The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Stereo Vision Based Swing Angle Sensor for Mining Rope Shovel

Li-Heng Lin, Peter D. Lawrence, *Member IEEE*, Robert Hall

***Abstract*— An easily retrofittable stereo vision based system for quick and temporary measurement of a mining shovel's swing angle is presented. The stereo camera is mounted externally to the upper swingable shovel house, with a clear view of the shovel's lower carbody. As the shovel swings from its 0° swing angle position, the camera revolves with the shovel house, seeing differing views of the carbody. In real-time, the camera position is tracked, which in turn is used to calculate the swing angle. The problem was solved using the Simultaneous Localization and Mapping (SLAM) approach in which the system learns a map of 3D features on the carbody while using the map to determine the camera pose. The contribution includes a locally maximal Harris corner selection technique and a novel use of 3D feature clusters as landmarks, for improving the robustness of visual landmark matching in an outdoor environment. Results show that the vision-based sensor has a maximum error of +/- 1° upon map convergence.**

I. 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 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.

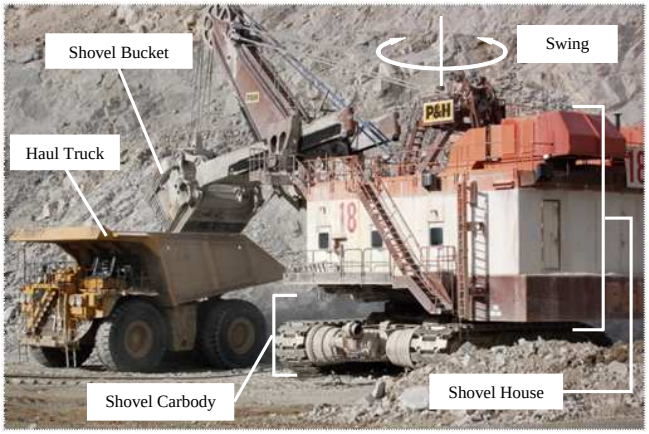

Fig. 1. Haul truck loading by a shovel.

A proof-of-concept collision avoidance system is being developed to provide advance warning of these collisions to the shovel operator [1]. The first objective is to design a set of shovel sensors that can obtain the necessary information for such a collision avoidance system. One necessary information is the shovel swing angle, which is the angle between the upper revolvable shovel house and lower carbody about the shovel's vertical rotation axis. This work focuses on a measurement subsystem for the shovel swing angle.

We would like to thank Highland Valley Copper and its employees for their invaluable assistance during this work. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under grant STPGP 321813-05.

An important design goal of the sensors is that they are easily and quickly retrofittable and removable, without significant modification to the shovel and without 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 at a mine. Sensors are removed after each experiment. A set of easily retrofittable and removable sensors provides the flexibility of quickly working with any available shovel at any mine. Several options have been explored for obtaining the swing angle:

A. Tracking Swing Motor Shaft Rotations with an Encoder

With this method, a custom-made stub shaft with a smaller shaft diameter must be first connected to the motor shaft, then the encoder can be attached to the stub shaft. The installation is highly time-consuming. The shovel swing motor must be taken apart and the stub shaft must be trued to minimize wobble when the motor spins. A re-design may also be needed for each shovel model due to the mechanical differences between models. Nevertheless, an encoder was installed on one shovel to use as a reference in measuring the accuracy of the swing angle sensing system reported here.

B. 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 shovel house revolves. This can be done using a pair of inductive sensors mounted underneath the house, separated 90° phase apart in terms of a tooth cycle. However, the immediate area near the swing gear is difficult to work in. Also, the inductive sensors must be adjusted as closely to the gear teeth as possible, and be placed precisely 90° degrees phase apart for proper measurement. Thus, installation time can be significant on each machine and it is not the ideal solution for our goal.

C. Measuring Swing Angle with Stereo Vision

This paper presents a vision-based swing angle sensor that is non-invasive, has little installation requirement, and can work on a wide variety of mining shovels. A Point Grey Bumblebee 2 stereo camera for 3D measurements is clamped externally to the bottom outer edge of the shovel house, aimed toward the carbody, so that the carbody fills the view of the camera. As the shovel swings from its 0° swing

Fig. 2. White box indicates the stereo camera location.

angle position, the camera revolves with the shovel house in a circular orbit about the shovel's vertical swing axis, seeing differing views of the carbody. The camera position relative to the start is tracked, which in turn is used to calculate the swing angle.

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

The Simultaneous Localization and Mapping (SLAM) approach is used to estimate the camera pose. While the shovel swings, the stereo camera records observed 3D features on the carbody as landmarks, and incrementally builds a global 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. This method has the advantage that an *a priori* model for each shovel type is not required. Also, unlike motion tracking methods, drift error is limited and can be reduced.

D. 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 method is used to select features evenly across the image and to select features more consistently in an outdoor scene. Secondly, "3D Feature Cluster" landmarks are used, contrasting with the standard practice of using a single feature as a single landmark. The 3D Feature Cluster landmark allows highly consistent and robust landmark measurements due to the large number of features per cluster. The ability to make reliable and consistent measurements for each landmark in turn reduces the number of landmarks needed for a robust SLAM algorithm. Thus, the computational cost of the SLAM filter can be lowered significantly.

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 on other rotating heavy machines.

II. Related Work

The general Simultaneous Localization And Mapping (SLAM) problem aims to localize a mobile robot in a global coordinate system. In this work, the robot is simply our stereo camera. The SLAM problem assumes that the environment is unknown and that the robot starts in a pre-defined pose in the global coordinate system. The robot is only equipped with sensors that take relative measurements of the environment. Examples include a laser scanner or a camera mounted to a robot which measures feature positions relative to the robot coordinate system. As the robot explores, it records observed features as landmarks and estimates landmark global positions, building a map of the environment. At the same time, these landmarks are re-referenced by the robot to help localize itself.

A key problem to solve in SLAM 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 global positions of new landmarks initialized by the robot. However, when the robot returns to its starting area and re-detects the initial landmarks, it should be able to correct the drift error in its pose estimate. This drift error correction should also propagate back to all previously observed landmarks for a consistent map.

An approach that allows this drift error correction was proposed by Smith *et al.* [2]. The solution is to estimate the robot pose and landmark global positions in a single state vector with a single covariance matrix, updated by an Extended Kalman Filter (EKF). This method has been shown to allow the map to converge monotonically to a perfect relative map with zero uncertainty in the limit as the number of landmark observations increases [3]. It is the method used in this work. However, a drawback of this method is its $O(N^2)$ computational complexity where N is the number of landmarks estimated in the EKF.

Solving the SLAM problem using the EKF and vision as the sensor has been demonstrated in works such as [4], [5]. In these works, each landmark is a single visual feature. Due to the EKF's $O(N^2)$ computational complexity, they are limited to estimating a maximum of 100 features in their map for real-time processing. For example, Davison *et al.* [5] constrain their algorithm to track at most 12 single feature landmarks at any one time to keep the total number of landmarks under 100. However, a problem is that vision algorithm robustness can degrade with a reduced number of tracked features. Furthermore, these vision algorithms have not been tested in outdoor environments with extreme directional sunlight causing shadows and highly non-uniform scene illumination.

III. METHOD

A. Algorithm Design and Overview

The algorithm flow is based on the operation of the EKF. 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: Feature Cluster landmarks are matched to observed 3D features. Landmark poses relative to the camera are found.
4) EKF Measurement Update: The measured Feature Cluster landmark positions are fed into the EKF. Camera pose and landmark position estimates are updated.
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 extracted from the current frame is added to the system.
7) Landmark Feature Management: Landmark features that have often failed to be matched are removed from the landmark database.

B. 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 $\hat{\boldsymbol{x}}$ is a large column vector in which the states are stacked:

$$\hat{\boldsymbol{x}} = \begin{bmatrix} \hat{\boldsymbol{x}}_{\boldsymbol{v}} \\ \hat{\boldsymbol{o}}_{\boldsymbol{l_1}} \\ \hat{\boldsymbol{o}}_{\boldsymbol{l_2}} \\ \vdots \\ \hat{\boldsymbol{o}}_{\boldsymbol{l_n}} \end{bmatrix} \tag{1}$$

where $\hat{\boldsymbol{x}}_{\boldsymbol{v}}$ is a non-landmark state vector which will be described in detail, and $\hat{\boldsymbol{o}}_{\boldsymbol{l_1}}, \hat{\boldsymbol{o}}_{\boldsymbol{l_2}}, \ldots, \hat{\boldsymbol{o}}_{\boldsymbol{l_n}}$ are landmark states. A landmark state $\hat{\boldsymbol{o}}_{\boldsymbol{l_i}}$ (3×1) represents the global 3D position of the i'th Feature Cluster landmark. Whenever a new landmark is added to the system, $\hat{\boldsymbol{x}}$ is enlarged by a landmark state.

The non-landmark state vector $\hat{\boldsymbol{x}}_{\boldsymbol{v}}$ (12×1) is:

$$\hat{\boldsymbol{x}}_{\boldsymbol{v}} = \begin{bmatrix} \hat{\boldsymbol{o}}_{\boldsymbol{c}} \\ \hat{\boldsymbol{r}}_{\boldsymbol{c}} \\ \hat{\boldsymbol{o}}_{\boldsymbol{r}} \\ \hat{\phi}_r \\ \hat{\theta}_r \\ \hat{\omega} \end{bmatrix} \tag{2}$$

Here, $\hat{\boldsymbol{o}}_{\boldsymbol{c}}$ (3×1) is the camera origin, and the camera rotation vector $\hat{\boldsymbol{r}}_{\boldsymbol{c}}$ (3×1) represents the camera frame $\underline{\boldsymbol{C}}_{\boldsymbol{c}}$ (3×3). The vector $\hat{\boldsymbol{r}}_{\boldsymbol{c}}$ can be transformed into $\underline{\boldsymbol{C}}_{\boldsymbol{c}}$ by matrix exponential: $\hat{\underline{\boldsymbol{C}}}_{\boldsymbol{c}} = e^{\hat{\boldsymbol{r}}_{\boldsymbol{c}}\times}$. Next, $\hat{\boldsymbol{o}}_{\boldsymbol{r}}$ (3×1) 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 $\hat{\boldsymbol{x}}_{\boldsymbol{v_{t|t-1}}} = \boldsymbol{f_v}(\hat{\boldsymbol{x}}_{\boldsymbol{v_{t-1}}}, w_{n_{t-1}})$ 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:

$$\hat{\omega}_{t|t-1} = \hat{\omega}_{t-1} + w_{n_{t-1}} \tag{3}$$

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

$$\begin{bmatrix} \boldsymbol{z_i} \\ 1 \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{C}}_{\boldsymbol{c}} & \boldsymbol{o_c} \\ 0\ 0\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{o_{l_i}} \\ 1 \end{bmatrix} + \begin{bmatrix} \boldsymbol{v} \\ 1 \end{bmatrix} \tag{4}$$

where $\boldsymbol{v}$ is a zero-mean white gaussian process modelling sensor noise.

C. Initialization and Calibration

After mounting the camera, $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_{\boldsymbol{v}}$ needs to be initialized. In this work, the initial camera pose $\{\hat{\boldsymbol{o}}_{\boldsymbol{c}}, \hat{\boldsymbol{r}}_{\boldsymbol{c}}\}$ is used to define the global coordinate system so its values are initialized to zeros. The rotation centre $\hat{\boldsymbol{o}}_{\boldsymbol{r}}$ 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 [3]. 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.

D. 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.

1) Locally Maximal Harris Corners: The Harris corner detector has been chosen for this work as feature scale invariance is not needed. Also, it has been shown to have high repeatability over a range of conditions such as viewpoint change and rotation, compared to other interest point detectors [6], [7]. However, these findings were conducted indoors and it has been found that in our outdoor setting,

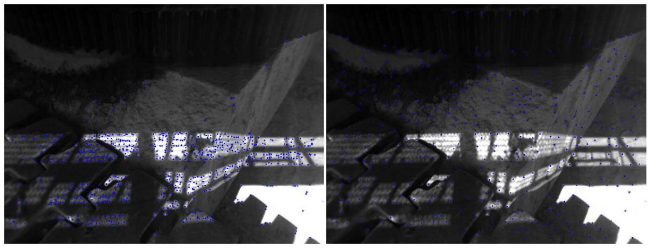

Fig. 3. Comparison of the two corner selection methods. Blue dots indicate corners. Left: Percent Threshold corner selection. Threshold set as 0.01 of highest corner measure in image. Right: Locally Maximal corner selection.

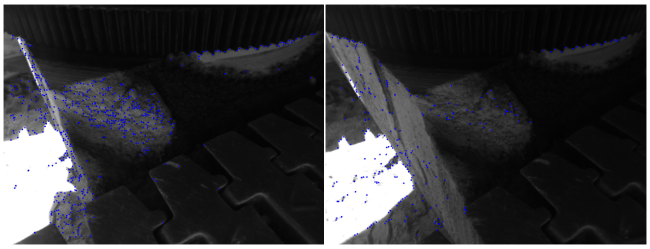

Fig. 4. Corner selection using the Percent Threshold method. Shown are 2 sequential snapshots, where left image is the earlier snapshot and the right image is latter. Snapshots were taken by the camera as it swings left toward the directly-lit scene region from the shadow scene region. The directly-lit scene region is located on left sides of the images and the shadow scene region is located on right sides of the images. Corner measures of features in the shadow scene region fall dramatically as camera swings toward the directly-lit scene region. Many features in the shadow scene region detected in the earlier snapshot are no longer detected in the latter snapshot.

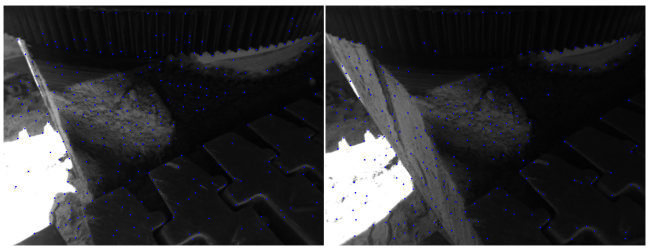

Fig. 5. Using the locally maximal criteria, corners are more consistently selected over changing image exposure.

the traditional Harris corner selection method does not work well. The problem and solution are explained below.

To select corner locations, the Harris corner measure is first evaluated for each pixel location. Then, corners are selected based on a corner threshold. However, the Harris corner measure is sensitive to image contrast. That is, given two identical image structure but with one image structure having a higher contrast, the higher contrast image structure will have higher Harris corner measure. Thus, selecting a suitable corner threshold for all lighting conditions can be difficult. Traditionally, the corner threshold is simply set as a percentage (e.g. 1%) of the highest corner measure found in the image [6], [8]. In this paper, this is referred to as the "Percent Threshold" corner selection method.

The Percent Threshold corner selection method has been found to be unsuitable for our outdoor scene where 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 high local contrast. Using the Percent Threshold corner selection method, most corners are selected on the shadow patterns. This is shown in the left image of Figure 3. The selection of most features on the shadow patterns is highly problematic because the shadow patterns move, so the selected shadow features cannot be reliably used to reference the camera pose.

A problem also occurs when the camera swings from the scene region under shadow to the scene region directly-lit by sunlight. The shadow boundaries between the two regions create "corners" of high corner measures, boosting the corner threshold. Yet, the camera shutter time shortens dramatically as the camera swings toward the directly-lit scene region, causing features originally detected in the shadow scene region to become less exposed and their local image contrast to fall. As a result, corner measures of these features quickly fall below the corner threshold and these features become undetected (Figure 4).

A novel metric is used to select corners more consistently and evenly without a corner threshold. Here, it is called the "Locally Maximal" Harris corner selection. 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 region centered about the pixel location. Compared to the method used in [9] which divides the whole image into square blocks and selects the strongest feature in each block, this method is more flexible and does not force the selection of a single feature when neighboring features from different image blocks become in the same block.

The parameter W of the Locally Maximal corner selection method controls the neighborhood size and can be used to tune the desired number of features per camera frame. The value of W has been set to 4, which results in the detection of roughly 600 features per frame. Processing this number of features has been found to provide a good balance between algorithm accuracy and speed.

Figures 3 and 5 show corner detection results using the new method. Although some corners are still selected on shadow patterns, they can be eliminated in latter processing. Many of these non-physical corners are non-distinctive 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.

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 5×5 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 [10]. 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:

$$\begin{aligned} {}^{c}z &= fB/d \\ {}^{c}x &= (u - u_0)\ {}^{c}z/f \\ {}^{c}y &= (v - v_0)\ {}^{c}z/f \end{aligned} \tag{5}$$

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.

3) Descriptor: Once the detected Harris corners are successfully stereo-matched, the image region centered about each corner is transformed into the corner's descriptor. The SIFT descriptor [11] has been selected for use, based on the finding that it has one of the best matching performances under changes in various imaging conditions such as view point [12].

E. Feature Cluster Landmarks

For all vision-based EKF SLAM works the authors are aware of, each landmark is a single visual feature. As features are initialized and tracked as landmarks, the EKF state vector expands and the computational cost of the EKF grows quadratically. Thus, for real-time processing the number of features tracked as landmarks is highly constrained. For example, the number of tracked landmark features at any moment was limited to 12 in [5].

However, single features are unreliable landmarks. 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 feature measurements. A landmark feature may be matched to the wrong observed feature, or the 3D position of the landmark feature's match may be poorly measured. Lastly, a landmark feature may not even be a real physical feature, such as a feature found on a moving shadow pattern. Such a non-physical feature cannot be reliably used as a reference for finding the camera pose.

Tracking many landmark features can improve the vision algorithm's robustness. If some landmark features are not re-detected in subsequent frames, there are numerous other landmark features that can be matched, providing references for finding the current camera pose. In addition, a large number of landmark feature matches allows confident 3D geometric verification which can eliminate erroneous feature matches. Yet, increasing the number of landmark features will significantly slow down the vision algorithm due to the EKF's quadratic computational complexity.

An ideal landmark is one that can be detected consistently across frames and measured reliably up to a large swing angle away from the camera position at which it was initialized. Given such ideal landmarks, the number of landmarks estimated in the EKF can be reduced. Unfortunately, no single feature is highly reliable. Thus, this inspired the use of a "higher level" landmark which can be consistently and robustly measured. Here, it is called the Feature Cluster landmark. The proposed landmark is a cluster of numerous 3D features detected in a single camera frame (keyframe). The cluster of detected 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 the cluster coordinate system (Figure 6). It is the cluster's relative pose to the camera that is measured and the cluster's global position that is filtered in the EKF, rather than the individual features. 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 cluster pose with respect to the camera estimated. In addition, cluster feature matches can be geometrically verified during the cluster's relative pose estimation. Erroneous feature matches would not agree with the cluster pose estimated from the majority of matches. Figure 7 illustrates the Feature Cluster landmark pose estimation.

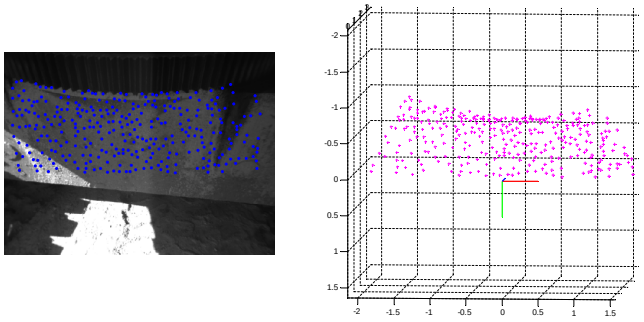

Fig. 6. Left: Detected cluster features from a keyframe. Roughly 350 3D features form the feature cluster. Right: The feature cluster and its coordinate system.

1) Feature Cluster Landmark Initialization: The system initializes a new Feature Cluster landmark when no existing landmark is well-tracked in the current frame. The first step is to select features observed in the current frame that will make up the feature cluster. Essentially, all detected features in the current frame are selected as cluster features, with some criteria. Firstly, features are selected using a slightly larger neighborhood block $W = 5$ rather than the normal $W = 4$ to allow for higher repeatability of their detection in subsequent frames. Lastly, only features in the top two-third portion of the camera image are selected as cluster features, to avoid selecting too many features from the shovel tracks.

Once cluster features are selected, a cluster coordinate system $\{\underline{\boldsymbol{C}}_{\boldsymbol{l}},\ \boldsymbol{o}_{\boldsymbol{l}}\}$ needs to be attached to the cluster and cluster feature positions need to be represented in terms of this cluster coordinate system. The cluster coordinate system can be placed anywhere relative to the cluster features. As long as it remains fixed relative to the cluster features, consistent landmark measurements can be made and the camera pose can be properly estimated by the EKF. The cluster coordinate system is simply placed at

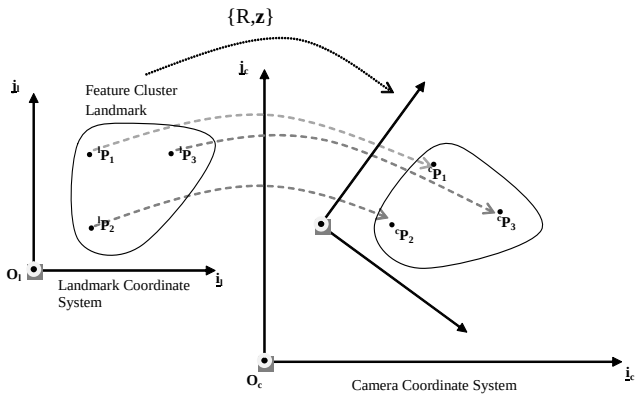

Fig. 7. Once a minimum of 3 non-collinear landmark features are matched to observed features, the landmark rotation R and position $\boldsymbol{z}$ relative to the camera can be found. In actual, a landmark consists of over 100 features so there will be numerous feature matches. Iterative Gauss-Newton least squares procedure is used to solve for the landmark's relative pose $\{\boldsymbol{R}, \boldsymbol{z}\}$ such that cluster features' landmark coordinates $\{{}^{l}\boldsymbol{P_1}, {}^{l}\boldsymbol{P_2}, {}^{l}\boldsymbol{P_3}, \ldots\}$ transform into their observed camera coordinates $\{{}^{c}\boldsymbol{P_1}, {}^{c}\boldsymbol{P_2}, {}^{c}\boldsymbol{P_3}, \ldots\}$. Geometric verification is performed simultaneously during this pose estimation as erroneous feature matches will become outliers.

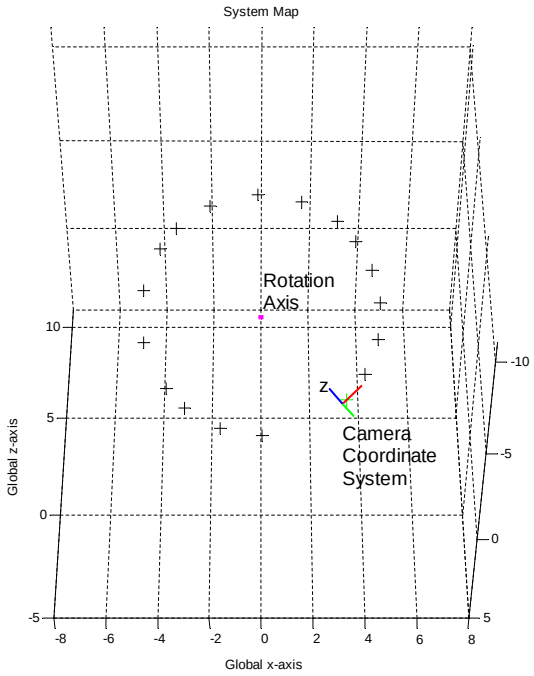

Fig. 8. Snapshot of the system map as camera swings clockwise before making loop closure. Map is plotted in the 3D global coordinate system, which is defined by the initial camera pose at $t = 0$. Axis scales are in meters. Plot view is set looking down the shovel house rotation axis. Note that the rotation 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.

the same location relative to the cluster features as the current camera coordinate system. Then, cluster features' landmark coordinates $\{{}^{l}\boldsymbol{P_1}, {}^{l}\boldsymbol{P_2}, {}^{l}\boldsymbol{P_3}, \ldots\}$ are simply their camera coordinates $\{{}^{c}\boldsymbol{P_1}, {}^{c}\boldsymbol{P_2}, {}^{c}\boldsymbol{P_3}, \ldots\}$ in the current frame. The cluster features' landmark coordinates and descriptors are stored to the system database as the full description of the new Feature Cluster landmark.

Next, the EKF state vector is expanded by size 3×1 for filtering the new cluster's global position (or origin). Given the selection of the cluster coordinate system as the current camera coordinate system, a cluster's global position is equivalent to the camera's global position at which the cluster was initialized. Thus, the cluster's estimated position in the EKF is initialized as the estimated current camera position. Figure 8 shows a snapshot of landmark positions as the camera swings clockwise before making loop closure; it can be seen that the landmarks are initialized along the circular trajectory of the camera.

F. Landmark Measurement

1) Active Landmark Feature Matching: After EKF prediction update and 3D feature detection, the system proceeds to match Feature Cluster 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 cluster feature is matched against 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 [11] is applied to remove ambiguous matches. In particular, it has been observed that many non-distinctive features such as those formed from the metal mesh walkway's shadow pattern are eliminated after this step.

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 cluster features that have been matched, their positions in landmark coordinates $\{{}^{l}\boldsymbol{P_1}, {}^{l}\boldsymbol{P_2}, {}^{l}\boldsymbol{P_3}, \ldots, {}^{l}\boldsymbol{P_M}\}$, and their corresponding positions as observed by the camera $\{{}^{c}\boldsymbol{P_1}, {}^{c}\boldsymbol{P_2}, {}^{c}\boldsymbol{P_3}, \ldots, {}^{c}\boldsymbol{P_M}\}$ are obtained. The goal is to find the feature cluster's rotation $e^{\boldsymbol{r}\times}$ and position $\boldsymbol{z}$ with respect to the camera, so that:

$$\begin{aligned} {}^{c}\boldsymbol{P_1} &= [\ e^{\boldsymbol{r}\times}\ ][\ {}^{l}\boldsymbol{P_1}\ ] + \boldsymbol{z} \\ {}^{c}\boldsymbol{P_2} &= [\ e^{\boldsymbol{r}\times}\ ][\ {}^{l}\boldsymbol{P_2}\ ] + \boldsymbol{z} \\ {}^{c}\boldsymbol{P_3} &= [\ e^{\boldsymbol{r}\times}\ ][\ {}^{l}\boldsymbol{P_3}\ ] + \boldsymbol{z} \\ &\vdots \\ {}^{c}\boldsymbol{P_M} &= [\ e^{\boldsymbol{r}\times}\ ][\ {}^{l}\boldsymbol{P_M}\ ] + \boldsymbol{z} \end{aligned} \tag{6}$$

Here an overdetermined non-linear system of equations is established. The relative landmark pose $[\boldsymbol{z}; \boldsymbol{r}]$ is found as the least squares solution of the above equation using the iterative Gauss-Newton algorithm [13] [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 dis-

carded 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. Figure 9 shows match results after geometric verification.

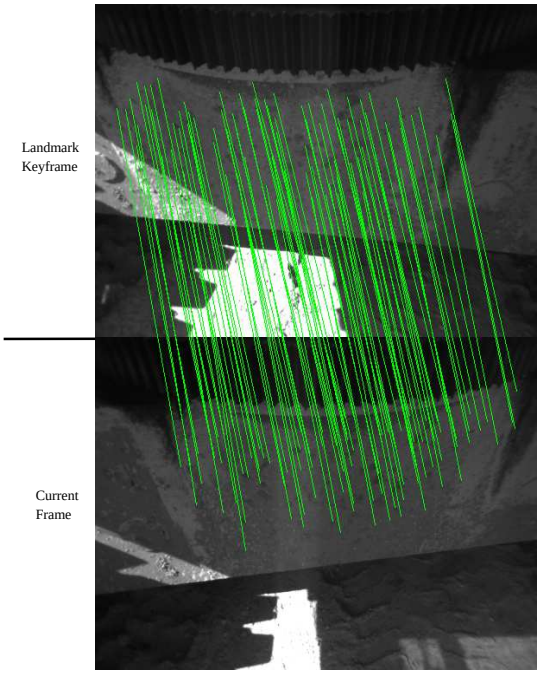

Fig. 9. Remaining feature matches after geometric verification.

It was observed that generally, the further the camera swings away from a landmark, the larger the landmark's pose estimation residual becomes. The average residual ranged from approximately 0.3cm when the camera viewed the landmark directly, up to 3.0cm 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.

G. 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 $\hat{\boldsymbol{o}}_{\boldsymbol{c}}$, the reference landmark position $\hat{\boldsymbol{o}}_{\boldsymbol{l_1}}$, the rotation center $\hat{\boldsymbol{o}}_{\boldsymbol{r}}$, and the rotation axis direction $\{\hat{\phi}_r, \hat{\theta}_r\}$.

First, the two vectors $\boldsymbol{v} = [\hat{\boldsymbol{o}}_{\boldsymbol{c}} - \hat{\boldsymbol{o}}_{\boldsymbol{r}}]$ and $\boldsymbol{w} = [\hat{\boldsymbol{o}}_{\boldsymbol{l_1}} - \hat{\boldsymbol{o}}_{\boldsymbol{r}}]$ are computed. Next, the unit vector $\hat{\boldsymbol{r}}_{\boldsymbol{r}}$ pointing in the direction of the rotation axis, is found from $\{\hat{\phi}_r, \hat{\theta}_r\}$. Then, $\boldsymbol{v}$, $\boldsymbol{w}$ are projected onto a plane perpendicular to $\hat{\boldsymbol{r}}_{\boldsymbol{r}}$, giving $\boldsymbol{v}_\perp$, $\boldsymbol{w}_\perp$ respectively. Finally, the swing angle θ_s is found as the angle between $\boldsymbol{v}_\perp$ and $\boldsymbol{w}_\perp$ (Figure 10).

H. 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

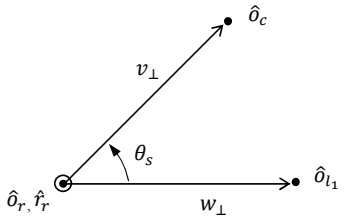

Fig. 10. View looking down $\hat{\boldsymbol{r}}_{\boldsymbol{r}}$, in direction of shovel rotation axis. Vectors $\boldsymbol{v}_\perp$, $\boldsymbol{w}_\perp$ are respectively the projections of $[\hat{\boldsymbol{o}}_{\boldsymbol{c}} - \hat{\boldsymbol{o}}_{\boldsymbol{r}}]$ and $[\hat{\boldsymbol{o}}_{\boldsymbol{l_1}} - \hat{\boldsymbol{o}}_{\boldsymbol{r}}]$ on a plane perpendicular to $\hat{\boldsymbol{r}}_{\boldsymbol{r}}$. Swing angle found as angle between $\boldsymbol{v}_\perp$, $\boldsymbol{w}_\perp$.

stays in the map permanently. 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 11 shows features of a landmark at initialization and after.

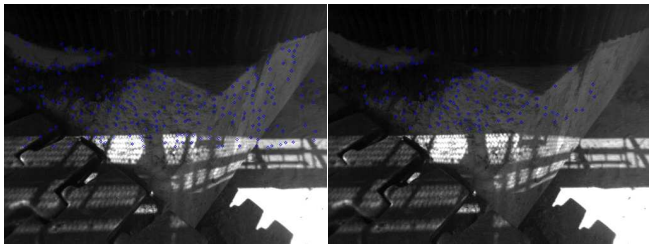

Fig. 11. Left: Initial features of a landmark. Right: After pruning of bad features over time. Note shadow features at the bottom have been pruned.

IV. Results

A. Swing Angle Results

The vision algorithm was tested on 3 separate video sequences taken on different days, with each sequence over 5000 frames (4min) long . Results for one sequence is discussed. During this sequence, the shovel swung 2 counter clockwise revolutions, then 2 clockwise revolutions, and then 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 12.

Initially, the vision system's error grew up to 5.5° 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 behavior, the drift error of the camera pose estimate was immediately corrected so the error in the estimated swing angle dropped to 0.5°. At the same time, this drift error correction 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°. Given that the camera was located 4.5m from the rotation center, the camera positional error was equivalent to

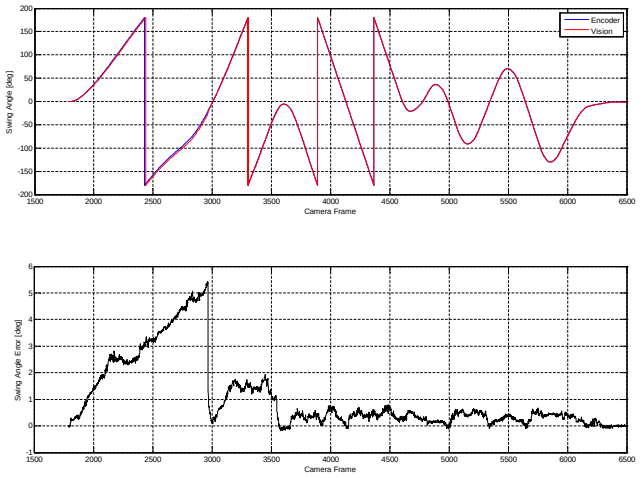

Fig. 12. Top: Vision and encoder outputs. Bottom: Difference between vision and encoder outputs.

+/- 0.08m over a 28m circular trajectory, or +/- 0.3%. This swing angular error was also equivalent to +/- 17cm shovel bucket localization error, assuming a truck loading distance of 10m. This was within the accuracy requirement of the aforementioned collision avoidance system.

In total, only 19 Feature Cluster landmarks were initialized along the entire circular path traversed by the camera. This was owing to the fact that each Feature Cluster landmark, consisting over 100 features, could be tracked reliably up to +/- 20° swing angle. Compared to previous works which filtered 100 single feature landmarks, the number of landmarks estimated in our EKF was much lower but yet the number of tracked features was much higher.

The algorithm is still in a prototype stage and is implemented in a mixture of C++ and Matlab. The algorithm ran at 4Hz on a single core of a 1.8GHz Intel Core2 CPU with 3GB RAM.

B. Comparison of Harris Corner Selection Methods

The Locally Maximal Harris corner selection method was compared to the traditional Percent Threshold corner selection method. In one comparison, it was found that the number of features detected in each frame fluctuated wildly between 200-2000 with the Percent Threshold method, versus the stable number of 450-700 with the Locally Maximal method.

In another comparison, the SLAM algorithm was run using the Percent Threshold method instead of the Locally Maximal 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 results using the Locally Maximal method. The test was performed for 3 different starting keyframes representing different lighting conditions. With the Percent Threshold method, the tracking results varied depending on the lighting condition of the starting keyframe. In the worst case in which the starting keyframe contained moving shadow patterns, cluster features were tracked to 2° swing angle only. With the Locally Maximal method, cluster features could be consistently tracked to 20° for all 3 starting keyframes.

V. 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 allows features to be selected evenly across the image and more consistently compared to the traditional Percent Threshold method. The use of Feature Cluster Landmarks allows the algorithm to make robust and consistent landmark measurements due to the large number of features per landmark. Moreover, the total number of landmarks estimated in the EKF can be reduced, lowering the EKF computational cost .

References

[1] N. P. Himmelman, J. R. Borthwick, A. H. Kashani, L. Lin, A. Poon, R. A. Hall, P. D. Lawrence, S. E. Salcudean, and W. S. Owen, "Rope shovel collision avoidance system," in *Application of Computers and Operations Research in the Mineral Industry*, Oct. 2009.

[2] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Autonomous robot vehicles*, vol. 1, pp. 167–193, 1990.

[3] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 3, pp. 229–241, 2001.

[4] P. Jensfelt, J. Folkesson, D. Kragic, and H. Christensen, "Exploiting distinguishable image features in robotic mapping and localization," in *European Robotics Symposium 2006*. Springer, 2006, pp. 143–157.

[5] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-Time single camera SLAM," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1052–1067, 2007.

[6] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 151–172, Jun. 2000.

[7] scar Mozos, A. Gil, M. Ballesta, and O. Reinoso, "Interest point detectors for visual SLAM," in *Current Topics in Artificial Intelligence*. Springer, 2007, pp. 170–179.

[8] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[9] A. E. Johnson, S. B. Goldberg, Y. Cheng, and L. H. Matthies, "Robust and efficient stereo feature tracking for visual odometry," in *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008*, 2008, pp. 39–46.

[10] P. Fua, "A parallel stereo algorithm that produces dense depth maps and preserves image features," *Machine Vision and Applications*, vol. 6, no. 1, pp. 35–49, Dec. 1993.

[11] D. G. Lowe, "Distinctive image features from Scale-Invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[12] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1615–1630, 2005.

[13] D. G. Lowe, "Fitting parameterized three-dimensional models to images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 5, pp. 441–450, 1991.

[14] P. Saeedi, P. Lawrence, and D. Lowe, "Vision-based 3-D trajectory tracking for unknown environments," *Robotics, IEEE Transactions on*, vol. 22, no. 1, pp. 119–136, 2006.