Enhanced Stereo Vision SLAM for Outdoor Heavy Machine Rotation Sensing

by

Li-Heng Lin

Bachelor of Applied Science, University of British Columbia, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Applied Science

in

THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

The University Of British Columbia

(Vancouver)

June 2010

© Li-Heng Lin, 2010
Abstract

The thesis presents an enhanced stereo vision Simultaneous Localization and Mapping (SLAM) algorithm that permits reliable camera pose estimation in the presence of directional sunlight illumination causing shadows and non-uniform scene lighting. The algorithm has been developed to measure a mining rope shovel’s rotation angle about its vertical axis (“swing” axis). A stereo camera is mounted externally to the shovel house (upper revolvable portion of the shovel), with a clear view of the shovel’s lower carbody. As the shovel house swings, the camera revolves with the shovel house in a circular orbit, seeing differing views of the carbody top. While the shovel swings, the algorithm records observed 3D features on the carbody as landmarks, and incrementally builds a 3D map of the landmarks as the camera revolves around the carbody. At the same time, the algorithm localizes the camera with respect to this map. The estimated camera position is in turn used to calculate the shovel swing angle. The algorithm enhancements include a “Locally Maximal” Harris corner selection method which allows for more consistent feature selection in the presence of directional sunlight causing shadows and non-uniform scene lighting. Another enhancement is the use of 3D “Feature Cluster” landmarks rather than single feature landmarks, which improves the robustness of the landmark matching and reduces the SLAM filter’s computational cost. The vision-based sensor’s maximum swing angle error is less than +/- 1° upon map convergence. Results demonstrate the improvements of using the novel techniques compared to past methods.
# Table of Contents

Abstract ................................................................. ii

Table of Contents ....................................................... iii

List of Tables ........................................................... vi

List of Figures ............................................................ vii

Acknowledgments .......................................................... ix

Statement of Co-Authorship ............................................... x

1 Introduction ............................................................. 1
   1.1 Introduction ....................................................... 1
   1.2 Background ....................................................... 3
      1.2.1 Pinhole Camera Model ........................................ 3
      1.2.2 Stereo Vision ................................................. 4
      1.2.3 Features ...................................................... 5
      1.2.4 Simultaneous Localization And Mapping ..................... 6
   1.3 Thesis Context, Outline and Contributions ....................... 7
      1.3.1 Thesis Goal .................................................. 7
      1.3.2 Thesis Outline ............................................... 8
      1.3.3 Contributions ............................................... 8
   1.4 References ....................................................... 9

2 Rope Shovel Collision Avoidance System ............................ 11
   2.1 Introduction ..................................................... 11
      2.1.1 System Overview ............................................ 11
   2.2 Arm Geometry .................................................... 12
      2.2.1 Distal: Joint Sensor Based ................................. 14
      2.2.2 Proximal: Laser Scanner Based ............................. 14
   2.3 Swing Measurement ............................................... 16
2.3.1 Encoder-Based Swing Measurement ................................. 16
2.3.2 Camera-Based Swing Measurement ............................... 16
2.4 Truck Localization ....................................................... 17
2.5 Discussion and Conclusions ........................................... 19
2.6 References ............................................................... 20

3 Enhanced Stereo Vision SLAM For Outdoor Heavy Machine Rotation Sensing 22

3.1 Introduction ............................................................... 22
  3.1.1 Contributions ....................................................... 25
3.2 Related Work ............................................................ 25
  3.2.1 Feature Types ....................................................... 25
  3.2.2 Approaches For Finding Camera Pose .......................... 26
3.3 Method ................................................................. 29
  3.3.1 Algorithm Overview ............................................... 29
  3.3.2 EKF State Vector ................................................. 30
  3.3.3 Initialization and Calibration .................................... 32
  3.3.4 EKF Prediction Update ......................................... 32
  3.3.5 EKF Measurement Update ...................................... 33
  3.3.6 3D Feature Detection ........................................... 34
  3.3.7 Feature Cluster Landmarks ..................................... 38
  3.3.8 Landmark Measurement ......................................... 42
  3.3.9 Swing Angle Computation ...................................... 47
  3.3.10 Landmark Feature Management ............................... 49
3.4 Results ................................................................. 50
  3.4.1 Comparison of Harris Corner Selection Methods ............ 56
  3.4.2 Accuracy Versus Number of Features per Landmark ....... 57
3.5 Conclusions .............................................................. 58
3.6 Acknowledgements ..................................................... 60
3.7 References .............................................................. 61

4 Conclusion ................................................................. 64

4.1 Summary and Conclusions ............................................. 64
4.2 Future Work .............................................................. 65
  4.2.1 Generic SLAM ..................................................... 65
  4.2.2 Algorithm Speed .................................................. 66
4.3 Potential Future Applications ......................................... 66
4.4 References .............................................................. 68

Appendices
A Stereo Vision Based Swing Angle Sensor for Mining Rope Shovel

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1 Introduction</td>
<td>69</td>
</tr>
<tr>
<td>A.1.1 Tracking Swing Motor Shaft Rotations with an Encoder</td>
<td>70</td>
</tr>
<tr>
<td>A.1.2 Counting Swing Gear Teeth with Inductive Sensors</td>
<td>70</td>
</tr>
<tr>
<td>A.1.3 Measuring Swing Angle with Stereo Vision</td>
<td>70</td>
</tr>
<tr>
<td>A.1.4 Contributions</td>
<td>71</td>
</tr>
<tr>
<td>A.2 Related Work</td>
<td>72</td>
</tr>
<tr>
<td>A.2.1 Feature Types</td>
<td>72</td>
</tr>
<tr>
<td>A.2.2 Approaches For Finding Camera Pose</td>
<td>73</td>
</tr>
<tr>
<td>A.3 Method</td>
<td>74</td>
</tr>
<tr>
<td>A.3.1 Algorithm Design and Overview</td>
<td>74</td>
</tr>
<tr>
<td>A.3.2 EKF Overview</td>
<td>75</td>
</tr>
<tr>
<td>A.3.3 Initialization and Calibration</td>
<td>76</td>
</tr>
<tr>
<td>A.3.4 3D Feature Detection</td>
<td>76</td>
</tr>
<tr>
<td>A.3.5 Feature Cluster Landmarks</td>
<td>79</td>
</tr>
<tr>
<td>A.3.6 Landmark Measurement</td>
<td>81</td>
</tr>
<tr>
<td>A.3.7 Swing Angle Computation</td>
<td>83</td>
</tr>
<tr>
<td>A.3.8 Landmark Feature Management</td>
<td>83</td>
</tr>
<tr>
<td>A.4 Results</td>
<td>84</td>
</tr>
<tr>
<td>A.5 Conclusions</td>
<td>86</td>
</tr>
<tr>
<td>A.6 References</td>
<td>87</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Computation time for a sample frame ........................................... 51
3.2 Accuracy versus number of features in landmark cluster ..................... 58
# List of Figures

1.1 Haul truck loading by a rope shovel .............................................. 1
1.2 Pinhole camera model ................................................................. 4
1.3 Virtual front image plane camera model ........................................ 4
1.4 Stereo camera geometry ............................................................... 5

2.1 Shovel movements and the locations of the hardware components installed on the shovel ......................................................... 12
2.2 The saddle block ........................................................................... 12
2.3 Biaxial accelerometers mounted to boom joint .................................... 14
2.4 Accelerometers and range finder ...................................................... 14
2.5 Rope shovel and laser scanner image ................................................ 15
2.6 Swing angle camera mounting location .............................................. 17
2.7 Swing angle camera image sequence ................................................ 17
2.8 Haul truck pose estimation hardware .............................................. 18

3.1 Haul truck loading by a rope shovel ................................................ 23
3.2 Camera installation ........................................................................ 24
3.3 Sample camera image sequence ....................................................... 24
3.4 Loop closure illustration ................................................................. 29
3.5 Algorithm flow overview ............................................................... 31
3.6 Corner measure versus image contrast ............................................. 35
3.7 Percent Threshold corner selection in shadow pattern scene ............... 36
3.8 Locally Maximal corner selection in shadow pattern scene .................. 36
3.9 Percent Threshold corner selection during transition from shadow to directly-lit scene .......................................................... 37
3.10 Locally Maximal corner selection during transition from shadow to directly-lit scene ............................................................ 37
3.11 A sample Cluster Feature landmark ................................................ 40
3.12 Relative pose estimation illustration .............................................. 41
3.13 A snapshot of the system map ....................................................... 43
3.14 Feature matching search windows ................................................ 44
3.15 Match results after geometric verification ....................................... 46
3.16 Motion vectors drawn using the found relative landmark pose estimation 47
I would like to thank my supervisors Drs. Peter Lawrence and Rob Hall for their guidance and encouragement throughout this work. I would also like to express my gratitude to my colleagues and friends James Borthwick, Nicholas Himmelman, Ali Kashani, Andrew Rutgers, and Nima Ziraknejad for the helpful discussions and invaluable feedback.

Also, special thanks to Teck Inc.’s Highland Valley Copper operation for letting us work with its mining rope shovels. I greatly appreciate its employees for their kind assistance during our field experiments.

I would also like to thank the Natural Sciences and Engineering Research Council of Canada for its financial support of this project.

Finally, I would like to thank my family for their continual encouragement throughout my studies.
Statement of Co-Authorship

This thesis presents research performed by Li-Heng Lin.

- The introduction chapter was written by Li-Heng Lin with feedback and suggestions from his supervisors Peter Lawrence and Robert Hall.

- Chapter 2 was written by Li-Heng Lin, Nicholas Himmelman, Ali Kashani, and James Borthwick with feedback and suggestions from Peter Lawrence and Robert Hall. Li-Heng Lin was primarily responsible for Section 2.3 which describes the concept and the hardware of the work presented in this research. He also collaborated to write the introduction.

- Chapter 3 was written by Li-Heng Lin with feedback and suggestions from Peter Lawrence and Robert Hall. The research and data analysis were performed by Li-Heng Lin.

- The conclusion chapter was written by Li-Heng Lin with feedback and suggestions from his supervisors Peter Lawrence and Robert Hall.

- Appendix A was written by Li-Heng Lin with feedback and suggestions from his supervisors Peter Lawrence and Robert Hall. The research and data analysis were performed by Li-Heng Lin.

The research program was identified and initiated by Peter Lawrence, Robert Hall and Tim Salcudean.
Chapter 1

Introduction

1.1 Introduction

This thesis presents a stereo vision Simultaneous Localization and Mapping (SLAM) algorithm for robustly determining the camera pose under outdoor lighting conditions. The algorithm has been developed in the context of a project related to mining shovel safety.

A key process at open-pit mines is the digging of mineral rich material and the loading of the material into haul trucks by large electric rope shovels (Figure 1.1). Two types of collisions can occur during this process: 1) the shovel bucket hitting the haul truck and, 2) the bucket hitting the shovel’s own protruding tracks while the bucket is being lowered for a dig.

![Figure 1.1: A shovel swung from the digging bank to its left to load a haul truck.](image)

A proof-of-concept collision avoidance system is being developed with the aim of providing advance warning of these collisions to the shovel operator [1]. A set of easily retrofittable sensors that measure the shovel kinematic variables and the truck pose is being developed. The instrumenta-
tion is divided into three measurement subsystems: 1) Truck Pose Estimation [2], 2) Shovel Arm Geometry [3], and 3) Shovel Swing Angle, i.e. the angle between the revolvable shovel house and carbody about the shovel’s vertical rotation axis. The stereo vision SLAM algorithm developed in this work has been applied to finding the shovel swing angle.

For research use, the sensors need to be easily retrofittable onto the shovel and easily removable. They should not require significant modification to the shovel or cause interference to existing shovel systems. The reason is that there is no dedicated research shovel. A shovel is only available for several hours each time it is down for scheduled maintenance. Moreover, the same shovel may not always be available. As well, sensors need to be removed after each experiment. A set of easily retrofittable and removable sensors allows experiments to run easily on any shovel, depending on availability.

For obtaining the swing angle, various mechanical and electrical options have been considered but all were deemed to be too invasive or time consuming for installation. For example, an encoder was installed on a shovel to track the shovel’s swing motor shaft rotations which in turn can be used to calculate the swing angle. The installation process took over 4 hours and included taking apart the motor, attaching a stub shaft to the motor shaft, truing the stub shaft, attaching the encoder to the stub shaft, and installing an assembly that keeps the encoder stationary while the swing motor shaft turns. This process will have to be repeated for each new machine that is available, or when maintenance is done on the swing motor. Moreover, if the new shovel is of a different model, a redesign may be required since not all shovels have the same motor design.

Computer vision i.e. the use of cameras as sensors, offers the ability to non-invasively measure a shovel’s swing angle with little installation requirement. It is also possible to design a vision-based swing angle sensor that works regardless of differences in the shovel mechanical design. The proposed novel swing angle sensing system consists of a stereo camera made by Point Grey Research, Inc. based in Vancouver, BC. The stereo camera is clamped externally to the bottom outer edge of the shovel house, aimed toward the shovel carbody, so that the carbody fills the view of the camera. As the shovel house swings from its $0^\circ$ swing angle position, the camera revolves with the house in a circular orbit about the shovel house’s vertical swing axis, seeing differing views of the carbody. Using the stereo vision SLAM algorithm, the camera pose relative to the start is tracked in real-time. The tracked camera position is in turn used to find the swing angle.

The SLAM approach for finding the camera pose does not require an a priori model of the shovel. While the shovel swings, the stereo camera records observed 3D feature points on the carbody as landmarks, and incrementally builds a 3D map of the landmarks as it revolves around the carbody. At the same time, the camera localizes itself by measuring its position with respect to the landmarks in the map. Further details regarding SLAM are discussed in Section 1.2.4.

A stereo camera has been chosen because it offers the ability to directly measure the 3D positions of observed features, as opposed to a monocular camera which cannot. For determining the camera pose, the SLAM approach has advantages over other methods such as Model-based Camera
Localization [4, 5, 6] or 3D Motion Tracking [7]. The disadvantage of the Model-based approach is that it requires an \textit{a priori} model of the shovel. For the 3D Motion Tracking approach, the camera pose estimate accumulates drift error over distance travelled by the camera. The SLAM approach has neither of these disadvantages. The 3D map of observed features is essentially a model of the shovel that is being built online. Moreover, the map accuracy improves as more measurements are made. Drift error is limited and is reduced as the map converges. Detailed comparisons are presented in Chapter 3.

1.2 Background

1.2.1 Pinhole Camera Model

A camera can be modelled using the pinhole camera model. A pinhole camera consists of an image plane/sensor (eg. CCD) enclosed within a box, located flat against the backside of the box. This box blocks out all light, except for a small pinhole located in the front of the box that allows light rays to enter and strike the image plane. This pinhole is infinitesimally small such that it allows only one light ray from each point in the world to enter the pinhole. The geometry of such a model is illustrated in Figure 1.2. A light ray from a point $P$ in the world enters through the camera pinhole and projects to point $q$ on the image plane. Notice that the projected image of the world is inverted on the image sensor. In fact, given a camera coordinate system defined with origin $O_c$ at the pinhole and $z$-axis pointing outward from the camera, and an image coordinate system as shown in the figure, the point $P$ at $\{x, y, z\}$ in camera coordinates, must have its image projection $q$ at the following image coordinates $\{u, v\}$:

$$
\begin{align*}
  u &= -fx/z + u_0 \\
  v &= -fy/z + v_0
\end{align*}
$$

Here, $f$ is the camera focal length defined as the distance between the pinhole and the image plane. The coordinates $\{u_0, v_0\}$ is the location where a line through the pinhole (optical axis) intersects the image.

Most digital cameras automatically invert the image coordinates when outputting a digital image so that the image appears upright. This is equivalent to having a virtual image plane in front of the camera pinhole instead of behind it (Figure 1.3). The equation describing the projected image position of world point $P$ is then changed to:

$$
\begin{align*}
  u &= fx/z + u_0 \\
  v &= fy/z + v_0
\end{align*}
$$

This virtual front image plane model is used throughout this work.
1.2.2 Stereo Vision

As seen from equation 1.2, it is not possible to recover the 3D position of the world point $P$ given its image location $q$ in a single camera. However, a pair of identical cameras in parallel, separated by a horizontal (baseline) distance $B$ allows $P$’s 3D position to be triangulated. This setup of two parallel cameras is called a stereo camera.

Figure 1.4 shows a stereo camera consisting of a left and a right pinhole camera, viewed downwards from top. The camera coordinate system is based on the right camera with the origin located at the right pinhole. Given a world point $P$ at camera coordinates $\{x, y, z\}$, its right and left image projections $q_r, q_l$ will be located at:

$$q_r : \{u_r = fx/z + u_0, \ v_r = fy/z + v_0\}$$

$$q_l : \{u_l = f(x + B)/z + u_0, \ v_l = fy/z + v_0\}$$  \hspace{1cm} (1.3)

Note that the vertical image coordinates of the two image projections are identical.

Conversely if the image coordinates of $P$’s left and right projections are known then $P$’s 3D coordinates can be found as:

$$d = u_l - u_r$$

$$z = fB/d$$

$$x = (u - u_0) \ z / f$$

$$y = (v - v_0) \ z / f$$  \hspace{1cm} (1.4)

where $d$ is called the disparity between the left and right image projections.

Thus, for an image point $q_r$ in the right image, if its correspondence $q_l$ in the left image can be found, the 3D position of the world point $P$ which $q_r$ corresponds to can be calculated. As mentioned, $q_l$ lie on the same horizontal image coordinate as $q_r$. Thus, to search for $q_r$’s correspondence $q_l$, one only needs to search along the same image row (scanline) in the left image.
This process is called stereo matching and more detail can be found in [8].

In conclusion, given the successful stereo matching of an image point in the right camera, the 3D position of the world point corresponding to the image point can be triangulated. Thus, a stereo camera can be considered as a 3D sensor.

Throughout this discussion the left and right cameras of a stereo camera are assumed to be identical and exactly parallel. The cameras are also assumed to be ideal pinhole cameras without radial distortion caused by their lenses. This is of course inaccurate in reality. However, the stereo images can be transformed as if they were taken by two parallel and ideal pinhole cameras. This process is called Stereo Rectification [9]. The stereo images from the Point Grey stereo camera used in this work have been rectified using Point Grey provided code prior to processing by our algorithm.

1.2.3 Features

Often, not all portions of a camera image contain useful information. For example, image pixels that are part of a white wall do not provide useful information. These pixels lie in a large uniform grayscale intensity image region and are nearly identical to their neighboring pixels so they cannot be well-localized, or matched in subsequent camera images when the camera moves. On the other hand, features are portions of the image that contain structure so they can be localized and subsequently matched. Examples of features are edges, or well-defined points in the image such as a dot. Most computer vision algorithms, including this work, first detect features in the image then process the detected features only.

Interest points are features having well-defined 2D positions in the image. These features are the most commonly used. The most important property of an interest point detector is its repeatability. It is the ability of the detector to consistently detect the same physical features
given changes in imaging conditions such as illumination, image rotation, and viewpoint. Various
interest point detectors were presented and compared in [10, 11]. The Harris corner detector [12]
was found to have the highest repeatability for a wide range of imaging conditions. However, even
its repeatability is limited, especially for viewpoint changes. With the Harris corner detector, only
40% of features were repeatably detected given a 50° viewpoint change [10].

The Harris corner detector was used in this work. This detector evaluates the “cornerness” of
each pixel location in the image. A corner measure threshold is set by the user. Only pixel locations
exceeding this threshold are selected as features. However, it was found that an appropriate corner
threshold is difficult to set given an outdoor environment with directional sunlight causing shadows
and highly non-uniform scene lighting. The reason is that corner measures can vary with image
exposure or contrast. This is explored in Chapter 3.

Once features are selected, for each feature an image region centered about the feature’s location
is used to compute the feature’s descriptor. This descriptor is an unique identifier for the feature.
It is used to establish correspondences between the feature and features detected in subsequent
images.

An ideal descriptor is one that is invariant to changes in image brightness level, contrast,
rotation, and view point. Different techniques can be used to make descriptors invariant or partially
invariant to these changes [13]. For example, the SIFT descriptor [14] is computed as a histogram
of image gradients in an image region centered about the feature. The image gradient removes the
effects of image brightness level. The histogram values are normalized to remove image contrast
effects.

1.2.4 Simultaneous Localization And Mapping

The Simultaneous Localization And Mapping (SLAM) problem assumes that a vehicle or a robot
starts in an unknown environment. The robot is only equipped with sensors that take relative
measurements of the environment rather than global sensors such as a GPS. Examples include a
laser scanner or a camera mounted to a robot which allows the robot to measure feature positions
relative to the robot’s coordinate frame. The goal for the robot is to incrementally build a consistent
map of the environment while simultaneously to localize itself with respect to this map. Observed
features are recorded as landmarks into the map. As the robot explores, it re-measures the relative
position of existing landmarks to localize itself, and adds newly observed features as landmarks
when it moves into a new area.

A key goal of SLAM is the ability to loop-close and build a consistent map. As the robot
leaves its starting position and traverses in a large loop, its positional estimate will accumulate
drift error. New landmarks initialized by the robot will have this drift error as well. However, once
the robot returns to its starting position and re-detects the initial landmarks, the robot should be
able to correct the drift error in its positional estimate. This correction should also propagate to
the positional estimates of the newer landmarks. The is called loop-closing. Any SLAM solution
must allow loop closure to occur.

The problem is well studied, and solutions that allow loop-closure exist. An overview of the problem and solutions can be found in [15, 16, 17]. A major approach is to use a single Extended Kalman Filter (EKF) to filter the robot pose and the landmark positions. This is the approach taken in this work. The EKF approach provides good loop-closing performance but its computational complexity is $O(N^2)$, where $N$ is the number of landmarks in the map. Thus, there is a limit on the number of landmarks that can be maintained for real-time performance. This limit can reduce an EKF SLAM algorithm’s robustness, as explained below.

A variety of sensors can be used in the SLAM problem, such as a laser scanner, a sonar, or a camera. However, the use of vision in SLAM is attractive due to the camera’s high data rate, and good resolution data. Solving EKF SLAM with vision has been demonstrated in [18, 19, 20]. However, these vision algorithms were not shown to work in outdoor environments with extreme directional sunlight. These works use a single feature as a single landmark. For real-time performance, the total number of landmarks is limited so only a few features can be tracked as landmarks. For example, the number of visible landmarks at any moment was limited to 12 in [20]. Thus, for robust operation, these algorithms require that the few features used as SLAM landmarks are reliable. That is, these single feature landmarks must be real physical features that are stationary, consistently detectable, and unique enough to be reliably matched. This is generally difficult due to the limited repeatability of the feature detector. Features can also become occluded. Non-distinctive features may also make up a large portion of the scene. Reliable camera localization is even more difficult to achieve in outdoor environments with extreme directional sunlight. Directional sunlight causes shadows and highly non-uniform scene lighting. In this kind of environment, landmark features can be features detected on shadow outlines or patterns. These non-physical landmarks can cause confusion to the SLAM algorithms as the physical world positions of these shadow features can shift. Moreover, with a non-uniformly illuminated scene, landmark features can easily become under- or over-exposed. For example, as the camera view shifts toward a directly illuminated scene region from a scene region under shadow, the camera dramatically reduces its shutter time to maintain a fixed average image exposure. Just previously observable landmark features in the shadow scene region can quickly become under-exposed and unmatchable. This reduction in the already low number of matchable landmarks can further degrade the robustness of these algorithms.

### 1.3 Thesis Context, Outline and Contributions

#### 1.3.1 Thesis Goal

The aim is to develop a real-time, stereo vision SLAM algorithm that can robustly determine the camera pose under outdoor lighting conditions. This algorithm should also offer the ability to quickly and easily obtain the shovel swing angle as required by the overall shovel collision avoidance
1.3.2 Thesis Outline

The thesis is organized as follows:

1. Chapter 2 presents a high-level overview of the sensing systems for the collision avoidance system. This chapter frames the thesis work in the context of the overall research project.

2. Chapter 3 describes the stereo vision based SLAM algorithm in detail. Results of using the algorithm to find shovel swing angle are reported.

3. Chapter 4 contains concluding remarks about the system, and discusses the strengths and weaknesses of the work and potential areas for future work.

4. Appendix A is an earlier, shortened report of this work that has been submitted to a conference. It was written before Chapter 3 and it contained earlier results only.

1.3.3 Contributions

The contributions of this work are:

1. An enhanced stereo vision SLAM algorithm that is robust even in the presence of directional sunlight causing shadows and non-uniform scene illumination. The enhancements include:
   - A “Locally Maximal” Harris corner selection method that is not directly based on a corner measure threshold. This corner selection method enables features to be selected more consistently in an outdoor lighting environment compared to a prevalent threshold-based corner selection method. The Locally Maximal corner selection method also selects corners evenly across the image and detects a stable number of features across frames.
   - The use of high level, 3D “Feature Clusters” as SLAM landmarks rather than single features, for robust and consistent landmark measurements in the outdoor lighting environment. The consistently matchable Feature Cluster landmarks reduce the number of landmarks needed in the SLAM map and so reduces the SLAM filter computational cost significantly.

2. A swing angle measurement system that is easily and quickly retrofittable to mining rope shovels or other large rotating machines.
1.4 References


Chapter 2

Rope Shovel Collision Avoidance System

2.1 Introduction

Rope shovels are used extensively in open pit mining to extract material from the earth and load it into haul trucks. The rate at which they are able to extract and load the material is often the limiting factor in a mine’s throughput, and as such, the shovels need to be run continuously in order to meet production targets. Unfortunately, the truck loading process is not without risk, as the dipper can collide with the haul truck during loading and self collisions are possible between the dipper and the caterpillar tracks. Although collisions do not typically occur on a daily or even weekly basis, when they do occur they can result in serious injury to the truck driver and expensive downtime and machine repairs for the mine. A system that is capable of warning the shovel operator when the dipper is on a collision course with a haul truck or with the shovel tracks could significantly reduce the likelihood of collisions, and therefore be of significant value to a mine.

2.1.1 System Overview

During the loading process, we assume that the truck is parked beside the shovel and that the shovel’s tracks are stationary. In order to avoid collisions during loading, we must determine the exact state of the shovel and the exact position of the truck. The state of the shovel is defined by the current position, velocity, and acceleration of the joints; the position of the truck is defined as its position relative to the shovel carbody. The task of determining the state of the shovel and position of the truck has been divided into three subsystems: shovel arm geometry, shovel swing angle, and truck location.

---

This is not the first attempt to instrument an excavator or mining shovel [1, 2, 3] or to locate a haul truck [1, 4, 5]. It is, however, to the best of our knowledge, the first published work describing a sensing system for a real-time collision avoidance system for a full-scale electric rope shovel.

In order to reduce the costs associated with installation and maintenance, the collision avoidance system is designed such that all the subsystems are installed on a shovel. This allows the data collected by each subsystem to be easily gathered by a central computer where collisions will be predicted. The information collected could be used to warn the operator of an impending collision, and if no corrective action is taken, briefly override the operator’s controls to prevent a collision. The following sections describe the hardware components and the data collected by each of the three subsystems. An overview of the hardware layout comprising all subsystems is shown in Figure 2.1.

**Figure 2.1:** Shovel movements and the locations of the hardware components installed on the shovel: A1) accelerometers placed on the rotary joints, A2) rangefinder placed on the saddle block prismatic joint, C) central computing system located in house underneath operator cab, L) planar laser scanner placed below boom foot, S) stereo camera placed underneath house looking inwards, and T) truck localization stereo camera mounted on boom.

### 2.2 Arm Geometry

Obtaining arm geometry variables is an important step towards building the collision avoidance system. The arm geometry of the shovel is determined by the position of the boom, the extension of the dipper handle, and the length of the hoist cables. The boom angle is set by adjusting the length of the boom suspension ropes whose length is kept constant during operation but can change slightly when the boom is lowered for maintenance.

During typical operation the arm geometry is controlled by the crowd and hoist functions. To determine the location of the dipper in space and the rate at which the joints are moving, the
arm geometry can be directly measured in two ways. The hoist rope length and the dipper handle extension can be used to determine the location of the dipper. One difficulty that arises when measuring the hoist rope length is estimating the stretch in the hoist rope as it depends on the load and arm geometry. Alternatively, the angle of the dipper handle with respect to the boom can be used with the dipper handle extension to locate the dipper. By measuring the angle of the dipper handle with respect to the boom the uncertainty associated with stretch in the hoist rope can be avoided.

The angle of the dipper handle with respect to the boom can be measured between the saddle block and the boom. The saddle block acts as a brace which keeps the dipper handle on the shipper shaft. As the dipper is moved, the saddle block pivots with the dipper handle, making it ideal for measuring the angle of the dipper. One problem related to measuring the dipper handle angle on the saddle block is that the saddle block can twist back and forth on the dipper handle as the hoisting direction changes or as the direction of the torque applied to the shipper shaft changes. The way the saddle block moves on the dipper handle is shown in Figure 2.2.

Traditionally, joint angle or extension sensors are used in forward kinematics to determine the workspace location of an end-effector [6]. In this work, the arm geometry was measured both directly and indirectly: Directly by placing joint angle and extension sensors at each joint (distal sensors) and indirectly by tracking the position of the dipper with a laser scanner (proximal sensor) and using inverse kinematics to determine the dipper handle angle, dipper handle extension, and hoist rope length. Advantages of using distal sensors are:

- Complex algorithms are not required for obtaining the arm geometry.
- The processing simplicity makes the required computing power minimal.
- Visibility of the dipper is not required as it is with a proximal sensor where occlusion can degrade the arm geometry measurements.
- Distal sensors may be less sensitive to environmental factors such as dust and lighting.

Advantages of using a proximal sensor are:

- Proximal sensors are less vulnerable to mechanical damage compared with distal joint sensors [7].
- Linkage singularities cannot cause numerical instabilities in calculations when using a proximal sensor, as they can for distal joint sensors [6].
- In some excavators, the dipper is not rigidly connected to the body, making forward kinematics impossible [8].
2.2.1 Distal: Joint Sensor Based

The first method for measuring the arm geometry used sensors placed at each of the joints. Pairs of biaxial accelerometers were used to measure the angles of the rotary joints (boom and dipper handle) and a laser rangefinder was used to measure the extension of the prismatic joint. The accelerometer-based angle measurement system used is part of a payload measurement system for hydraulic shovels called LoadMetrics, designed and manufactured by Motion Metrics International [9]. To measure the joint angle one biaxial accelerometer is placed on the proximal link of the joint and the other is placed on the distal link of the joint. Figure 2.3 shows the accelerometers installed on the boom joint. A second pair of accelerometers was installed on the saddle block joint to measure the dipper handle angle. The accelerometers are connected to a Motion Metrics International LoadMetrics computer that runs an algorithm which determines the difference in angle between the two sensors with an accuracy of $\pm 1^\circ$. An offset is used to adjust the measurement according to the placement of the sensors.

The laser rangefinder used to measure the extension of the dipper handle was a SICK DT 500. It has an accuracy of $\pm 3$mm between 0.2 and 18 m on a surface with 6% reflectivity (black). The rangefinder was mounted on the saddle block aiming along the dipper handle, away from the dipper. A target was mounted on the end of the dipper handle opposite the dipper. The rangefinder was connected to the same computer used for the angle sensors via an RS422 serial connection. An offset was used to adjust the distance measured by the rangefinder to make it correspond to the dipper handle extension. Figure 2.4 shows the installation of the rangefinder and target on a shovel.

![Figure 2.3: Biaxial accelerometers mounted to boom joint.](image)

![Figure 2.4: The left image shows the installation of the rangefinder (indicated by white bounding box) on the saddle block. The right image shows a target used on the end of the dipper handle with a white arrow depicting the laser beam path.](image)

2.2.2 Proximal: Laser Scanner Based

 Rather than using distal joint sensors, a proximally-mounted imaging sensor can be used to track the dipper. This section focuses on the development and implementation of a method for estimating the dipper location using a laser scanner.
To be operational on a mining shovel, a range sensor must be: reliable with a Mean Time Between Failures (MTBF) of greater than 1 year [10], accurate within 1% of the measurement range [10], and capable of range measurements up to 50m [11].

Computer vision, radar, and laser scanning have all been used for object tracking as well as dipper localization. While radars can provide the reliability and range, they are slow and expensive compared with laser scanners and cameras [10, 12]. Winstanley et al. successfully utilized a laser scanner for their research after finding cameras to be less reliable than laser scanners when faced with environmental factors such as rain, dust, and shadows.

Laser scanners provide information in the form of point-clouds which are a set or array of points that describe object outlines and surfaces in the workspace. Each data point represents a vector from the scanner origin to the intersection of the laser beam and an object in the workspace. Typically, as in the laser scanner used in this work (SICK LMS 221), the laser beam is rotated to produce 2D measurements in a scan plane. Hence, the output can be described in a 2D scanned plane of the environment [13].

The laser scanner was mounted vertically underneath the boom pointing toward the dipper. In this position, the laser scanner maintained a consistent view of the dipper and occlusions were minimal. A typical scan plane of the laser scanner at the given position is provided in Figure 2.5 b) where the shovel boom, dipper, ground, and truck are evident in the point-cloud. The laser scanner provides 40 readings per second via an RS422 serial connection with a resolution of 1cm, accuracy of 5cm, and maximum viewing range of 80m.

Figure 2.5: a) A P&H 2800 Electric Rope Shovel. b) A sample laser scanner image superimposed on equipment diagrams. The laser scanner is mounted vertically underneath the shovel boom. The laser scanner’s point cloud image, shown by a series of dots, represents a contour of the environment. Note that the above image is not taken during a normal digging cycle. Here, the laser sees the back of the truck, whereas during a normal loading sequence, the truck is located sideways and the laser only sees its side.
2.3 Swing Measurement

Swing angle, the angle between the house and the lower carbody, is one of the measurements required for the computation of the dipper’s 3D position in workspace. Without swing angle, the collision avoidance system cannot determine the dipper’s position and cannot predict common collisions such as the collision of the dipper with the excavator tracks or external objects such as a truck. Unfortunately, many shovels do not have a swing angle measurement system in place and one must be designed and retrofitted for this project.

2.3.1 Encoder-Based Swing Measurement

One method for obtaining the swing angle is to use an encoder to track the change in rotation angle that the swing motor shaft makes. For this purpose, a 100 line BEI HS35F encoder was attached to the shaft of one of the swing motors. In this configuration each quadrature increment on the encoder corresponds to a house rotation of 0.002°. The encoder was mounted to a stub shaft which was attached to the motor shaft, rising through the brake assembly. The brake assembly was removed to allow the stub shaft to be mounted to the motor shaft. The stub shaft had to be trued to the motor shaft to minimize wobble when the motor was spinning to prevent damage to the encoder. This laborious process would have to be repeated any time motor maintenance was required. A flexible coupling could be used to connect the encoder to the motor shaft but this would require a more complex mounting assembly which would in itself impede maintenance.

2.3.2 Camera-Based Swing Measurement

Given the drawbacks of using an encoder, a novel camera-based swing angle sensor which can be easily retrofitted and does not get in the way of regular maintenance was investigated. The proposed swing angle sensor consists of a Point Grey Bumblebee 2 stereo camera mounted on the bottom skirt of the excavator housing, looking down, toward the center of the carbody (Figure 2.6). As the house rotates, the camera will rotate with the house and revolve around the stationary carbody, seeing differing views of the carbody (Figure 2.7). An ethernet cable carries digital video images from the camera to a computer in the house. The computer analyzes the images in real-time and determines the angle from which the camera is viewing the carbody. The desired swing angle accuracy of the system is ±0.25 degrees which corresponds to approximately ±10 cm error in dipper localization.

For easy retrofitting, we are designing a system such that the camera need not be exactly positioned or angled when mounted to the excavator. As long as the lower carbody covers most of the view of the camera, the system will function properly. Further, the swing sensor should not need a prior model of the excavator as there are many variations of shovels. Thus the swing angle sensor automatically learns the 3D features on the carbody and calibrates the camera position and orientation with respect to the swing rotation axis.
2.4 Truck Localization

Once the shovel is fully instrumented, the task still remains of precisely locating the haul truck to avoid. As with the system developed by Stentz [1], we wish to determine the truck’s exact position, or “pose”, which can be fully described by six parameters – three for translation and three for rotation. However unlike [1], we require the system to work in real time without imposing new requirements or restrictions on how the shovel operator digs or loads a truck. The system must work in the background to quickly and accurately determine the six position parameters.

As stated previously, a goal was to place all equipment for the collision avoidance system on the shovel. This requirement restricts us from installing beacons or specialized GPS equipment on the trucks. As such, the use of a shovel-mounted 3D imaging sensor was seen as the best solution. Several 3D imaging sensors, namely stereo cameras, laser scanners and millimeter-wave radar units, were considered. Laser scanners and radar units were attractive because they deliver highly accurate 3D measurements, but unfortunately, also have slow image acquisition rates and low resolution images [1, 11]. Stereo cameras, meanwhile, deliver high-resolution 3D images with a high acquisition speed. However, stereo cameras suffer from the fact that their depth accuracy falls off exponentially with the distance between the camera and the measured object. This stems from the fact that stereo cameras work using the same triangulation principle as the human visual system. In order to mitigate this weak point, we selected a stereo camera with high-resolution sensors.
(1280 × 960) and a large separation between cameras (24 cm). This will allow for a triangulation depth accuracy on the order of ten centimeters at our working distance of about sixteen meters.

We chose to mount the camera high on the side of the shovel boom, far from potential sources of dust and damage. As the camera is not intended for use in outdoor or rough environments, we constructed a waterproof, heated camera enclosure and secured the camera on shock mounts. Figure 2.8 shows the camera’s location on the boom, its enclosure, and its view of a haul truck.

![Figure 2.8](image)

**Figure 2.8:** The left image shows a white arrow pointing at the location on the boom where the stereo camera is mounted. The middle image shows the stereo camera, mounted on the boom, in its protective enclosure. The right image shows the view of a haul truck delivered by the stereo camera.

The data produced by a stereo camera is called a “point cloud”, which is the same as a regular image except that for each pixel the precise \((x, y, z)\) location relative to the camera (and hence the shovel) is known. For the system to function, it must know which areas of the cloud represent the truck, as they must be avoided, and which represent the payload of the truck, as they may be gently brushed against by the dipper. Additionally, the \((x, y, z)\) measurements of the point cloud will contain errors which must not confuse or overwhelm the system.

What we wish to achieve is to be able to use this data to locate the truck from distances on the order of fifteen meters with an accuracy of about ten centimeters. Furthermore, this must be accomplished quickly enough to operate within the real-time time constraints of a collision avoidance system. We believe that the described hardware platform and resultant data will provide the basis for such a system.
2.5 Discussion and Conclusions

Three measurement subsystems have been described for a system to prevent collisions between the shovel’s dipper and a haul truck, or between the shovel’s dipper and its own tracks. Together, the three subsystems can provide to a collision avoidance system the position of the dipper, and the position of the haul truck with respect to the shovel’s carbody. The sensing subsystems are for:

- **Arm Geometry**, which measures the position of the dipper relative to the revolving frame. Two different approaches to obtain this information have been developed: a joint sensor based method which can be compared to the results from a laser scanner based method.

- **Swing Angle**, which measures the angle between the revolving frame and the fixed frame. This system relates the dipper position as found from the Arm Geometry subsystem, to the track body frame. We have also developed two approaches here: an encoder-based angle measurement subsystem which can be compared to the camera-based swing angle measurement subsystem.

- **Haul Truck Localization**, which measures the pose of a haul truck with respect to the revolving frame. A camera based approach to localize the truck has been developed.

All the sensors have been developed so that they could be easily retrofitted. They are all attached externally to the shovel without requiring the shovel to be disassembled or extensively modified. A practised team could install all the hardware components in one 12 hour shift.

These sensors have been installed and tested on a shovel at the Highland Valley Copper mine. The current installation described here forms the test bed for determining the most appropriate set of sensors, and for the development of the collision avoidance system itself. Results will be presented at the conference.
2.6 References


Chapter 3

Enhanced Stereo Vision SLAM For Outdoor Heavy Machine Rotation Sensing

3.1 Introduction

This paper presents a stereo vision Simultaneous Localization and Mapping (SLAM) algorithm that estimates the camera pose reliably in an outdoor lighting environment. The algorithm has been developed in the context of a vision-based swing angle sensor for mining rope shovels, as explained below.

In open-pit mines, large electric rope shovels dig mineral-rich material and dump it into haul trucks, which upon being filled carry the material to a mineral extraction plant (Figure 3.1). For experimental collision avoidance, automation, and performance evaluation of these mining rope shovels, a set of easily retrofittable sensors are being developed to measure a rope shovel’s kinematic variables and the haul truck pose [1], [2], [3]. For research use, the sensors need to be easily retrofittable onto the shovel and easily removable. They should not require significant modification to the shovel or cause interference to existing shovel systems. The reason is that there is no dedicated research shovel. A shovel is only available for several hours each time it is down for scheduled maintenance. Moreover, the same shovel may not always be available. As well, sensors need to be removed after each experiment. A set of easily retrofittable and removable sensors allows experiments to run easily on any shovel, depending on availability.

This paper focuses on a measurement subsystem for obtaining a shovel’s swing angle, i.e. the angle between the shovel house (upper, revolvable portion of the shovel) and the shovel carbody (lower, non-revolvable portion of the shovel) about the shovel’s vertical swing axis. Various me-

Figure 3.1: A shovel swung from the digging bank to its left to load a haul truck.

Mechanical and electrical options have been considered but all were deemed to be too invasive or time-consuming for installation. For example, it took 4 hours to install an encoder-based swing angle sensor that tracks the rotation angle of a shovel’s swing motor shaft. The design of the sensor may also need to be revised for different models of shovels that have different mechanical designs.

However, vision offers the ability to non-invasively measure a shovel’s swing angle with little installation requirement. It is also possible to design a vision-based swing angle sensor that works regardless of differences in the shovel mechanical design. The proposed novel swing angle sensing system consists of a Point Grey Bumblebee 2 stereo camera clamped externally to the bottom outer edge of the shovel house, aimed toward the shovel carbody, so that the carbody fills the view of the camera (Figure 3.2). As the shovel house swings from its $0^\circ$ swing angle position, the camera revolves with the house in a circular orbit about the shovel house’s vertical swing axis, seeing differing views of the carbody. In real-time, a computer in the shovel house analyzes the camera images and tracks the camera pose relative to the start. The tracked camera position can be in turn used to calculate the swing angle. Sample camera images can be seen in Figure 3.3.

The problem of estimating the camera pose is solved using the Simultaneous Localization and Mapping (SLAM) approach. An a priori model of the shovel is not required. While the shovel swings, the stereo camera records observed 3D feature points on the carbody as landmarks, and incrementally builds a 3D map of these features as it revolves around the carbody. At the same time, the camera localizes itself by matching observed features in the camera view to the landmarks in the map.

Such a camera system can be easily and quickly retrofitted. The camera mounting location is easily accessible from the shovel tracks. Furthermore, the vision algorithm is flexible such that it does not require the camera be mounted precisely at a pre-determined position or view direction. The camera can be mounted freely, as long as the carbody fills the majority of camera view. After
Figure 3.2: Clockwise from top-left to bottom-left. 1) White bounding box indicates the location of the stereo camera. 2) Closer view from the side of the shovel. 3) Side view of the stereo camera taken while standing on top of the shovel track. 4) Close-up view of the stereo camera.

Figure 3.3: Sequence of images from the camera as the shovel swings.

mounting, it is also unnecessary to make physical measurements of the camera position and view direction. Installation can be performed in 30 minutes. Moreover, the vision system can work on a variety of machines upon installation.
3.1.1 Contributions

This paper presents an enhanced stereo vision SLAM algorithm that permits reliable camera pose estimation in the presence of directional sunlight illumination causing shadows and non-uniform scene lighting. To achieve this, two novel techniques were conceived, implemented and evaluated. First, a “Locally Maximal” corner selection method for the Harris corners [4] is used. This novel method selects features evenly across the image and detects features more consistently in a non-uniformly illuminated scene compared to a previous corner selection method that based on a corner threshold. Second, “Feature Cluster” landmarks are used, contrasting with the standard practice of using single feature landmarks. Each landmark is a 3D feature point cluster extracted from a single camera frame (keyframe), allowing for highly consistent and robust landmark measurements. Also, the use of Feature Cluster landmarks reduces the number of landmarks that are needed in the map, and enables a significant reduction of the SLAM filter computational cost.

The work also demonstrates that an easily retrofittable swing angle sensor can quickly and flexibly be used for field measurements on large mining shovels and potentially on other rotating heavy machines.

3.2 Related Work

Obtaining the swing angle amounts to finding the camera pose with respect to the carbody of the shovel. Different feature types that can be used are considered, then various approaches to finding the camera pose are discussed.

3.2.1 Feature Types

3.2.1.1 Artificial Markers

The placement of markers on the shovel carbody was considered. In general, a marker consists of high contrast markings of a specific geometric shape such as a square [5] or a circle [6]. Grayscale thresholding or edge detection is applied to find the outline of a marker. The outline is then checked to ascertain if it matches the designed marker shape, before the marker is declared detected.

Although marker detection algorithms are computationally fast, their designs are generally heuristic and there is no guarantee that markers can robustly be detected under highly non-uniform scene illumination conditions. Furthermore, marker surfaces can easily be contaminated by dust and grease in the environment, which can cause the markers to become undetectable.

3.2.1.2 Edgels or Line Segments

Another feature that can be used is the natural edges in the scene. There are two main approaches for finding edgels. One approach filters the image with the Laplacian of Gaussian then finds edgels as the zero-crossings of the result [7]. The other approach calculates image gradients, performs non-
maximal suppression, then thresholds for strong edges [8]. There are difficulties in using edgels. First, both edge detection methods have shortcomings. The Laplacian of Gaussian often picks up false edges because it is a second order derivative operator which is highly sensitive to noise [9]. For the gradient method, an edge threshold must be set but no threshold is suitable for all imaging conditions. The second issue of using an edgel is that it is ambiguous along the direction of the edge so there is no exact correspondence of an edgel.

Edgels can be grouped into line segments instead of being used directly, but line segments present even more problems. Segments are difficult to detect reliably and consistently; line segments may sometimes be detected as separate pieces, or be merged with others [10]. Furthermore, segment end points are not stable. Lastly, in cases where 3D line segments are measured and filtered, choosing a suitable parameterization for the 3D line segments without singularities is not trivial [11].

3.2.1.3 Interest Points

Interest points on the other hand, are features that have well-defined positions. A traditional and popular interest point detector is the Harris corner detector [4]. To measure a pixel location’s “cornerness”, this detector shifts an image patch centred about the pixel by a small amount in different directions then evaluates the local signal changes using the sum of squared differences (SSD) [12]. Small local changes indicate that the pixel lies in a flat greyscale region, whose position is poorly defined. Large changes when the patch is shifted in one direction only, indicate that the pixel is on an edge, while large changes when the patch is shifted in both perpendicular directions indicate a corner. The traditional corner detectors search for features over image locations only, but more recent detectors find features over both locations and scales to allow for matching of features when viewed from different distances. Examples of recent detectors are the Scale Invariant Feature Transform (SIFT) [13] or Speeded Up Robust Features (SURF) [14] which find blob-like image structures of varying scales as features. Moreover, these detectors also include a descriptor for representing a feature.

Interest points have well-defined positions, exact correspondences, and require no special parameterization. Thus, they are easier to use compared to edge-based features. An interest point type feature is used in this work.

3.2.2 Approaches For Finding Camera Pose

3.2.2.1 Model-based Camera Localization

For this method, an \textit{a priori} model of the scene is required. Features are detected in the camera image and matched to model features. Then given the set of feature correspondences, the camera pose relative to the scene can be found by a non-linear iterative optimization algorithm such as the Gauss-Newton method. The camera pose parameters are initialized with a guess, then are iteratively corrected until projected model features fit onto their corresponding image features.
The model could be a wireframe CAD model, in which case edge features are detected and matched to the wireframe. Examples of such work include [15], [16]. In the present work, it is difficult to extract a clean set of edges that can be matched to the wireframe because the carbody is cluttered with erroneous edges created by shadow, or grease marking. Another drawback is that a CAD model must be obtained for each shovel type.

Rather than using a CAD model, it is possible to construct a model of the scene using several camera snapshots of the scene from different views, then use this model for online camera localization. In one such work [17], a SIFT feature-based model of the scene was built from roughly 20 snapshots. This model was then used for online camera localization in an augmented realities application. The limitation of this method is that the model is pre-built offline using a small set of images. During online camera operation, incoming camera images of the scene cannot be used to update and reduce errors in the model. Also appearance changes in the scene cannot be handled.

### 3.2.2.2 3D Motion Tracking

The camera pose is tracked from the start and there is no attempt to create a permanent model of the scene. An example of this work is [18]. A stereo camera was used to obtain natural 3D features in the environment which were assumed to be static. Starting from a defined camera pose, the 3D features were tracked over frames, and changes in feature positions relative to the camera were used to estimate the change in camera pose. New 3D features that came into the camera view were tracked while old features that had left the view were thrown away. The system has been shown to work robustly in outdoor environments. However, drift error grew unbounded as more distance was traveled, even in the case where the camera returned to its starting location after traversing in a large loop.

### 3.2.2.3 Simultaneous Localization and Mapping (SLAM)

SLAM can be considered as an extension to 3D Motion Tracking. Instead of forgetting features as they leave the camera view, features are recorded as landmarks into a “map”. At the same time, these landmarks are re-referenced by the camera for localizing itself. The landmark positions are filtered and are incrementally improved as more observations of the scene are taken. Most importantly, SLAM allows loop closure, i.e. the correction of drift errors in the camera pose and landmark position estimates when the camera returns to its starting location and re-detects the initial landmarks, after traversing in a large loop. Comparing SLAM to model-based approaches, an a priori model of the shovel is not needed. The map or “model”, is built and refined in an online manner as more observations are made. Thus, SLAM is the approach taken in this work.

The SLAM problem assumes that a vehicle or a robot starts in an unknown environment and the robot is only equipped with sensors that take relative measurements of the environment rather than global sensors such as a GPS. Examples include a laser scanner or a camera mounted to a robot which allows the robot to measure feature positions relative to the robot’s coordinate
frame. The goal for the robot is to incrementally build a consistent map of the environment, while simultaneously localizing itself with respect to this map.

The robot pose estimate and landmark position estimates are highly correlated. Since a new landmark’s global position is estimated using the robot’s current pose estimate, error in the new landmark position estimate correlates with the error in the robot pose, which in turn correlates with the positional errors of existing landmarks in the map. Smith et al. [19] proposed to account for all the correlations by estimating the robot pose and landmark positions within a single state vector and covariance matrix updated by the Extended Kalman Filter (EKF).

An important EKF behavior resulting from the maintenance of the correlations between the robot and the landmarks is drift error correction at loop closure. As the robot leaves its starting position and traverses in a large loop, the estimated robot pose accumulates more drift error and is more uncertain, as are the estimated positions of new landmarks initialized by the robot. But when the robot returns to its starting position and re-detects the initial landmarks, its estimated position is immediately corrected; and because of the maintained correlations, the correction propagates back to all previously observed landmarks. This is illustrated in Figure 3.4. The map converges monotonically to a perfect relative map with zero uncertainty in the limit as the number of landmark observations increases [20]. The drawback of using the EKF to maintain the correlations is that its computational complexity is $O(N^2)$ where $N$ is the number of landmarks.

Solving the SLAM problem using the EKF method with vision sensors has been demonstrated by various researchers. Se et al. [21] used a stereo camera as the sensor with 3D SIFT features [13] as landmarks. They were able to demonstrate an algorithm speed of 2Hz. In [22], a robot equipped with a combination of monocular camera and odometry sensors traversed a loop 30m long and created a map consisting of 113 landmarks. Near real-time performance was achieved. Using a monocular camera only, Davison et al. [23] have shown that it is possible to estimate the camera pose in a room-sized domain in real-time. However, in these works the vision algorithms were not shown to work in outdoor environments with extreme directional sunlight causing shadows and highly non-uniform scene illumination.

An alternative to the EKF is the Rao-Blackwellised Particle Filter (RBPF) that is more suited for building large maps with a high number of landmarks. One such RBPF based algorithm is FastSLAM [24], which has complexity $O(M \log N)$ where $M$ is the number of particles and $N$ is the number of landmarks. Each particle is a hypothesis of the robot trajectory. Given each particle, landmark positions are conditionally independent and can be estimated separately. However, its main drawback is its poorer loop-closing performance, which improves with the diversity of particles that is maintained [25]. The more particles maintained, the closer its loop-closing performance approaches that of the EKF.

The SLAM filter implemented in this work was EKF SLAM, for its good loop-closing performance and implementation simplicity. A small map is built since the camera follows a small, restricted circular trajectory as the shovel swings. Thus, real-time performance can reasonably be
Figure 3.4: Before loop closure, estimates of the robot pose and new landmark positions accumulate more drift error as the robot traverses a large loop path. When the robot returns to its starting location and re-detects the initial landmark, loop closure occurs. The estimated robot pose is immediately corrected. Due to the maintained correlations in the EKF, the correction also propagates back to all previously observed landmarks.

achieved with EKF SLAM. Moreover, most applications of RBPF are for 2D robot pose estimation, but in this work the problem is approached in 3D. Thus, the number of particles needed to adequately represent robot trajectory hypotheses will be increased significantly compared to previous applications. There may not be a computational speed advantage for adopting the RBPF here.

3.3 Method

3.3.1 Algorithm Overview

The algorithm flow (Figure 3.5) is based on the operation of the EKF SLAM filter. For each new camera frame, the following steps are performed:

1. EKF Prediction Update: The current camera pose is predicted from its previous estimate.

2. 3D Feature Detection: Using current stereo images, 3D features are detected.

3. Landmark Measurement: Feature Cluster landmarks are matched to the detected 3D features. The landmark poses relative to the camera are found. Geometric verification of individual
4. EKF Measurement Update: The EKF is updated with the measured landmark positions.

5. Swing Angle Computation: From the updated camera position estimate, swing angle is found.

6. New Landmark Initialization: If no Feature Cluster landmarks are well-tracked, a new Feature Cluster landmark is extracted from the current frame and added to the system.

7. Landmark Feature Management: Individual landmark features that are often matched incorrectly are removed from the landmark database.

### 3.3.2 EKF State Vector

The algorithm was designed with the assumption that the position of the shovel house rotation centre, and the shovel house rotation axis direction relative to the camera are not precisely known, and that only their approximate guesses are available. Thus, in addition to the camera pose and landmark positions, these states are filtered for improved estimates. The exact structure of the filter is described next.

As explained in Section 3.2.2.3, within the EKF SLAM framework, all states are estimated using a single EKF. That is, the EKF state vector \( \hat{x} \) is a large column vector in which the states are stacked:

\[
\hat{x} = \begin{bmatrix}
\hat{x}_v \\
\hat{o}_{l_1} \\
\hat{o}_{l_2} \\
\hat{o}_{l_3} \\
\vdots \\
\hat{o}_{l_n}
\end{bmatrix}
\]  

(3.1)

where \( \hat{x}_v \) is a non-landmark state vector which will be described in detail below, and \( \hat{o}_{l_1}, \hat{o}_{l_2}, \ldots, \hat{o}_{l_n} \) are landmark states. A landmark state \( \hat{o}_{l_i} \) is a \( 3 \times 1 \) vector representing the global 3D position of the \( i \)’th landmark. Whenever a new landmark is added to the system, the state vector \( \hat{x} \) is lengthened by \( 3 \times 1 \) to filter the new landmark position.

The non-landmark state vector \( \hat{x}_v \) (12 \times 1) is comprised of states such as the camera pose, the rotation center of the shovel house, and the rotation axis direction:

\[
\hat{x}_v = \begin{bmatrix}
\hat{o}_c \\
\hat{r}_c \\
\hat{o}_r \\
\hat{\phi}_r \\
\hat{\theta}_r \\
\hat{\omega}
\end{bmatrix}
\]  

(3.2)
Figure 3.5: Algorithm flow overview.
Here, $\hat{o}_c$ ($3 \times 1$) is the camera origin, and the camera frame $C_c$ ($3 \times 3$) is parameterized as the camera rotation vector $\hat{r}_c$ ($3 \times 1$). The camera rotation vector $\hat{r}_c$ can be transformed into the full $3 \times 3$ camera frame by a matrix exponential: $\hat{C}_c = e^{\hat{r}_c}$. Next, $\hat{o}_r$ ($3 \times 1$) is the rotation centre of the shovel house. The direction of the shovel house rotation axis is represented with its inclination angle $\hat{\phi}_r$ and azimuth angle $\hat{\theta}_r$. The unit vector $\hat{r}_r$ pointing in the direction of the rotation axis can be found as:

$$
\hat{r}_r = \begin{bmatrix}
\cos \hat{\phi}_r \cos \hat{\theta}_r \\
\cos \hat{\phi}_r \sin \hat{\theta}_r \\
\sin \hat{\phi}_r
\end{bmatrix}
$$

Lastly, $\hat{\omega}$ is the swing angular speed of the shovel house in radians per frame interval.

### 3.3.3 Initialization and Calibration

After mounting the camera, $\hat{x} = \hat{x}_v$ needs to be initialized with rough guesses of the camera pose, the rotation centre and the rotation axis of the shovel house. In this work the initial camera pose $\{\hat{o}_c, \hat{r}_c\}$ is used to define the global coordinate frame so its values are initialized as zeros. The rotation centre and the rotation axis are estimated by a vision-based calibration procedure, requiring no physical measurement. The shovel makes a small calibration swing of approximately $+/- 25^\circ$ during which the camera pose is tracked. The tracked trajectory arc of the camera is then used to estimate the rotation centre and axis.

EKF SLAM guarantees map convergence to a perfect relative map only. Thus, a swing angle reference landmark located at $0^\circ$ is needed. When computing the swing angle, the estimated camera position is compared to the estimated reference landmark position. For obtaining the reference landmark, the shovel operator is expected to swing the shovel to rest at $0^\circ$, before starting the system. When the system starts, the first camera frame is captured; the system initializes the first Feature Cluster landmark, which also becomes the reference landmark.

### 3.3.4 EKF Prediction Update

In the prediction update, current values of the EKF states are predicted from their estimates in the previous time step. The landmarks are assumed to be stationary in the global coordinate system so their estimates remain unchanged:

$$
\begin{align*}
\hat{o}_{t1|t-1} &= \hat{o}_{t1|t-1} \\
\hat{o}_{t2|t-1} &= \hat{o}_{t2|t-1} \\
\hat{o}_{t3|t-1} &= \hat{o}_{t3|t-1} \\
&\vdots \\
\hat{o}_{t{n}|t-1} &= \hat{o}_{t{n}|t-1}
\end{align*}
$$

(3.4)

To predict the camera pose, a non-landmark states prediction function $\hat{x}_{vt|t-1} = f_v(\hat{x}_{vt-1}, w_{n|t-1})$
is defined, where $w_n$ is a zero-mean white gaussian process noise modelling the prediction uncertainty. A constant shovel swing angular speed model is used in the prediction function:

$$\hat{\omega}_{t|t-1} = \hat{\omega}_{t-1} + w_{n|t-1}$$ (3.5)

which implies undetermined swing angular accelerations occur with a zero-mean Gaussian profile.

Setting a low standard deviation for $w_n$ implies a strong confidence that swing angular speed will change little between two frames. On the other hand, a high standard deviation implies high uncertainty in the constant angular speed model and that more weight will be given to the measurements for estimating the camera pose. The standard deviation of $w_n$ has been tuned to twice the maximum possible swing angular acceleration at $0.4 \text{ rad}/(\Delta t)^2$, where $\Delta t$ is the time interval between two frames.

To predict the current camera pose, the rotation of the shovel during the interval between the previous frame and the current frame is first estimated. This rotation is represented by the rotation vector $\hat{\mathbf{q}}_{t|t-1}$, estimated from swing angular speed $\hat{\omega}_{t|t-1}$ in radians per frame interval, and the shovel swing rotation axis $\{\hat{\phi}_{rt-1}, \hat{\theta}_{rt-1}\}$:

$$\hat{\mathbf{q}}_{t|t-1} = (\hat{\omega}_{t|t-1}) \begin{bmatrix} \cos \hat{\phi}_{rt-1} \cos \hat{\theta}_{rt-1} \\ \cos \hat{\phi}_{rt-1} \sin \hat{\theta}_{rt-1} \\ \sin \hat{\phi}_{rt-1} \end{bmatrix}$$ (3.6)

Using $\hat{\mathbf{q}}_{t|t-1}$, the predicted camera position and camera frame are:

$$\hat{\mathbf{o}}_{ct|t-1} = \hat{\mathbf{o}}_{rt-1} + [e^{\hat{\mathbf{q}}_{t|t-1} \times}] [\hat{\mathbf{o}}_{ct-1} - \hat{\mathbf{o}}_{rt-1}]$$

$$\hat{\mathbf{C}}_{ct|t-1} = [e^{\hat{\mathbf{q}}_{t|t-1} \times}][e^{\hat{\mathbf{r}}_{ct-1} \times}]$$ (3.7)

where $\hat{\mathbf{C}}_{ct|t-1}$ is then converted back into its rotation vector representation $\hat{\mathbf{r}}_{ct|t-1}$ in the EKF.

The shovel house rotation center and rotation axis remain the same:

$$\hat{\mathbf{o}}_{rt|t-1} = \hat{\mathbf{o}}_{rt-1}$$

$$\hat{\phi}_{rt|t-1} = \hat{\phi}_{rt-1}$$

$$\hat{\theta}_{rt|t-1} = \hat{\theta}_{rt-1}$$ (3.8)

For the update of the EKF covariance matrix, the Jacobian matrices $df_v/dx_v$ and $df_v/dw_n$ are computed numerically by the system.

**3.3.5 EKF Measurement Update**

An observation function $z_i = h(x_v, o_i, v)$ needs to be defined for EKF measurement update. The function models the measurement $z_i$ of the $i$’th landmark, given the camera pose in $x_v$, the landmark’s global position $o_i$, and the sensor noise $v$. Here, the measurement $z_i$ input to the EKF,
is defined as the measurement of the \( i \)‘th landmark’s position relative to the camera coordinate system, and the sensor noise is modelled as additive noise:

\[
\begin{bmatrix}
  z_i \\
  1
\end{bmatrix} = \begin{bmatrix}
  C_c & o_c \\
  0 & 0 & 0 & 1
\end{bmatrix}^{-1} \begin{bmatrix}
  o_l_i \\
  1
\end{bmatrix} + \begin{bmatrix}
  v \\
  1
\end{bmatrix}
\] 

(3.9)

For the computation of the Kalman gain, the Jacobians \( dh/dx \) and \( dh/dv \) are computed numerically.

### 3.3.6 3D Feature Detection

Input images come from a Point Grey Bumblebee 2 stereo camera consisting of a left and a right camera with a 12 cm baseline. The camera is pre-calibrated by the manufacturer. Stereo images are rectified using manufacturer-provided code before any processing by our algorithm, and are of size 512 × 384 pixels.

The right camera is the reference camera used by the system. Features and their descriptors are found using the right image. The left image is used during stereo matching only, to obtain the 3D positions of features detected in the right image. Both the feature detection and stereo matching are discussed in detail below.

#### 3.3.6.1 Locally Maximal Harris Corners

In this work, scale invariance of features is not needed as the camera does not move toward or away from the carbody. Thus, basic interest point detectors that detect only feature locations such as Harris corner detector or SUSAN [26] were considered over scale-invariant but more complex detectors such as SIFT and SURF. Of the basic detectors, the Harris corner detector was chosen as it has been shown to have high repeatability over a range of conditions such as viewpoint change, illumination variation, compared to other interest point detectors [27], [28].

To select corner locations, the Harris corner measure is first evaluated for each pixel location. Then, corners are selected based on a corner threshold. Prevalently, this threshold has been based on a percentage (e.g. 1%) of the highest corner measure found in the image. A pixel location is considered as a corner if and only if the location’s corner measure exceeds the corner threshold, and its corner measure is locally maximal compared to its 8 neighboring pixel locations. This is the method used in [27] and also implemented in OpenCV [29]. In this paper, it is referred to as the “Percent Threshold” corner selection method. However, this corner selection method was found to perform poorly in our outdoor setting where directional sunlight casts shadow patterns from metal-mesh walkways, or creates directly-illuminated and shadow regions of the scene. The problem and solution are explained below.

Auto-correlation based interest point detectors such as the Harris Corner detector evaluate a pixel location’s “cornerness” by looking at local signal changes as the image patch centred about the pixel is shifted in different directions. Consequently, this measure is sensitive to local contrast.
That is, for two structurally identical image patterns with one pattern having higher contrast than the other, the higher contrast pattern will have higher corner measures because the signal changes are larger. Figure 3.6 demonstrates this by comparing the corner measures of two images of the same checkerboard, of which one image is taken with a shutter time of 9ms for higher contrast while the lower contrast image is taken with a shutter time of 3ms.

This variance of the corner measures to contrast becomes problematic for our situation. Shadow patterns can be created from the sunlight casting through handrails and metal-mesh walkways behind the camera. “Corners” on these patterns have the highest corner measures because of their large apparent local contrast. Using the Percent Threshold corner selection method, as done in works such as [27], will result in almost all detected corners on these patterns (Figure 3.7).

A problem also occurs when the camera swings from the shadow scene region to the directly-illuminated scene region (Figure 3.9). The shadow outlines between the two regions also create
“corners” of extremely high corner measures, boosting the corner threshold. Exacerbating the issue is that camera shutter time shortens dramatically as the camera swings toward the directly-illuminated scene region, causing features in the shadow scene region to become less exposed and their corner measures to fall drastically. Thus, features in the shadow scene region quickly fail to be re-detected.

An attempt was made to identify shadow patterns or boundaries by finding image regions that were either over-saturated (image value 255) or completely under-exposed (image value 0). The found regions were then morphologically dilated and ignored during corner detection. However, this method did not work 100% of the time as not all shadow patterns or outlines were located at boundaries of over-saturated or under-exposed image regions. Moreover, this additional processing incurred significant computation time (28ms) due to the morphological dilation. Thus, it was not used considering its small benefit.

Rather than basing the corner selection on a corner threshold, the locally maximal criterion for selecting a corner can be extended to select corners more consistently. In this paper, this novel corner selection method is called the “Locally Maximal” corner selection method. A pixel location is identified as a corner if and only if its corner measure is locally maximal within a \((2W + 1 \times 2W + 1)\) neighborhood block, where \(W\) controls the neighborhood size and can be used to tune the desired number of features per camera frame. A large \(W\) gives sparse features while a small \(W\) gives many features. In this work, \(W\) is set to 4, detecting roughly 600 features per frame. To eliminate features from image regions of nearly constant grayscale intensities, a minimum cut-off threshold was still set. Figures 3.8 and 3.10 show corner detection results using the new criteria as compared to the earlier method. Although some corners are still selected on shadow patterns, they are not distinctive and most will not be matched during the feature matching process. The
Figure 3.9: Corner detection result using the Percent Threshold method for two snapshots as the camera swings from the shadow scene region to the directly-illuminated scene region. Left image is the earlier snapshot and right image is the later snapshot. The directly-illuminated scene region is located on the left sides of the images. The shadow scene region is located on the right sides of the images. The corner measures of features in the shadow region fall dramatically in the second snapshot because the camera drastically reduces its shutter time as it swings toward the directly-illuminated scene region. Many features in the shadow scene region detected in the first snapshot are no longer detected in the second snapshot.

Figure 3.10: Using the locally maximal criterion, corners are more consistently selected over changing image exposure.

Few matched non-physical shadow corners will not agree with the change in observed 3D positions of other physical features, and can be eliminated in a geometric verification step. The geometric verification step is described in Section 3.3.8.2.

3.3.6.2 Stereo Matching

Once features are detected in the right image, they are stereo matched to the left image for finding their 3D positions relative to the camera. For each feature in the right image, a $5 \times 5$ image patch around the feature is obtained and shifted along the corresponding scanline in the left image.
Using normalized cross-correlation [30] as the similarity measure, the feature’s best match in the left image is found. The match is cross-validated [31]. If the feature fails the cross-validation test, it is discarded. Finally, to obtain sub-pixel disparity a parabola is fitted to the similarity scores centered about the best match. The position of the parabola peak is used to determine the sub-pixel disparity. The 3D position of the feature relative to the camera is then calculated as:

\[
\begin{align*}
\hat{c}_z &= fB/d \\
\hat{c}_x &= (u - u_0) \frac{\hat{c}_z}{f} \\
\hat{c}_y &= (v - v_0) \frac{\hat{c}_z}{f}
\end{align*}
\] (3.10)

where \( f \) is the camera focal length, \( B \) is the camera baseline, \( d \) is the disparity, \((u_0, v_0)\) is the pixel coordinate of the image center, while \((u, v)\) is the feature’s image location.

Most features on the carbody lie 2–3m from the camera. Assuming the range of 3m and disparity accuracy of +/- 0.2 pixels, the stereo camera’s depth measurement accuracy is expected to be +/- 4cm given its baseline and focal length.

### 3.3.6.3 Descriptor

Once the detected Harris corners are successfully stereo-matched, their descriptors need to be computed. Since the Harris corner detector does not include a descriptor, the descriptor from SIFT [13] was borrowed. This descriptor has been shown to have one of the best matching performances [32].

SIFT computes a feature’s descriptor as a histogram of image gradients in a window centred at the feature’s location; the histogram values are then flattened into a 128-element vector which represents the feature. To keep the descriptor rotationally invariant, a dominant gradient direction is found as the feature orientation and image gradients are re-oriented with respect to the feature orientation for the histogram computation. A small modification to this descriptor as done in [22], is a “rotationally variant” descriptor in which this feature orientation is ignored and image gradients used for histogram computation are not re-oriented. Orientation invariance is not needed since the camera is aimed mainly horizontally and does not rotate much about its optical axis within the range of camera pose change between feature correspondences. Moreover, this descriptor is more distinctive than the original.

### 3.3.7 Feature Cluster Landmarks

In all previous vision-based EKF SLAM works that the authors are aware of, a landmark is simply a single feature. However, there are problems associated with using single feature landmarks. First, a single feature landmark may not be consistently re-detected in subsequent frames upon initialization. One reason is the feature detector’s limited repeatability. Also, the feature can sometimes fail its stereo match validation test. Moreover, the feature can quickly become occluded
as the camera swings to a different side of the carbody. Additionally, the feature can become over- or underexposed as the camera swings, due to the non-uniformly illuminated scene and the camera's limited dynamic range.

Tracking many features at any one time will alleviate above problems because some single feature landmarks can still be measured even if others are not detected. However, the EKF computational complexity grows quadratically as new landmark features are added to the map. Thus, for real-time processing, the number of tracked features is extremely limited. For example, in [23] the number of tracked landmark features at any moment was limited to 12 only. Yet, limiting the number of tracked features severely reduces the robustness and accuracy of a vision algorithm, given the unreliable nature of single features.

Another general problem is erroneous feature measurements. A landmark feature may be matched to the wrong observed feature as the descriptor of a single feature is generally not distinctive enough for reliable matching. Existing methods mitigate the problem by searching for landmark feature matches within regions of their predicted image locations only. Jenselt et al. [22] further associated each landmark feature with additional recognition features that appeared alongside the landmark feature; a landmark feature is matched only if its recognition features are matched as well. However, features that are similar and close to each other may still be mismatched, such as features found on the swing gear teeth. Also, it is necessary to detect the situation where the 3D position of the match is poorly measured. Lastly, a landmark feature may not even be a real physical corner, such as a feature found on shadow outlines. Such a feature's physical position in the world may shift. A full 3D geometric verification of the feature matches can eliminate erroneous or inaccurate landmark feature matches. However, given that so few features are tracked at any moment, and even fewer are successfully matched, a reliable geometric verification may be difficult.

Since no single feature can be consistently and reliably measured up to a large swing angle away from the camera position at which it was initialized, the use of a “higher-level” landmark was considered. The proposed landmark is a cluster of numerous 3D features detected in a single camera frame (keyframe). This cluster of 3D features is treated as a single rigid object. An object coordinate system is attached to the cluster landmark representing its pose, and individual cluster feature positions are represented in terms of this coordinate system (Figure 3.11). It is the pose of the cluster that is measured and filtered, rather than the individual feature positions. Such a landmark’s pose can be consistently and reliably measured. Even if some cluster features are undetected in subsequent frames, there are many others that can be matched, still allowing the feature cluster landmark to be recognized and its pose with respect to the camera measured. As long as a minimum of three non-collinear cluster features are matched, the cluster rotation $R$ and position $z$ with respect to the camera can be found (Figure 3.12). In addition, individual feature matches can be geometrically verified during this cluster pose estimation. Erroneous or inaccurate feature matches will not agree with the cluster pose estimated from the majority of matches. Thus, features that are erroneously matched, or non-physical features whose physical world positions
Figure 3.11: a) Cluster features detected from a keyframe. Roughly 350 3D features were detected to form this feature cluster. b) - d) The feature cluster and its coordinate system are plotted and viewed from different angles. Cluster feature positions are represented in terms of this attached coordinate system.

shift, can be identified. Lastly, the grouping of these features into a single feature cluster landmark allows the system to visually track many features while filtering one landmark only. Thus, the system can make much more robust landmark measurements with less SLAM filter computational cost compared to before.

This feature cluster landmark has a full pose (a frame and position) as compared to just a position for single point features. However, in this work, only the cluster’s global position is filtered. A full landmark pose \(\{R, z\}\) is still measured but only the relative position \(z\) of the pose measurement is needed by the EKF SLAM filter.

For improved accuracy, individual cluster feature positions can be filtered once the cluster pose has been measured. Each cluster feature can be filtered individually, since individual feature
positions relative to the cluster coordinate system are independent of each other. This incurs a computational cost linear to the number of features in a cluster only, as compared to the quadratic increase in computational cost if these features are estimated in the SLAM filter instead. However, for further algorithmic speed and simplification, this individual feature filtering is not implemented. In this work, a feature’s positional error with respect to its cluster’s coordinate system is expected to be less than 4cm, as calculated from the stereo camera’s 3D measurement accuracy in Section 3.3.6.2. This error is considered negligible, compared to the potential drift error of a cluster’s estimated global position, which is found to accumulate up to 50cm right before loop closure. Even if a cluster feature’s 3D position is poorly measured, it can be identified and discarded because the cluster feature would become an outlier during the cluster pose estimation.

3.3.7.1 Feature Cluster Landmark Initialization

The system initializes a new Feature Cluster landmark when existing Feature Cluster landmarks are no longer well-tracked in the current frame. The new Feature Cluster landmark consists of a subset of 3D features detected in the current frame. The landmark features are selected using a larger neighborhood block \( W \) and higher cut-off threshold than regular features. Using stricter criteria for selecting landmark features improves the repeatability of their detection in subsequent camera frames. Lastly, the image region from which the landmark features are selected was manually constrained to the top portion of the camera image. This is to avoid selecting too many landmark
features from the shovel tracks. Landmark features are selected with \( W = 5 \). This results the selection of over 100 features per cluster landmark.

Once 3D cluster features are selected, a landmark coordinate system \( \{ C_l, o_l \} \) fixed to the cluster needs to be chosen, and cluster feature positions need to be represented in terms of this landmark coordinate system. This landmark coordinate system can be chosen anywhere relative to the cluster features. As long as it remains fixed relative to the cluster, consistent relative landmark measurements can be made and the camera pose can be properly estimated by the SLAM filter. The landmark coordinate system is simply assigned as the current camera coordinate system. Thus, features’ landmark coordinates are simply their current observed camera coordinates. The cluster features’ landmark coordinates and their descriptors are stored to the system database as the full description of the new landmark.

Given this cluster coordinate system assignment, a cluster’s global position (or origin) is equivalent to the camera position at which the cluster was initialized. Figure 3.13 shows a snapshot of landmark positions as the camera is swung clockwise before making loop closure; it can be seen that these landmarks sit on the circular trajectory of the camera. This is convenient as the shovel’s swing angle can simply be found as the angle between the current camera position and the reference landmark position with the rotation center as the pivot point.

### 3.3.8 Landmark Measurement

#### 3.3.8.1 Active Landmark Feature Matching

After the EKF prediction update and the 3D feature detection, the system proceeds to match landmarks to the observed 3D features. Instead of a recognition approach in which observed features are looked up in the landmark database, a more efficient active search approach is taken. First, the visibility of each landmark in the current frame is predicted. Two cases exist for determining a landmark’s visibility. Case 1, the landmark has been properly tracked in the previous frame and in this case, the landmark is assumed to be visible in the current frame. The maximum swing angle to which the landmark is trackable is also recorded for deciding the landmark’s visibility in case 2. Case 2, the landmark is not tracked in the previous frame, in which case, the current swing angle between the camera and the landmark can be used to estimate the landmark’s visibility. If this swing angle is within the landmark’s recorded maximum trackable swing angle, the landmark is predicted to be visible.

Once the landmark is deemed visible, the process of matching its features to observed features starts. For each landmark feature, its predicted image location starts. For each landmark feature, its predicted image location is found. The image location is found by first predicting the feature’s 3D position relative to the camera \( C_P \):

\[
\begin{bmatrix}
  cP \\
  1
\end{bmatrix} = \begin{bmatrix}
  C_c & o_c \\
  0 & 0 & 0 & 1
\end{bmatrix}^{-1} \begin{bmatrix}
  C_l & o_l \\
  0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  lP \\
  1
\end{bmatrix}
\] (3.11)
Figure 3.13: Snapshot of the system map as camera swings clockwise before making loop closure. Map is plotted in the 3D global coordinate system, which is defined by the initial camera pose at $t = 0$. Axis scales are in meters. Plot view is set looking down the shovel house rotation axis. Note that the rotation axis is not aligned with any axes of the global coordinate frame because the camera was not mounted in alignment to the shovel rotation axis. Camera coordinate system at the time of the snapshot is indicated by the thick RGB lines. Landmark positions are indicated by ‘+’. As discussed, landmarks sit on the circular trajectory of the camera.

where $\{C_c, o_c\}$ is the predicted current camera pose, $\{C_l, o_l\}$ the landmark’s estimated global pose, and $^lP$ the landmark coordinates of the feature. It is noted that only the landmark position $o_l$ is filtered; however, the landmark frame can be estimated as $C_l = e^{\Delta \theta_s \hat{r} \times} C_c$ where $\Delta \theta_s$ is the estimated swing angle between the landmark and the camera, and $\hat{r}$ is a unit vector in the direction of the rotation axis. Upon finding $^lP$, the feature can be projected onto the image plane using a pin-hole projection model which is just the inverse of Equation (3.10):

$$u = f \frac{c_x}{z} + u_0, \quad v = f \frac{c_y}{z} + v_0$$ (3.12)
Each landmark feature is then matched to observed features within a search window centered about the landmark feature’s predicted image location (Figure 3.14). Small $20 \times 20$ search windows are used when the landmark is tracked, while large $80 \times 80$ search windows are used when a landmark has not been tracked. Variable-sized search windows based on state estimate uncertainties [23] can be implemented in the future.

In the matching process, the landmark feature descriptor is compared against descriptors of observed features within the search window, using the similarity measure: the angle between the landmark feature descriptor vector and a candidate descriptor vector. The candidate that gives the smallest angle is the best match. Then, a nearest neighbor ratio test [13] is applied, in which the similarity measure of the best match is compared to the similarity measure of the next best match. If the two similarity measures are too close, the match is discarded because it is ambiguous.
Only feature matches that pass the nearest neighbor ratio test are used for the subsequent relative landmark pose estimation. This test is effective at eliminating non-distinctive landmark features that have similar neighbors near them. In particular, it was observed that many non-distinctive features such as those formed from the metal mesh walkway’s shadow pattern had been eliminated after this step.

### 3.3.8.2 Relative Landmark Pose Estimation

If a feature cluster landmark has at least 16 of its features matched, its pose is estimated. Otherwise, the landmark is discarded from the pose estimation. For the \( M \) number of landmark features that have been matched, their positions in landmark coordinates \( \{ l_P^1, l_P^2, l_P^3, \ldots, l_P^M \} \), and their corresponding positions as observed by the camera \( \{ c_P^1, c_P^2, c_P^3, \ldots, c_P^M \} \) are obtained. The goal is to find the feature cluster rotation \( e^r \times \) and position \( z \) with respect to the camera, so that landmark features are transformed into their observed positions:

\[
\begin{align*}
  c_P^1 &= \left[ e^r \times \right] [l_P^1] + z \\
  c_P^2 &= \left[ e^r \times \right] [l_P^2] + z \\
  c_P^3 &= \left[ e^r \times \right] [l_P^3] + z \\
  \vdots \\
  c_P^M &= \left[ e^r \times \right] [l_P^M] + z 
\end{align*}
\]  

(3.13)

Here, an overdetermined non-linear system of equations is established. The relative landmark pose \( \{ r, z \} \) is found as the least squares solution of the above equation minimizing the sum of squared errors \( \sum_{i=1}^{M} \| e_i \|^2 \) where:

\[
e_i = c_P^i - \left( \left[ e^r \times \right] [l_P^i] + z \right)
\]

(3.14)

As in [13] and [18], the Gauss-Newton procedure is used to find the parameter vector \( p = [r; z] \) iteratively. At each iteration \( k \), the Gauss-Newton algorithm computes the errors \( e \) given a starting guess \( p_k \). The algorithm then finds a correction vector \( \Delta p_k \) and adds the correction vector to the original guess so \( p_{k+1} = p_k + \Delta p_k \). The correction vector \( \Delta p \) is set up as follows:

\[
\begin{bmatrix}
  e_1 \\
  e_2 \\
  e_3 \\
  \vdots \\
  e_M \\
  e
\end{bmatrix}
= 
\begin{bmatrix}
  \frac{\partial e_1}{\partial r} & \frac{\partial e_1}{\partial z} \\
  \frac{\partial e_2}{\partial r} & \frac{\partial e_2}{\partial z} \\
  \frac{\partial e_3}{\partial r} & \frac{\partial e_3}{\partial z} \\
  \vdots & \vdots \\
  \frac{\partial e_M}{\partial r} & \frac{\partial e_M}{\partial z}
\end{bmatrix}
\begin{bmatrix}
  \Delta r \\
  \Delta z
\end{bmatrix}
\]

(3.15)
Figure 3.15: Lines connect features in the keyframe to features in the current frame. These are the remaining feature matches after geometric verification.

and solved as linear least squares:

\[ \Delta p = (J^T J)^{-1} J^T (e) \]  

(3.16)

Geometric verification of the feature matches is simultaneously performed in each Gauss-Newton iteration. Erroneous feature matches will not agree with the parameter corrections and will have large residual errors compared to the average residual error. Feature matches with error residual twice the average are discarded from the Gauss-Newton estimation. The elimination of bad matches enables robust and accurate relative pose estimation of the landmark. The procedure is stopped once no match pairs need to be discarded and that the change in average residual stabilizes. Figure 3.15 shows match results after geometric verification and Figure 3.16 shows the feature displacement vectors drawn using the found relative rotation and position.

It is observed that generally, the further the camera swings away from a landmark, the larger the landmark’s pose estimation residual becomes. The average residual ranges from approximately
0.3cm when the camera viewed the landmark directly, up to 3.0cm when the camera swings 25° from the landmark. Past that angle, most landmarks cannot be measured due to an insufficient number of feature matches. The growing residual can be attributed to the fact that detected feature positions tend to shift as the camera view angle increases. Figure 3.17 plots all measured relative landmark positions $z$ as the camera swings. As expected, the dots form a circular arc. Also note that the plotted dots “scatter” more, the further they are from the arc mid-point (the camera origin). This is consistent with the observation that the average residuals grow as landmarks are measured from a larger swing angle away.

The covariance matrix of the least squares estimate of $p$ is estimated with a simplifying assumption that uncertainties associated with each data point are equal and independent:

$$R = \sigma^2 (J'J)^{-1}$$

where $\sigma^2$ is the average squared residual.

### 3.3.9 Swing Angle Computation

After the visual landmark measurements are made and the EKF measurement update performed, swing angle is found using the updated estimates of the camera position $\hat{o}_c$, the reference landmark position $\hat{o}_l_1$, the rotation center $\hat{o}_r$, and the rotation axis direction $\{\hat{\phi}_r, \hat{\theta}_r\}$.

First, the two vectors $v = [\hat{o}_c - \hat{o}_r]$ and $w = [\hat{o}_l_1 - \hat{o}_r]$ are computed. Next, the unit vector $\hat{r}_r$ pointing in the direction of the rotation axis is found from $\{\hat{\phi}_r, \hat{\theta}_r\}$ using Equation (3.3). Then, $v$, $w$ are projected onto a plane perpendicular to $\hat{r}_r$, giving $v_\perp$, $w_\perp$ respectively. Finally, the swing angle $\theta_s$ is found as the angle between $v_\perp$ and $w_\perp$ (Figure 3.18).
**Figure 3.17:** Measurements of landmarks’ positions are recorded over a period of time and plotted. Each dot is a measurement of a landmark’s position relative to the camera, at a single time instance. As expected, the dots form an arc, with the camera located at arc mid-point. The further from the camera a landmark is, the noisier the landmark’s measurements become, hence the scattering of the dots around the arc end points.

**Figure 3.18:** View looking down \( \hat{r}_r \), in the direction of the shovel rotation axis. The vectors \( v_\perp \), \( w_\perp \) are respectively, the projections of \( [\hat{\delta}_c - \hat{\delta}_r] \) and \( [\hat{\delta}_l - \hat{\delta}_r] \) on a plane perpendicular to \( \hat{r}_r \). Swing angle is found as the angle between \( v_\perp \) and \( w_\perp \).
3.3.10 Landmark Feature Management

Each Feature Cluster landmark can be consistently and reliably measured once added to the map. Thus, there is minimal management of landmarks; a landmark generally stays in the map permanently. However, the features of a landmark need to be managed. A landmark feature is removed if more than 50% of the time its match fails the geometric verification during the landmark pose estimation. Such a landmark feature is either non-distinctive so is often erroneously matched, or is a non-physical corner such as a shadow point whose world position is unstable. A landmark feature having a poorly initialized 3D position compared to others will be removed as well. Figure 3.19 shows the features of a landmark at initialization, and at a later time after some pruning of bad features.

Figure 3.19: Top: Initial features of a landmark. Bottom: Remaining landmark features after pruning of bad features over time. Note that most features on the shadow pattern were removed.
3.4 Results

The vision algorithm was tested on three separate video sequences. For verifying the output of the stereo-vision sensor, an encoder was installed on the shovel swing motor shaft, whose output served as the ground truth. The first sequence consisted of roughly 5000 frames (4min). In this sequence, the shovel first swung two full counter-clockwise revolutions, then two clockwise revolutions, and then randomly. Outputs of both sensors are shown in Figure 3.20.

Initially, the vision system’s error grew up to 5.5° as drift accumulated. At around Frame 2900, the camera was about to complete its first full revolution; the first landmark was re-detected and the loop was closed. As expected of the EKF SLAM behaviour, the drift error of the camera pose estimate was immediately corrected so the error in the estimated swing angle instantly dropped to 0.5°. At the same time, this drift error correction was also propagated to the estimates of the landmark positions and the rotation center. The estimates of the landmark positions and the rotation center continued to converge as more observations were made. Thus, subsequent swing angle estimates improved. After two full revolutions, the maximum error settled under 1°. Given that the camera was located 4.5m from the rotation center, the maximum camera positional error was equivalent to +/- 0.08m over a 28m circular trajectory, or +/- 0.28%.

In total, the system initialized 19 Feature Cluster landmarks along the circular path traversed by the camera, with each feature cluster consisting of more than 100 features. The number of landmarks filtered by our algorithm’s EKF was much lower compared to previous vision-based EKF SLAM works which filtered roughly 100 landmarks. Thus, the computational cost of the SLAM filter was much reduced compared to previous works. Yet, the number of features tracked in this work was much higher than in the past.

The second and third video sequences were taken during a separate mine trip from the one in which the first sequence was taken. The second sequence (Figure 3.21) was roughly 8000 frames (7min) during which the shovel first swung a full counter-clockwise then a clockwise rotation. After that, the shovel operator was asked to swing randomly at full acceleration. Again, after map convergence (2 full rotations) the swing angular error settled within 1°. The reader may notice that the initial error build up was much smaller than that of the first video sequence. This was likely attributed to a more accurate initial guess of the shovel rotation center, obtained from the offline vision-based calibration procedure. To verify this hypothesis, the initial guess of the shovel rotation center was purposely offset by 8cm from the estimate found from the calibration procedure. The algorithm was then re-run and Figure 3.22 showed that for this test, an initial error curve very similar to that of video sequence 1 also resulted.

In the third sequence, the shovel moved forward during Frames 6300-6600 (Figure 3.23). The forward motion was to test the effect of the shovel track movement on the algorithm. The keyframe being matched during Frames 6300-6600 is shown in Figure 3.24. Even though landmark features were selected using the top half of the image only, one-third of the features in the keyframe were still part of the shovel track. Despite the high proportion of track features, the algorithm remained
Table 3.1: Computation time for a sample frame.

<table>
<thead>
<tr>
<th>Process</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stereo Rectification</td>
<td>10ms</td>
</tr>
<tr>
<td>EKF Prediction Update</td>
<td>24ms</td>
</tr>
<tr>
<td>Corner Detection</td>
<td>27ms</td>
</tr>
<tr>
<td>Stereo Matching</td>
<td>67ms</td>
</tr>
<tr>
<td>Descriptor</td>
<td>78ms</td>
</tr>
<tr>
<td>Landmark Feature Matching</td>
<td>7ms</td>
</tr>
<tr>
<td>Landmark Pose Estimation</td>
<td>2ms</td>
</tr>
<tr>
<td>EKF Measurement Update</td>
<td>8ms</td>
</tr>
<tr>
<td>Swing Angle Computation</td>
<td>2ms</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>225ms</td>
</tr>
</tbody>
</table>

robust while the track moved. It was able to eliminate track feature matches in the relative landmark pose estimation (geometric verification) step. As a result, swing angular error remained nearly constant during this interval. The shovel swung randomly again after driving forward, and the swing angular error remained less than 1°.

In a separate test, the whole image was used when selecting landmark features. However, in this case the relative landmark pose estimation could not converge while the shovel track was moving. This was because the proportion of the track features exceeded 50% of all features in the keyframe being matched. Thus, as part of the system installation, it may be necessary to adjust the image region for landmark feature selection to avoid having the system choosing too many landmark features from the tracks.

The system currently is a prototype system and is implemented in a mixture of C++ and Matlab. All image processing, and landmark pose estimation functions are implemented in C++, while the EKF is implemented in Matlab. The algorithm ran at approximately 4Hz on a single core of a 1.8GHz Intel Core2 CPU with 3GB RAM. Table 3.1 shows the computation time required for a sample frame in which a landmark cluster (a total of 267 features) was matched.
Figure 3.20: Video sequence 1. Top: Vision and encoder outputs. Bottom: Difference between vision and encoder outputs.
Figure 3.21: Video sequence 2. Top: Vision and encoder outputs. Bottom: Difference between vision and encoder outputs.
Figure 3.22: Video sequence 2 with initial guess of the rotation center purposely offset by 8 cm from the estimate obtained from the offline vision-based calibration procedure. Top: Vision and encoder outputs. Bottom: Difference between vision and encoder outputs.
Figure 3.23: Video sequence 3. During frames 6300-6600 the shovel track moved but swing angular error remained stable. Top: Vision and encoder outputs. Bottom: Difference between vision and encoder outputs.
Figure 3.24: The keyframe matched during frames 6300-6600 of video sequence 3, while the shovel track was moving. Note roughly 1/3 features of this keyframe were selected on shovel track.

3.4.1 Comparison of Harris Corner Selection Methods

Detailed comparisons between the novel Locally Maximal Harris corner selection method and the prevalent Percent Threshold selection method [27] were performed. First, it was found that the number of features detected in each frame using the Locally Maximal method was more consistent compared to the Percent Threshold method. Figure 3.25 showed the number of features detected in each frame, from frames 1900-2250, during which the shovel swung approximately 180°. With the Percent Threshold method, the number of detected corners fluctuated wildly between 200-2000 features while the number of corners detected with the Locally Maximal method remained between 450-700.

Next, the SLAM algorithm was run using the Percent Threshold corner selection method instead of the Locally Maximal corner selection method. A feature cluster was detected from a selected starting keyframe and then tracked as the camera swung away. The fraction of the cluster features tracked as the camera swung from its initial position was recorded and compared to the original result using the Locally Maximal corner selection method.

The tests were performed on three separate starting keyframes. The scene in the first keyframe had no directional sunlight or shadows. Figure 3.26 showed that in this ideal imaging situation, the Percent Threshold method performed slightly better than the Locally Maximal method. For both methods, the cluster features could be tracked up to 20°.

The second keyframe (Figure 3.9) was taken as the camera swung from the shadow scene region to the directly-illuminated scene region. As mentioned in Section 3.3.6.1, with the Percent Threshold method, most cluster features in the shadow scene region quickly fell below threshold because the camera reduced its shutter time as it swung into the directly-illuminated scene region. It can be seen in Figure 3.27 that the fraction of tracked features fell to 0.1 by the time the camera had swung only 8° from its starting position. However, with the Locally Maximal method, the
cluster features were tracked up to 20°, comparable to the ideal imaging case of the first keyframe.

The third keyframe (Figure 3.7) contained shadow patterns created from the sunlight casting through handrails and metal mesh walkways behind the camera. Using the Percent Threshold method, most corners were selected on the shadow patterns. Thus, the feature tracking was completely unstable. However, the Locally Maximal method showed consistent feature tracking results and the cluster features could still be tracked up to 20° (Figure 3.28).

Figures 3.26, 3.27, 3.28 showed that although the Percent Threshold corner selection method performed slightly better in the ideal imaging situation, the Locally Maximal corner selection method performed consistently well in all situations.

3.4.2 Accuracy Versus Number of Features per Landmark

The effect of the number of features per landmark on the algorithm accuracy was examined. For each trial, the algorithm was run with different values of $W$ for selecting landmark features such that the number of features per landmark cluster was varied. The trials were run on video sequence 1. Results are shown in Table 3.2. The values of $W$ were varied from 15 to 5, providing an average number of features per landmark cluster from 52 to 278 respectively. The maximum error stated was the maximum error after map convergence, which was chosen as past frame 3700 after the
Figure 3.26: Fraction of initial features tracked. Scene in the starting keyframe had no directional sunlight or shadows. The Percent Threshold Corner Selection method performed slightly better than the Locally Maximal Corner Selection method.

Table 3.2: Accuracy versus number of features in landmark cluster.

<table>
<thead>
<tr>
<th>W</th>
<th>Avg # of Features per Landmark</th>
<th>Max Error</th>
<th>Computation Time of a Sample Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>52</td>
<td>2.0°</td>
<td>140ms</td>
</tr>
<tr>
<td>10</td>
<td>104</td>
<td>1.5°</td>
<td>172ms</td>
</tr>
<tr>
<td>5</td>
<td>278</td>
<td>0.8°</td>
<td>225ms</td>
</tr>
</tbody>
</table>

shovel has made two full revolutions. Results showed a trend of accuracy improvement as the number of features per landmark increased.

3.5 Conclusions

This paper has presented an enhanced stereo vision SLAM algorithm that works robustly in the presence of directional sunlight illumination causing shadows, and non-uniform scene lighting. Two novel techniques have been described. One is the Locally Maximal Harris corner selection method, which selects features evenly across the image and more consistently, compared to the prevalently used Percent Threshold method. This novel corner selection technique is applicable in works where the Harris corner detector is used in the outdoors. An example problem where this could be
Figure 3.27: Fraction of initial features tracked. The starting keyframe was taken as the camera was revolving from the shadow scene region to the directly-lit scene region. With the Percent Threshold method, most cluster features in the shadow scene region quickly fell below threshold so initial features were tracked up to $8^\circ$ only. With the Locally Maximal method, initial features could be tracked up to $20^\circ$, comparable to the ideal imaging case of the first keyframe.

Applicable is an outdoor mobile robot tracking its own position using a downward, ground-looking camera and detecting Harris corners.

The second novel technique is the use of Feature Clusters as landmarks rather than single features. The technique allows the EKF SLAM algorithm to make consistent, reliable, and accurate landmark measurements over large changes in the camera pose. A direct benefit of this is that the number of landmarks needed in the SLAM map can be lowered and the EKF computational cost can be reduced. The authors believe that the idea of Feature Cluster landmarks can be incorporated into future vision-based EKF SLAM works to improve algorithm robustness and to allow vision-based EKF SLAM to scale to larger regions than before.

The algorithm in this work currently runs at approximately 4Hz; however, the desired processing rate is 20Hz. Future improvements to the implementation include 1) full C++ coding, 2) confining feature detection to only search regions containing landmark feature matches rather than the whole image, 3) using variable-sized feature match search windows based on state estimate uncertainties, which could minimize the search regions when the camera pose and landmark positions are well-estimated, and 4) multi-threading the algorithm to take explicit advantage of the multi-core CPU. The authors are confident that these coding optimizations can improve the
Figure 3.28: Fraction of initial features tracked. Scene of the starting keyframe contained shadow patterns created by the directional sunlight casting through handrails and metal-mesh walkways behind the camera. With the Percent Threshold method, corners were mostly selected on shadow patterns so feature tracking was completely unstable. However, with the Locally Maximal method, initial features could still be tracked up to 20°.

algorithm speed sufficiently to achieve the desired processing rate.

3.6 Acknowledgements

We would like to thank Teck Inc.’s Highland Valley Copper operation and its employees for their invaluable assistance during this work, Teck Inc.’s Elk Valley Coal operation and Motion Metrics International for their support of this research. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under grant STPGP 321813-05.
3.7 References


Chapter 4

Conclusion

4.1 Summary and Conclusions

The shovel collision avoidance system that motivated this work was presented in Chapter 2. The sensing systems for the shovel collision system were presented and the swing angle sensor was placed into context. The motivation for the use of a vision-based swing angle sensor was explained and the hardware of such a system was described.

The enhanced stereo vision SLAM algorithm used in the vision based swing angle sensor was described in Chapter 3. Two novel techniques that enhanced the robustness of the SLAM algorithm were presented: 1) a “Locally Maximal” Harris corner selection method, and 2) the use of “Feature Clusters” as SLAM landmarks. Swing angle accuracy was reported to be under $1^\circ$. Moreover, improvements due to the novel techniques were demonstrated.

This research has made two main contributions. First, an enhanced stereo vision SLAM algorithm that allows the camera pose to be robustly estimated in outdoor lighting conditions was demonstrated. Two novel techniques are used in the algorithm. These techniques can be applied to other works. The first novel technique is the Locally Maximal Harris corner selection method, which performs better than the traditional Percent Threshold method [1, 2] in various outdoor imaging situations. The Locally Maximal method allows corners to be selected more consistently over changes in their local image exposure. For example, initial features in the shadow scene region can be tracked up to $20^\circ$ as the camera swings toward the directly-lit scene region, but only up to $8^\circ$ with the Percent Threshold method. Also, in the case that shadow or shadow patterns are observable, corners are selected evenly across the image as opposed to the Percent Threshold method which selects corners on shadow outlines or patterns only. Lastly, the number of corners selected in each frame is stable. The number ranges from 450-700 per frame using the Locally Maximal method, compared to 200-2000 per frame using the Percent Threshold method. This Locally Maximal corner selection technique can be applied to other works in which the Harris corner detector is used in an outdoor environment.
Another novel technique is the use of high-level Feature Cluster landmarks which allow robust and consistent landmark measurements. In this work, each landmark consists of over 100 features, allowing a high number of features to be matched as compared to a maximum of 12 in [3]. The high number of feature matches, and the subsequent geometric verification procedure enables robust and consistent landmark measurements over large changes in the camera pose. Moreover, Feature Cluster landmarks reduce the EKF SLAM filter computation significantly as fewer of them are needed for robust operation. In this work, only 19 landmarks are needed as compared to approximately 100 landmarks in works such as [3, 4]. The idea of Feature Cluster landmarks can be incorporated in future vision-based EKF SLAM works to improve algorithm robustness and to scale vision-based EKF SLAM to larger regions than before.

The other main contribution is a swing angle measurement system that is easily and quickly retrofittable to mining rope shovels or other large rotating machines. Summarizing from Chapter 3, a prior model of the shovel is not required and the swing angle sensor can be flexibly installed on any available large mining shovel. The camera need not be installed precisely in a particular position or orientation relative to the shovel. After clamping the camera, the time consuming process of physically measuring the shovel rotation center and axis relative to the camera coordinate system is also not needed. Instead, a simple requirement is that the shovel swings roughly +/- 25° so an offline vision based calibration procedure can obtain estimates of the shovel rotation center and axis. These estimates are then used to initialize the SLAM filter. Also, like any other swing angle measurement system, the shovel needs to be swung to zero degrees before starting the system to set the zero position. The last additional requirement is a quick two full shovel rotations to allow the map to be built and converge; then, the system can be used for accurate swing angle measurement. The vision-based system can be installed by a single person in less than 30 minutes, as compared to the 4 hours required for the installation of an encoder on the shovel swing motor shaft. The swing angle sensor, along with the other easily retrofittable sensors for shovel arm geometry and truck pose estimation provide the flexibility to run shovel collision experiments on any available shovel.

4.2 Future Work

4.2.1 Generic SLAM

The SLAM algorithm presented here has been adapted to the current application for better efficiency, and in its current form cannot be applied to general robot navigation. For example, the Harris corner detector used is not scale-invariant. As a result, the system will not be able to match features from different distances. Second, the SLAM filter prediction function assumed that the camera moves in a circular motion.

For future work, it would be interesting to make the described SLAM algorithm more generic. To do this, the feature detector can be switched to a scale-invariant feature detector such as SIFT or SURF so that features can be matched from different distances. Another necessary step is to
change the filter prediction function to a more generic motion model, but this is trivial.

The use of feature cluster landmarks reduced the number of required landmarks in the map. Another interesting future direction is to combine this with the submapping strategy proposed by Guivant et al. [5]. They have shown that the map can be divided into subregions, and that the robot only has to update the local submap, with a complexity $O(N_a^2)$ where $N_a$ is the number of landmarks in the subregion. A full map update with complexity similar to a normal EKF update is required only when the robot leaves the current subregion, to transfer information gained in the local subregion to the whole map. This submapping strategy combined with the use of Feature Cluster landmarks can scale vision based EKF SLAM to a much larger region compared to before.

### 4.2.2 Algorithm Speed

One main drawback of the work is that it is still not running at the desired speed for the application. As mentioned in chapter 3, optimizing the implementation such as 1) full C++ coding, 2) confining feature detection to only search regions containing landmark feature matches, 3) variable-sized feature match search windows based on state estimate uncertainties and 4) multi-threading, can significantly speed up the algorithm. However, an interesting area to explore is General Purpose Computation on GPUs (GPGPU). That is, to take advantage of a GPU to speed up the algorithm. Modern day GPUs are high powered, generally consisting over 128 cores so are extremely suited for computer vision algorithms which can be easily parallelized. For example, it was shown that for a $2048 \times 2048$ image, Sobel edge detection was 44.7 times faster, and Discrete Fourier Transform (floating point) was 44.5 times faster on a GeForce GTX280 graphics card compared to a Core2 Duo 2.13GHz CPU [6]. Bui et al. have also shown 2D image registration tasks to be 7 times faster for $500 \times 500$ images and up to 90 times faster for $3000 \times 3000$ images on a NVIDIA Tesla C870 GPU compared to an Intel Quad-Core Q6700 2.66 GHz CPU [7].

Parts of the current algorithm that are candidates for processing on a GPU include corner detection, stereo matching, and feature matching, which are all highly parallelizable.

### 4.3 Potential Future Applications

The SLAM algorithm in its current form can be used to find swing angle on other large rotating machine such as different types of shovels or even a crane, given enough clearance for the camera.

The range of applications greatly opens up if the described SLAM algorithm is made generic. A major application could be for outdoor autonomous robots that work within a confined region. For any autonomous robot a key requirement is for the robot to know its own position within its environment. The SLAM algorithm will allow the robot to estimate its position accurately within its environment, without the need to setup artificial landmarks, or requiring a model of the environment to be preprogrammed into the robot.

Another application is in the field of Augmented Reality [8, 9, 10]. Augmented Reality involves
the integration of virtual 3D objects into the real world, in real-time. A user wears a Head-Mounted Display (HMD) which can overlay images of virtual objects over what the user sees. For example, a virtual mug can be drawn on a real table in front of the user. The key is to know in real-time the user’s pose with respect to the real world so that the correct perspective of the virtual 3D objects can be drawn. A stereo camera can be integrated into the HMD and the generic stereo vision SLAM algorithm can be used to find the user’s pose in the real environment.
4.4 References


Appendix A

Stereo Vision Based Swing Angle Sensor for Mining Rope Shovel

A.1 Introduction

A key process at open-pit mines is the digging of mineral rich earth and the loading of the earth into haul trucks by large electric rope shovels (Figure A.1). Two types of collisions can occur during this process: 1) the shovel bucket hitting the haul truck and, 2) the bucket hitting the shovel’s own protruding tracks.

Figure A.1: Haul truck loading by a shovel.

A proof-of-concept collision avoidance system is being developed with the aim of providing advance warning of these collisions to the shovel operator [1]. The project objective was to instrument

---

the shovel to obtain the necessary information: 1) the truck pose with respect to the shovel, and 2) the shovel’s own joint and prismatic parameters. The instrumentation was divided into three measurement subsystems, all designed for temporary installation on rope shovels in several mines: 1) Truck Pose Estimation [2], 2) Arm Geometry [3], and 3) Swing Angle, i.e. the angle between the revolvable shovel house and carbody about the shovel’s vertical rotation axis. Swing angle measurement is the focus of this paper.

An important design goal of the sensing systems is that they needed to be easily and quickly retrofittable and removable, without significant modification to the shovel and without interference to existing shovel systems. The reason is that only limited shovel access was available for experiments. A shovel was only available when it was down for scheduled maintenance and for several hours each trip to the mine. Sensors were removed after each experiment. Furthermore, such a set of easily retrofittable and removable sensors can provide the flexibility of working with different shovels in a single mine or with shovels in different mines, depending on availability. Several retrofit options were explored for obtaining swing angle:

A.1.1 Tracking Swing Motor Shaft Rotations with an Encoder
With this method, a custom-made stub shaft with a smaller shaft diameter is first connected to the motor shaft, then the encoder can be attached to the stub shaft. This stub shaft must be trued to minimize wobble when the motor spins. An assembly holding the encoder stationary while the motor shaft turns also needs to be installed. These are time-consuming processes. Even as a permanent retrofit, the encoder and stub shaft hinder regular swing motor maintenance. Nevertheless, an encoder was installed on one machine to use as a reference in measuring the accuracy of the swing angle sensing system reported here.

A.1.2 Counting Swing Gear Teeth with Inductive Sensors
There is a large swing gear sandwiched between the house and the carbody, that is fixed to the carbody. A less intrusive way of measuring swing angle is to count the number of swing gear teeth that have passed by as the house revolves, using a pair of inductive sensors mounted underneath the house, separated 90 degrees phase apart in terms of a tooth cycle. However, the immediate area near the swing gear is difficult to work in. Furthermore, inductive sensors must be adjusted as close to the gear teeth as possible. The second inductive sensor needs to be installed precisely at 90 degrees phase apart from the first sensor for good measurement accuracy. Thus, the installation time can be significant on each machine and it is not the ideal solution for our goal.

A.1.3 Measuring Swing Angle with Stereo Vision
An alternative that could be easily and quickly retrofitted to a variety of different machines was sought. A prototype sensor consists of a Point Grey Bumblebee 2 stereo camera mounted externally
to the bottom outer edge of the house, aimed toward the carbody, so that the carbody fills the view of the camera. As the house rotates, the camera revolves with the house in a circular orbit about the house’s vertical rotation axis, seeing differing views of the carbody. A computer in the house analyzes the camera images in real-time and determines the position of the camera with respect to the carbody, which in turn is used to calculate swing angle.

This system can be easily and quickly retrofitted. The camera mounting location is easily accessible from the shovel tracks (Figure A.2). Furthermore, the vision algorithm does not require the camera be mounted precisely at a pre-determined position or view direction. The camera can be mounted freely, as long as the carbody fills the majority of camera view. After mounting, it is not needed to physically measure the camera position and view direction.

The problem of finding the camera pose with respect to the carbody was solved using the Simultaneous Localization and Mapping (SLAM) approach. While the shovel swings, the stereo camera records observed 3D feature on the carbody as landmarks, and incrementally builds a 3D map of these features as it revolves around the carbody. At the same time, the camera localizes itself by matching observed features in camera view to the landmarks in the map.

A.1.4 Contributions

This paper presents a visual SLAM algorithm that is robust under the presence of directional sunlight illumination causing shadows and non-uniform scene lighting. To achieve this, a locally maximal Harris corner selection technique is used to select features evenly across the image and to detect features more consistently in a non-uniformly lit scene. Secondly, a 3D feature point cluster is used as a landmark, contrasting with the standard practice of using a single feature as a
landmark; this allows highly consistent and robust landmark matching over changing camera pose, while reducing the computational complexity of the SLAM filter significantly. It has also been shown an easily retrofittable swing angle sensor that can quickly and flexibly be used for field measurements on large mining shovels, and potentially other rotating machines.

A.2 Related Work

Obtaining the swing angle amounts to finding the camera’s pose with respect to the shovel carbody. First, feature types that can be used are considered, then various approaches to finding the camera pose are discussed.

A.2.1 Feature Types

A.2.1.1 Artificial Markers

The placement of markers [4], [5] on the carbody was considered. However, their designs are generally heuristic and there is no guarantee that markers can be robustly detected under highly non-uniform scene illumination conditions. Furthermore, marker surfaces can easily be contaminated by the dust and grease in the environment, which can cause the markers to become undetectable.

A.2.1.2 Line Segments

Edgels can be detected and grouped into line segments [6]. However, segments are difficult to detect reliably and consistently; line segments may sometimes be detected as separate pieces, or be merged with others. Furthermore, segment end points are not stable. Lastly, in cases where 3D line segments are measured and filtered, choosing a suitable parameterization for the 3D line segments without singularities is not trivial [7].

A.2.1.3 Interest Points

Interest points are features that have well defined positions. A traditional and popular interest point detector is the Harris corner detector [8] that finds feature locations only. More modern detectors such as SIFT [9] and SURF [10] finds feature locations and scales to allow for matching of features when viewed from different distances. They also include a descriptor for representing a feature. Interest points have well-defined positions, exact correspondences, and require no special parameterization so are easier to use compared to line segments. Thus, in this work an interest point feature was used.
A.2.2 Approaches For Finding Camera Pose

A.2.2.1 Model-based Camera Localization
For this, an a priori model of the scene is required. Detected image features are matched to model features. Then given the set of feature correspondances, the camera pose relative to the scene could be found by a non-linear optimization algorithm.

The model could be a wireframe CAD model [11], [12]. However, it is difficult to extract a clean set of edges that can be matched to the wireframe, because the carbody is cluttered with erroneous edges created by shadow, sunlit spots or grease marking. Another disadvantage is that a CAD model must be obtained for each type of shovel.

Rather than a CAD model, a scene model can be constructed using several camera snapshots of the scene from different views; then use this model for online camera localization [13]. The limitation is that the model is fixed; during operation, incoming camera images of the scene cannot be used to update and reduce errors in the model, and appearance changes in the scene cannot be handled.

A.2.2.2 3D Motion Tracking
The camera pose is found by tracking the camera from the beginning, and there is no attempt to create a permanent model of the scene [14]. A stereo camera is used to obtain natural 3D features in the environment which are assumed to be static. Starting from a defined camera pose, the 3D features are tracked over frames. Changes in feature positions relative to the camera are used to estimate the change in camera pose. Features that come into the camera view are tracked while features that leave the view are thrown away. However, drift error grows unbounded as more distance is travelled, even when the camera returns to an area that it has seen before.

A.2.2.3 Simultaneous Localization and Mapping (SLAM)
SLAM can be considered as an extension to 3D Motion Tracking that eliminates the problem of unbounded drift error. Instead of forgetting features as they leave the camera view, features are recorded as landmarks into a “map” so that they can be re-referenced by the camera when the camera returns to an area it has seen. Moreover, landmark position estimates are filtered and incrementally improved as more images of the scene are taken. Thus, this map is essentially a scene model that is constructed online. Given SLAM’s advantages over the previously mentioned methods, this is the approach taken in this work.

The SLAM problem assumes a robot starts in an unknown environment and is only equipped with sensors that take relative measurements. A laser scanner or camera rigidly mounted to a robot which allows the robot to measure feature positions relative to the robot coordinate system are two examples. The goal for the robot is to acquire a map of the environment while simultaneously
localize itself with respect to this map, using the relative measurements only.

The robot pose estimate and landmark position estimates are highly correlated. Since a new landmark's position is estimated using the robot pose estimate, error in the new landmark position estimate is correlated with the error in the robot pose estimate, which in turn correlates with the positional errors of existing landmarks in the map. Smith et al. [15] proposed to account for the correlations by estimating the robot pose and landmark positions in a single state vector with a single covariance matrix, updated by an Extended Kalman Filter (EKF). It was proven that such a map filtered by the Kalman Filter converges monotonically to a perfect relative map with zero uncertainty in the limit as the number of landmark observations increases [16].

An important EKF behavior resulting from the maintenance of the correlations is drift error correction at loop closure. As the robot traverses a large loop from a starting position, the estimated robot pose accumulates more drift error and is more uncertain, as are the estimated positions of new landmarks initialized by the robot. But when the robot returns to its starting area and re-detects the initial landmarks, its estimated position is immediately corrected; and because of the maintained correlations, the correction propagates back to all previously observed landmarks. The drawback of using the EKF to maintain the correlations is that its computational complexity is $O(N^2)$ where $N$ is the number of landmarks.

Solving the SLAM problem using EKF and vision as the sensor has been demonstrated in [17], [18], and [19]. However, in these works the vision algorithms were not tested in outdoor environments with extreme directional sunlight causing shadows and highly non-uniform scene illumination.

A.3 Method

A.3.1 Algorithm Design and Overview

The algorithm flow is based on the operation of the SLAM filter. For each camera frame, the following are performed:

1. EKF Prediction Update: The current camera pose is predicted from its previous estimate.
2. 3D Feature Detection: Using current stereo images, 3D features are detected.
3. Landmark Measurement: Landmarks are matched to the observed 3D features and the landmark poses relative to the camera are found.
4. EKF Measurement Update: The EKF is updated with the measured landmark positions.
5. Swing Angle Computation: From the updated camera position estimate, swing angle is found.
6. New Landmark Initialization: If no landmarks are well-tracked, a new landmark extracted from the current frame is added to the system.
Landmark Feature Management: Landmark features that have often failed to be matched are removed from the landmark database.

In the following, the paper will first provide an overview of the EKF, discussing the estimated states, the prediction update, and the measurement inputs expected by the filter. Next, the 3D feature detection, the landmark, and the landmark measurement are described.

A.3.2 EKF Overview

Within the EKF SLAM framework, all states such as the camera pose and the landmark positions are estimated using a single EKF. That is, the EKF state vector $\hat{x}$ is a large column vector in which the states are stacked:

$$
\hat{x} = \begin{bmatrix}
\hat{x}_v \\
\hat{o}_{t_1} \\
\hat{o}_{t_2} \\
\vdots \\
\hat{o}_{t_n}
\end{bmatrix}
$$

(A.1)

where $\hat{x}_v$ is a non-landmark state vector which will be described in detail below, and $\hat{o}_{t_1}, \hat{o}_{t_2}, \ldots, \hat{o}_{t_n}$ are landmark states. A landmark state $\hat{o}_{t_i}$ $(3 \times 1)$ represents the global 3D position of the $i$'th landmark. Whenever a new landmark is added to the system, $\hat{x}$ is enlarged by a landmark state.

The non-landmark state vector $\hat{x}_v$ $(12 \times 1)$ is:

$$
\hat{x}_v = \begin{bmatrix}
\hat{o}_c \\
\hat{r}_c \\
\hat{o}_r \\
\hat{\phi}_r \\
\hat{\theta}_r \\
\hat{\omega}
\end{bmatrix}
$$

(A.2)

Here, $\hat{o}_c$ $(3 \times 1)$ is the camera origin, and the camera rotation vector $\hat{r}_c$ $(3 \times 1)$ represents the camera frame $C_c$ $(3 \times 3)$. The vector $\hat{r}_c$ can be transformed into $C_c$ by matrix exponential: $C_c = e^{\hat{r}_c \times}$. Next, $\hat{o}_r$ $(3 \times 1)$ is the rotation centre of the shovel house. The direction of the rotation axis is represented with its inclination angle $\hat{\phi}_r$ and azimuth angle $\hat{\theta}_r$. Lastly, $\hat{\omega}$ is swing angular speed in radians per frame interval.

For EKF prediction update, the landmarks are assumed to be stationary in the global coordinate system so their predicted positions remain unchanged from their previous estimates. To predict the camera pose, a non-landmark states prediction function $\hat{x}_{vt|t-1} = f_v(\hat{x}_{vt-1}, w_{nt-1})$ is defined, where $w_n$ is a zero-mean white gaussian process noise modelling the prediction uncertainty. A
constant swing angular speed model is used in the prediction function:

\[ \dot{\omega}_{t|t-1} = \dot{\omega}_{t-1} + w_{n_{t-1}} \] (A.3)

For EKF measurement update, a landmark measurement \( z_i \) is defined as the measured position of a landmark relative to the camera coordinate system. Thus, the measurement observation function \( z_i = h(x_v, o_{l_i}, v) \) is:

\[
\begin{bmatrix}
z_i \\
1
\end{bmatrix} = \begin{bmatrix}
C_c & o_c \\
0 & 0 & 0 & 1
\end{bmatrix}^{-1} \begin{bmatrix}
o_{l_i} \\
1
\end{bmatrix} + \begin{bmatrix}v \\
1
\end{bmatrix}
\] (A.4)

where \( v \) is a zero-mean white gaussian process modelling sensor noise.

### A.3.3 Initialization and Calibration

After mounting the camera, \( \hat{x} = \hat{x}_v \) needs to be initialized. In this work the initial camera pose \( \{\hat{o}_c, \hat{r}_c\} \) is used to define the global coordinate frame so its values are set to zeros. The rotation centre \( \hat{o}_r \) and the rotation axis \( \{\hat{\phi}_r, \hat{\theta}_r\} \) are estimated by a vision-based calibration procedure, requiring no physical measurement. The shovel makes a small calibration swing of approximately +/- 25° during which the camera pose is tracked. The tracked trajectory arc of the camera is then used to estimate the rotation centre and axis.

EKF SLAM guarantees map convergence to a perfect relative map only. Thus, a swing angle reference landmark located at 0° is also needed for computing the swing angle. For obtaining the reference landmark, the shovel operator is expected to swing the shovel to rest at 0°, before starting the system. When the system starts, the first camera frame is captured; the system initializes the first landmark, which also becomes the reference landmark.

### A.3.4 3D Feature Detection

Input images come from a pre-calibrated Point Grey Bumblebee 2 stereo camera consisting of a left and a right camera with 12 cm baseline. Stereo images are rectified using manufacturer-provided code before any processing by our algorithm, and are of size 512 × 384 pixels.

The right camera is the reference camera. Features and their descriptors are found using the right image. The left image is used during stereo matching only, to obtain the 3D positions of features detected in the right image. The feature detection and stereo matching are discussed:

#### A.3.4.1 Locally Maximal Harris Corners

The Harris corner detector was chosen for this work, as scale invariance of features was not needed and that it had been shown to have high repeatability over a range of conditions such as viewpoint change, illumination variation, compared to other interest point detectors [20], [21]. However,
contrary to these findings which were conducted indoors, it was found that in our outdoor setting the Harris corner detector failed to detect features consistently and that it detected features mainly on shadow patterns. The main issue was the corner threshold, which prevalently was based on the highest corner measure found in the image. It was resolved in this work by not directly using a corner threshold.

The Harris corner detector evaluates a pixel location’s “cornerness” by looking at local signal changes as the image patch centred about the pixel is shifted in different directions [22]. Consequently, this measure is sensitive to local contrast. There are shadow patterns created from sunlight casting through handrails and metal mesh walkways behind the camera. “Corners” formed by these shadow patterns have the highest corner measures due to their large local contrast. Selecting corners based on a threshold set as a percentage of the highest observed corner measure as done prevalently, results in most corners selected on these patterns (Figure A.3).

A problem also occurs when the camera swings from the shadow scene region to the directly-lit region. The shadow outlines between the two regions create “corners” of high corner measures, boosting the corner threshold. However, the camera shutter time shortens dramatically as the camera swings toward the directly-lit region, causing features originally detected in the shadow region to be less exposed and their corner measures to fall below threshold (Figure A.4).

It was attempted to identify shadow patterns or outlines by detecting image regions that were over-saturated (image value 255) or completely under-exposed (image value 0). The detected regions were then morphologically dilated and ignored during corner detection. However, this method did not work 100% of the time as not all shadow patterns or outlines were located within these regions. Moreover, this additional processing incurred significant computation time (28ms) due to the morphological dilation, so was not used considering its small benefit.
Figure A.4: Two camera snapshot taken quickly one after another. Shown are corners selected with a threshold set as 0.01 of the highest corner measure.

Figure A.5: Using the locally maximal criteria, corners are more consistently selected over changing image exposure.

A novel metric was used to select corners more consistently without a corner threshold. A pixel is identified as a corner if and only if its corner measure is locally maximal within a $(2W + 1 \times 2W + 1)$ neighborhood block, where $W$ controls the neighborhood size and can be used to tune the desired number of features per camera frame. To eliminate features from image regions of nearly constant grayscale intensities, a minimum cut-off threshold was still set. Figures A.3 and A.5 show corner detection results using the new metric. Although some corners were still selected on shadow patterns, they can be eliminated in latter processing. Many of these non-physical corners are undistinctive and most will not be matched during the feature matching process. The few matched non-physical corners will not agree with the change in observed 3D positions of other physical features, and can be eliminated in a geometric verification step, described in the landmark measurement section.
A.3.4.2 Stereo Matching

Once features are detected in the right image, they are stereo matched to the left image. For each feature in the right image, a $5 \times 5$ image patch around the feature is obtained and shifted along the corresponding scanline in the left image. Using normalized cross-correlation, the feature’s best match in the left image is found. The match is cross-validated [23]. Finally to obtain sub-pixel disparity a parabola is fitted to the similarity scores centered about the best match. The position of the parabola peak is used to determine the sub-pixel disparity. The 3D position of the feature relative to the camera is then calculated as:

$$
\begin{align*}
\hat{c}_z &= \frac{fB}{d} \\
\hat{c}_x &= \frac{u - u_0}{c_z / f} \\
\hat{c}_y &= \frac{v - v_0}{c_z / f}
\end{align*}
$$

(A.5)

where $f$ is the camera focal length, $B$ is the camera baseline, $d$ is the disparity, $(u_0, v_0)$ is the pixel coordinate of the image center, while $(u, v)$ is the feature’s image location.

Most features on the carbody lie 2–3m from the camera. Assuming the range of 3m and disparity accuracy of +/- 0.2 pixels, the stereo camera’s depth measurement accuracy is expected to be +/- 4cm.

A.3.4.3 Descriptor

Once the detected Harris corners are successfully stereo-matched, their descriptors are computed using the SIFT descriptor [9]. This descriptor has been shown to have one of the best matching performances [24].

A.3.5 Feature Cluster Landmarks

For all vision-based EKF SLAM works the authors are aware of, each landmark is a single feature. However, few features can be consistently re-detected in subsequent frames due to the feature detector’s limited repeatability. Also, a feature can sometimes fail stereo match validation, become occluded, or become over- or underexposed.

Another problem is erroneous measurements. A landmark feature may be matched to the wrong observed feature, or the match’s 3D position may be poorly initialized. Lastly, a landmark feature may not even be a real physical feature, such as a feature found on a shadow pattern. Such a feature’s physical position in the world shifts, and whose match does not provide reliable information for finding the camera pose.

Exacerbating the problem is that for real-time processing the number of tracked features is highly constrained as the EKF computational complexity grows quadratically with the number of landmark features. Thus, the vision algorithm’s robustness and accuracy become severely limited.

An ideal landmark is one that can be detected consistently across frames and measured reliably
Figure A.6: Left: Detected cluster features from a keyframe. Roughly 350 3D features form the feature cluster. Right: The feature cluster and its coordinate system.

up to a large angle away from the camera position at which it was initialized. Unfortunately, no single feature has this property. Thus, this inspired the use of a “higher level” landmark that is a cluster of 3D features detected in a single camera frame (keyframe). This cluster of 3D features is treated as a single rigid object. An object coordinate system is attached to the cluster representing its pose, and individual cluster feature positions are represented in terms of this coordinate system (Figure A.6). It is the pose of the cluster that is measured and filtered, rather than the individual feature positions. Such a landmark’s pose can be consistently and reliably measured. Even if some cluster features are undetected in subsequent frames, there are many others that can be matched, still allowing the feature cluster landmark to be recognized and its pose with respect to the camera measured. In addition, feature matches can be geometrically verified during the cluster pose estimation. Erroneous feature matches would not agree with the pose estimated from the majority of matches. Thus, features that are erroneously matched, or non-physical features whose physical positions shift, can be identified. Moreover, the grouping of these features into a single feature cluster landmark allows the system to visually track many features while filtering only one landmark. Thus, the system can make much more robust landmark measurements given the same SLAM filter computational complexity as before.

This feature cluster landmark has a full pose (a position and orientation) as compared to just a position for single point features. However, currently in this work only the cluster’s global position is filtered in the EKF. A full relative landmark pose \( \{ R, z \} \) is still measured but only the measured relative position \( z \) is fed to the filter.

For improved accuracy, individual cluster feature positions relative to the cluster coordinate
system can be filtered once the cluster pose has been measured. Cluster features can be filtered separately, since feature positions relative to the cluster coordinate system are independent of each other. This incurs a computational cost linear to the number of features in a cluster only. However, for further algorithmic speed and simplification, this individual feature filtering is still not implemented. In our case, a feature’s positional error with respect to its cluster’s coordinate system is expected to be less than 4cm, as found from the stereo camera’s 3D measurement accuracy. This error is considered negligible, compared to the potential drift error of a cluster’s filtered global position, which was found to accumulate up to 50 cm right before loop closure. Even if a feature’s position is poorly initialized, it can be identified and discarded because the feature would become an outlier during the cluster pose estimation.

A.3.5.1 Feature Cluster Landmark Initialization

The system initializes a new landmark when no landmark is well-tracked in the current frame. A new feature cluster landmark consists of a subset of 3D features detected in the current frame. These features are selected using a larger neighborhood block and higher cut-off threshold than regular features, allowing for higher repeatability of their detection in subsequent frames. Lastly, landmark features are selected from the top portion of the camera image only to avoid selecting too many features from the shovel tracks.

Once cluster features are selected, a landmark coordinate system \( \{C_l, o_l\} \) fixed to the cluster needs to be assigned, and cluster feature positions need to be represented in terms of this landmark coordinate system. This landmark coordinate system is assigned as the current camera coordinate system. Then, features’ landmark coordinates \( \{lP_1, lP_2, lP_3, \ldots\} \) are just their current observed camera coordinates \( \{cP_1, cP_2, cP_3, \ldots\} \). The cluster features’ landmark coordinates and descriptors are stored to the system database as the full description of the new landmark.

Given this cluster coordinate system assignment, a cluster’s global position (or origin) is equivalent to the camera position at which the cluster was initialized. Figure A.7 shows a snapshot of landmark positions as the camera is swung counter clockwise before making loop closure; it can be seen that these landmarks sit on the circular trajectory of the camera. This is convenient as the shovel’s swing angle can simply be found as the angle between the current camera position and the reference landmark position with the rotation center as the pivot point.

A.3.6 Landmark Measurement

A.3.6.1 Active Landmark Feature Matching

After EKF prediction update and 3D feature detection, the system proceeds to match landmarks to the observed 3D features. An active approach is taken where each landmark is predicted if it is visible to the camera in the current frame. Once a landmark is predicted visible, its features’ locations in the current image are predicted. Then each landmark feature is matched against
Figure A.7: Snapshot of system map as camera swung counter clockwise before making loop closure. View set looking down rotation axis. Map plotted in global coordinate system and axis scales in meters. Camera coordinate system at time of snapshot indicated by thick RGB lines. Landmark positions indicated by ‘+'. As discussed, landmarks sit on the circular trajectory of the camera.

observed features within a search window centered about its own predicted location. The similarity measure is the angle between the landmark feature descriptor vector and a candidate descriptor vector. The candidate that gives the smallest angle is the match. Then, a nearest neighbor ratio test [9] is applied to remove ambiguous matches. In particular, it was observed that many non-distinctive features such as those formed from the metal mesh walkway’s shadow pattern had been eliminated after this step.

A.3.6.2 Relative Landmark Pose Estimation

If a feature cluster landmark has at least 16 feature matches, its pose is estimated, otherwise the landmark is discarded from the pose estimation. For the \( M \) number of landmark features
that have been matched, their positions in landmark coordinates \( \{lP_1, lP_2, lP_3, \ldots, lP_M\} \), and their corresponding positions as observed by the camera \( \{cP_1, cP_2, cP_3, \ldots, cP_M\} \) are obtained. The goal is to find the feature cluster’s rotation \( e^r \) and position \( z \) with respect to the camera, so that:

\[
\begin{align*}
\overset{c}{P}_1 &= [e^r][\overset{l}{P}_1] + z \\
\overset{c}{P}_2 &= [e^r][\overset{l}{P}_2] + z \\
\overset{c}{P}_3 &= [e^r][\overset{l}{P}_3] + z \\
\vdots \\
\overset{c}{P}_M &= [e^r][\overset{l}{P}_M] + z
\end{align*}
\]

(A.6)

Here an overdetermined non-linear system of equations is established. The relative landmark pose \([z; r]\) is found as the least squares solution of the above equation using the iterative Gauss-Newton algorithm [12] [14].

Geometric verification of the feature matches is simultaneously performed in each Gauss-Newton iteration. Erroneous feature matches will not agree with the parameter corrections and will have large residuals compared to the average. Feature match pairs with residuals twice the average are discarded from the Gauss-Newton estimation. The elimination of bad matches enables robust and accurate landmark pose measurement. The procedure is stopped once no match pair is discarded and that the change in average residual stabilizes.

It was observed that generally, the further the camera revolves away from a landmark, the larger the landmark’s pose estimation residual becomes. The average residual ranged from approximately 0.3 cm when the camera viewed the landmark directly, up to 3.0 cm when the camera revolved 25° away from the landmark. Past that, most landmarks cannot be measured due to an insufficient number of feature matches. The growing residual can be attributed to the fact that detected feature positions tend to shift as the camera view angle increases.

A.3.7 Swing Angle Computation

After the visual landmark measurements are made and the EKF measurement update performed, swing angle is found using the updated estimates of the camera position \( \overset{\hat{o}}{c} \), the reference landmark position \( \overset{\hat{o}}{l}_1 \), the rotation center \( \overset{\hat{o}}{r} \), and the rotation axis direction \( \{\overset{\hat{\phi}}{r}, \overset{\hat{\theta}}{r}\} \).

First, the two vectors \( v = [\overset{\hat{o}}{c} - \overset{\hat{o}}{r}] \) and \( w = [\overset{\hat{o}}{l}_1 - \overset{\hat{o}}{r}] \) are computed. Next, the unit vector \( \overset{\hat{r}}{r} \) pointing in the direction of the rotation axis, is found from \( \{\overset{\hat{\phi}}{r}, \overset{\hat{\theta}}{r}\} \). Then, \( v, w \) are projected onto a plane perpendicular to \( \overset{\hat{r}}{r} \), giving \( v_\perp, w_\perp \) respectively. Finally, the swing angle \( \theta_s \) is found as the angle between \( v_\perp \) and \( w_\perp \) (Figure A.8).

A.3.8 Landmark Feature Management

Each feature cluster landmark can be consistently and reliably measured once added to the map. Thus, there is minimal management of landmarks; a landmark generally stays in the map per-
Figure A.8: View looking down $\hat{r}_r$, in direction of shovel rotation axis. Vectors $v_\perp$, $w_\perp$ are respectively the projections of $[\hat{o}_c - \hat{o}_r]$ and $[\hat{o}_l - \hat{o}_r]$ on a plane perpendicular to $\hat{r}_r$. Swing angle found as angle between $v_\perp$, $w_\perp$.

manently. However, the features of a landmark are managed. A landmark feature is removed if more than 50% of the time its match fails geometric verification. Such a landmark feature is either undistinctive so is often erroneously matched; or that it is a non-physical corner such as a shadow point whose world position is unstable. A landmark feature having a poorly initialized 3D position compared to others will be removed as well. Figure A.9 shows features of a landmark at initialization and after.

Figure A.9: Left: Initial features of a landmark. Right: After pruning of bad features over time. Note shadow features at the bottom have been pruned.

A.4 Results

The vision algorithm was tested on a sequence of roughly 5000 frames (4min). In the test duration, the shovel made two counter clockwise revolutions, then two clockwise revolutions, and then swung randomly. The stereo-vision based sensor output was verified with the output of an encoder
installed on the shovel swing motor shaft. Both outputs are shown in Figure A.10.

Initially, the vision system’s error grew up to $5.5^\circ$ as drift accumulated. At around Frame 3000, the camera made its first full revolution; the first landmark was re-detected and the loop was closed. As expected of the EKF SLAM behaviour, the drift error of the camera pose estimate is immediately corrected so the error in the estimated swing angle dropped to $0.5^\circ$. At the same time, this drift error correction was also propagated to the landmark position estimates. Thus, subsequent swing angle estimates improved. Landmark position estimates continued to converge as more observations were made and swing angle errors settled within +/- 1.1°. Given that the camera was located 4.5m from the rotation center, the camera positional error was equivalent to +/- 0.09m over a 28m circular trajectory, or +/- 0.3%. This swing angular error is also equivalent to +/- 19cm bucket localization error, assuming a truck loading distance of 10m. This is within the accuracy requirement of the aforementioned collision avoidance system.

In total, 25 landmarks were initialized along the circular path traversed by the camera. Each feature cluster landmark consisted of more than 100 features. Roughly, a total of 3750 features
were recorded by the system.

The algorithm is still in a prototype stage and is implemented in a mixture of C++ and Matlab. All image processing, landmark pose estimation functions are implemented in C++, while the EKF is implemented in Matlab. The algorithm ran at 4Hz on a single core of a 1.8GHz Intel Core2 CPU with 3GB RAM.

A.5 Conclusions

This paper has presented a visual SLAM algorithm that is robust in the presence of directional sunlight illumination causing shadows, and non-uniform scene lighting. The locally maximal Harris corner selection allowed features to be selected evenly across the image and more consistently.

The use of feature clusters as landmarks allowed the algorithm to track many features, significantly improving algorithm robustness, while keeping the EKF SLAM computational cost manageable.

Although the algorithm in its current implementation had not shown full real-time performance (20Hz), a much more optimized implementation in the future should bring the algorithm close to real-time, such as implementing it fully in C++, or taking advantage of all CPU cores.
A.6 References


87


