final report

Project code: A.TEC.0078

Prepared by: Strategic Engineering Pty. Ltd.

Date submitted: August 2011

PUBLISHED BY
Meat & Livestock Australia Limited
Locked Bag 991
NORTH SYDNEY NSW 2059

Beef hock location vision system

Meat & Livestock Australia acknowledges the matching funds provided by the Australian Government and contributions from the Australian Meat Processor Corporation to support the research and development detailed in this publication.

Contents

1 Scope

The aim of the project was to develop a vision system capable of locating and tracking the rear hocks of beef cattle in 3 dimensional space while lying in a cradle shortly after stunning. This report details the methodology used and results gained throughout the Beef Hock Location Vision System project. It covers system design, hardware selected, software implementation and testing, and results gained from system trials.

2 Project Overview

The Beef Hock Location Vision System project will produce a robust and reliable system for locating and tracking the rear legs of beef cattle in 3D space. The desired rear leg is to be identified, and its position and orientation tracked over time. The system is intended to be interfaced to an industrial robot, guiding the position of the robotic tool to perform the shackling procedure. Hardware components have been purchased in order to develop and test the Beef Hock Location Vision System. The components include a Time of Flight camera, a Laptop Computer, and all necessary cables.

The vision system will consists of the following two main parts:

1. **Scene capture –** A hardware sensor platform supplying digital measurements of ranges to objects in the vicinity, and

2. **Scene processing –** A software platform analysing captured scenes, detecting the presence and motions of an animal in the cradle then identifying and tracking the desired leg.

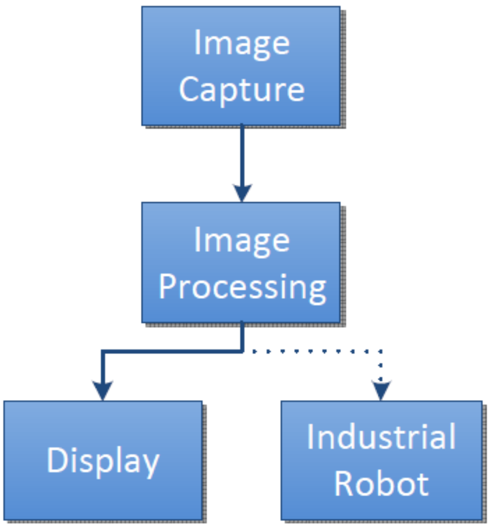

Figure 1: Vision System Block Diagram

As shown in the diagram above, a display will be an ancillary part of the system, used to display the output of the vision processing subsystem in real-time and to demonstrate the output that would be available to an interfaced industrial robot.

3 System Hardware

3.1 Image Capture Device

It was decided that the Time of Flight (ToF) camera provided the solution that best met the requirements of the project, primarily because it is capable of capturing an entire scene at once. The absence of such an ability is a major drawback of technologies such as LiDAR and Stereoscopic vision. Distance computation and correction for lens distortion are both performed on-board the camera and provided in real time with each frame. This results in superior spatial capturing of objects and allows rapid and accurate detection of objects' dimensions and positions. For these reasons, the Time of Flight camera was chosen.

A ToF camera is a camera system that captures vision of a scene and creates distance data using pulses of infra-red light. A pulse of infra-red light illuminates the scene and is reflected by the objects. The camera lens gathers the reflected light and images it onto the sensor plane. Depending on the distance to objects in the scene, the incoming light experiences a delay. This delay is measured and distance data is calculated.

As the distance data is provided natively by the ToF camera, no additional computation is required to post-process the captured data to provide distance measurement. As the entire scene is captured and distance data calculated at the one time, precise real-time tracking of the desired cattle leg is possible at the full frame rate of the selected camera.

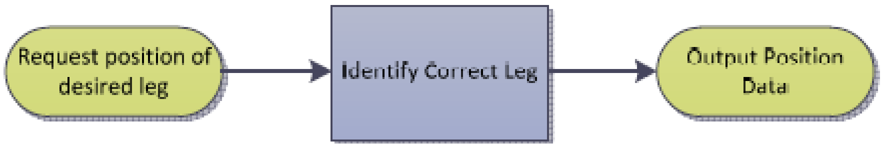

Figure 2: Basic control flow of the Vision System

The implementation of a ToF camera to generate a 3D display of the scene is very compact, being an all-in-one system. As the lighting is built into the camera housing, the system is not reliant on any external parts. Additionally, unlike Stereoscopic Vision, only one camera is necessary to acquire a three-dimensional scene. This minimises the footprint of the system on site, maximises reliability, and reduces the number of components and resulting maintenance burden.

The PMDTec CamCube3 Camera was selected as most suiting the requirements, having a faster sampling rate than the IFM Efector (54Hz compared with 25Hz) and providing a housing with a higher protection rating (IP67) than the SwissRanger (IP40) and Baumer (IP65) alternatives.

Additionally, the PMDTec camera has a higher resolution (200x200 pixels) and accuracy (<3mm at 1σ) at the intended range than the Swissranger and Baumer alternatives.

A +12V external power source is required to power the CamCube3 and optic expansion modules. A USB 2.0 connection is required to establish a communications link between the CamCube3 and the image processing hardware. USB 2.0 has been widely adopted in the computing industry, and is readily available on all modern computers.

Figure 3: PMDTec CamCube3

3.2 Image Processing Hardware

A commercial off the shelf laptop was used to execute the vision system software. A gaming laptop was preferable as they are generally higher performance machines with high performance CPUs, more Random Access Memory and a dedicated graphics adapter which may be used to perform a portion of the computational operations. Future revisions of the system will likely feature an embedded solution, such as a single board computer or equivalent.

The laptop was loaded with the GNU based Ubuntu Operating System. This Operating System was desirable as it is an open source distribution (free from any licensing fees) and contains POSIX compliant architecture. The operating system was configured to run the minimal services necessary whilst the vision system was operational. This approach dedicated the bulk of the system resources to the vision system, allowing it to run as quick and efficiently as possible.

The laptop screen is used for the visual output during the development and verification phase of the project to demonstrate the functionality of the system. When the system is to be deployed as a part of a wholly automated Beef Cattle Shackling System the display may not be required though this requirement will be revised.

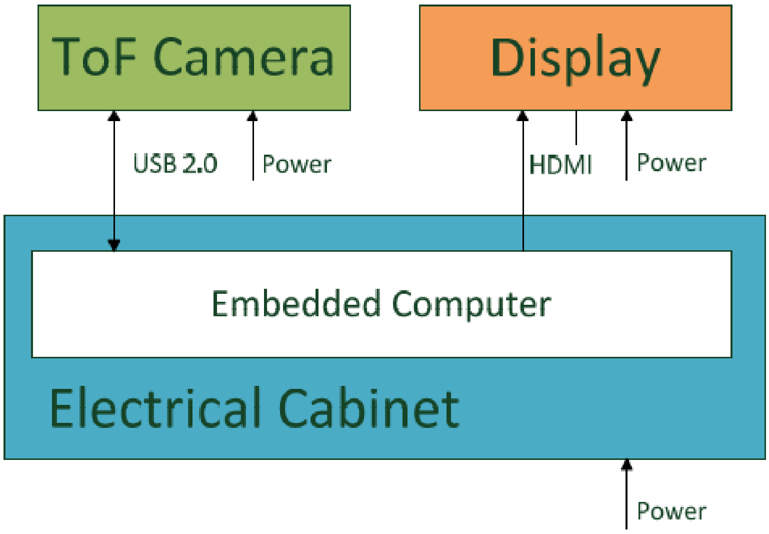

Figure 4: Vision System Hardware Diagram

4 Analysis Software Functionality

4.1 Status

The vision analysis software portion of the system has been successfully developed and implemented in the C++ programming language. The software has been packaged to be POSIX compliant as all test devices will be configured to run in such environments. This ensures that the software developed will be able to be directly ported to functional units with no need to modify the source code.

The software allows for a three dimensional scene capture via the Time of Flight camera. The scene is then analysed, Features of Interest (beef hocks) are then detected and relevant Features of Interest are then parsed. The now isolated features are then examined in order to determine the correct feature (in this case the rear leg of beef cattle) and once identified, distance, position and orientation information data is extracted.

This vision analysis consists of several steps, which are comparatively distinct from one another, so are treated separately and are discussed in detail below. There exist well-established industry standard algorithms for some (but not all) of these steps. The software has been tested extensively in-house and on-site under a variety of conditions. The software is currently configured to track rear cattle hocks in real time and correctly identifies and isolates them from the rest of the scene.

The results from preliminary trials have proven quite promising, correctly locating and tracking beef hocks with an accuracy of ±10mm. It is envisioned that such technologies may be easily adapted for use in other sectors of meat processing, partially or wholly automating repetitious and/or hazardous tasks.

5 Analysis Software Structure

The flowchart below shows the four stage high level process flow of the Beef Hock Location Vision System. Processes making up the analysis component of the software are highlighted. Once analysis of the captured images is complete and the candidate objects are enumerated, the leg will be identified and coordinates for shackling sent to the shackling system.

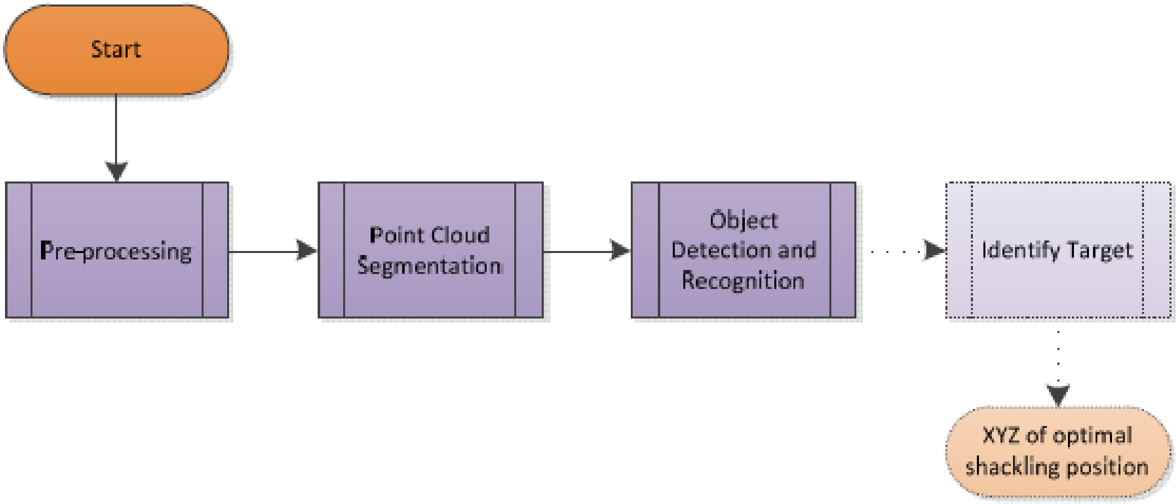

Figure 5: Vision System Process Overview

5.1 Platform

The image processing algorithms have been successfully implemented in C++ and fully comply with the POSIX interfacing standards. The POSIX platform is the ideal operating system for this application due to its low overhead, high customisability and its familiarity in industry.

Transitioning the software architecture from prototype (MATLAB) to production (POSIX compliant C++) allows for refinements to the software to be made in the environment that the final product will be implemented.

5.2 Libraries

In addition to custom written source code and algorithms the Beef Hock Location Vision System currently utilises several open-source software libraries. These libraries include the Point Cloud Library (PCL), wxWidgets and Visualisation Toolkit (VTK), which were used for data abstraction and image visualisation.

PCL is a highly optimised 3D point cloud library written in C++. The library supplies the necessary data abstraction to allow the image to be efficiently filtered, segmented and processed. It does this by providing a framework for n-dimensional point clouds and 3D geometry processing which then allows for the higher level cylinder detection and point cloud display to function.

VTK and wxWidgets provide the framework needed for quick and efficient visualisation of the point cloud data. They encompass visualisation algorithms and modelling techniques which allows the datasets to be easily displayed. The use of such tools encourages rapid refinement and development of core software modules.

6 Software Architecture

6.1 Pre-processing

The software architecture is designed to pre-process the point cloud data by utilising filters such as passthrough and outlier to isolate a region of interest. The region of interest, in this case the beast to be shackled, is then examined to determine leg location. The raw image must first be processed before the features may be identified. The dataset is filtered prior to analysis to remove data points that do not fall within a set region, or that fail to register a reasonable level of infrared intensity.

Performing these actions is appropriate as the beast will only be shackled while in the cradle, meaning that the vision system only needs to operate at a specific distance interval (1.5 - 2m for this particular application). This minimizes the amount of processing required as only data from the region of interest is processed and data from the frame in this distance, which may have passed through the background subtraction process, is ignored. This allows for the cell to be placed in an environment where the background image can change, such as having other employees or machinery operating in the background, and not adversely affect the vision system.

In addition to this, a high pass filter focusing on the intensity of the infrared signal is also used. This filter is set to remove any low confidence pixels whose intensity measurements fall below the acceptable threshold. Doing this eliminates erroneous system noise, and allows for a higher integration time to be set. Lengthening the integration time increases the ability of the system to detect black and glossy objects, but introduces significant noise into the system which can adversely impact the hock detection as well as the distance readings. Filtering out some of the noise allows for the integration time to be set relatively high without crippling the vision system with noise.

The data is then parsed through a planar model segmentation algorithm. This examines the data for any large flat or regular surfaces, and removes them from the dataset. This is done to remove large uninteresting areas and helps to lower the amount of data being processed, which increases speed.

The reason that this filter is implemented is to remove objects, such as the shackling cradle, the floor or a wall, from the frame. These features tend to be flat and regular which allows us to quickly recognise and disregard them.

Finally, the regioned data is checked for statistical outliers. This is done by comparing each data point with the remaining data set and determining its relevance dependant on it's the deviation with respect to the rest of the data set. Points which deviate too greatly are excluded as outliers. This removes noise that manages to make it through the regioning process and also serves to remove data points that, while located the correct distance away, are distinct from the main body of the image. This occurs rarely, such as when objects from the edge of the frame move into the shot.

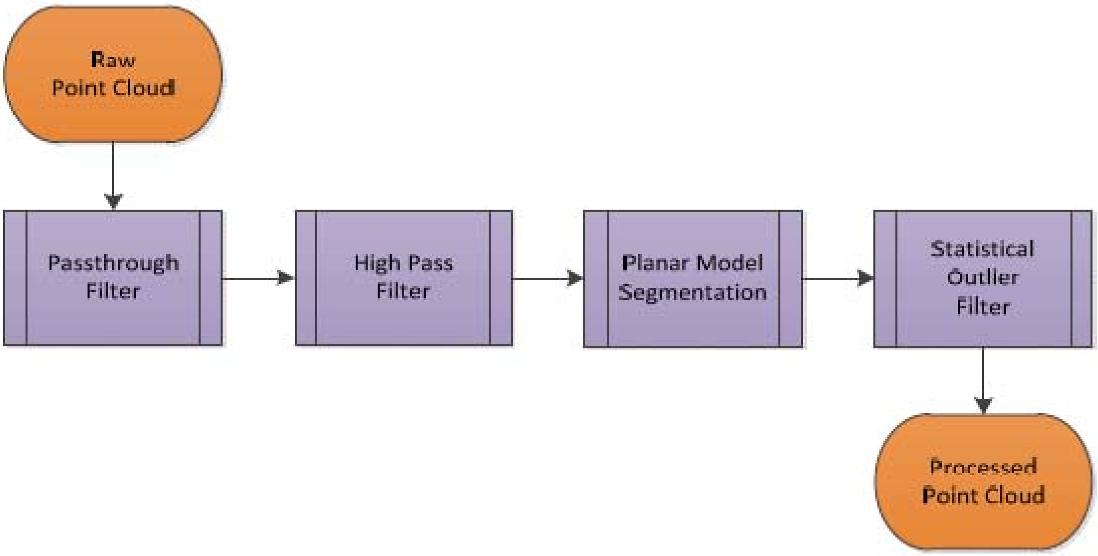

Figure 6: Process Flow to Determine the Region of Interest

Figure 7 below illustrates the contrast between processed and unprocessed point clouds. The left image contains significantly more noisy pixels in the scene which adversely impact the results of the vision system. Figure 8 depicts the pre-processing algorithms performed on sample cattle data.

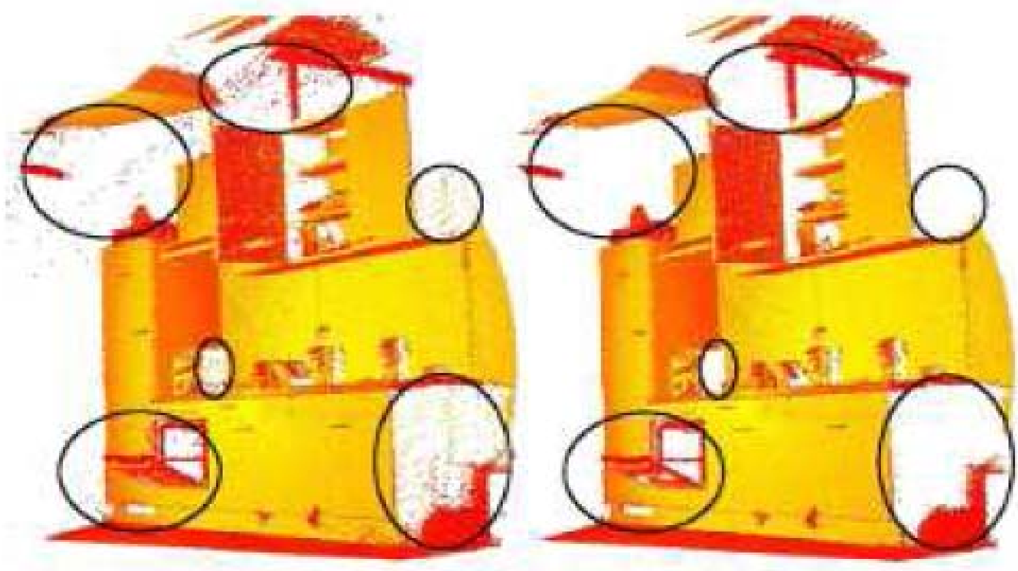

Figure 7: Unprocessed vs. processed point cloud

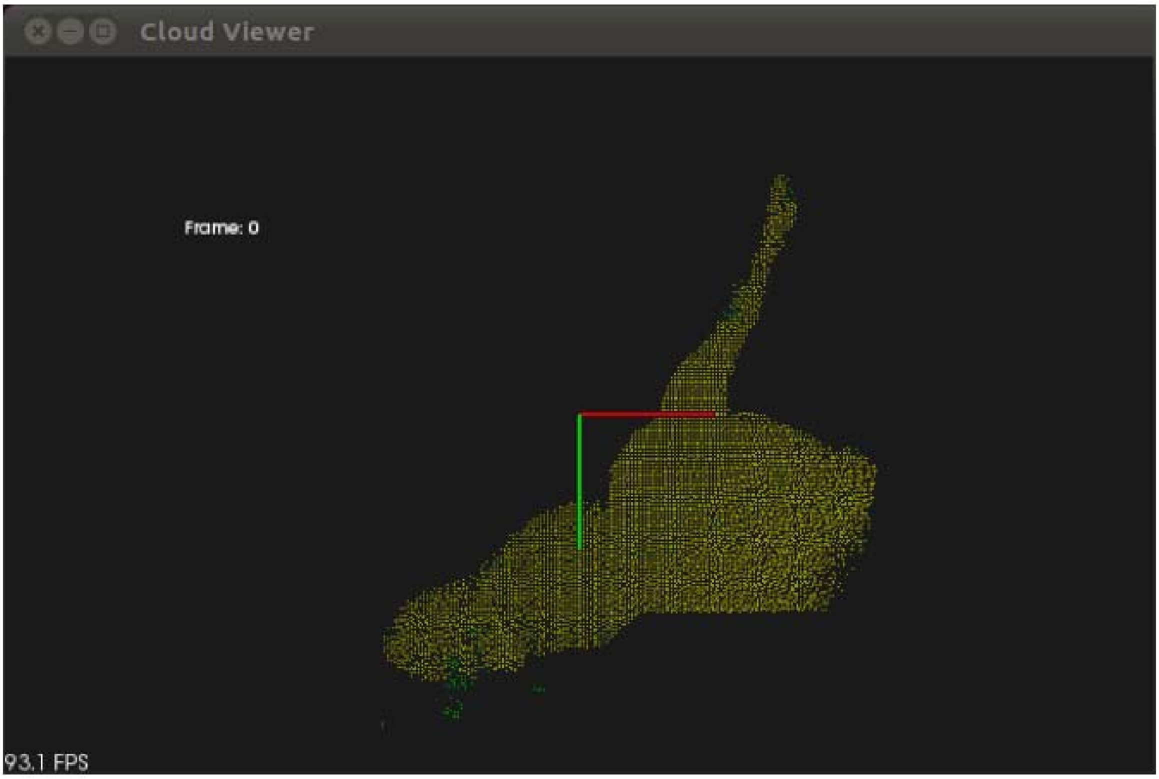

Figure 8: Filtered Point Cloud with Background Subtracted

6.2 Segmentation into Clusters

This step involves categorizing each range measurement within the given point cloud into smaller structured clusters. Each point is described by its three-dimensional Cartesian co-ordinate (x, y and z) and is categorized based on how well it conforms to the neighbouring points using strict selection criteria. This allows for the effective subdivision of the input point cloud to manageable, more meaningful subsets.

The constraint for classifying segments employed within this implementation is based on the 'smoothness constraint'. This requires the estimated surface normal of all points within one region not change too rapidly and that the points within a cluster be spatially connected. The use of only two constraint parameters provides easy adjustment of the system in the event of over-segmentation or under-segmentation in order to obtain the desired result.

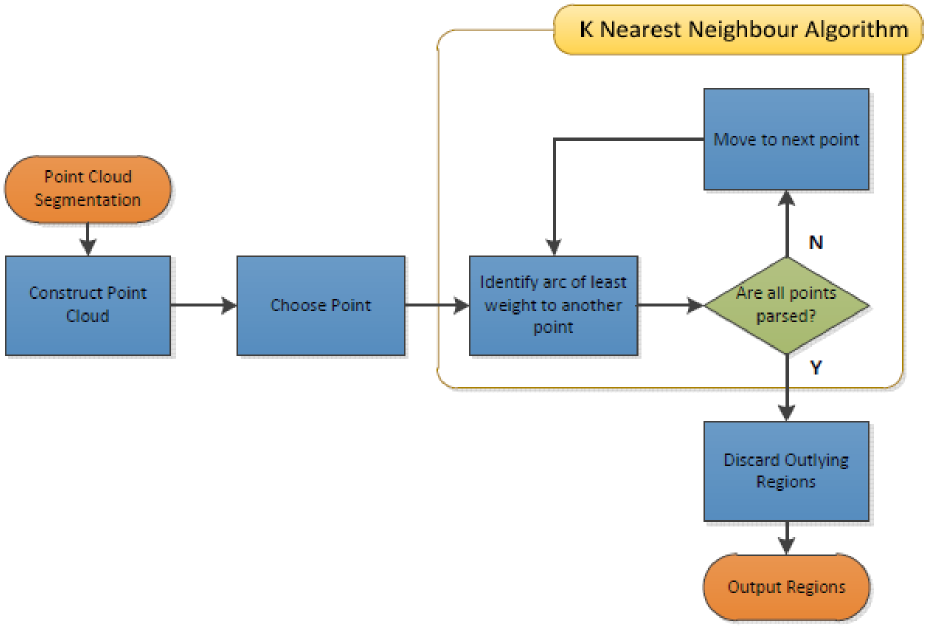

Figure 9: Point Cloud Segmentation Flowchart

Surface normals of each point are estimated using the K-Nearest-Neighbour algorithm. This process predicts the orientation of a surface plane based on a cluster of K neighbouring points and then calculates a normal vector and uncertainty factor. This is done to ensure that the system is able to distinguish between objects which are physically close to each other, but which are nevertheless distinct from each other. Among other uses, this step distinguishes the legs of the animal from the rest of its body, and each part of the animal from the cradle in which it lies. All of these regions are close to each other, but are not parts of the same object, so have radically differing surface normals at their closest points.

After completion of this step, many of the reconstructed region(s) can potentially be known not to be a part of an Object of Interest (OOI). Those regions, and regions containing points below a specified threshold may be discarded for greater efficiency. For example, this can occur when a given segment is wholly either too close or too distant from the camera to potentially be a part of an OOI.

In the general case, there will be multiple surfaces of multiple objects in the camera's field view at any given time, so the output of this step is multiple surfaces, some of which are not a part of the OOI.

Figure 10 below depicts the filtration and detection of a stunned bovine specimen on the slaughter floor of E.C. Throsby. The point cloud has been segmented into clusters and each cluster has been designated with a different colour.

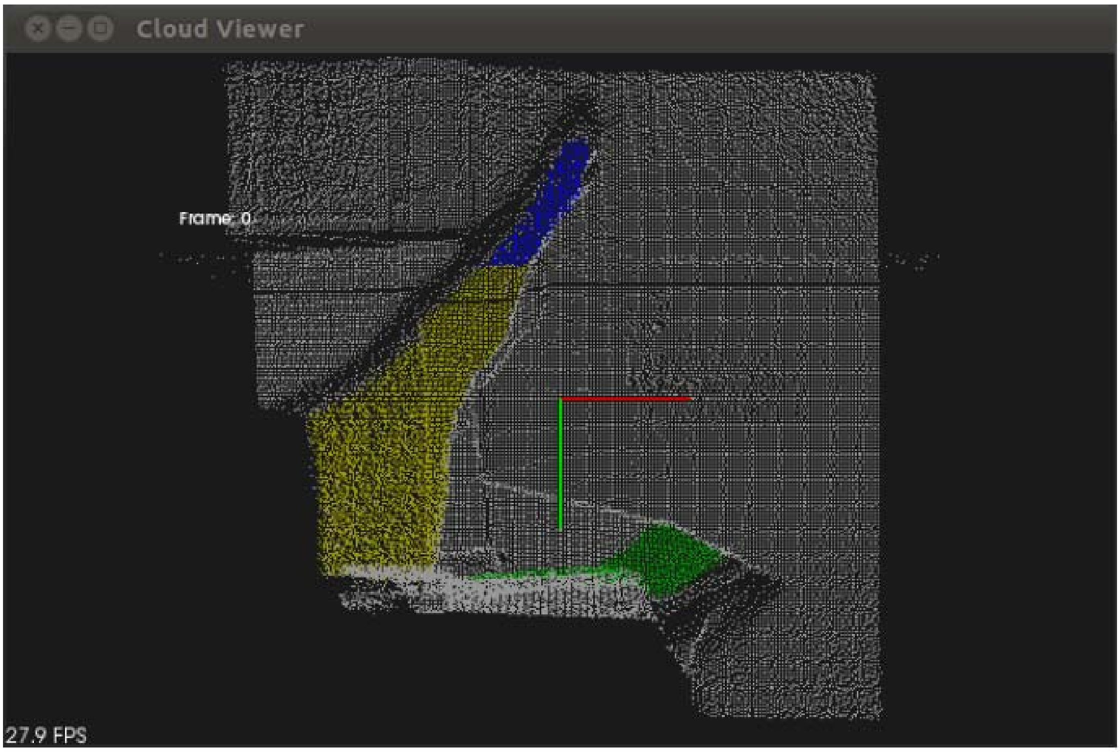

Figure 10: Point Cloud after Segmentation (colours designate segments)

6.3 Object Detection and Recognition

This step takes as input the set of connected regions yielded by the previous step, and compares each region to a bovine's leg (or part thereof). In the future, it may additionally need to specifically recognise and disregard uninteresting objects in the scene. This will almost certainly be necessary if such uninteresting objects partially occlude the Object of interest.

The leg of a bovine is a complex three-dimensional shape, and the precise shapes and sizes of bovines' legs differ by breed, gender and individual. It would therefore be both extremely computationally intensive and unreliable to directly search the input data for a specific and complex shape, such as a point cloud generated by a detailed high-resolution laser scan of the leg of an animal. A simpler and less specific test is necessary, for both efficiency and reliability reasons. A simplified geometric shape is a good choice: it is much more efficient, and it will more reliably detect OOI as OOIs, but will be correspondingly more likely to mis-detect other objects as OOIs.

Of the simple solid shapes which can be efficiently searched for by a cost-effective contemporary computer, a cylinder most closely resembles the leg of an animal. A frustrum (truncated cone) is also similar in shape, but incurs additional computational overhead without sufficient offsetting benefits.

Therefore, with some appropriate tolerance value, any region which is sufficiently dissimilar from a cylinder or cylinder fragment cannot be a part of the leg of an animal, so can be discarded from consideration at comparatively low computational cost.

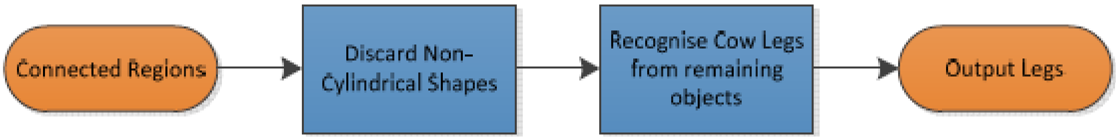

Figure 11: Object Detection and Recognition Flowchart

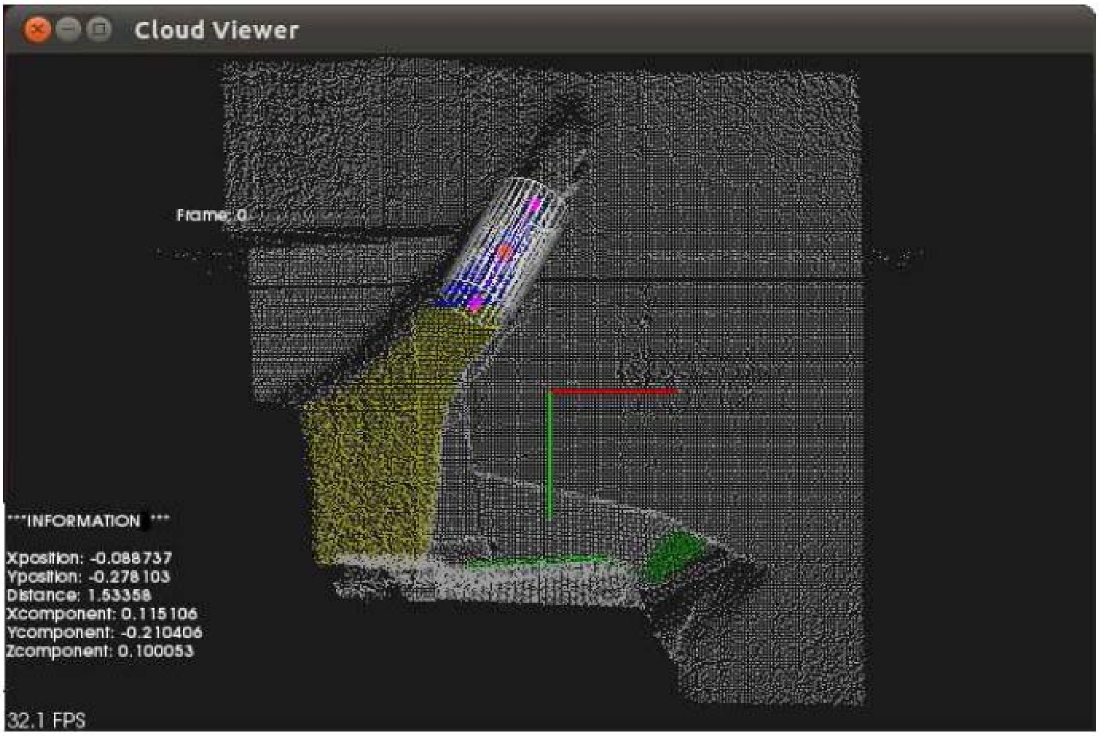

Figure 12: Processed Point Cloud with hock detected

Figure 12 above depicts the filtration, segmentation, and tracking algorithms being applied on a stunned bovine specimen. Two pink spheres have been drawn to depict the upper and lower extremities of the hock and a red sphere placed on the approximate shackling location. A cylinder has also been drawn to illustrate the position and orientation of the hock being tracked.

7 Conclusion

This project oversaw the successful design, development, and implementation of a bovine hock tracking system utilising state-of-the-art sensing technologies and algorithms. The developed system employed a PMDTec CamCube3 ToF camera to capture a 3D snapshot of a scene and successfully detect and track the position and orientation of hock like objects in three dimensional space within the captured scene.

The need for the vision system to produce predictable and repeatable results over extended periods is paramount to ensuring reliability and commercial viability. The software developed for this project has demonstrated the capabilities of the emerging technology of Time of Flight vision within dense, unstructured environments.

The software developed for the project comprises of sophisticated point cloud processing and visualization algorithms to aide in data analysis and representation. These tools helped determine the feasibility of such a system in an industrial setting. In its current form, the system was ultimately capable of performing at a rate of up to 2Hz and with successful classification of over 90% of samples taken.

Test Conditions:

During in-house and on-site testing, a sample was defined as a block of 100 sequential frames over a period of 10 seconds. Classification of the sample was deemed successful if the vision system was able to autonomously determine the correct pose and orientation of the beef hock over a majority of frames within given sample.

A sample was deemed successful regardless of if system parameters had to be tuned to cater for cattle falling in unfavourable positions in the cradle, or if the vision system became 'lost' due to external interference such as the introduction of foreign objects in the scene.

Unsuccessful classifications were primarily the result of cattle landing unfavourably within the cradle. Such occurrences produced a poor viewport and the camera was physically unable to generate suitable point cloud data.

A total of 82 samples were taken (8200 frames) over a two day period at E.C Throsby, Wittingham N.S.W. Of these samples, only 8 occurrences produced unsuccessful results.

7.1 Considerations

The implementation of the system was impeded by several issues beyond our control and scope of the project. These issues must be addressed upon commercialisation of the project to ensure the best possible results are achieved. Below are the issues which were encountered and our intended resolutions to address them during commercialisation:

- Susceptible to external interference

The introduction of external paraphernalia such as machine operators, chains and water jets within the region of interest had unpredictable effects on the tracking algorithm. These affects include getting 'lost' and/or 'trapped' within a region which was not necessarily be the hock.

An automated shackling system must be enclosed in a robot cell, preventing foreign objects from intruding the shackling area. Objects entering the field of view of the camera will be known to the system, allowing the software to ignore and/or filter the object from the scene.

Figure 13 below depicts the vision system incorrectly identifying the cattle hock with the introduction of a shackle, while Figure 14 depicts the system functioning correctly regardless of the external intrusion.

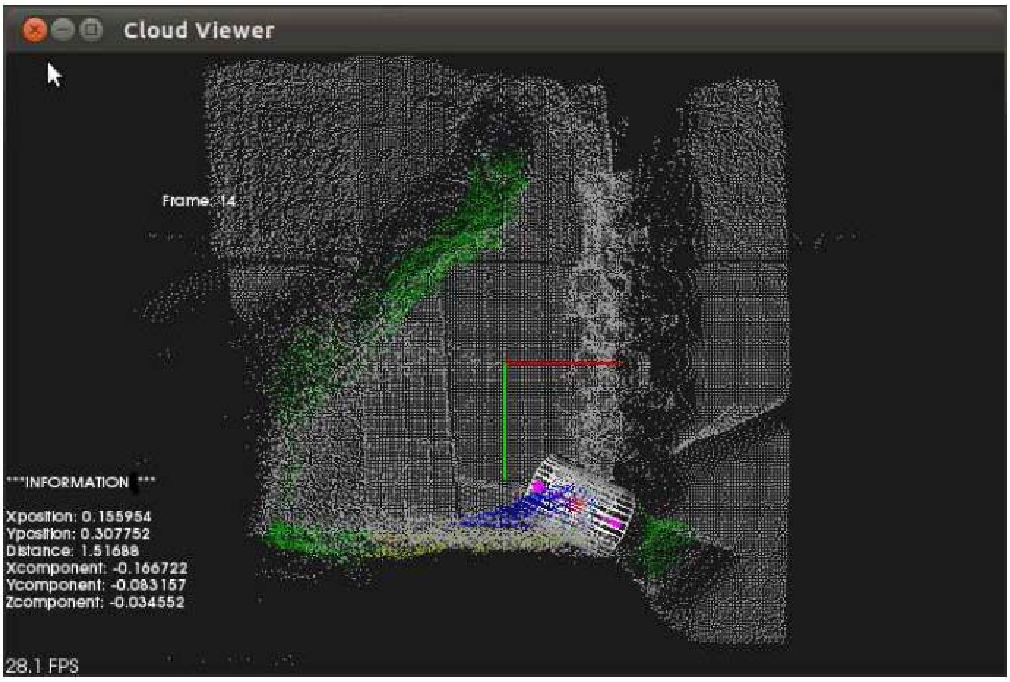

Figure 13: Vision System 'trapped' by introduction of chain on bleedroller

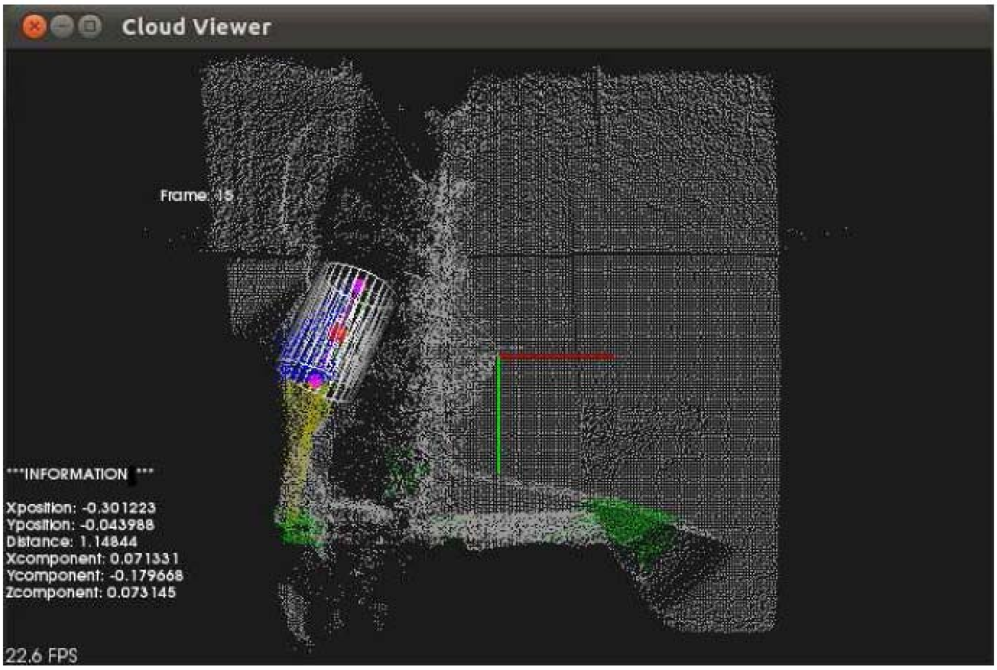

Figure 14: Vision system working correctly with introduction of chain on bleedroller

- Tight constraints of filters

Large datasets are presented for processing for each frame (40,000 data points per frame), resulting in tight constraints being enforced on the band-pass filter and KNN search algorithms to reduce the overall processing time required. These constraints have direct correlation with the

Olog(n) time complexity of the system, where more data points will, to an extent, increase the rate of successful hock detection. Results however do tend to saturate, and relaxed constraints begin to adversely affect the system.

Due to the nature of the algorithms used, the system may benefit from a multithreaded architecture. This was not implemented due to time constraints however performing multiple filtration processes simultaneously will dramatically improve system cycle time.

Increased repeatability in the means the cattle are loaded into the cradle would allow tight constraints to be imposed on the system without affecting the reliability and repeatability of the system.

- Oversaturation of point cloud via artificial light sources

ToF vision systems are not unlike traditional vision systems in that they are highly specific in the environments they may operate. Our system performed remarkably well in ideal conditions such as the slaughter floor of E.C Throsby due to the absense of abundance of natural and artificial light.

Abundant external lighting may interfere with the infrared light source the CamCube3 utilises to determine spatial awareness, resulting in noisy and inaccurate datasets.

Room lighting must be taken into account for all intended operating environments and conditions. Use of such a system in environments less than ideal, i.e. the ‘boning room’ at E.C Throsby, which has an abundance of artificial light, deteriorates the quality of the point cloud data dramatically.

Ultimately the plant’s lighting system may require alterations in order for the system to perform optimally.

7.2 Future Direction

The continual development of intelligent sensing architectures over the past decade has made such technologies viable solutions to the technical issues imposed on mainstream industry. ToF cameras have become an affordable solution in the field of 3D perception and machine vision, providing realtime 3D data at a cost of several thousand dollars.

We have demonstrated the use of ToF vision and state-of-the-art algorithms to successfully identify and track cattle hocks in real time. We feel that this technology could be readily adapted to benefit other sectors of the meat processing industry. Listed below are the several applications where the use of ToF vision may add value over traditional approaches:

- Beef Shackling

The purpose of this project was to develop a robust vision system capable of locating and tracking the rear hocks of cattle. This vision system would be integrated within works previously undertaken by Strategic Engineering Pty. Ltd. to produce a fully automated Beef Shackling System.

We believe that all relative systems (Beef Hock Location Vision System, Automated Shackle Loading System and Beef Shackling Tool) are at a phase in development that allows for a fully autonomous beef shackling system to be realised. The implementation of such a system will provide benefits such as reduced cycle time (6/min), reduced labour costs and lower OH&S risks.

- Carcass feature tracking

A vision system utilising ToF technology would be capable of identifying and tracking discernable features on carcasses in real time. This technology is an affordable solution to conventional machine vision problems and provides greater flexibility in scenarios where features or profiles must be precisely monitored in real-time.

Operations such as autonomous horn and hock removal may be automated by utilising a ToF vision system to track the both hocks and horns in real time. A robotic manipulator may then be guided into precise locations to detach these features from the carcass. Figure 15 below demonstrates the current vision software locating and tracking the front hock of a beef carcass after shackling. The carcass is swaying slightly while suspended from a chain lassoed around its hind hock. Such technology may also be utilised to robustly determine geometric properties of a carcass. Information relating to a carcass (i.e. length, width and orientation) can be easily determined and exploited while dissecting the carcass (i.e. brisket cutting).

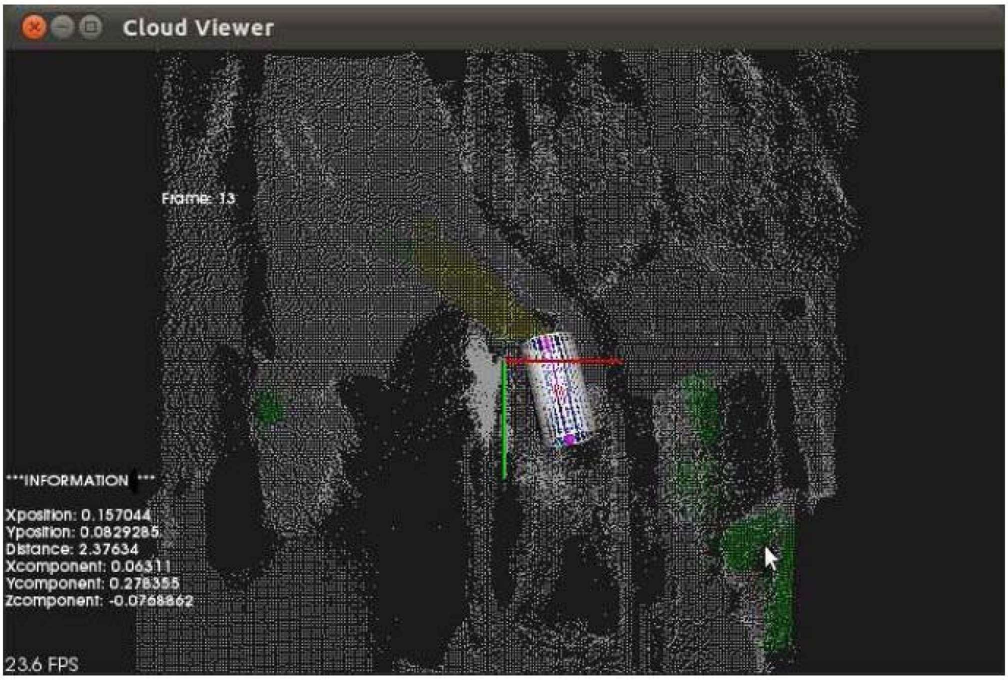

Figure 15: Front Hock Tracking from Dangling Carcase

- Primal Cuts Pick and Packing

Time of Flight vision technology can assist in the ‘pick and packing’ process of primal cuts in commercial abattoirs. A ToF vision system would be beneficial over conventional camera systems because they provide 3 dimensional information of the scene, reducing the need to integrate and collate multiple sensors. The ToF camera can directly relay position data to an industrial robot to guide the actuator in for picking.

Point cloud data will be used to classify cuts based on their features, profile and volume. The use of multiple attributes to classify primal cuts will reduce the rate of misclassification. Product profiles can then be used to determine the most optimal position and orientation the cut may be packed into cartons. It is envisioned that a ToF system will be robust enough to cater for classification of raw or vacuum sealed products.

The system may also employ a secondary ToF camera to identify clusters of primal cuts on the conveyor prior to packing and reorient/rearrange them into manageable subsets. This will significantly improve product identification and classification in dense, unstructured environments.

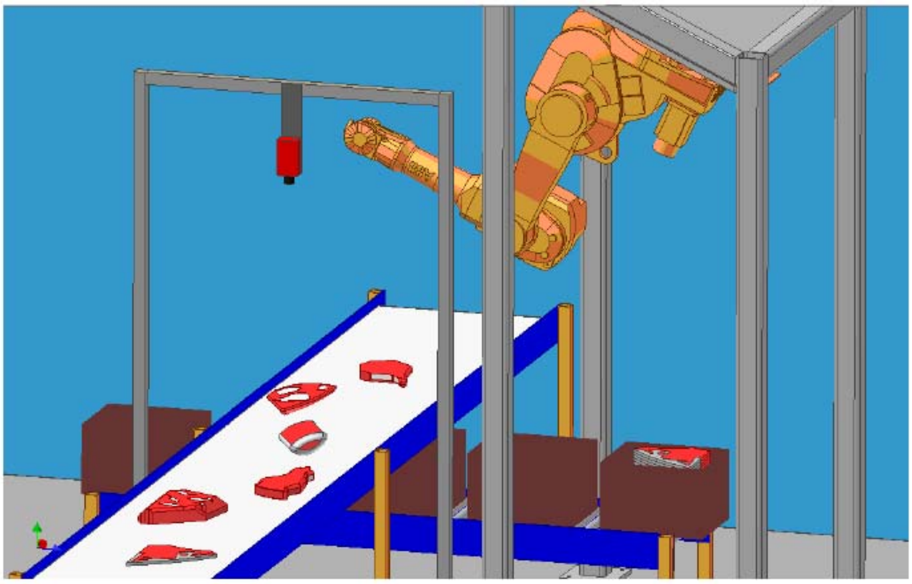

Figure 16: Proposed Robotic Pick and Place System

- Automation of Sheep Hock and Neck Tipping

Trimming of fore-hock and neck tipping is a slow and strenuous task performed in sheep processing facilities by a single labour unit. Optimising this process will add value to the sheep processing chain by reducing labour costs and lower OH&S risks. Automating this repetitious task is also beneficial as if this process is performed incorrectly it can reduce the value of the carcass.

The proposed system will scan the carcass with a ToF camera and generate a 3 dimensional map of the carcass. The system will distinguish the carcass from the gambrel and background, and then eliminate any unnecessary data points from the scene. The neck is then located and a cutting location determined. Similarly the hocks are located and cutting and gripping locations also determined.

The ToF camera may be directly interfaced with the robot and continually compute trimming positions. The ability to monitor and react to slight changes in position and orientation is advantageous as it allows the system to perform at minimal cycle times with not impact on performance.