} \) is the size of the object in any three dimensions and \( T_{\text{Scan}} \) is the scan rate of the sensor.

This suggests that it is possible to track an object if it is moving at less than its own size between frames in the scene. In practical terms an object 0.1 m in size being scanned at 10 frames /second will be able to move as fast as 1 m/s before the position tracking will fail.
4.8.2 Velocity Update Tracking

An extension of the position based tracking algorithm is Velocity Tracking. This algorithm is based on multiple sightings of a single object. The first two detections are connected initially with position based tracking or object recognition. Using at least two frames a, velocity estimation can be made. The velocity estimation can then be used to update the expected position of an object in the next frame and again use the collision detection algorithm to confirm the track.

4.9 Summary

The experiments were aimed at testing the different capabilities of the SSC process. The capabilities of the SSC process were focused on Environmental Analysis, Task Analysis and Path Planning.

The Environmental Analysis allows the process to:

- Match existing objects to a simulation
- Detect missing objects
- Identify unknown objects

The Task Analysis experiments were performed to determine if this data could be used to determine if there is interference with a pre-existing task or path.

A comparative analysis was performed using an occupancy grid populated with simulation data, scan data and a combination of both datasets. This was aimed at demonstrating the differences in the resulting occupied cells and demonstrates that simulation data can cover gaps in sensor data and can be used to distinguish between
fixed objects and the robot being controlled.

Path Planning experiments were performed to determine that if interference was detected with a path, that a new path could be generated. These experiments were aimed at generating a new path around the object interfering with the current path of a manipulator.

Experiments were performed to determine how well the unknown object detection system can detect objects by examining how accurate the size measurements of the unknown objects are.

Experiments were also performed to determine how well the unknown object data can track the unknown objects over time, as they move in the scanned environment.
5 Results

This section presents and summarises the results for the various experiments performed to evaluate the Simulation Scan Comparison (SSC) process, as described in Chapter 4. The SSC process has three main components Environmental Analysis, Task Analysis and Path Planning. Environmental Analysis compares a 3D simulation of an environment with a scan of the physical environment to detect known, missing and unknown objects. Task Analysis is based around detecting if a movement task can be completed or if it has interference from an unknown object. Finally the Path Planning component was designed to find a reroute path around an obstruction detected by the task analysis.

5.1 Environment Analysis Experiments Results

The Environmental Analysis experiments were aimed at demonstrating the capabilities developed from using a 3D simulation for background subtraction. The background subtraction uses an occupancy grid to match a simulation against scan data. The complexity of the matching process has been shown to be related to the size of the cells used. The smaller the cells used the more cells there will be in the occupancy grid, but the smaller the cells the fewer scan and simulation triangles in each cell, hence the fewer comparisons between triangles overall. The environment simulated is shown in Figure 4.1 of the experimental design, with dimensions of 1.8 m by 1.8 m by 1.2 m. Table 5.1 shows the data collected for two different simulations. The cell size varied from 1 m down to 0.08 m which resulted in between 18 and 8464 cells in the occupancy grid. One simulation had only 7300 triangles in all of the objects used and the other had
5 Results

a total of 23261 triangles. Both simulations were matched with a set of scans and the average time to perform the matching is presented in Table 5.1, along with the cell size and the resulting number of cells in the environment. Figure 5.1 shows a graph of the two simulations against the cell size and the resulting time the matching algorithm took to complete. These tests were performed on an Intel Core i7-4770K @3.5 GHz.

As described in Section 3.2, the background subtraction allows objects to be identified in three different categories:

- Known objects
- Missing objects
- Unknown objects

Known objects are simulation objects, they are correctly identified when scan data matches with the simulation data, in both projected distance and in surface normal direction. Missing objects are detected when the scan data conflicts with the simulation data. A conflict between simulation and scan data occurs when the distance between a scan triangle projected onto a simulation triangle is less than the matching tolerance, but the surface normal is in the opposite direction. Unknown objects are detected when areas of the scan do not match to any simulation data.

The Environmental Analysis experiments required a physical environment to be used which was also reconstructed in simulation using a standard industrial simulation software package, Robot Studio. This environment was then scanned with a fixed position scanner and the simulation and scan were compared using the SSC implementation software, as discussed in sections 4.1, 4.2, and 4.3.
5 Results

Table 5.1: Results of matching times with varying occupancy grid cell size

<table>
<thead>
<tr>
<th>Cell size (m)</th>
<th>Cell Count</th>
<th>Matching Time (s)</th>
<th>9322 Scan triangles</th>
<th>23261 Simulation triangles</th>
<th>7300 Simulation triangles</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>1</td>
<td>1.076</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td>2</td>
<td>1.059</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td>2</td>
<td>1.082</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>2</td>
<td>1.031</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>2</td>
<td>1.032</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1.036</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.9</td>
<td>2</td>
<td>1.036</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.8</td>
<td>2</td>
<td>1.033</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.7</td>
<td>18</td>
<td>1.025</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.6</td>
<td>18</td>
<td>1.03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5</td>
<td>18</td>
<td>1.164</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.4</td>
<td>18</td>
<td>1.02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>18</td>
<td>1.012</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>18</td>
<td>0.996</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>18</td>
<td>0.998</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>18</td>
<td>1.023</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.90</td>
<td>18</td>
<td>1.083</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.80</td>
<td>27</td>
<td>1.225</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.70</td>
<td>27</td>
<td>1.155</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.60</td>
<td>27</td>
<td>1.162</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>75</td>
<td>0.977</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.40</td>
<td>100</td>
<td>0.799</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.30</td>
<td>245</td>
<td>0.69</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.20</td>
<td>567</td>
<td>0.538</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.10</td>
<td>4693</td>
<td>0.269</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.08</td>
<td>8464</td>
<td>0.222</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

125
5 Results

Figure 5.2 shows the Robot Studio simulation on the left and the SSC results are shown on the right of the image. Several objects have been identified as part of the scan, including the components of the robot, the base of the safety cage and a small box placed in front of the robot. In the SSC results on the right, the green lines are objects from the simulation that have been correctly matched with the scan. The textured area of the scan shows the actual scene at the time of the scan.

Figure 5.1: Matching time vs cell size graph

Figure 5.2: Simulated objects matched to 3D scan
5 Results

Table 5.2 shows the results from the 1473 frames collected and analysed for the three evaluation scenarios: only simulation objects in the scene, missing objects in the scene and unknown objects in the scene. These scenarios were analysed to determine if the SSC process correctly matched the intended situation to the actual environment.

*Table 5.2: Results of SSC process matching*

<table>
<thead>
<tr>
<th></th>
<th>Simulation</th>
<th>Missing</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctly matched frames</td>
<td>1457 (98.9%)</td>
<td>1457 (98.9%)</td>
<td>1323 (89.8%)</td>
</tr>
<tr>
<td>Incorrectly matched frames</td>
<td>16 (1.1%)</td>
<td>16 (1.1%)</td>
<td>150 (10.2%)</td>
</tr>
<tr>
<td>Total</td>
<td>1473 (100%)</td>
<td>1473 (100%)</td>
<td>1473 (100%)</td>
</tr>
</tbody>
</table>

Table 5.3 shows the reasons for the discrepancies in the frames specifically the “Incorrectly matched frames” row from Table 5.2 and explains the reasons for these errors. By manually examining each frame, where the discrepancy occurred, it was possible to determine one of three reasons that objects were not correctly identified. The errors were either due to occlusion of the object, the object being too small to correctly detect or the object being outside the area of the scan.

Any object (simulated or unknown) that was occluded by another object in the scene, would result in no scan data either matching or conflicting with a simulation object. As such, it is not possible to identify the simulation object as being in the scene or not.

When simulation objects moved outside the scan area, they were not matched with any scan data. Unknown objects that moved outside the scan area could not be isolated from scan data.
Table 5.3: Errors in SSC process matching

<table>
<thead>
<tr>
<th>Reasons for errors</th>
<th>Simulation</th>
<th>Missing</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object occluded</td>
<td>10</td>
<td>10</td>
<td>75</td>
</tr>
<tr>
<td>Object too small</td>
<td>0</td>
<td>0</td>
<td>70</td>
</tr>
<tr>
<td>Object outside scan</td>
<td>6</td>
<td>6</td>
<td>95</td>
</tr>
</tbody>
</table>

If an unknown object was too small it would be impossible for the software to isolate enough of an area of the scan to create a bounding box. This could be due to both small objects and occasionally reflections from lighting causing a depth error on a small number of depth pixels which would be filtered out either by the Kinect (TM) itself or the scan filtering process. Multiple errors could occur in a single frame and as such, the totals in the “reasons for errors” will not necessarily be the same as “incorrectly identified” frames.

### 5.1.1 Known Objects

The SSC process was evaluated by analysing how well it matched the simulation and scan. In this case known objects were added to both the physical environment to be scanned and into the simulation, to determine if the scenes would match correctly. These objects included the robot components, the safety cage and occasionally extra boxes. This type of testing is aimed at determining if the system detects false positives for missing or unknown objects.

The robot was used as the main set of known objects, as this could be moved around the scene with a known position. A total of 1473 scans were captured with the robot in view in different positions. The frames correctly matched between four and seven of the seven components of the robot, including the end effector.
5 Results

Figure 5.3 shows a matched scan with no unknown objects detected. Several simulation objects, including the robot and the safety cage from the background, have been identified as part of the scan highlighted in green. In this image three objects (components of the robot) are coloured magenta showing that they were not matched to any data. The components that were not matched correctly were the small wrist components (joints 5 and 6) of the robot and the end effector. Also joint 2 of the robot was often not matched to scan data as it was close to the edge of the scan. Where the robot components were not correctly matched, the components were occluded by other objects in the scan scene, typically, either the gripper or the wrist components of the robot arm.

Table 5.4 shows an evaluation of the scan matching. The data shows the breakdown of the scan data matching process. The scans column shows how many scans were in the
5 Results
un. The other values are averaged over the entire run.

The area of the scan in Figure 5.3 is coloured by the match value on a scale between 1.0 and -1.0. The majority of the area successfully matched to simulation objects is coloured yellow which represents a value of approximately 0.75.

The match value is calculated from:

\[ M = (1.0 - \frac{D_{av}}{S}) \times P_d \]

Where \( M \) is the match value, \( D_{av} \) is the Average Projection Distance, \( S \) is the Scan tolerance and \( P_d \) is the Dot Product of Normals.

<table>
<thead>
<tr>
<th>Scans</th>
<th>Filtered Triangles</th>
<th>Average Matched Triangles</th>
<th>Average Match Value</th>
<th>Average Matched Triangles</th>
<th>Average Conflicting Triangles</th>
<th>Average Conflicting Value</th>
<th>Average Triangle Not Matched</th>
<th>Average Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1680 (18.02%)</td>
<td>7550.25 (80.99%)</td>
<td>0.87</td>
<td>1129.5 (12.12%)</td>
<td>-0.64</td>
<td>83.375 (0.89%)</td>
<td>104.00</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>1729.41 (18.55%)</td>
<td>7422.59 (79.62%)</td>
<td>0.87</td>
<td>1557.43 (11.86%)</td>
<td>-0.64</td>
<td>161.87 (1.74%)</td>
<td>104.00</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>529.308 (5.68%)</td>
<td>7356.97 (88.76%)</td>
<td>0.83</td>
<td>177.85 (1.91%)</td>
<td>-0.63</td>
<td>503.28 (5.4)</td>
<td>67.33</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>1743.68 (18.70%)</td>
<td>7409.31 (79.48%)</td>
<td>0.87</td>
<td>829.94 (8.90%)</td>
<td>-0.64</td>
<td>163.03 (1.75%)</td>
<td>102.44</td>
<td></td>
</tr>
</tbody>
</table>

The scan tolerance used was 60 mm for all scans. The average projection distance was calculated as shown in section 3.2. The dot product value was calculated between the normal of the scan for each triangle and the normal of the object triangle.

The Microsoft Kinect™ was used for these experiments. The Kinect™ produces a
5 Results

fixed array of depth values which translate into a fixed number of triangles for each scan. As such, there are exactly 9322 triangles in each scan.

The SSC process produces four possible results for each scan triangle: it can be filtered out, correctly matched to a simulation object, conflicting with a simulation object or not matched to any simulation objects. The filtered triangles column, in Table 5.4, shows the average number of triangles that were removed as they were too large. The filtering process is described in section 3.5.2. From testing a range of values, the match value of between 1.0 and 0.6 were found to best represent matching surfaces. The matching values between -0.6 and -1.0 were found to best represent conflicting surfaces. From testing a range of values, a match value between 0.6 and -0.6 was found to best represent when the scan surface had not been matched to any simulation surface. This analysis was aimed at demonstrating how the SSC process was able to correctly match various objects between the simulated environment and the scan, verifying their existence. This analysis demonstrates how well the scan matches the simulation. Areas of the scan not matched to any simulation objects are associated with unknown objects and areas conflicting with simulation objects represents missing objects.

The results presented in Table 5.4 represent the average of a set of scans from a run. The results represent the analysis of the scan matching process. This shows the number of triangles in the scan that were removed from filtering and the number of triangles matched to simulation objects.

Each scan triangle is matched to multiple simulation objects and the highest and lowest match values are used to determine the match and conflicting values. The number of triangles matched, conflicting and filtered will not add to 100% as some triangles will
5 Results

not correctly match to any object especially near the corners of objects and some triangles may conflict with multiple objects, but match correctly with another. In the models analysed in Table 5.4 there are no missing objects, which means that many triangles conflict with correctly matched objects. This can occur if objects are smaller than the matching tolerance, which would allow the front and back faces of an object to be matched to the same scan surface. Despite the conflicting areas it is still possible to identify all of the objects in the simulation as correct. With the scans presented there were no unknown objects detected in the scan areas as none of the unmatched areas were large enough to isolate an object. These results show that on average approximately 80% of the scan is correctly matched to simulation objects.

Table 5.5 shows the object analysis results of both known and unknown objects detected from the same scans as presented in Table 5.4. The Scans column again represents a set of scans or a run. The Table also shows how many objects were found in the scene and (in parentheses) how many should have been in the scene.

Table 5.5: Object analysis results

<table>
<thead>
<tr>
<th>Scans</th>
<th>Simulation Objects</th>
<th>Unknown Objects Found (Intended)</th>
<th>Average Matching Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Matched (Intended)</td>
<td>Not Matched (Intended)</td>
<td>Outside (Intended)</td>
</tr>
<tr>
<td>8</td>
<td>4.75 (9)</td>
<td>4.25</td>
<td>0 (0)</td>
</tr>
<tr>
<td>45</td>
<td>5.35 (9)</td>
<td>3.65</td>
<td>0 (0)</td>
</tr>
<tr>
<td>39</td>
<td>4 (4)</td>
<td>0</td>
<td>0 (0)</td>
</tr>
<tr>
<td>65</td>
<td>5.20 (9)</td>
<td>3.78</td>
<td>0 (0)</td>
</tr>
</tbody>
</table>
5 Results

Simulation Objects could be matched in one of four states: Matched, Not matched, Outside the Scan Area, and Missing. The scenes were created so that some objects were intended to be detected in a particular category, typically matched. In each frame the number of unknown objects was also measured. Again the scene was created so that the number of unknown objects was known before the test was run.

These results show that simulation objects can often be occluded even when inside the scan area i.e. the number of objects matched was less that the number of objects in the scene. The objects not matched to any data included the two smallest components of the robots wrist joint and often the gripper end effector. This was due to the parts being small and often occluded by other parts.

5.1.2 Missing Object Detection

To test the detection of missing objects, an object was added to the simulation that was not in the real environment. The result of a typical example of this SSC analysis in this situation is shown in Figure 5.4 where the simulation object shown in the Robot Studio simulation on the left, the missing object has been highlighted in the SSC display on the right. The object on the right has been highlighted in red wire-frame, as it does not match any of the existing scan data and conflicts with the surface data on the ground, hence can be assumed that it is not present in the scene.

Table 5.6 shows a comparison between two sets of scans with and without a missing object. As long as the ground underneath the object is visible the object was detected as missing. If the object was lifted off the surface by more than the matching tolerance it would not be detected as missing. Other incorrectly identified frames were due to
5 Results

occlusion and the object being outside of the frame.

Table 5.6 presents a scan matching analysis of scenes with and without a missing object. The analysis shows that missing objects do not change the number of filtered and matched triangles but there is a significant increase in the number of conflicting triangles from 1.91% to 4.98% and 11.86% to 13.13% respectively.

Table 5.7 shows an object analysis produced for the scan analysis shown in Table 5.6.

*Figure 5.4: A missing known object highlighted by the system.*
5 Results

*Table 5.6: Missing object matching results*

<table>
<thead>
<tr>
<th>Scans</th>
<th>Scan Triangles</th>
<th>Filtered Triangles</th>
<th>Average Matched Triangles</th>
<th>Average Match Value</th>
<th>Average Conflicting Triangles</th>
<th>Average Conflicting Value</th>
<th>Average Not Matched</th>
<th>Average Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>9322</td>
<td>529.31 (5.68%)</td>
<td>8274.51 (87.76%)</td>
<td>0.83</td>
<td>177.85 (1.91%)</td>
<td>-0.63 (5.40%)</td>
<td>503.28</td>
<td>104.00</td>
</tr>
<tr>
<td>41</td>
<td>9322</td>
<td>529.31 (5.68%)</td>
<td>8274.51 (87.76%)</td>
<td>0.83</td>
<td>464.23 (4.98%)</td>
<td>-0.76 (5.39%)</td>
<td>502.21</td>
<td>104.00</td>
</tr>
<tr>
<td>45</td>
<td>9322</td>
<td>1729.41 (18.55%)</td>
<td>7422.59 (79.62%)</td>
<td>0.87</td>
<td>1105.80 (11.86%)</td>
<td>-0.64 (1.74%)</td>
<td>161.87</td>
<td>67.33</td>
</tr>
<tr>
<td>45</td>
<td>9322</td>
<td>1729.41 (18.55%)</td>
<td>7422.67 (79.63%)</td>
<td>0.87</td>
<td>1223.85 (13.13%)</td>
<td>-0.68 (173%)</td>
<td>160.83</td>
<td>67.33</td>
</tr>
</tbody>
</table>

*Table 5.7: Missing object results*

<table>
<thead>
<tr>
<th>Scans</th>
<th>Simulation Objects</th>
<th>Average Matching Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matched (Intended)</td>
<td>Not Matched</td>
<td>Outside (Intended)</td>
</tr>
<tr>
<td>8</td>
<td>4.75 (9)</td>
<td>4.25</td>
</tr>
<tr>
<td>8</td>
<td>4.75 (9)</td>
<td>4.25</td>
</tr>
<tr>
<td>39</td>
<td>4 (4)</td>
<td>0</td>
</tr>
<tr>
<td>39</td>
<td>4 (4)</td>
<td>0</td>
</tr>
</tbody>
</table>
5 Results

5.1.3 Unknown Object Detection

Figures 5.5 to 5.8 show a typical example of a single unknown object, isolated in an environment with the robot and safety cage in the background. This example shows one unknown object, which is coloured transparent red from the scan matching results in Figures 5.5 from the top and 5.7 from the side. Figures 5.6 and 5.8 show textured views of the scan with the object highlighted by a bounding box.

Figure 5.5: SSC analysis of a scan with one unknown object top view
5 Results

Figure 5.6: Top view with one unknown object textured and highlighted

Figure 5.7: Side view of SSC analysis with one unknown object
Figure 5.9 shows a surface analysis with multiple screenshots. Several of the unknown objects detected are surrounded by black, which are areas of the surface that have been removed via filtering. In one case, the object has been completely isolated due to the filtering. The filtering was used as described in section 3.5.2.

Figure 5.10 shows the same frame from a slightly different angle with real texture and the objects isolated with bounding boxes. The largest detection, highlighted in blue, has been lifted off the ground by approximately 500mm, which is why it was isolated by the filtering. The other two objects are on the ground and have not been completely separated from the scan on the ground.
Figure 5.9 SSC analysis of a scan with multiple objects

Figure 5.10: Multiple unknown objects with bounding boxes
5 Results

Table 5.8 and 5.9 show that the more triangles that are not matched, the more areas exist that are part of unknown objects. The region growing algorithm searches for these areas to attempt to grow a region large enough to create an oriented bounding box around it.

Table 5.8: Unknown object scan matching results

<table>
<thead>
<tr>
<th>Scans</th>
<th>Filtered Triangles</th>
<th>Average Matched Triangles</th>
<th>Average Conflicting Triangles</th>
<th>Average Not Matched</th>
<th>Unknown Objects Found (Intended)</th>
<th>Average Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1680 (18.0%)</td>
<td>7550.2 (80.99%)</td>
<td>1129.5 (12.1%)</td>
<td>83.3 (0.8%)</td>
<td>0 (0)</td>
<td>100.1</td>
</tr>
<tr>
<td>45</td>
<td>1729.4 (18.5%)</td>
<td>7422.5 (79.62%)</td>
<td>1557.4 (11.8%)</td>
<td>161.8 (1.7%)</td>
<td>0.76 (1)</td>
<td>105.3</td>
</tr>
<tr>
<td>65</td>
<td>1743.6 (18.7%)</td>
<td>7409.3 (79.4)</td>
<td>829.9 (8.90%)</td>
<td>160.8 (1.7%)</td>
<td>0.97 (1)</td>
<td>102.4</td>
</tr>
<tr>
<td>39</td>
<td>529.308 (5.6%)</td>
<td>7356.9 (88.7%)</td>
<td>177.8 (1.9%)</td>
<td>503.2 (5.4%)</td>
<td>2.89 (4)</td>
<td>67.3</td>
</tr>
<tr>
<td>26</td>
<td>533.9 (5.7%)</td>
<td>8352.3 (89.6%)</td>
<td>84.4 (0.9%)</td>
<td>426.89 (4.5%)</td>
<td>3.14 (4)</td>
<td>71.70</td>
</tr>
</tbody>
</table>

Table 5.9: Unknown object detection results

<table>
<thead>
<tr>
<th>Scans</th>
<th>Average Matching Time (ms)</th>
<th>Simulation Objects</th>
<th>Unknown Objects Found (Intended)</th>
<th>Unknown object Isolation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Matched (Intended)</td>
<td>Not Matched</td>
<td>Outside</td>
</tr>
<tr>
<td>12</td>
<td>96.51</td>
<td>4.75 (9)</td>
<td>4.25 (0)</td>
<td>0 (0)</td>
</tr>
<tr>
<td>45</td>
<td>105.32</td>
<td>5.35 (9)</td>
<td>3.65 (0)</td>
<td>0 (0)</td>
</tr>
<tr>
<td>65</td>
<td>105.32</td>
<td>5.20 (9)</td>
<td>3.78 (0)</td>
<td>0 (0)</td>
</tr>
<tr>
<td>39</td>
<td>67.3</td>
<td>4 (4)</td>
<td>0 (0)</td>
<td>0 (0)</td>
</tr>
<tr>
<td>26</td>
<td>71.7</td>
<td>4 (4)</td>
<td>0 (0)</td>
<td>0 (0)</td>
</tr>
</tbody>
</table>
5 Results

5.2 Task Analysis Experiments

A total of 387 scan frames were collected where the robot moved through the environment on a simple path. These frames were captured in runs of between 10 and 75 frames in an online mode with the robot moving. During the runs, the full SSC process was running to detect unknown objects and perform the robot rerouting around any unknown objects that interfered with the path. The path of the robot is used to create a convex hull that encloses the volume that would be occupied by the robot as it moves. When unknown objects were positioned inside the movement hull they were detected by the collision detection algorithm and flagged as hazards. The Task Analysis was also performed off-line by examining the same frames. In the offline process the path could be modified so it would either move through one or more objects or miss all of the unknown objects in the scene.

Table 5.10 shows the results of the Task Analysis. This demonstrates how effectively the collision detection algorithm performs in detecting objects that interfere with a task.

<table>
<thead>
<tr>
<th>Collision detected</th>
<th>Object in path (Frames)</th>
<th>Object out of path (Frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collision detected</td>
<td>387</td>
<td>0</td>
</tr>
<tr>
<td>No Collision detected</td>
<td>0</td>
<td>387</td>
</tr>
</tbody>
</table>

Figure 5.11 shows two unknown objects that were detected in the scene. The yellow wireframe shape represents the path of the robot. The object on the left is highlighted in red to show that the object is a hazard to the manufacturing process as it is inside the
5 Results

movement hull. The object on the right is outside the movement hull and has not been highlighted; however, both have been identified by the SSC process as unknown objects. These results demonstrate that the SSC process can distinguish between hazardous objects and those that are not.

![Image](image.png)

*Figure 5.11: Unknown objects: one a hazard, the other clear of the movement*

The collision detection algorithm was timed when detecting collisions. From the timing measurements of the 387 frames, it was determined that the collision detection algorithm could perform collision checks between any objects in the scene and the predicted path of the robot within 2.9 ms, on average. The same frames could be tested in an off-line mode by adding an unknown object to the scan and retesting the same scan with a different simulation path applied. The off-line tests allowed the initial planned path to be significantly longer as the restrictions from the real robot were not present.
5 Results

5.3 Path Planning Experiments

As described in sections 3.2 and 3.5, the SSC process uses an occupancy grid to assist with the matching process. Path planning can also be performed using the same data structures. To demonstrate the difference in the resulting cells, an occupancy grid was populated several times using different sets of data including scan data only, simulation data only and combined simulation and scan data. Table 5.11 shows the results of the total number of cells in the grid.

Three data combinations were tested; scan only, simulation only and combined simulation and scan. For each case, the total number of cells for each dataset was analysed to determine how many cells resulted in being occupied for path planning purposes, how many cells were occupied by the robot directly and finally the number of cells that were available for path planning, i.e. the free cells.

Table 5.12 shows an analysis of the contents of the cells using the three combinations of data sources. This shows the different combinations of data that appear in each cell once the occupancy grid is populated.

Table 5.11: Occupancy grid cells with different data sources

<table>
<thead>
<tr>
<th>Overall cell count</th>
<th>Total Cells</th>
<th>Cells Occupied</th>
<th>Cells with Robot</th>
<th>Cells free</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan data only</td>
<td>4501</td>
<td>442</td>
<td>0</td>
<td>4058</td>
</tr>
<tr>
<td>Simulation data only</td>
<td>8464</td>
<td>810</td>
<td>266</td>
<td>7388</td>
</tr>
<tr>
<td>Both simulation and scan data</td>
<td>8464</td>
<td>819</td>
<td>266</td>
<td>7379</td>
</tr>
</tbody>
</table>
5 Results

Table 5.12: Occupancy grid cell content with multiple data sources

<table>
<thead>
<tr>
<th>Cell content</th>
<th>Scan data only</th>
<th>Simulation data only</th>
<th>Both simulation and scan data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan data only</td>
<td>442</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>Simulation data only</td>
<td>0</td>
<td>810</td>
<td>518</td>
</tr>
<tr>
<td>Both simulation and scan data</td>
<td>0</td>
<td>0</td>
<td>292</td>
</tr>
<tr>
<td>Robot Only</td>
<td>0</td>
<td>266</td>
<td>167</td>
</tr>
<tr>
<td>Robot and Scan data</td>
<td>0</td>
<td>0</td>
<td>99</td>
</tr>
</tbody>
</table>

Figures 5.12, 5.13 and 5.14 all show screenshots of an environment. Each screenshot uses one of three combinations of data. The green dots represent cells in the occupancy grid that are free of obstructions. The red dots represent cells that have an obstruction.

Figure 5.12 shows the occupancy grid populated using only scan data. Notably the cells associated with the robot are highlighted in red as there is no method that can be used to discriminate between an obstruction and the robot when only using scan data.

Figure 5.13 shows the occupancy grid populated using the simulation data only. In comparison with Figure 5.12 there are several differences. Cells associated with the robot have not been set as occupied as they need to be free so that the path planning can start from the current position without being “self” obstructed. Cells associated with the object on the right of the robot are all considered obstructed in the simulation-only data, but in the scan-only data only the front surface detected by the scan are shown to be occupied. Also, several areas of the ground of the work-cell are not flagged as occupied due to occlusion by other objects in the scene.
Figure 5.14 shows the occupancy grid populated using the combination of the simulation and scan datasets with unknown objects highlighted. In this case all of the areas associated with simulation objects are flagged as occupied including areas occluded in the scan data. Again, the cells associated with the robot are not flagged as occupied, which allows the path planning to proceed from the current position. There are also two unknown objects that have been identified in the scene, one outside the safety cage and one close to the robot arm highlighted in red. In Table 5.12 nine cells were detected with only scan data associated when using both simulation and scan data. These nine cells are the ones directly associated with the unknown object highlighted in red in Figure 5.14.
5 Results

Figure 5.13: Occupancy Grid with only simulation data

Figure 5.14: Occupancy Grid with both simulation and scan data
5 Results

Path Planning testing experiments were used to determine if a path planner could find a reroute path around an unknown object detected by the SSC process using the occupancy grid. Figure 5.15 shows the path planned around an obstruction. This example took 0.47 ms to generate the path. The green robot wire-frames show configurations of the robot for each point of the path. The reroute path is shown as the green nodes labelled T_ROB1 End connected by a cyan line. The original path is shown as the yellow line which goes through the unknown object highlighted in red.

![Figure 5.15: Path Planning results](image)

5.4 SSC Process Experiments

Experiments were performed on the SSC process to determine the combined performance of the Environmental Analysis, Task Analysis and Path Planning modules. These experiments tested the combined performance of the components and the overall timing of the entire process.
5 Results

The overall testing required the robot to complete a simple task developed in Robot Studio. The robot performed the task if there was no obstruction in its path. This task was interrupted by a dynamic obstacle, which moved in and out of the path of the robot as the robot was moving. An unknown object was manually moved into the path of the robot. When the unknown object moved into the path, the robot either stopped moving, if the task could not be rerouted, or moved around the obstacle. To test the stability of the overall process it was run multiple times. Samples of the runs are presented in Table 5.13.

Table 5.13: Overall system testing results

<table>
<thead>
<tr>
<th>Run</th>
<th>Result</th>
<th>Average Frame time (s)</th>
<th>Total time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>The robot stopped then when the object was removed, then continued on the specified path</td>
<td>0.49</td>
<td>24.02</td>
</tr>
<tr>
<td>34</td>
<td>The robot stopped then when the object was removed, then continued on the specified path</td>
<td>0.45</td>
<td>32.02</td>
</tr>
<tr>
<td>35</td>
<td>The robot paused then successfully rerouted around the object</td>
<td>0.46</td>
<td>31.34</td>
</tr>
<tr>
<td>36</td>
<td>The robot successfully detected obstruction and rerouted.</td>
<td>0.43</td>
<td>25.88</td>
</tr>
<tr>
<td>37</td>
<td>The robot successfully detected obstruction and rerouted.</td>
<td>0.48</td>
<td>30.59</td>
</tr>
<tr>
<td>38</td>
<td>The robot successfully detected obstruction and rerouted.</td>
<td>0.46</td>
<td>30.14</td>
</tr>
</tbody>
</table>

These tests were performed on an Intel Core i5 2400 @ 3.10 GHz. The average timing for the algorithms used in the SSC were found to be 460 ms with a standard deviation of 89 ms. The large standard deviation were produced by the delays in performing the Path Planning and updating the new path to the robot controller. The details of this timing are expanded in table 5.14, which shows the average processing times for
various internal components of the SSC process. Notably, the internal processing only
takes 250 ms on average which leaves as much as 210 ms for processes external to the
SSC process to complete. Typically this includes as much as 160 ms for
communicating with the robot to find its current position and update the path if
required. The average time between frames also includes a polling lag of up to 100 ms.

\[\text{Table 5.14: Overall system testing times}\]

<table>
<thead>
<tr>
<th>Initialisation times</th>
<th>Total 150 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan collection and formatting</td>
<td>1.7 ms</td>
</tr>
<tr>
<td>Simulation data applied to occupancy grid</td>
<td>138 ms</td>
</tr>
<tr>
<td>Movement hull generation</td>
<td>10 ms</td>
</tr>
<tr>
<td><strong>SSC average total time</strong></td>
<td><strong>101 ms</strong></td>
</tr>
<tr>
<td>(Environment Analysis) Simulation scan matching time</td>
<td>98.2 ms</td>
</tr>
<tr>
<td>(Environment Analysis) Object Isolation</td>
<td>1.2 ms</td>
</tr>
<tr>
<td>(Task Analysis) Collision detection</td>
<td>2.2 ms</td>
</tr>
<tr>
<td>(Feedback) Reroute Path Planning</td>
<td>0.5 ms</td>
</tr>
</tbody>
</table>

**5.5 Object Measurement**

To test the reliability of the unknown object isolation algorithms, a set of single models
were made of ten objects at various resolutions. In these models a single object was
isolated from the test area and its size calculated using the SSC method discussed in
sections 3.2.6, 3.2.7, 3.5.4 and 3.5.5. The initial resolution was chosen so that the
object would be successfully detected at lower resolutions and would effectively
demonstrate the stability of the size measurements with different filter sizes. The
resolutions are shown in Table 5.15.
5 Results

Table 5.15: Scan resolutions

<table>
<thead>
<tr>
<th>Model</th>
<th>Scan Resolution (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.96</td>
</tr>
<tr>
<td>2</td>
<td>1.31</td>
</tr>
<tr>
<td>3</td>
<td>1.92</td>
</tr>
<tr>
<td>4</td>
<td>3.84</td>
</tr>
<tr>
<td>5</td>
<td>7.68</td>
</tr>
<tr>
<td>6</td>
<td>15.36</td>
</tr>
</tbody>
</table>

As the measurement algorithm uses a convex hull of the visible areas of the object, the point of view of the object from the 3D scan affects the number of faces visible in the final scan. For example, if a single face of a cube is presented to the scanner two dimensions can be measured. With two faces of a cube present all three dimensions can be measured with two measurements in at least one dimension. At most three faces will be visible which allows all three dimensions to be measured twice. The results shown in Table 5.16 split the measurements of the oriented bounding box into “Maximum”, “Intermediate” and “Minimum” axes. A total of six scan resolutions shown in Table 5.15 were used.

Table 5.16 shows the model resolution, the appropriate filter size for the resolution and the average ratio between the measured sizes of the oriented bounding box of the object divided by the real sizes of each axis. There were in total 80 objects measured for each resolution. Notably in Table 5.16 model six is missing as no objects were detected at this resolution.

Comparing actual measurements of known objects with the objects isolated using the SSC process shows that the intermediate measurements are the most reliable on
5 Results

average with an error of 5% at the extreme. The smallest axes are the least reliable because if only one face is detected on the object only two axes can be measured, essentially the width and height of the face. Typically these measurements are the maximum and intermediate size of the object. In this case the smallest axis represents the amount of noise in the surface.

Table 5.16: Scan resolution tests measuring the size ratio (Measured / Real)

<table>
<thead>
<tr>
<th>Model No</th>
<th>Scan Resolution (mm)</th>
<th>Average Bounding box size ratio (Measured / Real)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Maximum</td>
</tr>
<tr>
<td>1</td>
<td>0.96</td>
<td>1.12</td>
</tr>
<tr>
<td>2</td>
<td>1.31</td>
<td>1.11</td>
</tr>
<tr>
<td>3</td>
<td>1.92</td>
<td>1.14</td>
</tr>
<tr>
<td>4</td>
<td>3.84</td>
<td>1.14</td>
</tr>
<tr>
<td>5</td>
<td>7.68</td>
<td>1.19</td>
</tr>
</tbody>
</table>

Table 5.17: Standard deviations of real to measured object size values

<table>
<thead>
<tr>
<th>Scan Resolution (mm)</th>
<th>Standard deviation of (Measured / Real)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Maximum</td>
</tr>
<tr>
<td>0.96</td>
<td>0.14</td>
</tr>
<tr>
<td>1.31</td>
<td>0.14</td>
</tr>
<tr>
<td>1.92</td>
<td>0.24</td>
</tr>
<tr>
<td>3.84</td>
<td>0.21</td>
</tr>
<tr>
<td>7.68</td>
<td>0.42</td>
</tr>
</tbody>
</table>

Figure 5.16 shows a graph of the results presented in Table 5.17. The Table and graph shows that the standard deviation of the intermediate axis measured from the oriented bounding box is almost linear with the scan resolution (R). As scan resolution approaches the size of the object the standard deviation of the measured object’s size
5 Results

increases which suggests that there will be a minimum object size that will be detectable for a given scan size.

5.5.1 Object Separation Detection

The objective of the object separation test is to determine the separation of objects at the various resolutions required, to find the minimum distinguishable separation distance and the relationship to filtering.

Table 5.18 shows the filter size used for the experiment, the distance the two objects were separated and the number of objects that were successfully separated in software. The results indicate that the probability of objects being successfully detected improves greatly when the separation is significantly larger than the filter distance. This is due to the fact that triangle filtering removes the triangles connecting objects. Having an appropriate point of view assists the process, but this will not always occur in practice.

Figure 5.16: A graph of the object size measurement reliability
5 Results

Table 5.18: A sample of object separation tests

<table>
<thead>
<tr>
<th>Triangle Filter Size (mm)</th>
<th>Object Separation (mm)</th>
<th>Objects Detected successfully</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>14.5</td>
<td>0 of 5</td>
</tr>
<tr>
<td>16</td>
<td>14.5</td>
<td>0 of 5</td>
</tr>
<tr>
<td>10</td>
<td>14.5</td>
<td>5 of 5</td>
</tr>
<tr>
<td>8</td>
<td>14.5</td>
<td>5 of 5</td>
</tr>
<tr>
<td>7</td>
<td>14.5</td>
<td>5 of 5</td>
</tr>
<tr>
<td>16</td>
<td>8.5</td>
<td>0 of 5</td>
</tr>
<tr>
<td>20</td>
<td>8.5</td>
<td>0 of 5</td>
</tr>
<tr>
<td>10</td>
<td>8.5</td>
<td>0 of 5</td>
</tr>
<tr>
<td>8</td>
<td>8.5</td>
<td>3 of 5</td>
</tr>
<tr>
<td>7</td>
<td>8.5</td>
<td>4 of 5</td>
</tr>
</tbody>
</table>

Figure 5.17 shows an original model before filtering with two real objects separated by a small distance. The large triangles connecting the two objects are clearly visible in the image.

Figure 5.18 shows two objects that have not been detected as separate objects. The objects have been isolated and measured using the yellow bounding box. The objects are connected by a small amount of unfiltered surface slightly smaller than the filter resolution.

Figure 5.19 shows two objects that have been successfully detected as separate objects. The objects are no longer connected by any unfiltered surface as the filter size is small enough to remove the connecting triangles. Each object has been isolated and measured using separate bounding boxes.
5 Results

Figure 5.17: Object separation pre-test

Figure 5.18: Object separation partial separation
5 Results

5.6 Tracking Results

Experiments were conducted in a controlled environment that included several known objects and several unknown objects which were scanned in three dimensions. One or more unknown objects were added to the scene and moved to determine if they could be tracked. In these experiments the SSC process was expected to identify any unknown object in the simulated area. The unknown objects were then isolated from the rest of the scan using simulation background subtraction. Three different methods of tracking unknown objects (position, velocity and an optimised method) were tested to compare the capabilities of each tracking method given the unknown objects isolated using the SSC algorithms.

During these experiments the tracking system used a frame rate of 3-4 Hz. For each test one or more objects were moved through the scene at a constant velocity. This experiment simulates the motion that is expected from a conveyor belt in a

Figure 5.19: Object separation successful separation
5 Results

manufacturing system. Each object was a different size ranging from 60 mm up to 225 mm. Over the five tests with each object, the velocity of movement was slightly increased until the position based tracking system failed. The same data was used with the different methods (position, velocity and optimised tracking algorithms). This allowed a direct comparison between the three methods.

Figure 5.20 shows successful tracking using the position based tracking technique where the object overlaps in three dimensions between successive captured frames. The solid lines show that the object was successfully tracked from its initial to final positions.

Figure 5.21 shows an example where the object does not overlap between frames. The coloured line segments show different sections of the path where the tracking was successful. By measuring the distance between object detections on successive frames it is possible to show the amount of movement that occurred relative to the size of the object and correlate the movements with successful and failed tracking.
The graph in Figure 5.22 shows all of the tracking tests for a set of different sized objects. The expected limit, as described in section 3.5.6, is overlaid on the graph. It can be seen in the data that as the object moves a distance that approaches the object size, the position based tracking is more likely to fail.

Using the velocity based tracking method, larger movements were successfully tracked. Although large accelerations (such as at the start of a movement) could cause the velocity method to fail. The technique is capable of tracking objects far above the limit of position based tracking. From the graph shown in Figure 5.23, it can be seen that the velocity based tracking approach can successfully track objects under much broader conditions than the position based method.
The optimised method determines the shortest, straightest path between detections for objects that closely match in size which allowed a single object to be tracked irrespective of how far or fast it moved. The results for the optimised method are shown in Figure 5.24. The graph shows that the optimised method successfully tracks a single object in all of the cases where one or both of the other two methods failed.
5 Results

To test the reliability of the different tracking methods under different conditions, a single object suspended on a string was moved through the environment. As long as the
5 Results

distance moved was smaller than the size of the object all of the tracking methods performed effectively.

However, under conditions where the object moved significantly further that the object’s size as shown in Figure 5.25 the position based method failed. For the same experiment the velocity and optimised methods successfully tracked the moving object as shown in Figures 5.26 and 5.27.

Figure 5.25: Moving object failed to track using position based tracking
5 Results

When the objects were thrown into the environment, they bounced off an object in the background, which created a large change in velocity. In this situation neither position nor velocity based tracking systems could successfully track the object over the entire path, as shown in Figure 5.28. The optimised method was able to track a single object under these conditions, shown in Figure 5.29.
5 Results

A final set of ten tests used multiple objects moving in the same scene as the previous experiments. An example is shown in Figure 5.30, where a total of four unknown objects were present in the scene. One object was suspended on a string above the others and moved through the air above the rest. Two objects are initially moving close to each other and one object remained stationary. When object movements were small the position based tracking method often proved sufficient to track each object. The

Figure 5.28: Unsuccessful tracking of a bouncing object with position tracking.

Figure 5.29: Successful tracking of a bouncing object with optimised tracking.
5 Results

optimised tracking method performed better than the position and velocity based methods with single objects. However, when multiple objects were introduced into the scene, the optimised method often connected incorrect objects. Conversely, the position and velocity methods did not become confused between different objects.

5.7 Summary of Results

Experiments were performed to validate the SSC process and to demonstrate its capabilities in terms of Environmental Analysis, Task Analysis and Path Planning.

A comparative analysis of the size of the cells used in the occupancy grid was made that demonstrated the advantage of using smaller grid cells over larger grid cells for the matching process between scan data and simulation data.

The Environmental Analysis demonstrated that it is possible to match scan data to existing objects from a simulation, detect missing objects from the simulation and
5 Results

identify unknown objects that were not specified in the simulation. The Environmental Analysis data was collected from experiments, which were also used with Task Analysis to determine if any unknown objects interfered with a pre-existing task or path.

A comparative analysis of the occupancy grid was performed using scan-only data, simulation-only data and combined simulation and scan data. This showed that the combined simulation and scan data was able to cover gaps in the scan data resulting from object occlusions and was able to distinguish between fixed objects and the robot itself. The path planning component of the SSC process demonstrated that it was possible to generate a new path around an interfering unknown object using the occupancy grid populated with both simulation and scan data.

The unknown object detections have also been validated by examining how accurately the unknown objects are measured. The accuracy was determined by comparing the size of the oriented bounding boxes measured by the SSC process to measurements of real objects. Experiments were also performed to demonstrate that the unknown object data can be used to track dynamic unknown objects in a test environment over time.
6 Discussion

The main focus of the research presented here is the development of the Simulation Scan Comparison (SSC) process. The SSC process has been developed to address limitations in current approaches to monitor manufacturing processes in industrial environments. The SSC process allows unknown objects to be detected in a controlled environment and uses this information to detect if changes to a movement task are required.

The previous three chapters have presented theories, experiments and results that describe and examine the performance of the SSC process. The SSC process uses an occupancy grid for matching between a 3D simulation and a 3D scan. The capabilities of this process have been explored by considering the performance of matching and path planning. Experiments were performed to investigate the three modules of the SSC process: Environment Analysis, Task Analysis and Path Planning. Experiments performed on the Environmental Analysis module have tested the capabilities of using 3D simulation background subtraction to identify known, missing and unknown objects. Task Analysis has considered pick and place tasks using collision detection to determine if an unknown object interferes with the movement of a robot.

The use of the occupancy grid was analysed using scan only data, simulation only data and combined simulation and scan data. The resulting cells using these datasets have been compared to show the differences in interpreting the obstructions in an environment. This showed that the combined simulation and scan data was able to cover gaps in the scan data resulting from object occlusions and was able to distinguish between fixed objects and the robot being controlled. The Path Planning component of
the SSC process demonstrated that it was possible to generate a new path around any interfering unknown object using the occupancy grid populated with both simulation and scan data.

Two other sets of experiments were performed to analyse the performance of the Environmental Analysis module. The first examined the reliability of unknown object size measurements. The second determined the robustness of the unknown object detection algorithm by considering the tracking of unknown objects over time.

### 6.1 Environment Analysis Results

Environmental Analysis refers to subdividing and analysing scan data from a real environment to determine what is in the environment. The Environment Analysis experiments were performed to determine if the use of a simulation for background subtraction was a useful technique for analysing a well-controlled environment.

The complexity of the matching process was analysed by comparing the performance of the matching process and varying the cell size of the occupancy grid. As described in section 3.2.5 the use of the occupancy grid in theory allows the performance of the matching process to be improved. Table 5.1 and Figure 5.1 show the results of analysing the matching process with different cell sizes. The results demonstrate that the smaller the cell size the faster the matching process occurs. The simple simulation with 7300 triangles using the smallest cell size of 0.08 m was 6.6 times faster than the slowest time for the larger cells. For the complex simulation with 23261 triangles, the smallest cells were 5.5 times faster than the slowest time with larger cells.

Notably the complexity of the simulation has a significant effect on the overall time.
6 Discussion

There are 3.2 times more triangles in the complex simulation and the time to complete matching increases by 280% on average. This suggests that the relationship between matching complexity and simulation complexity is close to linear. The final matching time will be heavily dependent on where the simulation and scan objects match not simply the size of the cells. The significant result is that the matching process is significantly faster using smaller cells.

While the occupancy grid structure has been used by other researchers for collision detection [165, 166], none of the literature reviewed has considered the use of this concept for matching objects, object isolation or object recognition as described in section 3.2. While this may not be a completely novel concept, it is demonstrated to be useful for simplifying the simulation background subtraction process.

The main function of the Environmental Analysis module is to detect different categories of objects, specifically:

- Known objects
- Missing objects
- Unknown objects

To examine this process, simulations of an environment were created to match to scans of the physical environment. The environment included a safety cage, a robot and other boxes positioned to be consistent with objects in the simulation. To test for the different categories, objects were added to the physical environment and / or the simulation to determine if the Environmental Analysis module could identify them correctly.

To test known object detection only the objects specified in the simulation were added
6 Discussion

to the physical environment. To test missing object detection, objects were added to the simulation, but not the physical environment. Unknown object detection was tested by adding objects to the physical environment but not the simulation.

The results of the SSC process matching (Table 5.2) shows the overall results of matching simulation objects, missing objects and unknown objects in the total number of frames collected. This table shows that the analysis correctly detected 90% of unknown objects and 98% of simulated and missing objects.

These results were further analysed to determine why there was a discrepancy between the expected and actual results. Table 5.3 shows the reasons for the discrepancies in expected versus actual results. There were three main reasons identified that objects were not correctly detected:

- Not inside the area of the scan but in the environment
- Occluded by another object in the environment
- Too small to be detected correctly

In terms of the first two reasons, not in the scan area and occluded, there is no scan data to match with the simulation object, hence the SSC process is not capable of correctly determining the state of the object.

Table 5.2 shows that 1473 frames of scan data were analysed to detect unknown objects of various sizes and positions. Of these, 75 frames had occluded objects, 70 frames had objects that were too small to be detected or affected by reflections that caused sensor errors and 95 had objects near or outside the scan area and were not detected correctly. Of the total 1473 frames, 1323 (89.8%) frames were correctly analysed, identifying all visible unknown objects.
6 Discussion

It was noted that if unknown objects are too small they cannot be isolated accurately by the SSC process. The results of the object measurement experiments, as described in section 5.5 show that objects smaller than three scan resolutions will often not be detected.

Some limitations were identified in relation to the SSC process. They relate to:

- Unknown object tracking with occlusions
- Partial matching with a known object

Unknown objects can be occluded in a scene just as easily as known objects. This can create a problem for controlled objects if they move into a position that occludes an object, but this problem can be addressed using tracking to identify the last position or potentially a predicted position of an unknown object.

Partially matching objects have not been significantly investigated in the current research. This situation is shown in Figure 6.1.

![Diagram: Unknown Object Detection](image)

*Figure 6.1: Incorrect objects*
6 Discussion

If an object is offset from the expected position in the simulation by more than the matching tolerance the object can be matched correctly to some areas of scan data and leave others unmatched which would create an unknown object.

6.1.1 Known Object Detection

The SSC process was evaluated using the simulation background subtraction process as described in section 3.2.4. The background subtraction process matches the simulation against the scan and indicates a final match value for each triangle in the scan. From analysing the matching values it is possible to determine how much of the scan surface was matched correctly to the simulation objects. Table 5.4 shows an analysis of the surface matching of the SSC process. The simulation background subtraction process produces a match value of between +1.0 and -1.0 showing how well the scan surface has matched or conflicts with simulation data. A value of +1.0 shows that the scan has matched perfectly with a simulation triangle. A value of -1.0 shows that the scan and simulation match in position perfectly, but have surface normals in opposite directions, as such, the scan and simulation are in conflict. A matching value close to 0.0 shows that the scan triangle did not match with any simulation data.

The scan matching results (Table 5.4) show that it is possible to successfully match up to 90% of scan data (with an average match value of 0.87) to simulation objects in a well aligned scan. In these cases 5-20% of scan data will be filtered. This demonstrates that the matching process is capable of eliminating the vast majority of background objects that intentionally exist in a scene.

The object analysis results (Table 5.5) analyse the same scans as shown in the scan
6 Discussion

matching analysis. The object analysis results show that, overall of the runs, the simulation objects were typically detected correctly or not matched. A few frames were found where an object was incorrectly identified as missing in the scene, as it was partially occluded.

These results demonstrate that the vast majority of frames analysed had successfully matched simulation objects with scan data. The results show that as much as 90% of a correctly oriented scan could be matched to a simulation. Approximately 5-20% of a scan will be filtered; typically these will be triangles connecting foreground objects or the robot, assuming there is one, to the background. This suggests that any other objects added to the scene (either missing simulation objects or unknown objects) would be detectable as the match area would change.

Without a missing object in an environment, 5-20% of the scan could conflict with existing simulation data. This may occur for thin objects where the dimension of the object is less than the matching tolerance.

Typically 1-3% of a surface will not match to any simulation object without any unknown objects existing in the scene. The scan will often not match near the edges of the simulation objects. At these locations a scan triangle will cover multiple surfaces of a simulation object, but the surface normal of the scan will not be the same as either simulation surface. Essentially the corners of objects will be clipped during scanning as shown in Figure 6.2. The corner clipping becomes significantly more pronounced at lower resolutions or with small objects. The clipping can cause variations in the orientation of the bounding box and the overall size measurements of unknown objects.
6 Discussion

Notably, the areas of the scan values (Filtered, Matched, Conflicting and Not Matched) add up to over 100%. This occurs as some areas of the scan can conflict with one or more simulation objects and correctly match to other areas of the simulation.

Other researchers that have considered detecting known and unknown objects in an environment include Fischer and Henrich [53], Stueckler [59] and Kaelbling et al. [129]. However, none of these authors have presented a metric for analysing the reliability of the scan data matching process they have used.

6.1.2 Missing Object Detection

The experimental approach for testing the SSC process for detecting missing objects was to add an object to the simulation without adding the object to the real world. The SSC process was then run on the environment and simulation to determine if the simulation objects would be detected as missing. The scan analysis for missing objects (Table 5.6) shows a comparison of a scan analysis with and without a missing object. The object analysis (Table 5.7) was performed using the same scans from the scan analysis. When an object was added to the simulation, but not the scan, it would be...
detected as missing. This would also cause the conflicting data to increase by 2-5% without changing any of the matching data. The object analysis shows that when a missing object is added to the simulation, it was correctly detected as long as the missing object was not occluded. These results demonstrate that the missing object detection can be performed using a simulation background subtraction process and missing objects can be detected with minimal influence on other objects in the scene. Notably none of the literature reviewed demonstrated any concept relating to missing object detection.

### 6.1.3 Unknown Object Isolation

The unknown object isolation was tested by using the SSC matching process to analyse an environment with one or more objects added to the environment but not the simulation. The matching analysis of unknown objects (Table 5.8) shows that when the environment has unknown objects the total area not matched increases compared to the environment when no unknown objects are present. These results show that the simulation background subtraction is capable of eliminating 80-90% of the scan surface and filtering will typically eliminate 5-20% of the rest of the surface. Any remaining areas not matched to the simulation can be analysed to determine if unknown objects can be identified in the scene.

In one example the “area not matched” of the scan for a particular environment was 0.89%. When an unknown object was added this value increased to 1.73, a 0.84% increase. This suggests a small object, approximately 0.84% of the scan surface in size was added to the scene. As the scans had 9322 triangles, 1% of this area represents an
object with 93 triangles. The final change in area depends on the size and number of unknown objects in the scene. In another case, much larger areas (4.58% and 5.4%) of the scan data were not matched to the simulation. These areas would then be grouped into the unknown object detections shown in Table 5.8. When unknown objects are added to a scene, the area of the scan not matched to the simulation will increase. For most of the objects used this value increased by 1-2%. This value will be different depending on the size of the object. This shows that when unknown objects are added to an environment, as long as the object is large enough, the object isolation process will be able to identify it.

Table 5.9 shows the number of objects matched and the number of objects isolated from the scene. These results also show the time taken for the entire matching process versus the time spent just in the object isolation algorithm. On average this algorithm takes 1.5 to 2 ms to isolate a single unknown object. This time will vary depending on the size of the object and ultimately the number of triangles. Irrespective of the size of the object this part of the Environmental Analysis typically takes less than 10% of the time taken for the matching process.

The corners of the scan would often have distortion larger than the matching tolerance in the depth image due to vignetting or a darkening of the scan depth at the corners of the depth image. This would result in unknown object detections in the corners of the scan. This could be compensated for by using an appropriate calibration with the Kinect (TM) sensor.
6 Discussion

6.1.4 Object Measurement

The object’s size detection is performed by examining the convex hull of the grouped points in an unknown object. The reliability of this measurement needs to be considered as only a single point of view is used to generate it.

The object size measurement experiments were tested by adding several objects of a known size to a simple environment. This environment was then scanned and the objects detected using the unknown object isolation of the SSC process. The measurements of the unknown objects were then compared against the size of the physical objects.

The object measurement results shown in section 5.5 demonstrate that the reliability and accuracy of measurements is closely related to the scan resolution and the objects size. The objects being measured were block of 15 by 30 by 60 mm, 30 by 30 by 60 mm or a cube of 30 mm. The measurements of the oriented bounding box of the objects were separated into three axes: maximum, intermediate and minimum. Typically the intermediate axis was the most reliable measurement. The maximum axis was more likely to be larger than the real measurement due to noise. The smallest axis was likely to be shorter than the real measurement as this axis was often self-occluded. Table 5.15 shows the list of resolutions the objects were analysed at. These resolutions ranged from 0.96 mm to 15.36 mm. Notably, a scan resolution of 15.36 mm did not produce any measurements of the objects. Objects of size 30 mm are only two scan resolutions wide (2R) and in model 5 objects at 30 mm would be four resolutions (4R) which were detected. From this result it can be assumed that objects need to be at least four resolutions to be detected reliably. Figure 5.16 shows a graph of the measurement...
reliability compared to scan resolution. The measurement reliability is directly related to the triangle size of a scan when compared to the size of the object. The standard deviation of measurements shows that as the average triangle size approaches the object size the reliability of the object measurement decreases. A scanner will have a known resolution \( R \), where \( R \) is the average distance between scan points at a set distance from the scanner.

From analysing the standard deviation data in Table 5.17:

\[
S = 0.89 \times \left( \frac{R}{L} \right) + 0.03
\]

Where \( S \) is the expected standard deviation of the measurement (reliability), \( R \) is the scan resolution and \( L \) is the object axis measurement. From this the minimum reliable object detectable with a 95% confidence level (i.e. a standard deviation of 0.33 or within 3 standard deviations) can be estimated at 3\( R \). Objects larger than 6.8 scan resolutions (i.e. a standard deviation of 0.16, or within 6 standard deviations, six sigma) should be detected 99.9999998% of the time. The measurement of objects at 6 scan resolutions or more should produce a size measurement within 5% of the real object on at least one axis.

6.1.4.1 Noise

Noise in the 3D scan can play a significant role in any over-estimates of unknown object size. Figure 6.3 shows an example of an object which has several “spikes” on its edges which are due to stereo vision miscorrelations on the object. Any scan includes noise, which will be incorporated into the isolated shape of an unknown object. This noise error will always increase the measured size of the unknown object,
as such, measurements should never be smaller than the object.

It is possible to reduce the noise in 3D scanning by using filters or an average of multiple scans, but this adds to processing times. With more reliable scanning devices noise is less likely to be a problem.

![Figure 6.3: Object noise](Image)

6.1.4.2 **Bounding Box Orientation Stability**

One notable result is that the measurements tend to be oversize either due to noise or orientation problems as shown in Figure 6.4. The rotating callipers method will determine a reasonable estimate of the size of the ideal oriented bounding box, but it may not identify the ideal measurement. The combination of the axies can be oversized, but is rarely undersized. The measurement will be at most a factor of \( \sqrt{2} \) above the ideal measurement from simple geometric calculations as shown in Figure 6.5.
The general trend as the size of the object is reduced, when compared to the filter size, is that even small errors from noise, orientation errors and detection fragmentation reduces the reliability of measurements.

There are several aspects of these experiments relevant to path planning. Path planning cells need to be an appropriate size for comparison with scan data to be effective in detecting objects. If cells are too large the path planning will produce a coarse path that will be obstructed unnecessarily by unknown objects. If cells are too small they may not allow the scan to match with simulation objects correctly and this makes path planning more computationally expensive.
6.1.5 Tracking

Tracking experiments were performed to determine if the unknown object detection methods from the Environmental Analysis module could track dynamic unknown objects over time. The results for position tracking shown in section 5.6 (Figure 5.22) show that the trend described for position based tracking in section 3.5.6 is not followed exactly, but it can be clearly seen that the larger the object the faster it has to move before the tracking fails. The differences between the expected tracking trend and the real measurements with the position based method can be explained by the variation in size measurements of objects. The unknown objects are not always detected as the same size, but they will always be within a factor of 1.4 above the real dimension as discussed in section 6.1.4.2. This is a useful result despite the failures in tracking as the basis of the method is to assume objects are not moving. Even with multiple objects in

Figure 6.5: Maximum bound error
6 Discussion

the scene, the position based tracking system successfully tracks objects when objects are moving slowly.

The results of velocity based tracking (Figure 5.23) show that this method tracks more objects than position based tracking, by estimating the movement of an object. Many of the successful tracks are above the trend described for position based tracking. As such, the movement of an object between frames can be increased before velocity based tracking loses a track. The velocity based tracking system will still fail if an object undergoes a significant change in velocity, such as an impact. This is demonstrated in the throwing tests where the object bounces off an object in the physical scene.

Notably, there were two tracking failures of the velocity method below the predicted position based trend line. These failures were due to large changes in velocity of the object being tracked. With multiple objects in a scene, it is possible for the velocity based tracking method to be confused by objects, if they are the same size and change their direction of movement.

Both position and velocity based methods are expected to have improved robustness when the frame rate improves. As these two methods use timing to estimate object positions, the faster the frame rate of the scanning system, the less objects will move between successive frames. The less the objects move the more likely they will overlap their original volumes.

The optimised tracking method uses the position and velocity of tracked objects, but it also calculates the probability that a particular object is the same object in the next frame. The optimised tracking is aimed at determining the correct object even if the object moves along a variable path. As such, it is expected that it can accommodate
larger changes in velocity. The main difficulty with the optimised method is that it could be easily confused between multiple objects. During operation, tracking could swap to incorrect objects when multiple objects were present.

6.2 Task Analysis

The Task Analysis module was aimed at determining if the SSC process could detect if any unknown objects detected from the Environmental Analysis would interfere with the movement of a manipulator robot. The results shown in section 5.2 demonstrate that for simple movement tasks such as pick and place operations collision detection can be used to determine if an object interferes with the path of a manipulator robot. The Task Analysis could fail to detect a real object that would intersect its path if the object passed in an area occluded from the scan i.e. behind the object. In this case the manipulator may clip the object as shown in Figure 6.6.

![Diagram showing limitations of task analysis with unknown objects due to sensor shadows.](image)

*Figure 6.6: Limitation of Task Analysis with unknown objects due to sensor shadows*
6 Discussion

This type of problem could be reduced by using multiple scanners from different directions or simply making sure any movements do not go past any obscured areas of a scan.

6.3 Path Planning

The Path Planning is based on an occupancy grid as described in section 3.4. The occupancy grid is populated using both simulation and scan data from the matching process as described in section 3.2 and section 4.3. Section 5.3 presented an analysis of the occupancy grid with three sets of input data: scan only, simulation only and a combination of simulation and scan. An analysis of the contents of the cells in the occupancy grid (Table 5.12) shows that each cell can be occupied by scan data, simulation data or a combination of both. Simulation data is also broken down between static objects in the scene and the robot being controlled.

The cell analysis (Tables 5.11 and 5.12) shows that with only simulation data it is not possible to detect unknown or unexpected conditions in the environment. The cell content breakdown (Table 5.12) shows that the combination of simulation and scan data can distinguish between the different combinations. In this example 810 cells contained simulation data, but only 292 cells contained both simulation and scan data. This means that 518 cells are occupied by objects from the simulation that would not have been identified with only scan data. Some of these 518 cells would represent the self-occluded areas of objects in the scene.

The cell content breakdown shown in Table 5.12, shows that a total of 266 cells are directly associated with the robot. Of these cells, 99 are matched with both simulation
and scan data. If only scan data is used, a robot being controlled will appear in scan data, but without the simulation to compare to, it is impossible to identify which parts of the scan represents the robot. As such, the robot should not appear in any scan data without filtering, otherwise that path planning will always be obstructed. Hence, the use of both simulation and scan data is required, if a robot is expected to be in the scan data.

Notably, in this example, only nine cells had scan-only data. These nine cells are the result of an unknown object in the scene. As expected, in a well-controlled environment the vast majority of the cells should be able to be taken into account using only the simulation, but the combination of the two datasets is required to be able to take into account all conditions (known and unknown). This shows that the combination of simulation and scan data is both important and necessary for path planning.

The experiments performed show that when the Task Analysis detected the interference, the Path Planning module would reroute around the obstacle using a standard A* path planning algorithm. As each of the cells used in the occupancy grid was assigned a set of robot configurations this allowed an appropriate configuration for the robot to be used without requiring the use of a more complicated path planner specifically for manipulators. The online testing was limited to a distance of 400 mm which would typically contain 5 to 6 cells and could be completed in 2.9 ms.

6.4 SSC Process

This research has presented a process for analysing an environment, analysing a simulated task and from this analysis, determine if the task needs to be altered to be
6 Discussion

completed correctly. While there may be alternative methods for some of the components, the SSC process as implemented can be used to identify problems in a physical manufacturing environment that involve unknown or missing objects.

While the SSC process can correctly detect interference in a task and reroute a manipulator, an overall processing time of 450 ms as shown in Table 6.1 is probably not sufficient for a safety critical environment, but will be sufficient for general monitoring of an environment. Given that this is the first time this process has been demonstrated it is likely that future versions will be more suitable for safety critical work.

Table 6.1: Results of the average overall processing times

<table>
<thead>
<tr>
<th>SSC processing</th>
<th>250 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Processes</td>
<td>150 ms</td>
</tr>
<tr>
<td>Polling Lag</td>
<td>50 ms</td>
</tr>
<tr>
<td>Total</td>
<td>450</td>
</tr>
</tbody>
</table>

6.5 Summary

The literature review and discussion have demonstrated that the SSC is a novel concept. While some of the methods used to perform the SSC process have been used previously, the combination of elements enables several capabilities that are not present with other methods. The results and discussion demonstrate that the SSC process has several capabilities that are useful in manufacturing environments.

Several capabilities of the SSC process have been demonstrated including:

- 3D Background subtraction using a simulation
6 Discussion

- Simulation object matching
- Missing object detection
- Unknown object isolation
- Task interference detection using collision detection algorithms to determine if unknown objects interfere with movement hulls
- Path planning using both simulation and scan data

Additionally, use of the occupancy grid for reducing the complexity of the matching algorithm has been evaluated. The evaluation has shown that with a small enough cell size the performance can be increased by a factor of 9.4 compared to using a larger size cell.

The unknown object measurement has been validated by experiments that show the accuracy of size measurements is proportional to the size of the object relative to the resolution of the scan. It has also been demonstrated that unknown objects can be tracked using collision detection algorithms and predictions based on how the object is moving.

The complete SSC system has been demonstrated to be able to detect an unknown object interfering with a movement task and reroute a manipulator around the object in a real-time environment.
7 Conclusions

The Simulation Scan Comparison (SSC) process was developed to allow a manufacturing process to be monitored and modified if required. The SSC process has been developed based on the scientific process of Hypothesise (Simulate), Observe (Scan) and Test (Compare) [4]. To detect differences between the simulation and a real world scenario, the SSC process uses a 3D simulation and a 3D scan of a physical environment. This allows an environment to be investigated, a task to be analysed (based on the state of the environment) and corrective action performed if required.

Experiments were carried out to examine the capabilities of the components of the SSC process and their combined performance. These experiments demonstrate that the SSC process is capable of analysing simulation data and live scans of an environment to detect known simulation objects, unknown objects and other unexpected conditions which could lead to failures in the manufacturing process. The SSC process was capable of performing at 2.1 Hz which is sufficient for general monitoring of an environment.

7.1 Environmental Analysis

7.1.1 Matching Complexity Analysis

The main feature of the Environmental Analysis module as implemented in the SSC process is the use of the simulation for background subtraction of the scan. This subtraction or matching process uses an occupancy grid to reduce the complexity of the matching process and thereby improve the performance of the matching.
7 Conclusions

The results show that by using small sized cell with the occupancy grid the performance improved by a factor of almost 5.5 with the complex simulation and as much as 6.5 with the simple simulation. The exact factor of improvement is dependent on the complexity and state of both simulation and environment, but there is a definite improvement in matching performance when small cells are used.

Other researchers have used occupancy grids or other spatial subdivision strategies to simplify collision detection management such as Teschner et al. [165] and Eitz and Lixu [166]. No researchers examining object isolation or object recognition have been found that have used occupancy grids for matching purposes.

7.1.2 Object Detection

The simulation background subtraction technique allows three types of objects to be detected; known simulation objects, missing simulation objects and unknown objects.

The results demonstrate that the Environmental Analysis module using a simulation for background subtraction allows up to 90% of a scan surface to be matched with known simulation objects. This result shows that the majority of the scan can be eliminated without effecting the detection of missing or unknown objects.

Missing objects are detected when the surface matching shows an increase in the conflicting data between simulation and scan for a given environment. Conflicting data occurs where the simulation and scan share the same position, but the surface normals are in opposite directions. The conflicting data in a scan can increase by as little as 1% for an object to be detected as missing.
7 Conclusions

Unknown objects are found where the scan data does not match to any simulation objects. The object measurement results discussed in section 6.1.4 show that when objects larger than three scan resolutions are added to a scene, the SSC process can isolate them from the scan data and measure them. This also results in an increase in the area not matched between simulation and scan.

Other researchers have considered detecting known and unknown objects in an environment. No other published literature reviewed has demonstrated the idea of detecting missing objects. Notably Fischer [140], Stueckler [125] and Kaelbling et al. [129] have considered object isolation and use simulations to plan tasks or analyse environments. However, none of these authors have presented a metric for analysing the reliability of object matching or isolation from scan data and none have performed an in depth analysis of a simulation scan comparison process.

7.1.3 Object Measurement

This research demonstrates the effects of scan resolution against object size detections. A convex hull is created around isolated objects and using a rotating callipers measurement an oriented bounding box is selected. The three axes of the oriented bounding box can then be used as measurements of the object’s dimensions. The consistency of the sizes detected for the objects at various resolutions shows that the intermediate size measurement is typically within 5% of the real object size, which demonstrates the reliability of the isolation process using a single scanner.

The measurement reliability was found to be directly related to the scan resolution of a scan when compared to the size of the object. The standard deviation of measurements
shows that as the average scan resolution approaches the object size the reliability of 
the object measurement decreases. From analysing the standard deviation data of size 
measurements the following relationship was established:

\[ S = 0.89 \times \left(\frac{R}{L}\right) + 0.03 \]

Where \( S \) is the expected standard deviation of the measurement (reliability), \( R \) is the 
scan resolution and \( L \) is the object axis measurement. From this an object must be at 
least 3R in size to be detected with 95% confidence. Objects larger than 6.8 scan 
resolutions, 6.8R, should be detected 99.9999998% of the time. The isolated object 
measurement, essentially a rotating callipers algorithm of a convex hull, produces a 
size measurement within 5% of the real object on at least one axis.

From the object separation tests it is possible to conclude that if the object separation is 
less than the scan resolution (R) the separation will not be detected. Typically the 
triangle filter will be based on a factor (typically 1.5 to 2.0) of the scan resolution (R). 
When setting up a system for an SSC process the minimum size of objects and the 
minimum separation used for both path planning and tolerances need to be established. 
The analysis of the minimum size and minimum separation show that these values can 
be calculated based upon the resolution of the scanner available. These values can be 
used as a minimum clearance for any path planning operations. This will ensure that 
any object that is not successfully detected will be smaller than the clearance and hence 
should not cause a problem. Few researchers have considered the concept of object 
isolation and none were found that considered the measurement of such objects.
7 Conclusions

7.1.4 Object Tracking

The tracking results show that the unknown objects isolated from scan data are stable enough to be tracked over time. The tracking methods developed are based on collision detection algorithms. The upper limit on position based tracking can be expressed as shown below:

\[ V_{\text{object}} < \frac{S_{\text{xyz}}}{T_{\text{Scan}}} \]

Where \( V_{\text{object}} \) is the velocity of the object, as a three dimensional vector; \( S_{\text{xyz}} \) is the size of the object in three dimensions; and \( T_{\text{Scan}} \) is the scan rate of the sensor.

Velocity based tracking is capable of tracking objects moving faster than their own size per frame.

Very few other researchers were found that had considered the concept of object isolation and none were found that considered tracking of such objects.

7.2 Task Analysis

Task Analysis uses collision detection between the simulated movement hull of the component objects of a robot and any unknown objects in an environment isolated from scan data. The results demonstrate that collision detection can be used to determine if a movement task needs to be replanned. The limitation of this approach is that a collision may not be detected if the robot is moving into an area occluded from the sensor by the unknown object. To avoid this problem multiple cameras can be used to scan the work area as demonstrated by other researchers [52–55, 97].

Many researchers have considered the concept of collision detection, but few
7 Conclusions

publications have been found that have considered the concept of collision detection between a simulated task and physical objects from scan data in a manufacturing environment.

7.3 Path Planning

The technique demonstrated for path planning analysis used the occupancy grid for path approach planning using an A* search. Each cell in the grid contains a set of configurations that allow a manipulator to be used with the path planning. The occupancy grid is populated with data from both simulation and scan data. This determines which cells are considered occupied and which cells are free and can be used to perform path planning. A comparison of the datasets was performed to show the differences in using only scan data, only simulation data and combined simulation and scan data.

This comparison showed that scan data alone can show which areas are occupied in the grid from visible scan data. However, there are many cells of the scene that are occluded by objects which results in gaps in scan data. Scan data alone is not able to distinguish between the environment and the robot that requires path planning. Hence path planning may not be possible if the robot appears in the scan data. Using a simulation allows the robot to be taken into account in the occupancy grid. The simulation of the robot will occupy the appropriate cells, but simulation data alone does not take into account any unknown objects. By using a combination of both simulation and scan data it is possible to identify the robot in the scan data and avoid flagging the appropriate cells as occupied to allow a path to be planned. Also unknown objects can
7 Conclusions

be fully accounted for and cells that are associated with objects that have been matched can also be considered occupied. Other researchers have usually focused on either purely simulated data or purely scan data.

7.4 SSC Process

The combined results of this research demonstrate that the combination of simulation and scan data can be used to analyse an environment. The results show that the SSC process is capable of identifying unknown objects in a known environment from a 3D scan. These objects can be reliably measured and tracked over time. Once unknown objects have been isolated, interactions with the objects can be planned or avoided using the Task Analysis module. Task Analysis for movement tasks can be performed using collision detection between the simulated movement of the robot and the unknown object detections. The results of the Task Analysis can be used to determine when to update a path if interference has been detected with an unknown object. The path planning system using both simulation and scan data in an occupancy grid is demonstrated and it is shown that this technique can assist in identifying areas that are occupied by the robot being controlled as well as other expected and unexpected obstacles in the scene. The results show that the SSC process can run at up to 2.1 Hz. This is fast enough to be useful for an online monitoring system, but may not be fast enough for a safety critical system.

Other researchers have considered a system using both simulation and scan data that can detect unknown obstacles and plan paths around the obstacles [53, 97]. However, none of the literature reviewed has demonstrated the combined capabilities of the SSC
7 Conclusions

process. In addition, none of the literature reviewed has presented analysis on the reliability of their matching process between simulation and scan, unknown object detections size measurement or unknown object tracking.

7.5 Future Work

Through the experiments and analysis of the operation of the SSC process several concepts have been identified where future work would be useful.

7.5.1 Simulation Development

To improve the performance of the SSC process it is possible to create simulation models at more appropriate resolutions for the matching process. It is unnecessary to model every component to 1 mm resolution when the scan resolution is only 20 mm. Results have shown that the more complex the simulation used, the longer the matching process takes. Many of the components used with the simulation are directly from ABB’s object library, which have no optimisation for matching purposes. Other models, for example the gripper, were over complicated and required manual modelling to simplify the object to a reasonable level of detail to compare to a scan at 20 mm resolution. In future it should be possible to create models at appropriate resolutions for optimised matching performance.

7.5.2 Object Recognition and Pose Estimation

Once unknown objects have been isolated from the scan it should be possible to perform object recognition on the isolated objects. As the Kinect (TM) uses both 3D and 2D data, both 3D shape analysis and 2D feature recognition techniques could be used to further identify objects. The processes of object recognition and pose estimation should
be much simpler using isolated objects rather than attempting to identify the objects purely from scan data as the objects have already been isolated from the background information in the scene.

7.5.3 Experimental Set-up
The experimental test area was designed to be a simple work-cell similar to what would be found in a typical manufacturing environment. The main focus of this environment was to demonstrate the basic capabilities of the SSC process, not necessarily its application to a complex task. Hence, it is expected that a real environment would have much more complexity in terms of both machinery and movements. This in turn would require much more simulation detail and far more interaction with other equipment in the environment. In future the SSC process needs to be tested in a real production environment to determine the effect on production performance and error reduction.

7.5.4 Performance
There are several possibilities for improving the performance of the entire SSC process. A faster CPU or a hardware implementation, such as an FPGA implementation would make a significant difference in performance. Alternatively, the performance may be improved using a GPU based approach. Any of these approaches will only affect the internal components at present represents 250 ms of a 450 ms duty cycle.

The current approach was optimised for speed but limited by the ABB network interface with up to 150ms delay due to network latency. For safety critical work ideally another interface is required with significantly less latency.
7 Conclusions

7.6 Final Comments
The Simulation Scan Comparison (SSC) process developed in this research demonstrates the value of the combined use of simulation and scan of the physical environment. The combination of these two datasets, when accurately aligned, allows the scan data to be interpreted based on the simulation. This allows decisions to be made about the tasks being performed. The SSC concept has been analysed by examining the output of the unknown object detection and overall performance to determine its reliability and accuracy. It is expected that this process can be applied in industrial and more general applications to enable them to operate “intelligently” to be more robust to changes in the environment and ultimately more productive.
References

References


[34] M. Carlberg, P. Gao, G. Chen, and A. Zakhor, “Classifying urban landscape in


References


[62] B. L. Theisen, “The 16TH annual intelligent ground vehicle competition:
References


[77] R. C. Luo and M. G. Kay, “Multisensor integration and fusion in intelligent
References


References

References


References


References


Appendix

A list of Manufacturers who produce 3D scanning equipment or software, from Bi and Wang [158] with additional entries.

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
<th>Range</th>
<th>Cycle time</th>
<th>Accuracy</th>
<th>Web site</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optech</td>
<td>ILRIS-3D</td>
<td>3–1500 m</td>
<td>Minutes to hours per scan</td>
<td>10's of cm over long range</td>
<td><a href="http://www.optech.ca/">www.optech.ca/</a></td>
<td><a href="http://www.geo-konzept.de">www.geo-konzept.de</a></td>
</tr>
<tr>
<td>Shape Grabber</td>
<td>Scan heads, portable classic, automated system Model Make D50,100,200</td>
<td>0.33–1.75 m</td>
<td>18,000 to 100,000+ points/sec</td>
<td>µm</td>
<td><a href="http://www.shapegrabber.com">www.shapegrabber.com</a></td>
<td>Used for scanning small objects Not useful for area scans</td>
</tr>
<tr>
<td>INO 3D Scanners Metres</td>
<td>Laser profiling sensor Model Make Z35,70,140 (handler scanner)</td>
<td>0.3, 0.05–0.2 m</td>
<td>19200 pts/s</td>
<td>8 µm</td>
<td><a href="http://www.ino.ca/">www.ino.ca/</a></td>
<td><a href="http://www.3dscanners.com/">www.3dscanners.com/</a></td>
</tr>
<tr>
<td>Tyzx</td>
<td>DeepSea camera 3cm, 6 cm,22cm, DeepSeaG2, G3</td>
<td>0.2–2.7 m</td>
<td>Up to 30 Hz</td>
<td>Dependant on camera, lens, and distance</td>
<td><a href="http://www.tyzx.com/">www.tyzx.com/</a></td>
<td>Stereo camera system. Used for security purposes And has Person detection capabilities</td>
</tr>
<tr>
<td>Point Grey Research</td>
<td>Bumblebee, 2 Bumblebeen</td>
<td>1 – 10 m</td>
<td>3-4 Hz, up to 100Hz for small images</td>
<td>Dependant on camera, lens, and distance</td>
<td><a href="http://www.ptgrey.com">www.ptgrey.com</a></td>
<td>Stereo camera system.</td>
</tr>
<tr>
<td>Videres Design</td>
<td>Aparen</td>
<td>1 – 10 m</td>
<td>3-4 Hz, up to 100Hz for small images</td>
<td>Dependant on camera, lens, and distance</td>
<td><a href="http://www.videredesign.com">www.videredesign.com</a></td>
<td>Stereo camera system.</td>
</tr>
<tr>
<td>Surveyor</td>
<td>SVS</td>
<td>1 – 10 m</td>
<td>3-4 Hz, up to 100Hz for small images</td>
<td>Dependant on camera, lens, and distance</td>
<td><a href="http://www.surveyor.com">www.surveyor.com</a></td>
<td>Stereo camera system.</td>
</tr>
<tr>
<td>Konica Minolta</td>
<td>Ranger 5 Ranger 7 VIVID 9i VIVID 910</td>
<td>450 - 800mm 450 - 800mm 0.5 to 2.5 m 0.5 to 2.5 m 0.5 to 2.5 m</td>
<td>Laser scanner 2 seconds per scan</td>
<td>±40 µm</td>
<td><a href="http://www.se.konicaminolta.us">www.se.konicaminolta.us</a></td>
<td>Laser scanners</td>
</tr>
<tr>
<td>Genex Technologies</td>
<td>Rainbow 3DCamera</td>
<td>0.2–0.37 m</td>
<td>5-30 sec</td>
<td>0.025 mm 0.25 mm</td>
<td><a href="http://www.genextech.com/pages/601/Rainbow_3D_Camera.htm">www.genextech.com/pages/601/Rainbow_3D_Camera.htm</a></td>
<td>Best guess Structured light camera</td>
</tr>
</tbody>
</table>
## Appendix

<table>
<thead>
<tr>
<th>Company</th>
<th>Model/Description</th>
<th>Measurement Range (m)</th>
<th>Scan Time (per scan)</th>
<th>Resolution (mm)</th>
<th>Website/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3rdTech</td>
<td>DeltaSphere-3000 3D Digitizer</td>
<td>0.5–15</td>
<td>2 - 15 minutes/scan</td>
<td>5</td>
<td><a href="http://www.3rdtech.com">www.3rdtech.com</a> Laser scanner</td>
</tr>
<tr>
<td>Trimble</td>
<td>Trimble CX, GX, FX,</td>
<td>80</td>
<td>50,000 pts/sec</td>
<td>Up to 7</td>
<td><a href="http://www.trimble.com">www.trimble.com</a> Laser scanner</td>
</tr>
<tr>
<td>Leica Geosystems</td>
<td>Scan Station</td>
<td>0.3</td>
<td>Up to 700m</td>
<td></td>
<td><a href="http://www.leica-geosystems.com">www.leica-geosystems.com</a> Laser scanners</td>
</tr>
<tr>
<td>FARO</td>
<td>LS 420, LS840, LS880</td>
<td>20 - 80</td>
<td>976,000 points/sec</td>
<td>2</td>
<td><a href="http://www.faro.com">www.faro.com</a> Laser scanner</td>
</tr>
<tr>
<td>Riegler Laser Measurement Systems</td>
<td>LMS-Zxxx series</td>
<td>0.35–1.0</td>
<td>Minutes to hours per scan</td>
<td>10's of cm over long range</td>
<td><a href="http://www.riegl.com">www.riegl.com</a> Laser scanner</td>
</tr>
<tr>
<td>Zoller+Frohlich GmbH</td>
<td>IMAGER5006, IMAGER 5003</td>
<td>80</td>
<td>508,000 pix/sec</td>
<td>1</td>
<td><a href="http://www.zf-laser.com">www.zf-laser.com</a> Laser scanner</td>
</tr>
<tr>
<td>Roland Corp Inus Technology</td>
<td>LPX-60/600 3D</td>
<td>0.3–0.4</td>
<td>Seconds per scan</td>
<td>0.1</td>
<td><a href="http://www.rolanddda.com/Object">www.rolanddda.com/Object</a> Scanner</td>
</tr>
<tr>
<td>Bytewise Measurement Systems</td>
<td>CTWIST</td>
<td>0.03</td>
<td>5 min</td>
<td>0.008</td>
<td><a href="http://www.bytewise.com/">www.bytewise.com/</a> Tire scanner</td>
</tr>
<tr>
<td>Micro-Epsilon</td>
<td>LLT2800-25, 100100</td>
<td>0.025–0.1</td>
<td>640 pts / scan</td>
<td>0.05</td>
<td><a href="http://www.me-us.com/">www.me-us.com/</a> Profile scanner</td>
</tr>
<tr>
<td>Steinbichler</td>
<td>COMET 5</td>
<td>0.42–1.7</td>
<td>0.6 sec</td>
<td>50 – 800 µm</td>
<td><a href="http://www.steinbichler.de">www.steinbichler.de</a> Laser scanner</td>
</tr>
<tr>
<td>ARIUS3D</td>
<td>ARIUS3D</td>
<td>6</td>
<td>Minutes per scan</td>
<td>Sub mm</td>
<td><a href="http://www.arius3d.com">www.arius3d.com</a> Object scanner</td>
</tr>
<tr>
<td>Breuckmann</td>
<td>OptoScan, Smartscan, 3D-Alignment</td>
<td>0.36 - 0.72</td>
<td>&lt; 1 sec mm, Sub mm</td>
<td></td>
<td><a href="http://www.breuckmann.com">www.breuckmann.com</a> Stereo camera</td>
</tr>
<tr>
<td>MicroScribe</td>
<td>MicroScribe MX, MicroScribe MLX</td>
<td>0.63 - 0.84</td>
<td>Single point measurement µm</td>
<td><a href="http://www.microscribe.com">www.microscribe.com</a> Metrology measurement device</td>
<td></td>
</tr>
<tr>
<td>GOM mbH</td>
<td>ATOS 3D Digitizer</td>
<td>1.6</td>
<td>seconds per scan</td>
<td>Sub mm</td>
<td><a href="http://www.gom.com/EN/">www.gom.com/EN/</a> Object scanner</td>
</tr>
<tr>
<td>MEL Mikroelektronik GmbH</td>
<td>M2D, M2DW, M20D-XF</td>
<td>1.2</td>
<td>Seconds to minutes per scan</td>
<td>Sub mm</td>
<td><a href="http://www.melsensor.de">www.melsensor.de</a> Object scanner</td>
</tr>
<tr>
<td>Cyberware</td>
<td>Model shopWholebody4, and X</td>
<td>1</td>
<td>7 seconds</td>
<td>mm</td>
<td><a href="http://www.cyberware.com">www.cyberware.com</a> Object scanner, full body scan of a human</td>
</tr>
<tr>
<td>Laser Design</td>
<td>DS, RE, PS</td>
<td>1</td>
<td>Minutes per scan</td>
<td>mm</td>
<td><a href="http://www.laserdesign.com">www.laserdesign.com</a> Object scanner</td>
</tr>
<tr>
<td>Vitronic</td>
<td>Vitus</td>
<td>1</td>
<td>12 seconds</td>
<td>mm</td>
<td><a href="http://www.vitronic.de">www.vitronic.de</a> Object scanner, full body scan of a human</td>
</tr>
<tr>
<td>Polhemus</td>
<td>FastScan</td>
<td>0.8</td>
<td>Hand held line scan</td>
<td>0.1</td>
<td><a href="http://www.fastscan3d.com">www.fastscan3d.com</a> Hand held scanner</td>
</tr>
</tbody>
</table>
### Appendix

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Model</th>
<th>Type</th>
<th>Resolution</th>
<th>Scan Speed</th>
<th>Website</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC2</td>
<td>Body scanner</td>
<td>0.8 m</td>
<td>8 seconds</td>
<td>3 mm</td>
<td><a href="http://www.tc2.com">www.tc2.com</a></td>
<td>Object scanner, full body scan of a human</td>
</tr>
<tr>
<td>Nextec</td>
<td>Hawk</td>
<td>0.3 m</td>
<td>Minutes per scan</td>
<td>2 µm</td>
<td><a href="http://www.nextec-wiz.com">www.nextec-wiz.com</a></td>
<td>Object scanner</td>
</tr>
<tr>
<td>Kreon</td>
<td>ZEPHYR KZ25,50,100</td>
<td>0.2 m</td>
<td>30 000 points per second</td>
<td>3 µm</td>
<td><a href="http://www.kreon3d.com">www.kreon3d.com</a></td>
<td>Metrology Hand held scanner</td>
</tr>
<tr>
<td>Perceptron</td>
<td>Contour Probe Sensor</td>
<td>0.1 m</td>
<td>23,000 points per second</td>
<td>50 µm</td>
<td><a href="http://www.perceptron.com">www.perceptron.com</a></td>
<td>Metrology Hand held scanner</td>
</tr>
<tr>
<td>PMD tec</td>
<td>PMD vision</td>
<td>1.0 – 50 m</td>
<td>15- 30 Hz</td>
<td>10 mm</td>
<td><a href="http://www.pmdtec.com">www.pmdtec.com</a></td>
<td>Phase Modulation Device (PMD)</td>
</tr>
<tr>
<td>Mesa Imaging</td>
<td>Swiss ranger</td>
<td>5– 10 m</td>
<td>15- 30 Hz</td>
<td>10 mm</td>
<td><a href="http://www.mesa-imaging.ch">www.mesa-imaging.ch</a></td>
<td>Flash LIDAR</td>
</tr>
<tr>
<td>Advanced Scientific concepts</td>
<td>Tiger eye</td>
<td>10 – 5000 m</td>
<td>15- 30 Hz</td>
<td>10 mm</td>
<td><a href="http://www.advancedscientificconcepts.com">www.advancedscientificconcepts.com</a></td>
<td>Flash LIDAR</td>
</tr>
<tr>
<td>Microsoft</td>
<td>Kinect (TM)</td>
<td>1.2-5 m</td>
<td>15- 30 Hz</td>
<td>20 mm</td>
<td><a href="http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/">http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/</a></td>
<td>Infra-red dot projection</td>
</tr>
</tbody>
</table>