University of Plymouth

PEARL **https://pearl.plymouth.ac.uk**

04 University of Plymouth Research Theses 01 Research Theses Main Collection

2019

ACTIVE STEREO VISION FOR MANIPULATOR ROBOT

Mohamed, Abdulla

http://hdl.handle.net/10026.1/14674

http://dx.doi.org/10.24382/437
University of Plymouth

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.

Copyright Statement

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the author's prior consent.

UNIVERSITY OF
PLYMOUTH

ACTIVE STEREO VISION FOR MANIPULATOR ROBOT

By

ABDULLA MOHAMED

A thesis submitted to University of Plymouth

in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing, Electronics and Mathematics

October 2018

Acknowledgements

I would like to start by thanking Allah for giving me this opportunity and send the person without him I cannot reach this point. I would like to thank him for his financial support and his spiritual support. This person made this work possible, thank you, my dad, Hasan Al-Qassim. I would like to thank my wife for her support and her patients all the time she spent with me during my study and give me the best gift in my whole life Hasan and Mariam. I would like to thank my family and special thanks to my youngest brother Jaafar Al-Qassim for his support in all stages of my thesis and his engagement in my work and to my sister Safa Al-Qassim for her carefulness and managing my finance.

I would like to thank Dr Phil Culverhouse, for being a consistent source of support and encouragement. His guidance and help have made my PhD program a smooth and enjoyable one.

Last but not least I would like to thank all people in the lab that share with me their time and chats and if I list their name it will fill number of pages.

Author's declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award without prior agreement of the Doctoral College Quality Sub-Committee.

Work submitted for this research degree at the University of Plymouth has not formed part of any other degree either at the University of Plymouth or at another establishment.

Publications (or public presentation of creative research outputs):

1. A. Mohamed, P. F. Culverhouse, A. Cangelosi, and C. Yang, Integrate a Visual Attention Model with a Binocular Platform for Harvesting Tomatoes", *IEEEAccess*, 2018c (In press).
2. A. Mohamed, P. F. Culverhouse, A. Cangelosi, and C. Yang, "Depth Estimation Based on Pyramid Normalized Cross-correlation Algorithm for Vergence Control", *IEEEAccess*, 2018b. DOI: 10.1109/ACCESS.2018.2877721

3. A. Mohamed, P. F. Culverhouse, A. Cangelosi, and C. Yang, "Active Stereo Platform: Online Epipolar Geometry Update," *EURASIP J. Image Video Process.*, vol. 2018:54, no. 1687–5281, p. 16, 2018a. DOI: 10.1016/j.ifacol.2017.12.030

Presentations at conferences:

4. Mohamed, P. F. Culverhouse, R. De Azambuja, A. Cangelosi, and C. Yang, "Automating Active Stereo Vision Calibration Process with Cobots," IFAC-PapersOnLine, vol. 50, no. 2, pp. 163–168, Dec. 2017. DOI: 10.1016/j.ifacol.2017.12.030

5. A. Mohamed, C. Yang, and A. Cangelosi, "Stereo Vision based Object Tracking Control for a Movable Robot Head," *IFAC-PapersOnLine*, vol. 49, no. 5, pp. 155–162, Jan. 2016. DOI: 10.1109/ACCESS.2018.2877721

Word count of main body of thesis: 50,155

Signed _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Date _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Abstract

This thesis describes a novel active stereo vision platform that was developed to provide visual guidance to fruit harvesting robots. A five degree of freedom camera platform was built in order to identify and localize tomatoes grown in greenhouses in farms. A novel cognitive model of attention with three main features map is used to control the gaze of the camera system. The three feature maps are (A) an online epipolar geometry update to produce a disparity map and affordance of grasping, (B) a vergence controller to provide a map of the depth of tomatoes at high accuracy and (C) a bottom-up saliency map to detect and identify ripe tomatoes. The properties and performance of each of these maps is studied in detail. The final system, tuned to the visual properties of the tomato, has been assessed and reveals a detection performance of 82% and operates over a range of 200+/-0.2cm. The proposed system is not limited to tomato detection, since the attention system can be adjusted to different type of crops such as apply, orange etc. The camera system is designed to integrate with any robotic manipulator arm.

.

Content

List of tables

List of Figures

ABBREVIATIONS

2D two Dimensional

3D three Dimensional

CMOS Complementary Metal-Oxide Semiconductor

DOF Degree of Freedom

DPN Deformable Parts Model

FOA Focus of Attention

ICP Iteration Close Point

IOR Inhibition of Return

MAD Maximum of Absolute Differences

MAP Maximum Posterior Probability

MGM More Global Matching

MSR Most Salient Region

NCC Normalized Cross-Correlation

OpenCV Open Computer Vision

PCL Point Cloud Library

PNCC Pyramid Normalized Cross-Correlation

PPM Perspective Projection Matrix

RGB Red Green Blue

RMS Root Mean Squared

SAD Sum of Absolute Differences

SIFT Scale-Invariant Feature Transform

SSD Sum of Squared Differences

ZDF Zero Disparity Filter

ZNCC Zero Normalized Cross-Correlation

Chapter 1

Introduction

The main objective of this PhD project is to build, validate and calibrate an active binocular vision platform that can be attached to manipulator robot for use in harvesting tomato fruit. The platform will help improve the accuracy and robustness of the depth system beyond that offered by current solutions. This will be achieved by studying and understanding the properties of binocular vision with a vergence controller and a variable baseline (the horizontal distance between the cameras) that estimate the depth based on a motor encoder. Simultaneously, the project will aim to study and explore the accompanying error in the active binocular vision system. Moreover, a relationship between the motor angle and image space will be obtained to update the epipolar geometry, which ultimately leads to a rectified image.

Robots usually require a system to update the location information of the fruits/vegetables within the camera's field of view. However, in many cases, the fruit/vegetable is not within the range of the gripper, or the fruit or vegetable may not be mature enough for picking. Therefore, an adaptive model is needed to control the robot in such situations. In this thesis, a saliency model that combines colour, edge intensity and orientation and information theory is adopted in a manner that is extensible and that emulates the behaviour of human fruit pickers. The model developed by Itti et al. (1998) was extended to suit the conditions for picking tomatoes in a greenhouse better, and then, it was integrated with a visual attention model combined with active binocular vision.

1.1 Motivation

According to a United Nations report from 2017, the global population in 2017 was 7.5 billion and that number was predicted to increase to 9.7 billion by 2050 (United Nations, 2017). Farmers will be required to increase their production, which has opened up a need for machines that can help increase the productivity of famers. This high demand motivates the development and improvement of advanced machinery in agriculture, a field in which many tasks can be systemised using robots. Harvesting, weed control, autonomous mowing, pruning, seeding, spraying and thinning, phenotyping, sorting and picking are all tasks that can be done by programmed robots (Reddy et al., 2016). Ultimately, all of the aforementioned tasks rely on computer vision. Generally, computer vision is one of the main components required for automated tasks in robotics due to the complex information provided by vision.

For example, in harvesting, computer vision is responsible for three main processes, namely fruit detection, fruit classification and computation of the 3D position of the fruit. Much research has been done on fruit detection and classification to identify the maturity of fruit before it is picked (Constante et al., 2016; Rakun et al., 2011; Xiang et al., 2014). Kapach et al. (2012) presented a review of the state of the art in fruit detection and classification. The processes discussed required a single camera to accomplish the task. In many harvesting robots, identification and picking of the fruit uses onboard cameras (Häni and Isler, 2016; Hayashu et al., 2002). This process is time-consuming since it requires cycling through the fruit one by one to determine whether the fruit is ready for picking. Using multiple cameras provides information on depth, allowing the position of the fruit to be localised in relation to the robot, as well as allowing the size of the fruit to be determined. Depth is as important as detecting the maturity of fruit for accelerating the picking process (Gao et al., 2017; Tejada et al., 2017). The increase in accuracy resulting

from depth data reduces the time required for the visual serving mechanism to reach and pick the fruit (Jiménez et al., 2000; Zhao et al., 2016).

In a related example, depth data is not only used to locate fruit; it is also used to survey crops such as cauliflower, cabbage and broccoli. Depth information is used to reconstruct 3D maps of the crops to measure their maturity or to look for damage to the product (Kise and Zhang, 2008; Underwood et al., 2016). Depth-measurement systems are also used to monitor plant growth, which includes such features as leaf size and shape, which requires a measurement resolution of millimetres (Kazmi et al., 2014). The leaf angle and direction of a plant can also be monitored using stereo vision (BISKUP et al., 2007).

A fourth industrial revolution is predicted, which is based on automating the entire manufacturing process (Drath and Horch, 2014; Pfeiffer, 2017). For example, in welding, a vision system can be used to monitor seam welding and detect defects. Some systems utilise depth information to reconstruct the welding joint at different welding passes to monitor the quality of the weld (Li et al., 2010; Wu et al., 1996). Currently, in welding robots, identification of the trajectory of the welding process is accomplished by hard coding the coordinates or by using time-consuming teaching methods (Du et al., 2018; Maeda and Nakamura, 2015). Researchers have recently implemented a new intelligent system able to track the welding seam, enabling the trajectory of the robotic arm to be determined without human intervention (Fang et al., 2011; Sicard and Levine, 1989). Detecting the welding joint requires a high-precision system to maintain the quality of the joint (Timings, 2008). In spite of the fact that industrial technology in computer vision has reached an advanced level, agricultural automation is still considered a challenge. The main reason for this challenge is the factors and variables involved in industrial manufacturing processes are manageable and can be controlled, unlike the factors that affect computer vision processing in agriculture. For example, colour detection in a controlled environment requires a simple threshold, whereas outdoors, the problem of

tracking a colour remains a challenge due to constant changes in the lighting conditions (Zhao et al., 2016b). Moreover, a slightly change in the camera position lead to different colour value.

Based on the discussion above, it is clear that computer vision lies at the core of many robotics and automation processes that require image processing and depth data for automation.

1.2 Contribution

In this work the problem of detecting a tomato and compute the 3D location using an active stereo vision platform is study in depth and providing a reliable system that work outdoor. The contribution of this thesis is divided into three categories which are:

1. The relationship between the motor angle and image space angle was established to update the epipolar geometry as the platform configuration changes accordingly in real time. It requires an offline calibration to extract the relation between both angles. The proposed methods eliminate the accumulated error in the gearbox by integrating the encoder direct with the camera rotating shaft.
2. This thesis describes the use of course-to-fine vergence controller to increase the performance and the reliability of the fixation point between the master and slave cameras. An integrated Gaussian pyramid with Normalize cross-correlation algorithm was introduced to perform the vergence control. The proposed system worked in complex outdoor environment (i.e. farms and greenhouse environment).
3. Finally, this thesis proposes a novel cognitive map based on visual attention model used in tomato fruit harvesting by providing information about the tomato probability, the position and affordability of grasping. A saliency map,

vergence controller, and 3D shape analysis are all algorithms used to assess in generating the cognitive map. The proposed cognitive map helps to minimise the computational error and avoid using a large dataset to train the system.

1.3 Papers associated with this thesis

6. A. Mohamed, P. F. Culverhouse, A. Cangelosi, and C. Yang, Integrate a Visual Attention Model with a Binocular Platform for Harvesting Tomatoes", *IEEEAccess*, 2018c (In press).
7. A. Mohamed, P. F. Culverhouse, A. Cangelosi, and C. Yang, "Depth Estimation Based on Pyramid Normalized Cross-correlation Algorithm for Vergence Control", *IEEEAccess*, 2018b (In press).
8. A. Mohamed, P. F. Culverhouse, A. Cangelosi, and C. Yang, "Active Stereo Platform: Online Epipolar Geometry Update," *EURASIP J. Image Video Process.*, vol. 2018:54, no. 1687–5281, p. 16, 2018a.
9. Mohamed, P. F. Culverhouse, R. De Azambuja, A. Cangelosi, and C. Yang, "Automating Active Stereo Vision Calibration Process with Cobots," IFAC-PapersOnLine, vol. 50, no. 2, pp. 163–168, Dec. 2017.
10. A. Mohamed, C. Yang, and A. Cangelosi, "Stereo Vision based Object Tracking Control for a Movable Robot Head," *IFAC-PapersOnLine*, vol. 49, no. 5, pp. 155–162, Jan. 2016.

Chapter 2

Literature Review

Computer vision is an interdisciplinary field that seeks to extract information from images and videos. In computer vision, images are processed to understand the content of the image, not unlike the manner in which humans or animals see. This process helps automate many tasks. For example, computer vision gives robots the ability to pick up objects. Computer vision can be used to extract 2D information from an image, such as colours, edges and shapes. Computer vision can also be used to extract 3D information from a scene using two or more 2D images. Using two images to extract 3D information of a scene is referred to as a stereo vision or binocular vision. In stereo vision, the understanding of the external geometry of a system is sought to be able to compute the 3D information of the scene or target.

This chapter presents a historical overview of the state of the art in binocular stereo vision. The literature review analyses previous research on different algorithms and processes used in orthogonal stereo vision. Then, the work on active stereo vision in the literature is described. Finally, the chapter concludes with a discussion on how to better relate the biological understanding of human vision and similar topics to computer vision.

2.1 Photogrammetry

Photogrammetry is the process of non-contact measurement using photography (Jiang et al., 2008). The concept of photogrammetry was first introduced in 1480 by Leonardo da Vinci when he explained the projection of geometry (Doyle, 1964). Based on da Vinci's work, in 1525 Albrecht Duerer built a device that was used to create perspective drawings (Gruner, 1976). The mathematical principle

of perspective was first introduced by Johan Heinrich in 1759 (Jensen, 2007). In the 1840s, a French geodesist introduced the use of photogrammetry using a daguerreotype process (Baqersad et al., 2017). After the introduction of the camera in 1830s, the relationship between projective geometry and photogrammetry was developed by R. Sturms and Guido Hauck (Doyle, 1964). The Frenchman Laussedat is known as the father of photogrammetry because he was the first person to use terrestrial photographs for topographic maps. Laussedat also did the first aerial photogrammetry using balloons in 1862 (Jiang et al., 2008). In the 1890s, Deville, a Canadian surveyor, created the first stereoscopic device, which was used to create a map from a stereo photograph. In early of the 20 century, when airplanes were created , photogrammetry become very popular and has been widely used for land surveying and military purposes by creating a 3D maps for lands (Doyle, 1964). In the 1970s, photogrammetry entered the digital world where the development and improvement of cameras and computers have helped to increase the accuracy and quality of photogrammetry. Photogrammetry is the process of computing the three-dimensional (3D) using two or more photographs. Today, this process is used in different applications from 3D scanning of a small object to using the system in self-driving car.

The principle of photogrammetry is to use more than one image where the relative position between the images is known. A triangulation is used to compute the distance between the camera and the target. For instance, the point that needs to be measured must be visible in both images, while the position between both cameras is known; a triangle can be drawn between the points in both images and the actual point. Figure 2-1 illustrates the principle of photogrammetry. The process of using images to measure objects developed significantly when the calibration process was first introduced to compute the relative position between

the cameras and the internal parameters of the cameras more precisely. A step further in photogrammetry that is referred to as stereo vision where stereo vision produces a depth map computed by matching the features in the left and right images to create a detailed map of the object or environment.

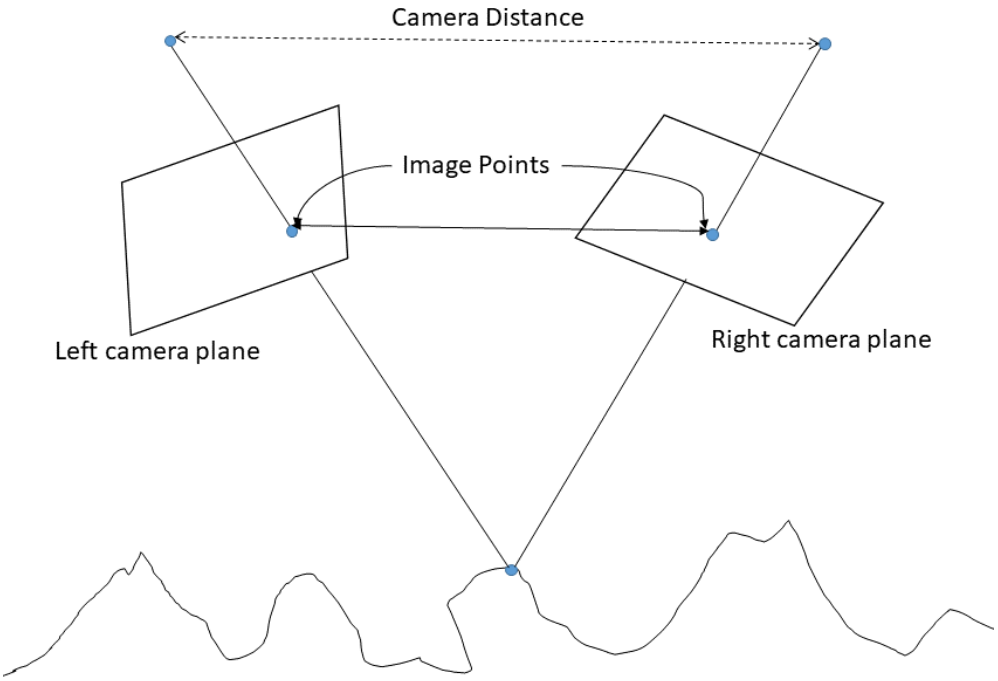

Figure 2-1: Stereo photogrammetry example.

2.2 Stereo vision feature matching

A stereo vision system requires four steps to compute the depth of a scene: (1) system calibration, (2) image rectification, (3) correspondence and (4) reconstruction (triangulation) (Bradski and Kaehler, 2008). Correspondence, which is the most complicated aspect of orthogonal stereo vision, is an area of actively ongoing research. Correspondence is the process of computing the differences between pixels belonging to the left image and the same pixels in the right image. In most of the existing studies, these differences are primarily computed along the horizontal axis. These differences are termed the disparity, which is defined as the differences in location between a feature in the left image and the same feature in the right image along a horizontal line (Fua, 1993).

Figure 2-2 shows a common method to perform matching in rectified images, which involves selecting windows in the left image, scanning along the corresponding horizontal line in the right image, and calculating the similarity using a cost function. The cost function is a mathematical expression used to compute the similarity between the left-image window and the right-image sliding window. Table 2-1 lists different cost functions that may be used to compute the differences. Figure 2-2(b) shows the similarity computed from the scan along the y-axis. Selection of the best match depends on the mathematical cost function used. For example, in normalised cross-correlation (NCC), the bigger the output, the better the match.

Table 2-1: Block-matching costs (Forsyth and Ponce, 2012)

Cost Function	Mathematical Definition
Sum of Squared Differences (SSD)	$\sum_{u.v}\left(I_l(u,v) - I_r(u+d,v)\right)^2$
Normalised Sum of Squared Differences (NSSD)	$\sum_{u,v}\left(\frac{(I_l(u,v)-\overline{I_l})}{\sqrt{\sum_{u,v}(I_l(u,v)-\overline{I_l})^2}} - \frac{(I_r(u-d,v)-\overline{I_r})}{\sqrt{\sum_{u,v}(I_r(u+d,v)-\overline{I_r})^2}}\right)$
Sum of Absolute Differences (SAD)	$\sum_{u.v}\left\lvert\left(I_l(u,v) - I_r(u+d,v)\right)^2\right\rvert$
Normalised Cross-Correlation (NCC)	$\frac{\sum_{u,v}(I_l(u,v)-\overline{\mathrm{I}_\mathrm{l}}) \times (I_r(u+d,v)-\overline{I_r})}{\sqrt{\sum_{u,v}(I_l(u,v)-\overline{\mathrm{I}_\mathrm{l}})^2 \times (I_r(u+d,v)-\overline{I_r})^2}}$
Rank	$\sum_{u.v}\left(\acute{I}_l(u,v) - \acute{I}_r(u+d,v)\right)$ $\acute{I}_k(u,v) = \sum_{m,n} I_k(m,n) < I_k(u,v)$

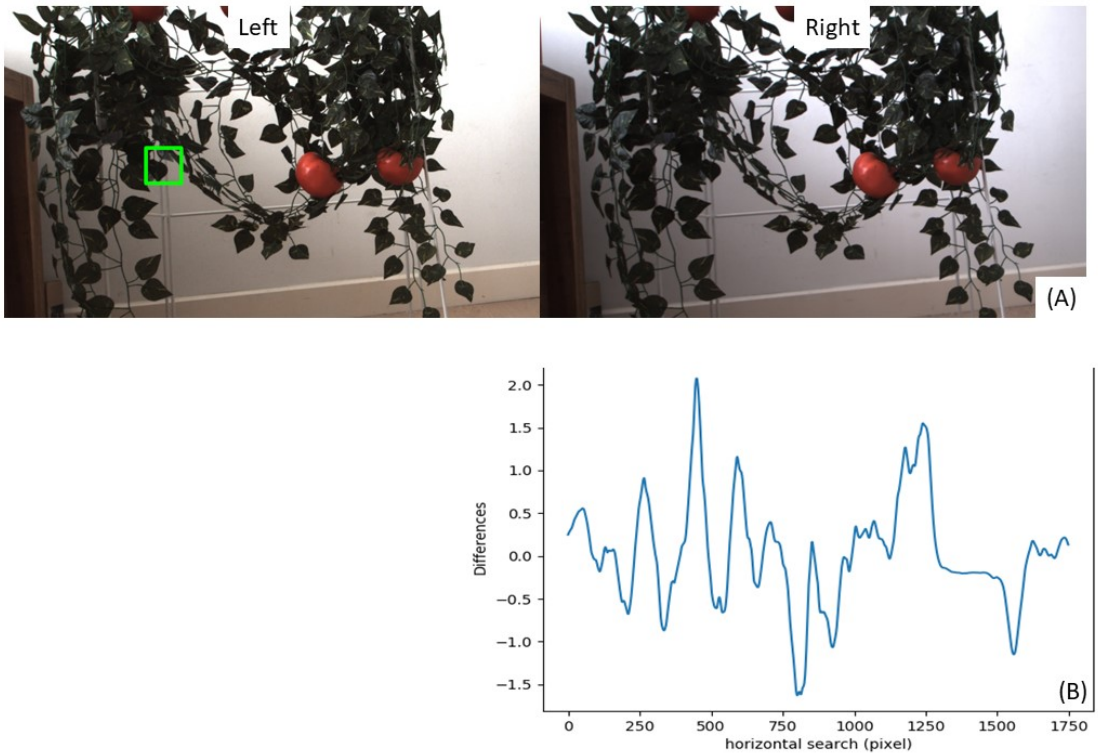

Figure 2-2: Stereo matching. (a) A window from the left image taken and scanned along the same horizontal axis in the right image. (b) The plot of differences generated from the scan process

There are many studies that compare stereo matching methods and classify the process. For example, Scharstein & Szeliski (2001) examined the stereo matching process and presented a taxonomy of stereo matching and specify two class of correspondence algorithm which are local matching and global matching . Scharstein & Szeliski (2001) also examined parallel computing by implementing the stereo matching algorithm on multiple cores. The experiment concluded that block matching methods based on the sum of squared differences (SSD) produce the finest disparity map, while for real-time applications, matching processes based on the sum of absolute differences (SAD) were found to be the fastest. SAD produce a moderate disparity map with fast computation due to the lower amount of computation required by the process.

Wang et al. (2006) and Ben-Tzvi and Xu (2010) suggested a taxonomy comprising four steps: matching cost computation, cost aggregation, disparity computation and disparity refinement. M.Z. Brown et al. (2003) and Hirschmüller and Scharstein (2007) evaluated the performance of a matching algorithm based on the quality of the disparity and the

speed of the computation. The results of these studies were similar to the results obtained by Scharstein & Szeliski (2001). However, M.Z. Brown et al. (2003) described a third class to the classification that encompassed algorithms that handle occlusion. Occlusion occurs when a point is visible in one image but is not visible in the other (see Figure 2-3).

M.Z. Brown et al. (2003) defined three methods to work with occlusions: occlusion detection, reducing the sensitivity to occlusion and modelling occlusion geometry. The detection of occlusions is achieved by searching for discontinuities in the disparity map. Hirschmüller and Scharstein (2007) took this comparison of the matching algorithms further by applying an extra effect to one camera, such as additional lighting, to test the performance of the matching process.

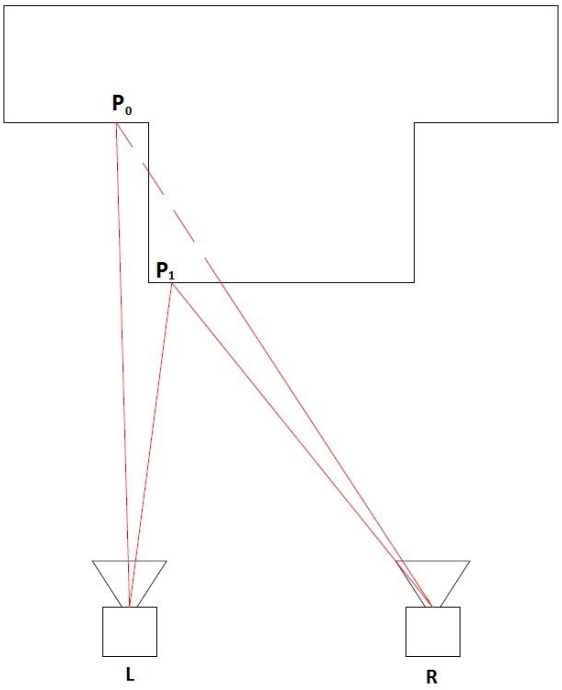

Figure 2-3: An example of occlusion where point Po is visible to the left camera only.

Scharstein & Szeliski (2001) introduced four steps to compute the disparity map: (1) matching cost computation, (2) cost aggregation, (3) disparity computation and optimisation and (4) disparity refinement. Stereo correspondence algorithms can be grouped into two classes: local matching algorithms that use a block-matching cost

function and those that directly use the intensity of pixels. Local matching is usually used in steps (1), (2) and (3) and, in some cases, only in steps (1) and (2) (Szeliski, 2009). The second category of disparity computation is a global matching algorithm that computes the disparity and occlusion by optimising cost values (Cyganek and Siebert, 2009). The result of this global matching is a smooth disparity map with a fully reconstructed scene. Steps (1), (3) and (4) are used in global matching. The reason that the second step is not used in the computation is the fourth step utilises similar functions (Szeliski, 2009).

2.2.1 Local matching

Many studies have been conducted to improve the quality of the disparity map and improved the speed of the computation in the local stereo matching process. For example, Stefano et al. (2004) proposed a single-matching phase based on detecting unreliable matching features. Wang et al. (2006) and Patil et al. (2013) performed a study on cost aggregation, which is the second step in the computation of disparity according to (Scharstein and Szeliski, 2001). Wang et al. (2006) and Patil et al. (2013) works focused on evaluating the methods used in local-matching cost aggregation. Tippetts et al. (Tippetts et al., 2011) investigated an algorithm that uses the intensity of the profile of the rows of an image to find the shape by identifying discontinuities.

Muhlmann et al. (2001) evaluated a colour space–matching algorithm. The algorithm used the SAD matching cost in the RBG colour space. The matching process generated the best match across the three layers. The algorithm works on rectified images; in which it calculates the disparity of only the left image with windows sliding along the right image. The results demonstrated that matching based on the colour space yields reliable results and that, if an occlusion exists, it can be found. However, the algorithm may face obstacles in the real world due to the effects of lighting, namely that changes in the intensity of the lighting may produce changes in colour, particularly if the cameras are not synchronised.

Another investigation was performed to improve the disparity computation by examining the effects of variable window size and shape (Veksler, 2003). The window size used in the matching process changes the quality of the disparity map depending on the texture in the scene. The fixed size of the windows can generate systematic errors. To address this, a window of variable size was used during corresponding, with the window size changing depending on the features in the image. This paper sought to characterise the systematic error generated by windows of fixed size or shape. The window cost computation utilised an integral image to calculate the cost of each window and find the average error. The experiment found that the variable-window method produced an error of 2.35% on the Tsukuba dataset. In the same dataset, the error near the discontinuities was 12.17%. This result was based on the error generated from differences between the ground truth disparity and the disparity computed by the method. It was found that changing the window size could improve the disparity map.

Feature-based matching has been implemented in many studies, particularly those focused on embedded systems and real-time applications. Features are extracted using pre-process techniques such as SIFT and Canny edge detection. Ben-Tzvi and Xu (2010) integrated a stereo vision system into a mobile robotic platform using a feature-based algorithm. Their system made use of Canny edge detection to locate the edges in the image. A maximum of absolute differences (MAD) method was used for the matching cost aggregation to produce the disparity map. To select the algorithm, an experiment was conducted to compare the performance of four methods: DPN, MAD, SAD and SSD. The investigation found the speed of MAD (26.516 s) to be faster than that of both SSD (28.766 s) and SAD (28.563 s), a result that contradicts the results of certain previous studies (Myron Z. Brown et al., 2003; Scharstein and Szeliski, 2001). Regardless, the quality of the disparity map produced by SSD and SAD is better than that produced by MAD.

In another study, Stefano et al. (2004) used a local algorithm based on area matching, called single matching phase. This algorithm comprised four steps: (1) pre-processing the input image, during which the images are normalised, (2) calculating the disparity based on SAD matching, (3) testing the disparity to determine the reliability of the matching at locations where poor texture points were rejected and (4) obtaining sub-pixel measurements to refine the disparity map. When the algorithm was run on the Tsukuba dataset, the root-mean-square error was 5%, which was the same error produced by the bidirectional matching algorithm.

2.2.2 Global matching

Global stereo matching builds on local stereo matching by using advanced refinement algorithms. Specifically, it utilises the global smoothness, for which many global stereo global matching use energy minimisation (Szeliski 2011). Global matching algorithms rely upon different types of refinement functions, such as belief propagation, (which is based on probabilistic algorithms), dynamic programming, graph cut algorithms and hierarchical algorithms (Cyganek and Siebert, 2009). The advantage of global matching over local matching is it generates a smoother disparity map, which allows for better 3D reconstruction at the cost of slower computation. On the other hand, the accuracy of depth estimations decreases (Sabater et al., 2011). This constitutes a trade-off for both of the algorithms, and the choice of one algorithm over the other depends on the requirements of the task.

The random walk algorithm is an algorithm that uses a series of steps, each of which is probabilistically determined, and which was first introduced by Pearson at the turn of the twentieth century (PEARSON, 1905). Later, in the 1970s, the random walk algorithm was first used in computer vision (Wang et al., 2017). Unger and Ilic (2014) used an algorithm based on a random walker to locate sharp details in an object. The random walker scans the image based on colour similarity and records the matches, after which it

votes for the best matching feature in the disparity based on its density. The algorithm matches both of the images, then uses cost aggregation based on the random walker and, finally, uses voting to determine the best match. The result is the random walker describes a path consisting of random steps using an image with partial labelling (Grady, 2006). This algorithm was experimentally evaluated based on the Middlebury benchmark, and the output of the analysis was compared to the ground truth of the dataset. On the Tsukuba dataset, the error was 8.68% for the overall image and 17.4% near discontinuities. On the Teddy dataset, the errors were 5.98% and 15.0%, respectively. The algorithm performs better than the previous algorithms at discontinuities and provides a shape boundary around the edges.

Much research has been conducted with the aim of improving global matching algorithms. This is achieved by combining said algorithms with other acceleration methods to reduce the computation time and produce high-quality disparity maps (Brunton et al., 2006; Facciolo, 2015; H. Hirschmüller, 2005; Hermann and Klette, 2013; Hirschmüller, 2011; Wang et al., 2007).

2.2.3 Calibration

The calibration process in a stereo vision system consists of calculating both the internal and external parameters of the system, such as pixel size, focal length and image size etc. External parameters define the orientation and position of the cameras relate to each other (i.e. the transformation from the left camera to the right camera). In an orthogonal stereo system or a fixed system, the calibration is well-defined using Zhang's calibration algorithm (Zhang, 2000). Zhang's calibration algorithm uses a checkerboard calibration pattern to detect the corners within the pattern and, then, uses these corners to compute the camera parameters using projection equations. The output of the calibration process is used in the rectification process. Rectification is used to transform the left and right images to be parallel to the epipolar plane and collinear with the baseline (Trucco and

Verri, 1998). This transformation simplifies the next step, correspondence, in which the search across the scanning line becomes 1D instead of 2D.

Vergence cues are used by humans to focus visual attention on a target, i.e. to keep both of the eyes focused on the same object. Because, in an active system, the external parameters change whenever the target moves or changes, disparity maps generated by active stereo vision depend on updating the epipolar geometry. One way to compute the new epipolar geometry is using a feature-based algorithm (e.g. SIFT algorithm) involves to compute the fundamental matrix[1] (Bjorkman and Eklundh, 2002; Krotkov et al., 1990; Luong and Faugeras, 1997; Sang De Ma, 1996). Studies that utilise the aforementioned technique focus on matching features between the left and right images upon every change in a system to compute the fundamental matrix. A drawback of this method is the potential for failures when matching features. This leads to errors in the computation of the fundamental or homography matrix.

Another approach combines image features and the motor angle to correct errors in feature matching. Thacker and Mayhew (1991) used a Kalman filter on encoder readings to predict the position of an object in subsequent frames (Thacker and Mayhew, 1991b).

Changes in the epipolar geometry occur as a result of changes in the camera angle and the position of the camera. These changes are measured by shaft encoders and are used in the control of camera position. Dankers et al. (2004) developed an online calibration process for the CeDAR head (Dankers et al., 2004). Their work was built on static system rectification (Fusiello et al., 2000) where the perspective projection matrix (PPM), obtained from the standard calibration process for the left and right cameras, was used to determine the mapping between the two images. The PPM was decomposed and a new PPM and transformation between the left and right images were used to update the

[1] Fundamental matrix is a 4×4 matrix contain the internal and external information of the stereo system.

epipolar line parallel to the baseline. Dankers et al. (2004) modified the algorithm so that the rotation angle of the left and right images was replaced by the angle of the encoder of each camera (Dankers et al., 2004). Both the motor angle and the image were captured simultaneously. However, though this process was quite fast, it required a system manufactured with extremely high accuracy, since it is very difficult to correctly place the axis of rotation of the motor with regard to the camera origin, which can lead to errors in the baseline size.

Kwon et al. (2007) designed another approach to calibrate active stereo vision. Their method treats the system as a kinematic chain that links the camera to its pan and tilt joints. By creating a kinematic chain between the joints and the camera and initialising the system, a calibration at the zero position of the system can be used to generate calibration matrices for the new positions. The motor angle is transformed to image coordinates via the transformation matrix between the image and the motor. Even though this method takes into account the position of the origin of the camera, if it is not intersected by the axis of rotation, the error accumulates during the running time because of the integration of the differences computed between the old and new angles.

Hart et al., (2008) developed a calibration algorithm using a humanoid head and controlling the stereo vergence angle. The algorithm starts with an offline process in which the essential matrix of each camera at two different orientations is computed before the properties of the system are decomposed from the fundamental matrix. Rodrigues' rotation formula is used to calculate the centres of each camera and the rotation matrix (Szeliski, 2009). These parameters are used at run time to compute a new epipolar geometry using the motor angle by inverting the process offline. An experiment was performed to compare the standard calibration algorithm to the new algorithm. The result demonstrated a mean difference of 2.38 pixels between the two methods. However, their

algorithm used the difference between the encoder readings of each orientation rather than the absolute angle. This led to the accumulation of errors with time.

Sapienza et al. (2013) investigated, in real time, the parameters of a stereo vision system during operation. Their system maps the angle of the motor encoder to the image space by calibrating the system offline. The offline calibration finds a linear equation that maps the value of the motor angle to the image space angle. The algorithm in their study calculates the homograph of each image (left and right) and decomposes the matrix to find the value of the angle in image space. This process was repeated at a range of motor angles and the results were used to determine the relationship between the motor angle and the image area. In addition, they determined the properties of a common homograph with the same features at different angles. All of this was performed during an offline process. A linear equation was generated to map the motor space to the image space using the motor angle as its input. During operation, the homographs of both the left and right images were calculated using this equation and the motor angle. From the homographs, the fundamental matrix was calculated and used to rectify the images. This equation works linearly within the range of −20° to 20°, with a transfer error of 1.03 pixels at 0° that increases to 3.28 pixels at 20°. These results were compared to the conventional calibration process. In this study, the model coefficient was fixed throughout the range of angles. This assumption requires a high-precision manufacturing process to maintain the origin of the camera as close to the rotating angle as possible.

Both Kwon et al. (2004) and Sapiens et al. (2013) have performed similar studies of transformations from the motor angle to the image angle. Kwon et al. (2004) worked with larger angles (from −45° to 45°) compared to Sapiens et al.(2013), whose work was limited to −20° to 20° for each camera. However, Kwon et al. (2004) included the tilt angle, and their study placed the origin of the camera better. Hart et al. (2008) used the angle of the motor encoder to estimate the essential matrix, which resulted in an error in

the value of the matrix; conversely, Sapiens et al. (2013) corrected the motor angle via pre-processing.

2.3 Active stereo vision

In this work, active stereo vision refers to the stereo vision system in which the geometry of the camera changes dynamically. There are many configurations of active stereo vision. For example, the following parameters can be adjusted: pan and tilt of the stereo camera or pan and tilt each camera individually, variable baseline and focal length. Active stereo vision is used to control the angle of each camera dynamically to extend the field of view, improving object tracking. It can also be used to fix the view of both cameras on a point of interest while controlling the baseline to improve distance measuring between the system and the object. Controlling the focal length helps to improve the focusing. Selecting the right parameters is crucial for the design of a stereo vision system.

There are several characteristics of an active stereo vision system that improve its performance relative to orthogonal or fixed stereo vision systems. An active stereo vision system narrows the correspondence process (small disparity) to allow it to focus on an object of interest in a scene by increasing the overlap between the left and right images (Dankers et al., 2007). The vergence angle simplifies the depth measuring process by keeping the fixation point on the object (Zhang and Tay, 2011). The fixation point is the point at which both focal axes intersect. This fixation remains on the object as either it or the system move. Another characteristic of active stereo vision is the variable baseline, which refers to the fact that the depth of the object from the system is proportional to the baseline. Increasing the intensity enhances the accuracy of depth measurement.

2.3.1 Vergence controller

Stereoscopy techniques have been widely used to measure distance by focusing two devices, typically cameras, at different positions on a single point. Here the distance

between the devices is known, forming a triangle consisting of both cameras and the point. Geometric triangulation is used to calculate the distance between a camera and the target point. Disparity calculation between two fixed cameras is an alternative. However, this method suffers from known sources of error (Dang et al., 2009; Kanatani, 2005; Xiong and Matthies, 1997). A previous study (Sabater et al., 2011) employed a block-matching algorithm to investigate the error generated by the disparity calculation.

It is clear that disparity map quality can be affected by changing the verge angle. (Krotkov et al., 1990) carried out an investigation to explore the relationship between the verge angle and the quality of the disparity map. The two cameras in the system were able to pan independently, making it a 2-DOF system. The baseline was fixed at 13 cm. The experiment involved measuring the distance from the camera to the object while varying the vergence angle. A feature-matching algorithm using edge detection was used in this work to compute the disparity. Three methods were utilised to control the performance of the matching: (1) supervised matching, (2) unsupervised matching and (3) partially supervised matching. The depth was measured based on the vergence angle and the disparity. Supervised matching produced the best performance with regard to measuring depth. The experiment yielded an error of 5% with regard to measuring depth (distance from the camera centroid to the object) at a distance of 3 m.

In (Sahabi and Basu, 1996), the difference error was studied by varying both the vergence angle and the spatial resolution. Spatial resolution was determined using a log-polar transformation that increased the resolution at the centre of the image while decreasing the resolution at the periphery. The paper presents two experiments. In the first experiment, a single camera with a focal length of 8.37018 mm took images from two different locations 112 mm apart. Objects were placed in front of the system at known distances. The key contribution of the paper was the disparity error was not affected by

the particular angle, based on the complete image. The result of this experiment was similar to that performed by Krotkov et al. (1990).

In the second experiment, spatial resolution was used in the analysis, as it mimics the function of the human eye. The experiment showed that, when both of the cameras focused on the same point, the disparity error at that point was minimised, a result that agrees with the theoretical prediction. In Figure 2-4, the minimum error occurs when both projection rays of the cameras meet at the centre of the object.

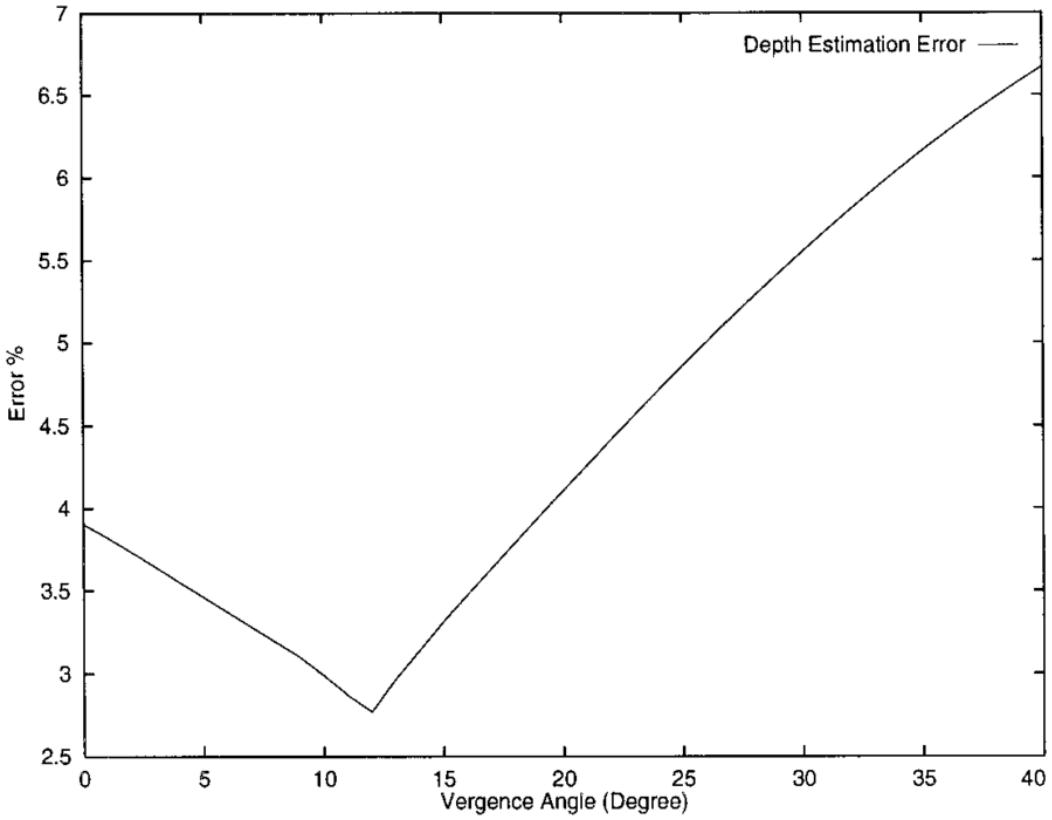

Figure 2-4: Average depth error using spatial resolution over vergence angle[2].

Another study that analysed the performance of active stereo vision parameters was performed by Das and Ahuja (1995), in which the performance of three-cue vision was examined based on (1) parallel focal length, (2) vergence angle and (3) focus[3]. The characteristics of the system were analysed and the error was divided into systematic error

[2] Reprint by permission of Elsevier
[3] Focus refers to the motorisation that allows the focal length to zoom in and zoom out.

and rounding error. This work was done to study the limitations of cue-based vision (Figure 2-5).

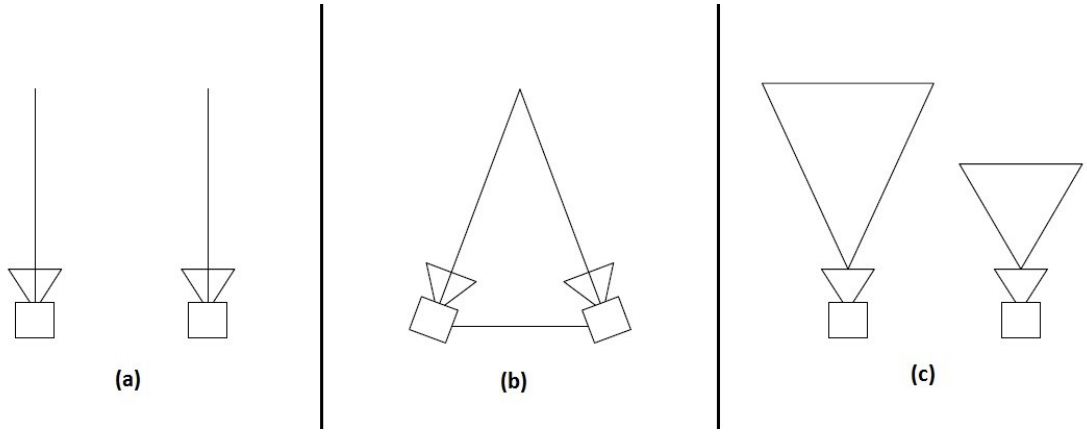

Figure 2-5: Three signals in the stereo vision system. (a)Parallel focal length setup. (b) Configuration of the vergence angle with the camera rotating on the pan axis. (c) Focus setup with the focal length controlling the field of view to increase the focus on the point.

The experiment highlighted the advantages and disadvantages of each system. Use of the focusing cue leads to obtaining fine features of the object. However, adjusting the focus of both cameras to converge at the same point is a difficult task and the paper did not describe the process used in the experiment. The vergence angle affects the quality of the disparity map, and a mixed focus cue and vergence angle lead to less error in the disparity map as a result of increasing the resolution of the object when the system zooms in. Therefore, a more accurate centroid can be computed. A system using these vision cues requires a precise controller to position both of the cameras and control the lens. The result of this work is similar to that of (Sahabi and Basu, 1996), which found that focusing on an object in the scene causes the depth error drop to a minimum.

The advantage of using the vergence angle over orthogonal in a stereo vision system is an increase in the quality of depth estimation and an increase in the overlap of the field of view between the two cameras. This leads to better accuracy with regard to resolving an object in the scene at the expense of full image reconstruction. Georgoulas and Andreadis (2010) built a system to control two cameras able to pan independently, producing disparity map in real time using a field-programmable gate array. The system

consists of two parallel processes. The first process is setting the angle of each camera and the second process is producing the disparity map. The first process uses a pyramid reduction algorithm to reduce the computation time of calculating the ZNCC cost function to determine the optimal verge angle that allows both of the cameras to focus on the same object. The process begins with a parallel focal axis and involves changing the vergence angle in 1° increments until it attains a value of 45°. The value of the ZNCC at each angle is stored and the vergence angle corresponding to the minimum value is selected. This process seems to work for an entire scene as opposed to focusing on an object, and the resolution will always be highest in the middle of the scene.

The second process Georgoulas & Andreadis (2010) utilised involved the use of a colour-SAD support window to construct a disparity map and estimate the distance to an object. The system operates at 320 FPS with a resolution of 640x480 pixels. Standard test samples were used to compare the results with other systems. It achieved 87.12% accuracy on the Tsukuba dataset (384x288, disp. level = 16) and 89.28% accuracy on the Teddy dataset (450x375, disp. level = 6). These values are obtained by comparing the output of the correspondence process to the ground truth image.

Rougeaux et al. (1993) built an active stereo vision system that used two joints to control two cameras to track moving objects against a complex background and developed the idea of a virtual horopter. The horopter is the circle that passes through the centre of both of the cameras' 3D views and the target, wherein the target has zero disparity (Cyganek and Siebert, 2009). A virtual horopter was designed to continue tracking the object even if it strayed outside the actual horopter by shifting the image to the left or right to increase or decrease the size of the horopter, respectively. A ZDF-based edge detection algorithm was used for this task. The system used by Rougeaux et al. operated at an image resolution of 230x130x8 bits and a frame rate of 30 FPS. The resolution with which the pan angle could be controlled was ±0.4°, and the system could track an object in the horopter circle

at a rate of 50° per second. The depth error calculated at a distance of 112 mm was ±6 cm. Their experiment demonstrated that this shifting approach enabled a virtual horopter to track objects. However, shifting the image can lead to error because the baseline is fixed and the image is taken at the baseline distance. When the image is shifted, the baseline increases, which was not considered in the final depth calculation.

Marefat et al. (1997) evaluated gaze stabilisation for object tracking to calculate the disparity map and improve the fixation point of the slave camera. Their system employed two cameras with independent pan and tilt control. The focus of their work was to control the vergence angle. Marefat et al. investigated the disparity of the object of interest rather than exploring the disparity of the entire scene. To calculate disparity, a Fourier phase-based approach was used to distinguish the subject matter between both images. This transformation was estimated by calculating the phase difference between the images. The transformation process was performed in binary space to increase computation speed. Their results indicate that the disparity error depends on the vergence angle, similar to the previously mentioned studies on the fixation point. Gibaldi et al. (Gibaldi et al., 2017) employed local phase differences between the left and right images to control the vergence angle. Here, a Fourier-shift theorem based on a population of oriented disparity detectors was applied in a 2D search, where disparity was computed using both horizontal and vertical search.

Marefat et al. (1997) and Georgoulas & Andreadis (2010) have done similar work with vergence control. Georgoulas & Andreadis (2010) searched for methods to determine the optimal vergence angle by starting the system with parallel focal length and then increasing the angle by a 1° visual angle. On the other hand, Marefat et al. (Marefat et al., 1997) used tracking methods to keep the focus on the fixation point. It is clear that the approach advocated by Marefat et al. has the advantage, since it can concentrate on the object during tracking even when the object is not situated between the two cameras.

However, the difference in processing speed was due to advances in hardware between the experiments. In addition, both of the systems have a weakness in using ZDF due to the virtual horopter that depends on the image shifting horizontally in pixel space.

For the most part, active stereo vision systems with vergence angle controllers are used in object tracking. Shibata and Honma (2002) designed an active stereo system with three joints: two joints to independently control the pan of the cameras and one joint to control the pan joint of the base. The system was designed to track an object in 3D space ensuring that the object was always located in the centre of the cameras. The system used the cameras as feedback sensors for the motor to track the object and keep it within the centre of the field of view. A basic threshold algorithm was used to find the centroid of the object in the scene. The main purpose of this work to focus on the controller component of the vergence angle. A Kalman filter was used to stabilise the controller due to slow image processing (33 ms per stereo pair). The system was tested by hanging a black ball and swinging it left and right. The tracking was smooth, and the camera became stable in four seconds, after which it continued to track the object. However, the system was slow and it was difficult to track an object in real time. Moreover, depth measurements obtained with it were not accurate.

In object tracking, probability knowledge can improve tracking performance. The maximum a posteriori probability (MAP) is an estimate of the unknown point that maximises the distribution (Prince, 2012). Dankers et al. (2007) used active stereo vision to track a hand using maximum a posterior probability combined with ZDF (MAP ZDF). Their platform used the ZDF to track and segment the hand gesture. The difference value between Gaussians and NCC was utilised in the algorithm to smooth images and compute the disparity map. Note that Dankers et al. (2007) used the same methods as Rougeaux et al. (1993). MAP ZDF was used to track the hand, control the camera joint to follow the hand and maintain the centre of gravity within the cameras' centres. This system operated

at an average of 25 FPS and achieved ample working space. The object tracking performance of this system was robust and accurate. However, occasionally, the system failed to track fast moving hands. Dankers et al. (2007) employed the same standard methods as Rougeaux et al. (1993), i.e. both of the studies used ZDF at the same resolution at nearly the same speed (25 and 30 FPS, respectively).

The search speed of matching feature-based correlation vergence control has been improved by implementing a coarse-to-fine pyramid algorithm (Yim and Bovik, 1994; Zhang and Tay, 2011). Zhang and Tay (2011) reported a solution based on a coarse-to-fine (pyramid image) search algorithm, where the image is transformed to log-polar space prior to constructing the pyramid. An NCC search algorithm was run on the log-polar images, and the output was used to set the vergence angle. Implementing this log-polar transformation in the search algorithm improved tracking performance in a compound sense with multiple disparities. Zhang and Tay's study (2011), the image resolution was 200x200, the window size was 84x84 and the baseline was 24 cm. Here, a three-level pyramid was used. The average accuracy of the depth measurement for different objects was 90%.

There are many more studies on object tracking using active stereo vision (for example Tanaka et al. (1994), Yu and Baozong (1996) and Tsang and Shi (2006)). The algorithms used in each study are different but the results are similar. They focus on tracking an object and keeping it within the camera centres to simplify depth measuring without implementing an extra tracking filter such as the Kalman filter.

2.3.2 Active baseline

Another variable in active stereo vision is the baseline. Baseline refers to the distance between the origins of both cameras. Klarquist and Bovik (1997) developed a system consisting of a variable baseline and two cameras with a pan joint. They introduced a

method to improve the quality of the depth map by using a variable baseline. The process starts with a short baseline to simplify the matching process, after which the baseline is increased to explore the depth resolution of the scene. The new baseline is chosen based on the result of the previous baseline, and the cycle repeats until a satisfactory resolution is attained. The experiment was run with different objects at different distances, and it was found that a minimum distance of 50 cm was required to produce excellent resolution. The result of the experiment shows that the process provided an excellent depth map with smooth reconstruction, though the authors published no specific details regarding the baseline.

Another study on variable baselines was performed by Nakabo et al. (2005), who built an active stereo vision system with a variable baseline and rotary joints that allowed each camera to pan and tilt independently. The work was used to achieve a uniform depth error by controlling the baseline and vergence angle during object tracking. The speed of the baseline travel was 4 m/s, the system was run on images with 120x120x8 bit resolution, and the image processing speed was 30 FPS. The system could be used to track an object, estimate the distance and reconstruct the object only if the object was near the platform. The experiment made use of SAD during the matching process with a window size of 5x5 pixels. The goal of the experiment was to track an object moving in a circle. Three experiments were conducted at fixed baselines of 400 mm and 800 mm, as well as with a variable baseline. The error generated by the variable baseline was 30% lower than the error generated by the fixed baselines. This result demonstrates that the potential exists for developing a system that can achieve low depth error using a variable baseline.

2.4 Detection and Classification

2.4.1 Deep learning

Deep learning has been successful in many applications, especially in computer vision where the technique of using a dataset to train a vision system to detect and identify objects in an image motivated researchers to apply deep learning techniques to agricultural processes. One of the motivations is the challenge of outdoor detection, where the lighting conditions are difficult to control (LeCun et al., 2015). Deep learning adds more depth to the model in a hierarchical way that allows the model to extract the features without pre-processing algorithms (Kamilaris and Prenafeta-Boldú, 2018).

There are 16 areas where deep learning is being used in agriculture, of which fruit classification (e.g., Steen et al., 2016; Christiansen et al., 2016; Sakai et al., 2017), weeding, fruit counting (e.g., Rahnemoonfar and Sheppard, 2017; Sa et al., 2016), and plant recognition are the most popular. Deep learning has been implemented in plant disease classification where the crop or the leaf is scanned to identify the disease (Barbedo, 2018; Rangarajan et al., 2018). One of the most popular deep learning implementations is a proposed detection system based on a Faster Region Convolutional Neural Network (R-CNN) that works in real time to detect fruits such as sweet peppers and rock melons (Sa et al., 2016). Zhang (2014) has proposed another fruit classification using a basic neural network that is the forward neural network.

Deep learning requires a huge amount of data to work perfectly, but many small farms cannot spend much on such a system, especially if required to generate data and label it, which is time consuming. Moreover, there are tradition processes that can solve the classification and detection problems without requiring a huge amount of data.

2.4.2 Visual Attention

According to Rosenblum et al. (2010), humans acquire 83% of their total sensory information visually. This amount of information needs to be processed in less than a millisecond, which requires an efficient operation.

When a human observes a scene, a single target is often the focus of attention and can be described as the most interesting part of the scene. This phenomenon is referred to as visual attention, where a complex process occurs in the human brain to identify the target. Visual attention has been studied since the time of Aristotle (Shields, 2016), including intensive studies by psychologists, and many models have been proposed to describe the mechanisms that human brains use in visual attention (Frintrop, 2006). Heinke et al. (2004) provides a review of cognitive psychological models of visual attention.

One of the best-known models that describes a possible visual attention mechanism was developed by Treisman and Gelade (1980) and is called feature integration theory (FIT). In FIT, the lower levels of the features are computed from a scene by separating the features into different map-like colour maps and brightness and edge orientation features; these are referred to as feature maps (Treue, 2003). Each feature map shows the locations of the special feature in that space (e.g., a red flower in a green field). The special features in each map are registered in parallel processes. These features are combined into a master map to locate the target with the highest value. In later studies, this master map is referred to as the saliency map, indicating the most salient features in an input scene (Bichot, 2001). Another model proposed by Cave and Wolfe (1990) extends the Treisman and Galade model, which split the colour features into separate channels (red–green–blue). Cave and Wolfe (1990) tried to answer some of the questions raised by the FIT model and conducted a guided search in the master map by fusing the features (Frintrop, 2006).

Many studies have been conducted on visual attention models that can be implemented in computer vision systems. Visual attention can be classified into two types: bottom-up and up-down. A bottom-up model is a mechanism by which the visual scene drives the saccadic attention, resulting in attention being focused on a target. Conversely, the second attention mechanism, known as up-down attention, is a conscious mechanism (Connor et al., 2004). For example, if a person is hungry his/her attention will be focused on food.

One of the first visual attention models for computer vision was proposed by Koch and Ullman (Koch and Ullman, 1985) based on the work of Treisman and Galade (Treisman and Gelade, 1980) and included an algorithm to process the input feature maps in parallel to form a saliency map. The most salient region in the saliency map was selected dynamically using a winner-take-all (WTA) network; then the network selected the next most salient region. Driscoll et al. (Driscoll et al., 1998) proposed a visual attention model referred to as the feature gate model. Milanese (1993) introduced conspicuity maps for visual attention models, which were later implemented by Itti el at. (Itti et al., 1998). Frintrop et al. (Frintrop et al., 2010) presents a collection of previous studies done on visual attention.

Itti el at. (1998) describe a visual attention model based on a saliency map. Figure 2-6 shows the general architecture of the proposed model. The model is based on the work of Koch and Ullman (Koch and Ullman, 1985). The algorithm adopted in Figure 2-6 starts by computing the feature maps from the input image using a linear filter. In the same study, Itti el al. used three initial feature maps: the intensity map, the RGB colour opponent maps (red–green and blue–yellow) and the orientation feature maps $(0°, 45°, 90°, 135°)$ (Figure 2-6). A centre-surrounding operation is introduced to enhance the small features in the centre and the surrounding regions (e.g. the edges) (Westheimer, 2004).

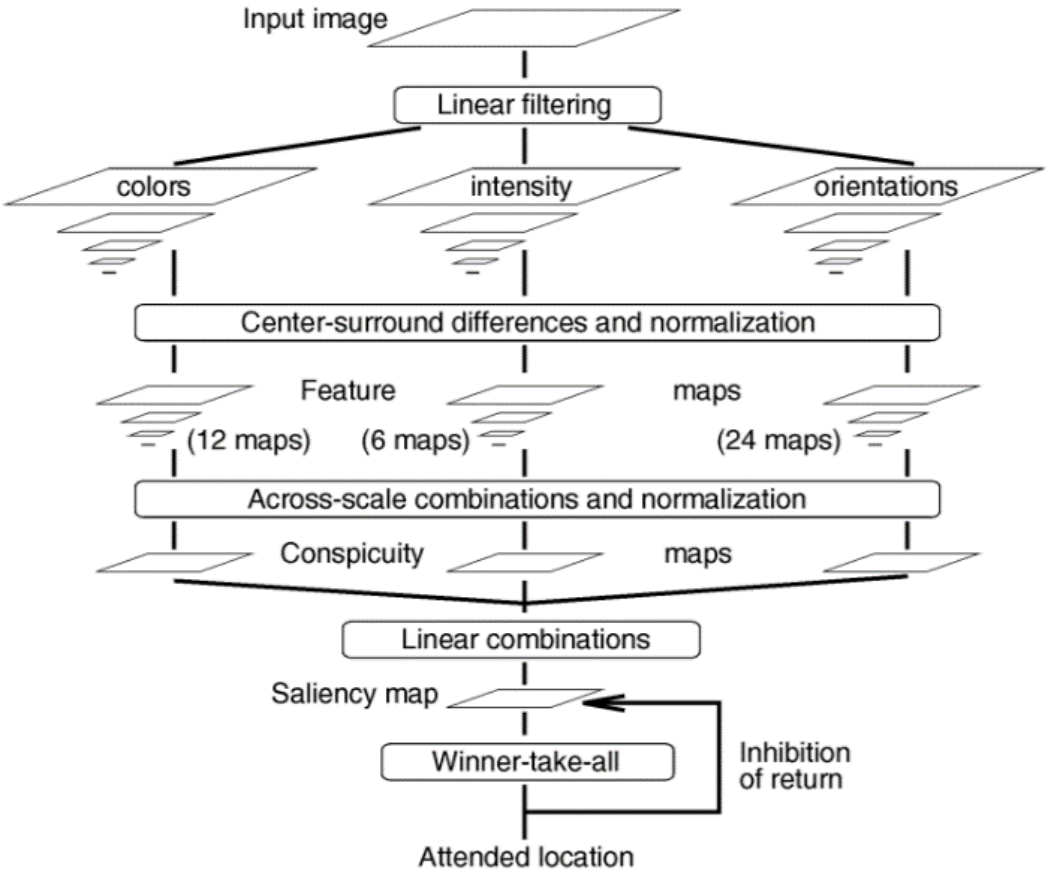

Figure 2-6: Itti et al., (1998) Saliency map architecture [Taken with permission from copyright © 2011 IEEE]

The initial feature maps are processed by the centre-surrounding operation, where the output consists of six feature maps from each initial feature map. In a study by Itti el at. (1998), the centre-surrounding operation was performed by computing the difference between the pyramid layers, and the output of the operation was normalised. The output feature maps from the centre-surrounding operation consist of multiple scale maps; therefore, these maps combine together in an across-scale combination operation, where the final feature has the same size as the input image size and the output maps are referred to as conspicuity maps. This operation is done by resize the feature maps to actual input image size, then subtract them element by element to create the final conspicuity maps. Finally, the saliency map is formed by combining the conspicuity maps. Itti el at. (1998) used WTA for the operation of the inhibition of return. Inhibition of return is referring to the process of selecting the salient region from the highest to the lowest features. The

feature maps generated from the input image are colour maps used to compute the colour opponents red–green and blue–yellow, the intensity map, and the local orientation information using a Gabor filter and a Gaussian pyramid. This model has good performance for detecting colours and shapes when these features are already included in the algorithm. However, it cannot detect features that are not included in the model, such as contour completion or tracking objects in motion.

There are multiple published studies on the integration of visual attention with an active binocular vision platform. Choi et al. (2004) used an attention model to control a vergence platform, using a selective attention model to improve the attention process by ignoring unwanted output from the saliency map and focussing on the desired feature using fuzzy adaptive resonant theory (Fuzzy ART). The visual attention in this study is based on a bottom-up model that computes four feature maps (the colour opponents red–green and blue–yellow, the intensity, and the edge features). These feature maps are processed to determine the centre-surrounding differences and normalised before computing the saliency map. The system first finds the most salient region in the saliency map and then passes it to Fuzzy ART to determine whether the target is good using the feature maps of the centre-surrounding differences. This process occurs in parallel for the left and right cameras; the outputs of both saliency maps are compared to determine the dominant salient region for the master camera (e.g., there is no fixed slave camera in the system). The verge of the system is determined by comparing the saliency region of the master camera to that of the slave camera. This could lead to an incorrect verge on the target if the same object occurs twice in the scene.

Several studies have been performed on visual attention using different approaches. In a study by Aragon-Camarasa et al. (2010), an active stereo vision system with two pan-tilt cameras was used. The proposed visual attention model used a scale-invariant feature

transform (SIFT) algorithm to determine the salient feature in the scene. These features were used for the gaze, vergence controller, and object recognition. The features were computed in both images to compute the 2D disparity in the x- and y-axes with respect to the image. The vergence controller depended on the global SIFT features between both images in the verge process; this is referred to as nonselective vergence. The proposed vergence controller using SIFT resulted in a maximum error generated by the system of ±6.5 pixels. The visual attention model was based on a pre-attention model. The pre-attention model was trained using the salient features of different objects at various orientations and locations. The model was trained to recognize targets in search mode. The drawback of this type of system is that the entire process depends on the SIFT feature of the target for the scene to verge correctly.

Improvements and extra testing on the above algorithm were performed by the same lab (Khan et al., 2016). An extra operation was added to memorize the target visited by the system to avoid revisiting targets. The architecture of the model was redesigned to be based on sensor fusion effects and to work with a robot operating system (ROS) (Quigley et al., 2009). In this study, three feature matching algorithms were evaluated in the model, i.e., SIFT, SURF, and KAZE. The proposed algorithms were also evaluated on two platforms. The experimental results indicated that the performance of the system depends on the feature extraction algorithm, where KAZE produced the best recognition rate. Moreover, their results demonstrated that the position of the object being tracked using extracted features was affected by the position of the object relative to the system.

In studies by both Aragon-Camarasa et al. (2010) and Khan et al. (2016), a visual attention model integrated with an active stereo vision system was proposed, in which the cameras moved using motors. The proposed model is based on feature extraction, which is used to control the verge angle and track the object with the most salient features. The model

was also integrated with object recognition using the same features computed in the first step. This type of model performs better in environments with many features, such as indoor environments. In our model, we use a visual attention model to identify and locate tomatoes, which are texture-less and cause the feature extraction algorithm to fail even if the feature is extracted from the surroundings. Therefore, another algorithm was used to detect the tomatoes; this is referred to as the saliency map.

2.5 State of art conclusion

There is a considerable amount of literature on stereo vision correspondence methods. The literature seemed to be divided into studies on local matching and global matching, with the focus primarily on reducing computational time and producing high-quality disparity maps that improve 3D measurements. As demonstrated by these studies, advances in hardware and lower hardware costs have resulted in many improvements, allowing the research on stereo vision correspondence to focus increasingly on improving quality instead of focusing on decreasing computational speed. New algorithms have been introduced that improve the accuracy of the depth map, such as the random walker algorithm, MGM and RGB-SAD. These algorithms contribute to the production of high-quality disparity maps and analysis.

2.5.1 Online epipolar geometry

The literature on active stereo vision is divided into three categories: the study of vergence angle, object tracking and variables baselines. Though these studies make use of the same principles of active stereo vision, the targets of their work are different. In papers that study on vergence angle, the focus is on reconstructing the target object by altering the vergence angle of the cameras to achieve the most significant overlap in their fields of view where dese disparity map generated with high-quality details. Some papers investigate the differences between fixed stereo vision systems and active stereo vision

systems concerning vergence angle (Das & Ahuja 1995), with the results of these studies showing that there is a disparity between some of these systems due to the low computational power of hardware power before 2000.

There are no complete methods that can be used in active stereo vision platform to generate fast and accurate calibration data after the change in the position of the camera. Most of the continues epipolar geometry work has a weak point such as the limited in verge angle (e.g. ± 25 degrees), error accumulative between encoder reading and image's angle, compute wrong essential matrix due to the fewer number of features in the image and compute the internal and external parameters of the system up to scaler which cannot be used to in measurement. The most apparent thing in published works is that the disparity map computation to evaluate the result of the calibration methods has not been used and tested with the online calibration algorithms. In other words, the disparity map has not been generated using the proposed algorithms.

2.5.2 Vergence Controller

Other studies falling into the second category focus on object tracking and gaze control, specifically the challenge of keeping both cameras focused on the same point. Different approaches, such as the master and slave system which have been introduced above; uses the correspondence algorithms to match the images from the two cameras, or the ZDF algorithm, which utilises the horopter. ZDF introduced the so-called virtual horopter, which enhances object-tracking performance. The methods used in these studies are limited in terms of their ability to calculate distance and improve the depth map. The proposed methods in the literature on vergence vision were focused on controlling the slave camera to focus on the fixed point as well as stability of the controller. Ultimately, the mentioned algorithms are limited to focusing on stabilising the controller and features matching which in sequence lead to the use of low image resolution that will ultimately effect negatively on the depth estimation. In fact, depth estimation plays a major role in

addition to the two factors mentioned above to improve the vergence vision. Moreover, there is no mathematical analysis for vergence controller that study how the parameters in the system contribute to the error in depth measurement. Most of the published works are focusing on testing and evaluating the vergence vision inside laboratories but not yet proposing useful application that can be use in real life. Therefore, this thesis will utilise the vergence vision system to allocate the relative position of tomato fruit in green house.

2.5.3 Detection System

Finally, the literature review chapter introduced the visual attention model. The literature on visual attention is focus on understanding human vision behaviour and implementation into computer vision and robotics. There are different approaches to implement visual attention, such as saliency map and feature matching algorithm to track the salient features in the scene. Both studies focus on the performance of the detection of notable feature in the scene where the feature matching studies required an intense feature to detect the salience feature where it is not reliable to use in the harvesting process. Moreover, the research of saliency map shows that there are so-called the return of inherits (ROI) which processes the salient features from the highest to lowest in term of salient strength. There are two approaches introduced in the published work which (I) find the highest value in the saliency map then cover it with zeros and (II) find the highest value then apply a flood fill algorithm to cover region belong to that point. In both approaches, the system needs to be tweak for different application; furthermore, the information of the target such as size and shape cannot be computed. In this work, the focus on visual attention is to create an application that able to detecting tomato in a greenhouse by integrating the information theory and watershed algorithm into the proposed algorithm. Unlike the other works, the focus in this thesis is to design a useful application that can be used outside the lab.

Chapter 3

System Design and Overview

This chapter provides an analysis to the previous chapter where the gap in the state of the art is addressed. Later the chapter, an overview of the stereo vision platform will be presented, as well as its specifications and the mechanical design and system controller. This chapter concludes with an overview of the experiments reported in subsequent chapters.

3.1 The gap in state of the art

3.1.1 Online epipolar geometry update

In stereo vision, the internal and external parameters are needed in order to be able to estimate the depth of an object. The process of computing the external and internal parameters of the system is referred to as the calibration process (Szeliski, 2009). Intense research has been conducted in order to estimate the parameters of the system or update the parameters of the system when there is any modification of the parameters in the fixed stereo vision system. For example, an increase in the temperature leads to extension of the overall dimensions of the system. In active stereo vision, the research on updating the external parameters can be divided into three categories: (1) feature-based matching (e.g., SIFT), (2) integrating feature matching with encoder reading, and (3) using the motor encoder to update the parameters.

In the first approach, the parameters are updated to scale as a result of using image space only, and at the same time the process is computationally expensive and the mismatched features lead to incorrect parameter estimations that affect the output (Bjorkman and Eklundh, 2002; Krotkov et al., 1990; Luong and Faugeras, 1997; Sang De Ma, 1996). The

second category overcomes the problem of estimating the parameters in a scale dimension but is still prone to feature mismatching and is computationally expensive (Cyganek and Siebert, 2009). The third category involves direct use of the encoder reading to update the geometry. This approach requires a precise manufacturing platform. Many studies have been conducted to solve this problem; for example, Hart et al. (2008) have computed the differences between the old position and the new position and then updated the differences which this approach leads to accumulate of the error in the encoder. One different approach was to directly integrate the motor encoder into the parameters, but this approach requires precise manufacturing (Dankers and Zelinsky, 2004; Kwon et al., 2007). Sapienza et al. (2013) have proposed using an offline calibration to determine the relationship between the encoder and the image space using a homograph. However, this approach has a limited small angle range of +/- 20 degrees, and no further results, such as rectification image output, have been presented.

Most of the proposed approaches have not provided sufficient results on the online geometry update or shown further process after the rectification. In this study, the problem of updating the parameters of the external parameter is addressed by proposing an offline calibration to estimate a relationship between the encoder angle and the image angle. This approach helps to improve the rectification process while generating data that helps to identify the accuracy and reliability of the platform. Furthermore, the algorithm is used to compute the disparity map and used to identified the 3D shape of tomato fruit.

3.1.2 Vergence controller

A vergence controller is used to control the slave camera to verge correctly on the fixation point of the master camera. There are three classes of vergence controller algorithms: (1) zero disparity, (2) feature-based matching, and (3) template matching.

The first class of vergence controller algorithm computes the disparity between the slave and master images, and then moves the motor using the disparity value. This method is computationally expensive, and the major issue is the update of the external parameters (Krotkov et al., 1990). The second class provides a good result but is not reliable especially because this algorithm requires a textured background and target to manage to verge on the target, which means that this method will not work in a harvesting process due to the texture less nature of the tomato and some fruits. The final class of the algorithm is the fastest algorithm and requires less computational power. This class has been studied using different approaches to control the vergence angle; for example, cross correlation, sum of absolute differences based on edge detection and cross-correlation in log polar space (Dankers et al., 2007; Georgoulas and Andreadis, 2010; Rougeaux et al., 1993). These algorithms have been tested and evaluated indoors (i.e., inside a lab) where the lighting conditions are controllable. In this thesis, a vergence controller algorithm is developed and implemented on an active stereo vision platform to detect tomatoes. The algorithm is developed to work outdoors and overcomes the issue of the uniform background (i.e., the uniform leaves and sky), but the proposed algorithm suffers from the sunlight effect, which leads to incorrect vergence on the target. Moreover, in this study the depth estimation based on the vergence controller is measured and evaluated in order to provide an accurate depth estimation. There are few studies on vergence controllers that evaluate the depth estimation, which will be compared with the present research.

3.1.3 Visual attention model for fruit detection

The visual attention model has been studied and implemented in general application where the behaviour has been studied to detect the most salient feature in the scene. Different applications of the visual attention model have been proposed, such as detecting the motion of the object, detecting the most salient object in the scene, or detecting the strongest colour in the scene. Moreover, visual attention models have been developed to

search for objects based on a few input parameters or using the system to memorise the targets (Khan et al., 2016). Few researchers have implemented the visual attention model in an active stereo vision to mimic the human cognitive system in tracking the most salient object in the environment (Aragon-Camarasa et al., 2010; Choi et al., 2004). Other studies have used the visual attention model to control the vergence on the targets (Khan et al., 2016).

The majority of the published works on visual attention have taken different approaches to focus of attention FOA. FOA is the process of isolating the most salient feature in the scene, then moving on to the second, and so on. Most studies have used a fixed circle or rectangle to cover the salient feature (e.g., Itti et al., (1998), while others have used a flood algorithm (Frintrop, 2006). Both approaches require parameter tuning.

It is apparent that the majority of visual attention applications have been general or have focused on developing the model to be more accurate. In this study, the visual attention model is developed to be implemented in a specific application that considers a challenging problem in computer vision, which is fruit detection outdoors. The model is developed to detect tomatoes and identify the size of the tomato by implementing a watershed algorithm within the FOA process. The algorithm is developed to generate a cognitive map that integrates with a robotic arm to accelerate the picking process.

3.2 Active stereo vision platform

To evaluate the active stereo vision system in detail and overcome the issue addressed above, a research platform was designed and established, allowing for a variety of active stereo vision configurations. Two versions of the platform were constructed. The first version of the stereo vision platform has three degrees of freedom (DOFs), that is, two DOFs that control the angle on each camera (pan angle) and one DOF that is associated with the distance between the cameras (baseline). Meanwhile, the second version has five

degrees of freedom, that is, four DOFs to control the pan and tilt of each camera independently and one DOF to control the baseline distance.

3.2.1 Platform configuration

Both versions of the platform have common materials, overall dimension and cameras. The platform was constructed using 3D-printed parts[4], with aluminium-extruded tube used as the rail. Two carriers were used to carry the two cameras and their motors; the horizontal position of these carriers is controlled by stepper motors (Figure 3-1). The parts used in building the platform are shown in Table 3-2, as well as the suppliers and the price. Integrated stepper motors with encoders are used to control the baseline distance. Stepper motors are frequently used in most automated machines, such as computer numerical controllers and many manipulator arms, because these motors divide the rotations into small steps that aid for a precise movement. However, the drawback of these motors with very small steps is the decline in torque, thereby leading to missing steps. Hence, an encoder was integrated with the stepper to address the missing steps.

The carriers on the baseline have a linear velocity of 3 m/s, in comparison with that of Nakabo et al. (2005) where the camera speeds were 4 m/s. The baseline was integrated with an endstop and a linear encoder. The endstop is used to identify the starting distance between the carriers once the rig is turned on. Meanwhile, the linear encoder consists of two parts: (1) a magnet incremental quadrature encoder with a resolution of 12 bits and (2) a magnetic strip which is fixed on the rail with a resolution of ±40 µm/m. Both parts are used to determine the position of the carriers with respect to one another, where the baseline is the distance between the left and right origins of the cameras. The more precise

[4] The materials are Polylactic Acid (PLA) printed using a Prusa i3 mk2 3D printer with a resolution of 0.2 mm.

the encoder used in this platform is, the more accurate the measurement estimation will be.

In the camera, the origin is located at the camera aperture. All physical measurements are related to this origin. The cameras are attached to the motor using a 3D-printed bracket (Figure 3-3). The design of the bracket was chosen carefully to ensure that the rotating axis of the motors intersects with the origin of the cameras. Both cameras' origins should intersect with each other on the x-axis (Figure 3-1). This option is important to minimise the offset between the axes, thereby simplifying the kinematic chain of the platform. Table 3-1 describes the overall dimensions of the platform.

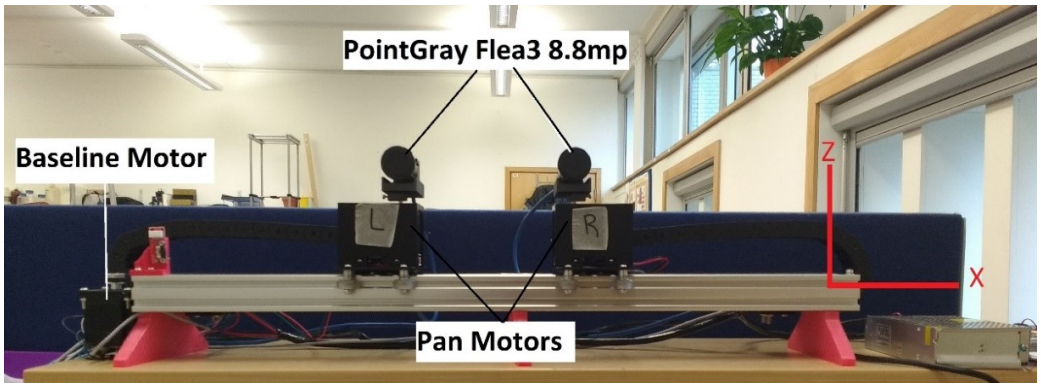

Figure 3-1: Version 1 of the stereo vision platform.

Table 3-1: Stereo vision rig specification and overall measurement.

Variable	Unit	Value
Overall dimensions		
Width	mm	860
Depth	mm	180
Height	mm	247
Working range		
Min. baseline	mm	46.7
Max. baseline	mm	500
Min. cameras' pan angle	degree	−90
Max. cameras' pan angle	degree	90
Min. cameras' tilt angle	degree	−48
Max. cameras' tilt angle	degree	70

Table 3-2: Part list, price and suppliers.

Part No.	Description	Qty	Price	Total Prices	Supplier
Version 1					
1	Stepper Motor NEMA 17	2	£12.50	£25.00	Ooznest.co.uk
2	Stepper Motor NEMA 23	1	£23.00	£23.00	Ooznest.co.uk
3	Ustepper Controller Board	3	£43.20	£129.60	Ooznest.co.uk
Version 2					
	Dynamixel XL-420	2	£50.00	£100.00	Amazon.co.uk
	Dynamixel AX-12A	2	£45.00	£90.00	Amazon.co.uk
	USB2Dynamixel	2	£38.00	£76.00	Amazon.co.uk
Commonly used in versions 1 and 2					
4	Endstop	1	£1.40	£1.40	Amazon.co.uk
5	Power Supply Transformer AC 110–240 V and DC 12 V/24 V	1	£12.99	£12.99	ebay.co.uk
6	C-BEAM LINEAR RAIL – 1000 MM	1	£26.20	£26.20	Ooznest.co.uk
7	V-Slot Gantry Plate Kit	2	£35.50	£71.00	Ooznest.co.uk
8	Smooth Idler Pulley	1	£4.80	£4.80	Ooznest.co.uk
9	Tee Nuts M5	1	£5.26	£5.26	Ooznest.co.uk
10	Idler Pulley Plate	1	£6.50	£6.50	Ooznest.co.uk
11	NEMA 23 Motor Mounting Plate	1	£7.50	£7.50	Ooznest.co.uk
12	GT3 Pulley and Timing Belt Kits [2 m]	1	£25.00	£25.00	Ooznest.co.uk
13	Magnet strip 400 mm in length	2	£15.91	£31.82	rls.co.uk
14	Linear encoder AS5304 AB	2	£12.34	£24.68	mouser.co.uk
15	3D printing filament and service	1	£25.00	£25.00	University of Plymouth
Total				£ 959.75	

Version 1: three DOFs

Encoders were integrated in the platform to increase the accuracy of the stepper motor measurements. They were assembled with the shaft of the stepper motors. The encoder used in controlling the rotation angle is an absolute magnet encoder with a resolution of 0.08° (12 bits), whereas that in previous studies was $\pm 0.1°$ (Gibaldi et al., 2015) and $\pm 0.01°$ (Dankers and Zelinsky, 2004). The resolution of the encoder is important for the final output depth measurement in which the accuracy and the repeatability of the platform depend on. The value of the encoder was set to 0 to maintain the focal axis of the cameras perpendicular with the baseline.

The motors, which are attached to the cameras, were set with an angular velocity of 100 rpm. This angular speed was selected to minimise motion blurring to 0.43 mm at a working distance of 2 m. The cameras used in this project have a shutter speed of 0.021×10^{-3} s, which is sufficient to process the images. The stepper motors are PID controlled, in which the system was set to minimise the damping to avoid clash between the cameras.

Version 2: five DOFs

Version 2 was constructed to incorporate tilting in the platform. The initial idea was to add a common tilt for the platform, but it was not feasible owing to the weight and difficulty to install a platform with a manipulator arm. This version has an independent tilt for each camera. The stepper motors that control the pan were replaced by the equivalent digital servomotors Dynamixel XL. The new digital motor has a higher torque. Moreover, the controller and performance were better than those of the stepper motor. Version 2 of the platform (Figure 3-2) is different from version 1 owing to the following:

1. The tilt joints of each camera use Dynamixel AX-12A, with an encoder resolution of 8 bits (0.29°) and a motor speed of 60 rpm.
2. The range of motion in the tilt angle is from $-45°$ to 75°).

3. The pan motors were replaced with a Dynamixel XL motor with an encoder resolution of 12 bits, but the angular speed of the motor remains constant at 100 rpm.

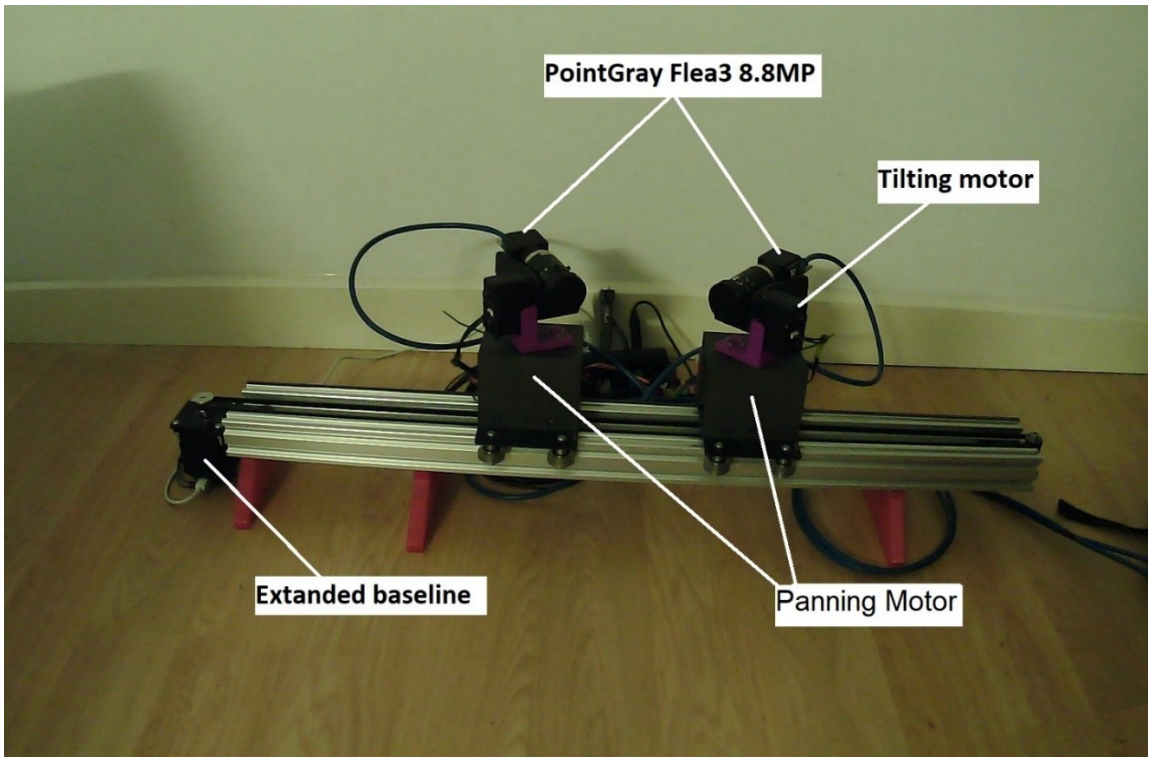

Figure 3-2: Version 2 of the platform.

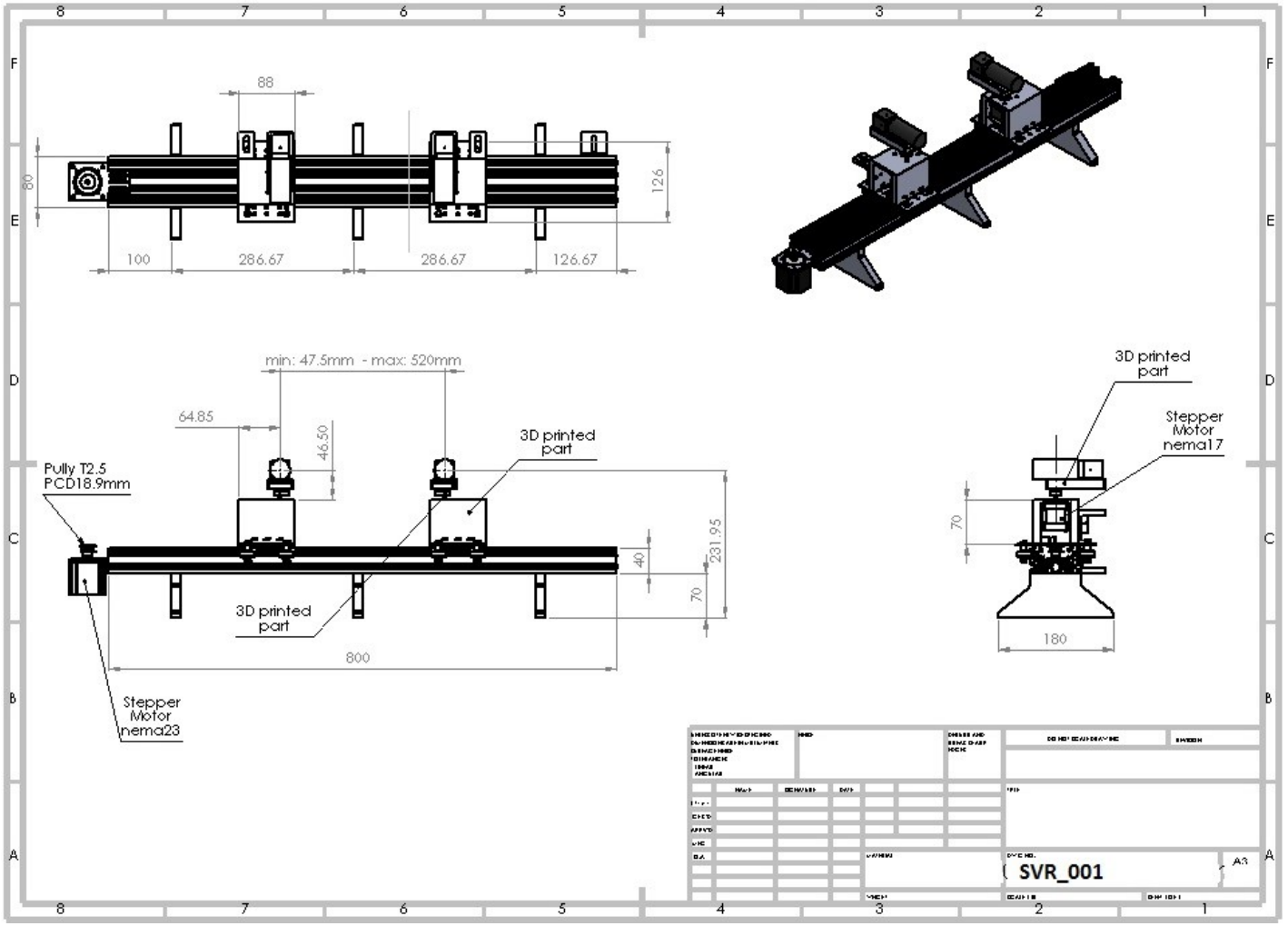

Figure 3-3: Overall dimension drawing of the stereo vision rig (version 1).

3.2.2 Camera specification

The cameras used in the platform are colour Point Grey Flea 3 with 8.8 MP and a frame rate of 21 fps. The camera has different mode that control the resolution of the image and the frame rate. The sensor is Sony IMX121 with a resolution of 4096 × 2160 and 12-bit ADC. The pixel size is 1.55 µm, and the cameras have a high dynamic range (HDR) of 10.71 bits (see Appendix A for further details). Note that the HDR of the cameras is not optimal when used outdoor but is suitable for indoor use because the plan was to use the platform indoor. However, in a future work, we should consider replacing the cameras.

The camera lenses were selected to have a depth of view within 1–2 m. The focal length is 8.5 mm, providing a diagonal field of view of 54.0°. This selection includes the overlap in the stereo vision between the two cameras at a maximum baseline range of 500 mm that leads to 77% overlap at 2 m. Figure 3-4 illustrates the criteria used for the selection of the camera lenses, where the maximum baseline in the platform is 500 mm, the working distance from the platform is 2 m and the specified CCD sensor size is 10.1 mm. The figure shows the guide for the camera lens selection considering the parameters and variables. The lens was selected to provide a maximum overlap (grey line) with a small angle of view. The angle of view was selected to be small to increase the pixel density (Uliano, 1992). The horizontal pixel density of the system for each camera is 75.85 pixels per degree, in comparison with the human eyes 60.0 pixels per degree (Uliano, 1992).

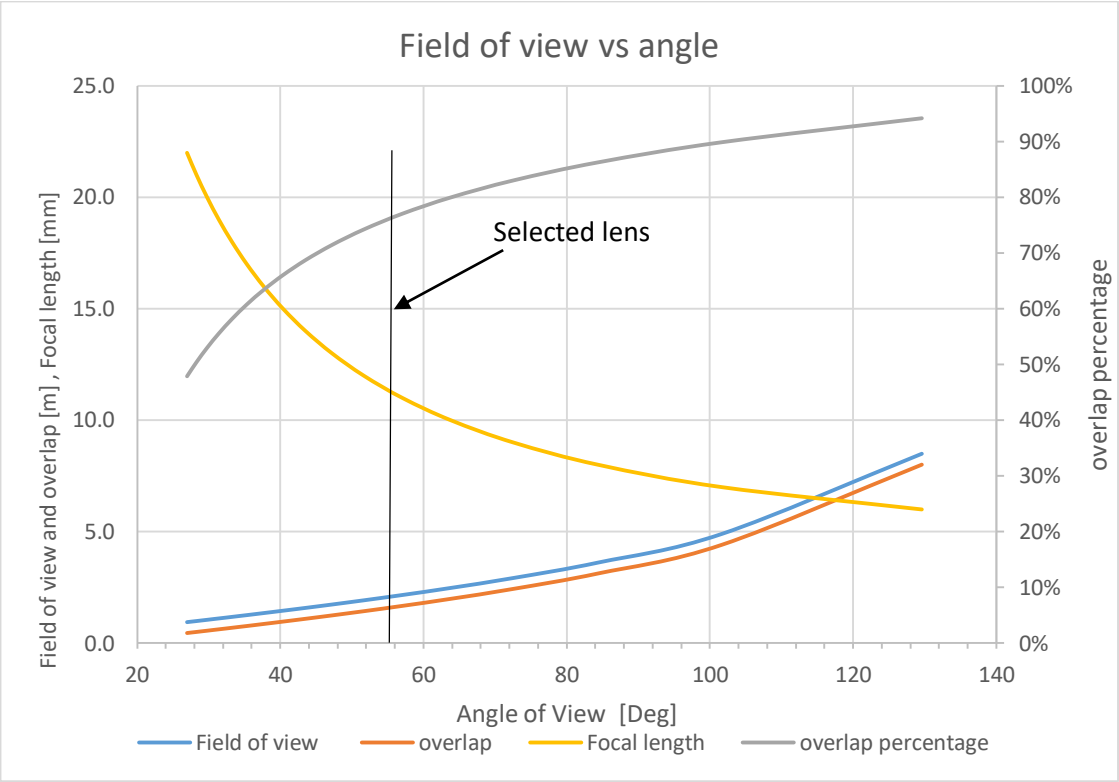

Figure 3-4: Lens selection to provide a maximum overlap in the stereo vision. The baseline used to generate this result is 500 mm.

3.2.3 Depth resolution in the fixed stereo vision setup

The depth measurement resolution depends on the geometrical dimension of the system (Cyganek and Siebert, 2009). The depth resolution is computed based on the baseline (b), focal length (f), horizontal pixel resolution (R_h), view angle (α) and depth (Z)

$$R = \frac{Z^2}{\left[\frac{R_h b}{2\tan\left(\frac{\alpha}{2}\right)}\right] - Z} \tag{3.1}$$

Eq.(3.1) (Cyganek and Siebert, 2009) is used to compute the depth measurement resolution for a fixed setup. Figure 3-5 shows the output of eq.(3.1) for 4 baseline size and over a variety of depth. The working range for the propose platform is 1-2m. Therefore, for the maximum resolution using baseline bigger or equal to 0.2 m in the depth estimation is 5 mm at 2 m and 0.62 mm at depth of 1 m. However, this resolution is based on the external geometry.

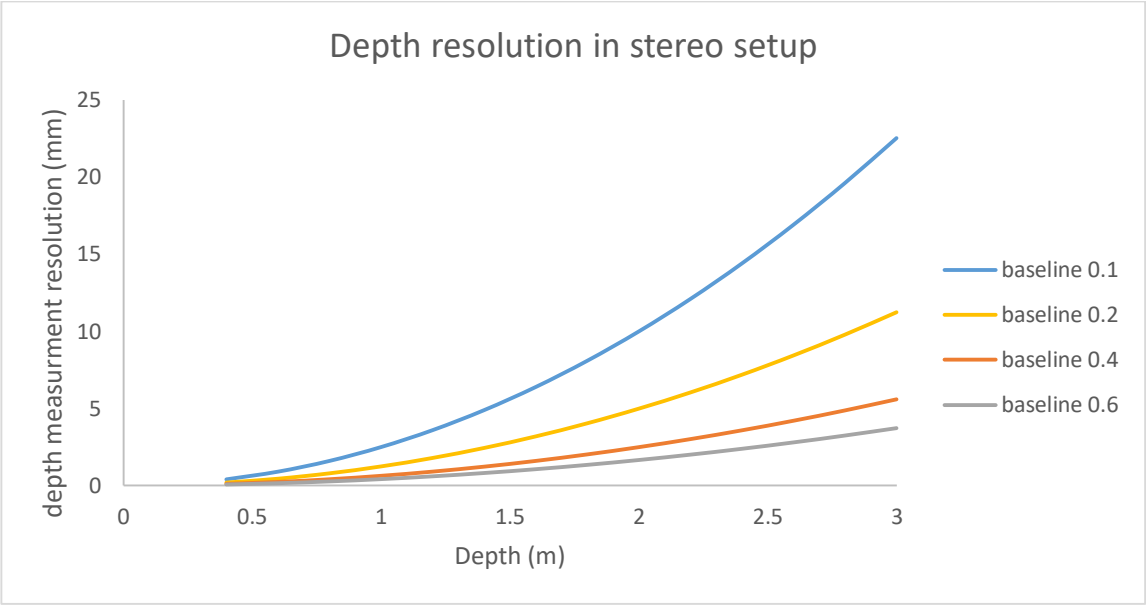

Figure 3-5: Depth measurement resolution for 4 baseline size over variety of depth.

3.2.4 System controller

A robot operating system (ROS) (Quigley et al., 2009) was adopted as the software architecture of the platform. Four low-level nodes are present in the basic structure of the platform, which directly communicate with the hardware, that is, three nodes to control the joints and one node to control the cameras (Figure 3-6). The nodes that control the motors are the baseline, left and right controller nodes. The right and left nodes control the pan and tilt joints, respectively. Each node has a subscribe topic (e.g. /left/pan/move) and a publish topic (e.g. /left/pan/angle). The values of these topics are in degrees, whereas the baseline is in meters.

The stereo node is responsible for the synchronisation of the left and right cameras, generates different types of images (e.g. greyscale and rectified) and controls the parameters of the cameras (Table 3-3). The ROS manages the control command and feedback and then converts raw signal into useful data (e.g. speed in rpm).

Table 3-3: Platform low-level nodes with their topics (the stereo controller node shows the main topics only).

Nodes	Subscriber	Publisher	Units
Right Controller	/right/pan/move	/right/pan/angle	degrees
	/right/tilt/move	/right/tilt/angle	degrees
Left Controller	/left/pan/move	/left/pan/angle	degrees
	/left/tilt/move	/left/tilt/angle	degrees
Baseline controller	/baseline/move	/baseline/position	meter
Stereo controller	/stereo/right/parameters	/stereo/right/image_raw	
	/stereo/left/parameters	/stereo/right/image_gray	
		/stereo/left/image_raw	
		/stereo/left/image_gray	
		/stereo/camera_info	

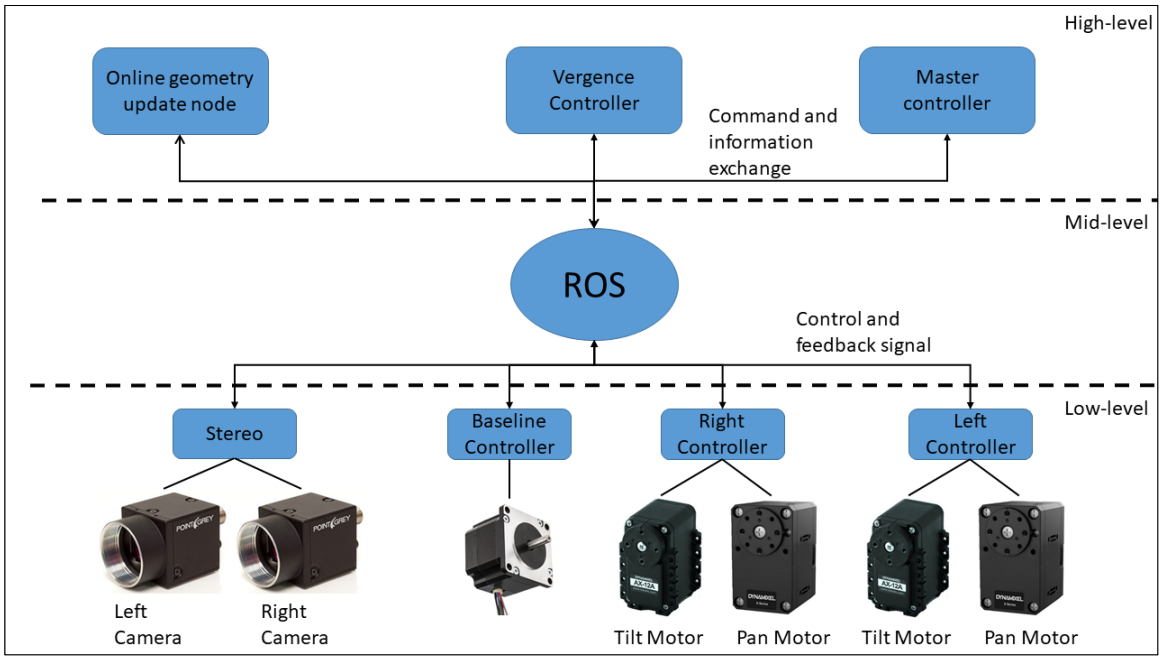

Figure 3-6: Architecture of the platform and controller levels.

The high-level nodes are responsible for applying the algorithm and generating the results and control commands. The node in the level has three types, namely, the online geometry update node, vergence controller for the slave camera (right) node and master controller for the left camera node. These nodes are explained in detail in the following chapters.

In the online geometry update node, the images and angles from the left and right cameras are processed to rectify both images and generate a disparity map (Chapter 4). The vergence controller node controls the fixation point of the slave camera and computes the

depth of the fixation point using vergence depth (Chapter 5). Meanwhile, the master camera node (the "brain" of the system) is responsible for detecting the tomato fruit. The rest of the high-level node actions are based on the change of the master camera (Chapter 6). The high-level nodes are integrated with the cognitive map node which is used in detecting and generating the 3D shape and estimating the depth of the tomato.

3.3 External hardware and software libraries

3.3.1 Computers

Two computers were used in this project, namely, an ASUS ROG GL551JW with Intel® Core™ i7 4720HQ, 2.6–3.6 GHz processor, 16 GB DDR3 1600 MHz memory and NVIDIA GeForce GTX 960M 2 GB GDDR5, which was used in the field experiments; and a desktop with Intel® Core™ i7-7700K, 4.2–4.5 GHz Turbo Quad Core, 8 thread processor, 32 GB 3000 MHz DDR4 memory and NVIDIA GeForce GTX 1080 Ti 11 GB. This desktop was used inside the lab for platform testing and internal experiments.

3.3.2 Computer vision library

Apart from the ROS, we fully used the open computer vision (OpenCV)[5], which is an open-source library. OpenCV has multiple available interfaces. In this work, both C++ and Python were utilised. The version used in this work is OpenCV 3.3.1.

3.3.3 Integration with manipulator arm

The system was designed to work with any manipulator arm. Appendix C (page 186184) shows the integration with a GummiArm robot.

[5] https://opencv.org/

3.4 Experimental procedure

Various sets of experiments were planned in this project to improve the active stereo vision rig. Each set of experiments is discussed in detail in separate chapters with regard to its theory, experiment and results, as indicated in the following section.

3.4.1 Calibration system (chapter 4, page 71)

Calibration is the most important aspect of the entire project. The calibration process was divided into two parts. The first part is offline calibration, and the second one is the online geometry update process. In the first part, a set of 30 different configurations of the platform was setup, in which the baseline and the verge angle were altered in every run. In each run, a calibration process was performed using the algorithm of Zhang (2000) to determine the internal and external properties of the rig. A checkerboard was used during the calibration process. After the calibration process is completed, the external parameters were decomposed to identify the raw data of the rotating angle yaw, roll and pitch.

The raw data were used to analyse the geometry of the rig and evaluate the error generated owing to the manufacturing process and misalignment during assembly. These errors were employed to obtain more accurate movements of the rig during operation.

3.4.2 Vergence controller (chapter 5, page 100)

The vergence cue is the ability to estimate the depth by focusing both cameras on one point. In chapter 5, the vergence controller is evaluated and tested in details using two different approaches which are quantitative and qualitative experiments. In the qualitative experiments, the system examines using a multiple target template. The template contains a 16 target with ArUco at a different height. The designer of the template was to test the reliability and the accuracy of the system to verge on the targets. Then the system tested under un-balance lighting condition where the master camera input light was reduced while the slave maintains the same lighting input.

In quantitative experiments, a depth measurement setup used to analysis the accuracy and repeatability of the platform under different configuration (i.e. at different baseline). Finally, the platform and the vergence controller tested to verge on a tomato fruit in an outdoor environment (greenhouse).

3.4.3 Tomato detection based on visual attention model (chapter 6, page 136)

The final application of the thesis was introduced in this chapter, where a visual attention model was designed and implemented for the harvesting application. The model was designed to identify a tomato fruit and compute the 3D location of the fruit in relation to the manipulator's arm. The algorithms designed and implemented in Chapters 3 and 4 were integrated together with a saliency map to form visual attention for tomato detection.

The saliency map was designed to detect the salient feature in the scene: the tomato fruits are more salient than the ground leaves and the sky. The saliency map reduces the search for special objects in the scene to a few that deemed to have the most salient features. Therefore, a Bayes model was implemented to identify the most salient feature from the saliency map.

The visual attention algorithm generates a cognitive map that contains information related to the target, such as 3D position, probability of the target being a tomato and the grasping affordance. The visual attention experiment is more qualitative than quantitative where a data set of 120 image was capture and used to evaluate the performance of the saliency map. The other experiment was test the entire system in a greenhouse where the system was evaluated by counting the number of success in identify the entire information regarding the targets and then update them in the cognitive map.

Chapter 4

Motor Controller

4.1 Introduction

Visual tracking is a classical problem has been studied in computer vision and has many applications. The classical visual tracking is used a statistic camera where an object tracked within the field of view of the camera; such process uses in industrial especially on the conveyor belt. Many algorithms were developed for the static camera such as background subtracting that assume the background is static and the foreground is changing (Barnich and Van Droogenbroeck, 2011; Ren and Wang, 2016). Background subtraction has many disadvantages like the introduction of illumination and light changing (Xu et al., 2016), various moving backgrounds like moving trees or slow moving of the foreground where these two issue has been studied in (Lin et al., 2017).

Another approach in tracking an object is optical flow. Optical flow is an algorithm depends on feature extraction of the target like using corner detection, Scale-Invariant Feature Transform SIFT (Lowe, 1999), then track these feature in the next frames (Kale et al., 2015). Many works have been done on this approach to improve the quality and the speed of the tracking (Bota and Nedevschi, 2011; Denman et al., 2010; Salmane et al., 2011).

In sport the camera tracking the players or the ball where the camera in these case is moving or in humanoid case the head is tracking the moving object. In this case, the problem of object tracking gets more complicated when the issue of moving camera introduced. The object tracking problem introduces to the control system, which required

to design a controller suit the camera specification. Many works have been done on this type of issue using different techniques and based on the required task. Kim and Kweon (2011) implement object tracking for multiple targets using the homography based motion detection (Loan et al., 2015) to detect an individual target then an online boost tracker was integrated to combine the separate targets.

In (Chen et al., 2014), a detection algorithm to track an object in a moving camera was studied. The algorithm base on feature correspondences between frames, then using the information generated from feature matching the properties of motion is computed. Hu et al. (2015) studied a multiple object detection in moving camera. Which the algorithm was presented in their work is using feature detection in the frames. The features are classified into background and foreground where the foreground represent the target.

In (Mohamed et al., 2016), an algorithm of object tracking in 3D coordinate was studied. The controller used was based fuzzy logic that gives the performance to track the object in 3D coordinate relate to the robot. Where the focus was to control the motor that attached to the camera in order to keep the target within the field of view and the centroid of the camera.

In this chapter, the work is focusing on the controller system that controls the camera during the tracking process where the primary focus on keeping the target within the field of view. In this work, we are interested in keeping the centroid of the target matching the center of the image during the tracking process. This chapter focuses on the control side of the system by integrating an exponential function with the motor controller. To provide

smooth object tracking and control the blur that generated during the movement of the camera.

The chapter is organized as follow: the next section introduces the methodology of the control system, then the setup of the experiment will be shown in section three. In the fourth section, the result and discussion are presented and, finally the conclusions and future development closing this work.

4.2 Background and Preliminaries

4.2.1 Camera model

The pinhole camera model is the standard model used to describe a space point relate to the camera origin (Figure 4-1). Point P is a world point in front of the camera. This point coordinate $[X\ Y\ Z]^T$ is relate to camera origin. A projection of this point $p = [u, v]^T$ is shown on the camera plane π when a connected line between P and the origin of the camera O. (u, v) is the corrected coordinate of the image plane which transform the centroid to lay on the z axis of the camera origin. The relationship between the point P and the camera origin O describe using trigonometry, where two right angle triangle is form and describe in eq. (1) and eq. (2)

$$\frac{\boldsymbol{u}}{\boldsymbol{X}} = \frac{\boldsymbol{f}}{\boldsymbol{Z}} \tag{4.2}$$

$$\frac{\boldsymbol{v}}{\boldsymbol{Y}} = \frac{\boldsymbol{f}}{\boldsymbol{Z}} \tag{4.3}$$

f is the focal length of the camera and describe the perpendicular distance between the origin O and image plane π. Using these two equations help to determine a scaler position of point P, due to the missing of Z.

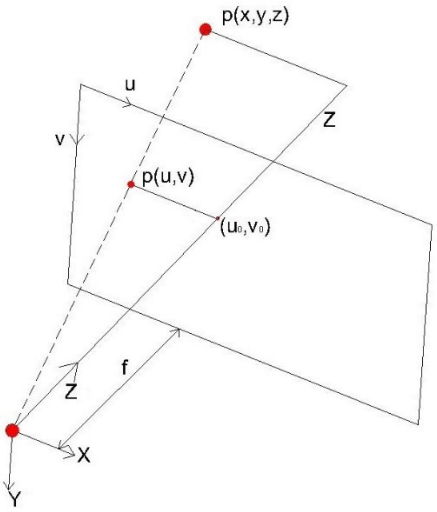

Figure 4-1 single camera model

The ability to move the eyes focus on a fixation point is referred to as vergence vision. This cue is used by human and different kinds of animals. In this work, the control system was designed to work with an active stereo vision that has five degrees of freedoms. The system was designed and built to keep the origin of the cameras collide with the rotating axis of the motors. Because if the rotating axes are not aligned with the origin of the camera a large displacement will occur during the tracking and this will lead to losing the target.

4.2.2 Object tracking

In object tracking, the moving speed of the target affects the detection part in the system, where this depends on the shutter speed of the camera. Shutter speed is defining the time the camera exposed to the light. The higher the shutter speed leads to less motion blur (Szeliski, 2009). The motion blur can be computed using eq.**Error! Reference s ource not found.**

$$B_m = \nu \times ST \qquad (4)$$

Where B_m is motion blur size, ν is the velocity of the moving object, and ST represent the shutter speed. Eq.**Error! Reference source not found.** use to computer the blur occur d ue to the motion of the object or the camera's motion.

The motion blur uses to determine the maximum speed of the moving camera or the object in order to match it with the detection algorithm. The quality of the detection algorithm depends on the features in the image where a large motion blur could lead to mixed the features or erase them (Figure 4-2). As shown in Figure 4-2 the feature of the image has been mixed and erase, where the detections algorithm fails to detect the fundamental features such as corner and lines. From this, we know that the motion blur has a significant effect on the detection algorithm.

Figure 4-2 Motion blur generated due to the fast camera motion

A standard object tracking algorithm is used in this work because the focus on implementing the control system is the task of this paper. An open-source library was used in this paper as a tracking algorithm. The library used in this work is Aruco (Garrido-Jurado et al., 2014) which used a pattern to identify the centroid of the pattern. This algorithm depends on detecting the corners and lines which will have a major effect on the detection when the camera speed passes the limited speed.

The control system used in this work is based on exponential function. The reason for using an exponential function is the behaviour generated using it. As described in motion

blur that affects the quality of the object detection algorithm, where at substantial differences the camera can move fast but when the target gets close to the center a more resolution image required which need a smooth movement. At the same time using a camera as a feedback sensor, the oscillation behaviour is unwanted.

The control system in this work represented as motor and camera used as feedback sensor. The block diagram is shown in Figure 4-3 where the output of the system is the position of the object about the camera centroid. The input to the motor is the angular velocity $\dot{\theta}$ in rpm, which computed by the exponential function eq.(6.11).

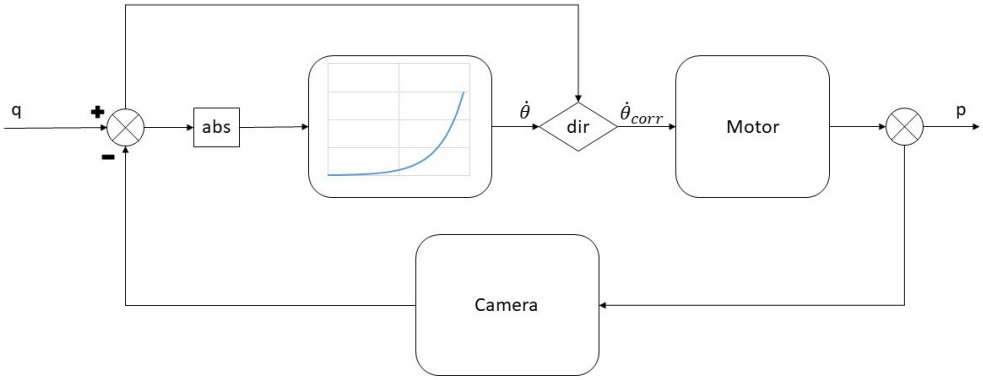

Figure 4-3 The block diagram of the object tracking

$$\dot{\theta} = \exp^{(q \times \lambda)} \times \beta \tag{5}$$

Where $\dot{\theta}$ is the angular velocity in rpm, β is control constant that control the range of the output to meet the range of the motor. While λ is the constant describe the shape of the output. q is the input to the exponential function where this has to be always positive? The direction of the output velocity $\dot{\theta}$ is corrected using the sign of the q before taking the absolute of this value eq.(6.12).

$$\dot{\theta}_{corr} = \frac{q}{|q|} \times \dot{\theta} \tag{6}$$

Where $|q|$ is an absolute q always positive.

4.3 Experiment setup

The control system was evaluated using a static target and moving target, where the static target test places the target in front of the platform at a different position then start the system from the default position (zero position of the motors). This will give information about the system as when introducing a step input where the behaviour of the system will be studied. The second test it a moving object. Where an object fixed on a cantilever attached to the motor. The speed of the rotating of the motor can be control which will define how fast the target can move.

The size of the image was used in this setup is $2084\,x\,1042$ (4.4 MP). The object detection returns the centroid q_{target} of the target in image coordinate which the origin is at top left corner of the image. Therefore, a remapping to these coordinate was applied to move the origin coordinate to the center of the image plane; which is the origin of the camera and the rotating axis. Using the new coordination, the output of the $q(u, v)$ has range of $(-1024,1024)$ pixel.

The maximum speed of the motor that should move at to avoid blur image was calculated using eq.**Error! Reference source not found.** where in the worse scenario the m aximum distance between the target and the platform is 2 meters. The maximum angular speed of the motor $\dot{\theta}_{max} = 60\,rpm$. Convert the angular speed to linear speed using motion equations $v = 12.5\,m/sec$. Now put these value with the camera speed shutter $ST = 0.021\,ms$ into eq.**Error! Reference source not found.** the blur motion $B_m =$

$0.26\ mm$. This value converted to a pixel in order to check the detection algorithm requirement to detect the object in motion.

The exponential function was designed to meet the input pixel size and the output angular speed. From eq.(6.11), there are two parameters need to optimize λ and β, one responsible on controlling the behavior of the trajectory of the motion and the other parameter responsible to control the speed of the trajectory respectively.

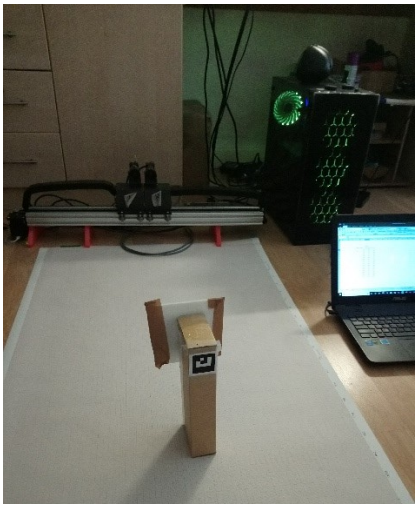

Figure 4-4 The experiment setup with the static target

Figure 4-4, shows the experimental setup the static target where the target place in front of the platform at different positions in x range from (400 to 2500) mm and y from (-500 to 500) mm.

The second experiment is to track a moving target. This experiment will determine the performance of the controller. A Stepper motor with speed controller is connected to a cantilever, and on the other end of the cantilever, the target is fixed. The Stepper motor is controlled by speed in rad/sec, and the linear velocity of the target was calculated as well in order to compare it with the blur motion. The setup of the experiment is shown in

Figure 4-5where this time the detection algorithm was used is based on colour detection just to simplify the process.

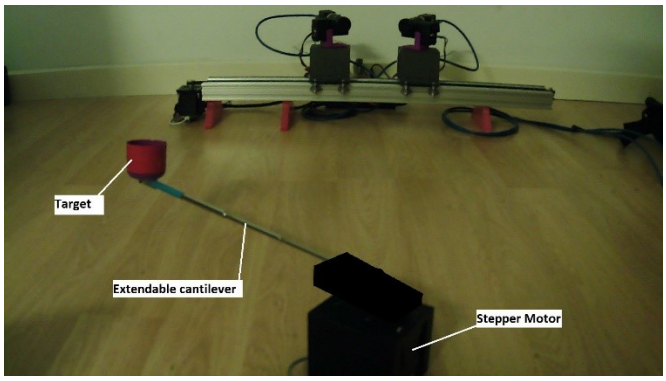

Figure 4-5 Moving object tracking experiment setup.

The cantilever is used in the experiment can be extended at different length which helps to generate more data that help to improve the performance. The length of the cantilever is set to 200,400, and 500 mm.

4.4 Result and discussion

In this part, the result of the testing is presented. The output result of the system was divided into three different lambda λ (0.0010, 0.0015, 0.0020). In each λ, there are 4 control β that responsible on controlling the angular speed of the camera motor.

Figure 4-6 to Figure 4-8 show the results of the static experiments where each figure compare the different running of different β.

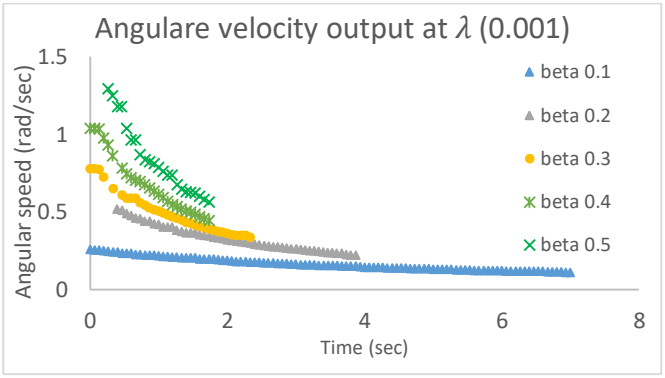

Figure 4-6 Angular velocity of the motor at lambda 0.0010

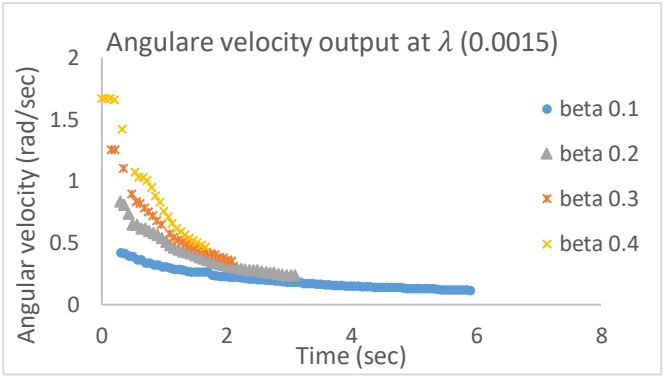

Figure 4-7 Angular velocity of the motor at lambda 0.0015

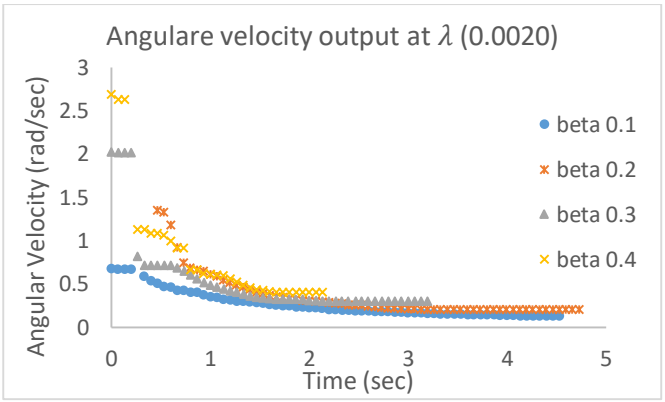

Figure 4-8 Angular velocity of the motor at lambda 0.0020

The output of the experiment shows that the system can track the target at the maximum speed of the motor where $\beta = 0.5$. However, the time take for the system to center the target with the camera is range from 1.5 to 7.5 seconds, which depend on the λ and β. The increase of β lead to increase in the tracking speed. This increase in β leads

the system to change the behaviour of the exponential function output where the output become more step.

In Figure 4-7 and Figure 4-8, shows the increase of the beta leads to increase in the speed of tracking the object in the same time this cause the camera to lose the target as a result of the increase of the blur that effects the feature detection at the beginning of the tracking process. Moreover, based on this output the maximum angular velocity should be set to 2 rad per seconds, which is half the speed of the theoretical velocity that calculated in the experiment setup section.

The output of the system in all cases has followed the exponential output as is wanted with a confidential of 90%. Using the exponential in object tracking system help the system to track the object at a different position, and fix the gaze point with the centroid of the target with an error of $\pm$ 5 pixels, which this improve the performance a lot in the next process.

There is an issue with the lambda, where the increase of the lambda leads to steep in the angular velocity when the object is far away from the center; then the angular velocity drops to a minimum while still, the object is far from the centre. Difference setup of lambda is shown in Figure 4-9. The increase of the value of lambda leads to the drop of the speed control value β to maintain the speed within the limited.

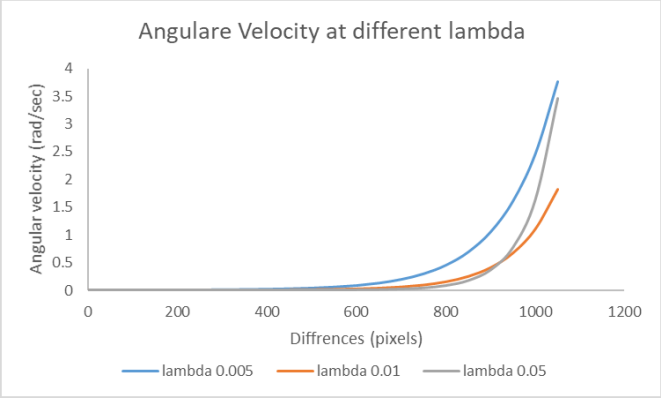

Figure 4-9 Exponential function at different lambda

Moreover, the result shows that using exponential function help to improve the settlement time of the system and avoiding fluctuating. In object tracking using the movable camera, the fluctuating of the camera leads to increase in the error which the system loses the target.

Therefore, the selected lambda and beta was chosen in order to give the best performance was to select a small lambda ($\lambda = 0.001$) and combine it with large beta ($\beta = 0.5$). This combination was selected due to the output performance which help the system to keep tracking the target without losing the target.

The results of tracking moving object are shown in Figure 4-10 - Figure 4-12. These results show the differences between the object center and zero coordinate of the image. Three cantilever length was used in the experiment where it leads to maximum linear velocity of the object to 0.76 m/sec. As figures shows that the controller keep the target within the field of view. The increase in object speed lead to extend the distance of the object to the origin of the image.

At speed of 90 deg/sec with cantilever length of 500 mm the control system start to suffer lose in the target as shown in Figure 4-12. These occur due to the presents of large motion blur that effect the output of object tracking.

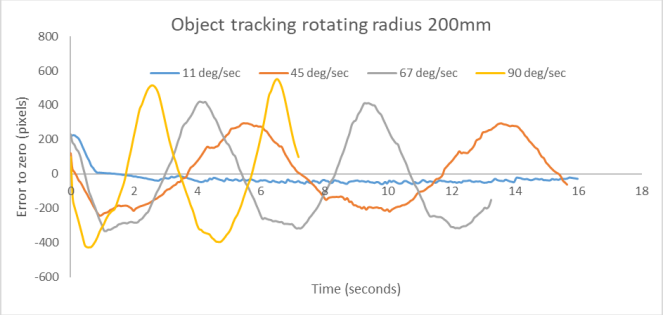

Figure 4-10object tracking using cantilever length 200 mm

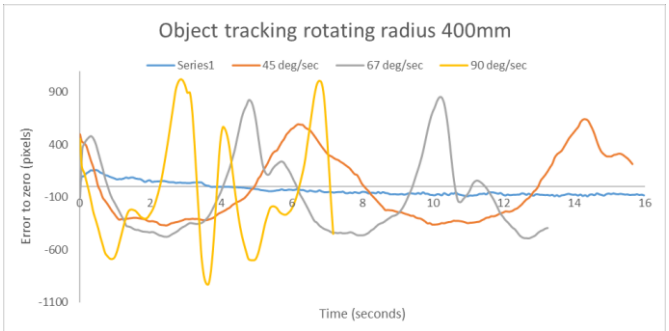

Figure 4-11object tracking using cantilever length 400 mm

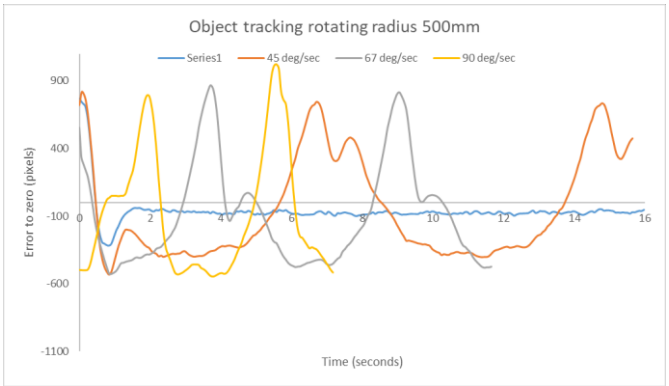

Figure 4-12 object tracking using cantilever length 500 mm

4.5 Conclusion

To conclude, this chapter present a work was done on an active stereo vision with five degrees of freedoms. The control system of object tracking using the portable camera was studied. The controller was designed to use the exponential function in order to track the object. The functions show the ability to track the object with a good speed that maintains the image with no blur. Moreover, the control system help to improve the track the object with an accuracy of ± 5 pixels which lead to improvement in deploying the vergence system. This improvement was due to the nature of the exponential that show linear behaviour when the target gets closer to the center of image.

Chapter 7

Active Stereo Platform: Online Epipolar Geometry Update

This chapter presents a novel method to update a variable epipolar geometry platform directly from the motor encoder based on mapping the motor encoder angle to the image space angle, avoiding the use of feature detection algorithms. First, an offline calibration is performed to establish a relationship between the image space and the hardware space. Second, a transformation matrix is generated using the results from this mapping. The transformation matrix uses to update the epipolar geometry of the platform to rectify the images for further processing. The system has an overall error in the projection of ±5 pixels, which drops to ±1.24 pixels when the verge angle increases beyond 10°. The platform used in this chapter is version one, that has three degrees of freedom to control the verge angle and the size of the baseline.

5.1 Introduction

In this chapter, the problem of updating the epipolar geometry in active stereo vision directly from a motor angle is solved using Projective Projection Matrix (PPM) to rectify the images. An improvement to the algorithm used by Dankers et al. (2004) is presented in this chapter. The idea to find the relationship between the image space and the motor space was taken from Sapienza et al. (2013) where in this work the PPM is used instead of computing the homography transformation between both images. When the raw data of the system are extracted using the image space and the actual geometry data, a linear relationship is drawn to perform conversions between the motor angles and the image angle, including the error in the manufacturing process. The configuration of the system

is studied in depth to allow an accurate rectification process for the images generated by the system under different arrangements.

The rest of this chapter is organized as follows. The epipolar geometry and the process of computing the parameters in image space are presented in Section 5.2. Section 4.3 presents the process of collecting the data using a stereo calibration algorithm and the setup used to evaluate the algorithm. In Section 5.4, the results and discussion are presented, and finally, the chapter is concludes in Section 5.5.

5.2 Epipolar geometry analysis

This section introduces the algorithm used to produce the disparity map and depth measurement while the camera tracks an object without the need to constantly recalibrate the system. The process of updating the epipolar geometry online is described in this part. The method of updating the configuration of the system has two stages. The first stage is the offline calibration process using Zhang's calibration algorithm (Zhang, 2000), where the output of this algorithm is the PPM and distortion matrix for each camera. The PPM and distortion matrix contain the internal and extrinsic parameters for each camera, and the internal and distortion parameters are assumed fixed at all times.

Figure 5-1 shows the outer parameters of the system. The origin of the system is set, as is frequently done in computer vision, with the left origin as the origin of the system (Bradski and Kaehler, 2008). Therefore, the essential matrix describes the rotation and translation from the left image to the right image. In the offline calibration stage, the calibration was done under different geometric configurations. This process is used to find the relationship between the rotation angle in the image space and the motor space.

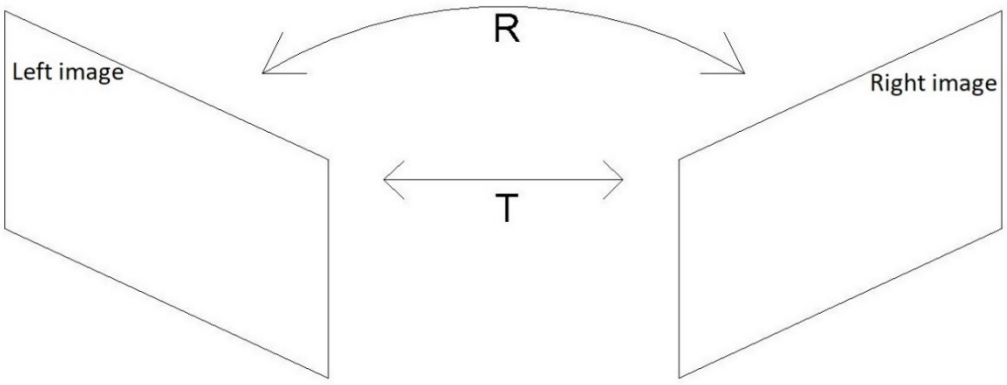

Figure 5-1: The relationship between the left and right cameras described by the essential matrix, which contains the rotation and the translation measurements.

The second stage of the calibration is online calibration, where the generated relationship between the image space and the motor space is used to update the PPM.

5.2.1 Single-camera model

We start with a single-camera model that describes a pinhole camera system. This model is also used to describe the Complementary Metal-Oxide Semiconductor (CMOS) sensor in the cameras used in this project. The center of the camera is O, which is the center of the Euclidean coordinate system. The image plane π coincides with the Z-axis, and the distance between the origin and the image plane is the focal length f.

Suppose a point W with coordinates $W = [X\ Y\ Z]^T$ set in the front image plane. A projection point $w = [x\ y]^T$ on the image plane will form when we draw a line from W to the origin of the camera O. This creates a mapping from 3D space to 2D space. Using a homogeneous coordinate to map between points, we get Eq. (5.1):

$$w = PW \tag{5.1}$$

where $W = [X\ Y\ Z\ 1]^T$ and $w = [x\ y\ 1]^T$ are homogenous vectors and P is the camera projection matrix.

The camera projection matrix P contains the internal and external parameters:

$$P = AR[R|t] \tag{5.2}$$

where A is a 3 × 3 matrix describing the internal properties of the camera (Eq. (5.3)), where α_x and α_y are the focal lengths in pixels in the x and y directions, respectively, and s is a skew parameter, which, in most new cameras, is zero (Szeliski, 2009). R and t are external parameters that refer to the transformation between the camera and world coordinate, where R is a 3 × 3 rotation matrix of rank 3 and t is a translation vector.

$$A = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.3}$$

5.2.2 Stereoscopic model

Stereo model represents two cameras observe the same view from different location. In this section, the parameters with subscript letters l and r are used to refer to the left and right camera models, respectively. Figure 5-2 shows the model that is studied in this section. The distance between the two origin cameras is b and is referred to as the baseline. Supposing that both cameras look at the same point in the world $W = [X\ Y\ Z]^T$, a point w_i will be projected onto both image planes $w_l = [x_l\ y_l]$ and $w_r = [x_r\ y_r]$.

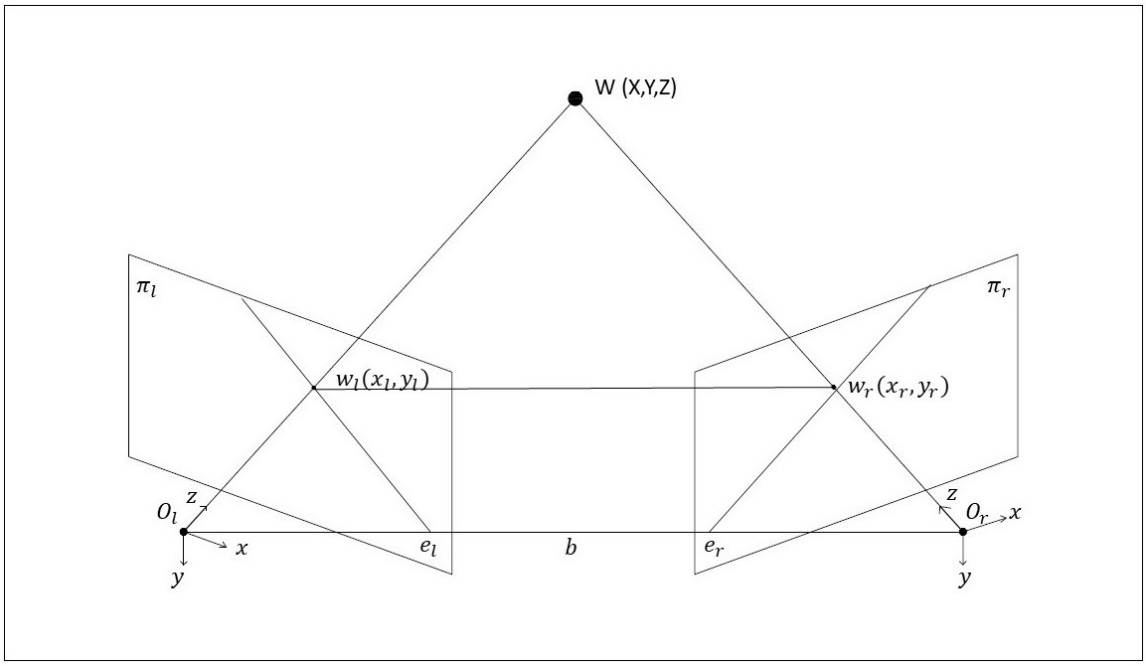

Figure 5-2: Stereo model represent the epipolar geometry.

From the model, a plane is formed when $O_l, W,$ and O_r are connected. This plane is called the epipolar plane. Epipolar point e_i, is the point occur when the line $O_r O_l$ cross the image plane π_i. However, in some cases there is no cross between $O_r O_l$ line and the image plane which result e_i lays in infinite (Cyganek and Siebert, 2009). If w_l is known, w_r will be excite laying on line $l_r = e_r \times w_r$. This line is called the epipolar line. From the epipolar line, $l_r = e_r \times w_r = [e_r] \times w_r$, where $[e_r]$ is the cross product, and because we know that w_r is mapping to w_l, we get the relation $w_r = H\, w_l$. H is a 3 × 3 homography matrix of rank 3 that describes the mapping between two points. By combining both equations, we get $l_r = [e_r] \times H\, w_l = F\, w_l$, where $F = [e_r] \times H$ and is called the fundamental matrix (Hartley and Zisserman, 2003).

The fundamental matrix (F) can be extended to include the camera projection matrix, as shown Eq. (5.4), where P_l^+ is the pseudoinverse of P_l. The fundamental matrix describes the internal and external parameters of the stereo vision system. F is a 3 × 3 matrix of rank 2.

$$F = [e_r] \times P_r\, P_l^+ \tag{5.4}$$

For a stereo vision system, the projection camera matrix satisfies Eqs. (5.5) and (5.6), where R and t represent the rotation and translation between the left and right origins. O_l is the origin of the system.

$$P_l = [I\ |0] \tag{5.5}$$

$$P_r = [R\ |t] \tag{5.6}$$

The fundamental matrix should satisfy Eq. (5.7), where w_l lies on the epipolar line $l_r = Fw_l$ (Hartley and Zisserman, 2003):

$$w_r F w_l = 0 \tag{5.7}$$

Equations (5.5) and (5.6) are in normalized coordinates, and solving them, we obtain Eq. (5.8):

$$E = [t]_\times R = R[R^T t]_\times \tag{5.8}$$

The essential matrix (E) describes the transformation between the left and right origins in normalized image coordinates. The E matrix has similar properties to the F matrix in its correspondence between $\widehat{w}_l$ and $\widehat{w}_r$ in normalized coordinates (Hartley and Zisserman, 2003):

$$\widehat{w}_r E \widehat{w}_l = 0 \tag{5.9}$$

The essential matrix is used to compute the coordinate of the point $W(X, Y, Z)$ seen by both cameras. Using the essential matrix means that there will be six degrees of freedom: three degrees from the rotation angle and three degrees from the translation. In our system, the rotation angle around the y-axis and the translation along the baseline are not fixed. These two parameters were selected because they change the visual view of the camera.

5.2.3 Calibration

The calibration camera is the process to estimate the camera parameters the intrinsic and the extrinsic parameter. The principle of the calibration process is to define a point in the scene that correspondence to the image points. A checkerboard is used due to the simplicity to detect the corners and the easier to measure their position (Cyganek and Siebert, 2009).

First Eq.(5.1) is convert to homogeneous form Eq.(5.10).

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{5.10}$$

The calibration process use the define points in the scene and the correspondence points in the image to solve eq.(5.10). The only missing in eq.(5.10) is the projection matrix. Therefore, for each point ($i = 1,2,.,.$) in the scene a two equation is computed eq.(5.11) (5.12)

$$x_i = \frac{P_{00}X_i + P_{01}Y_i + P_{02}Z_i + P_{03}}{P_{20}X_i + P_{21}Y_i + P_{22}Z_i + P_{23}} \tag{5.11}$$

$$y_i = \frac{P_{10}X_i + P_{11}Y_i + P_{12}Z_i + P_{13}}{P_{20}X_i + P_{21}Y_i + P_{22}Z_i + P_{23}} \tag{5.12}$$

There are two equations with 12 unknown variables from projection matrix therefore, 6 point required in the scene to find the projection matrix variables. Eq.(5.11) (5.12) split into a vectors with known and unknown variables, then the solution done using the singular value decomposition (Cyganek and Siebert, 2009).

The extrinsic parameters in stereo vision is solving eq.(5.9) where w_r and w_l is the normalized correspondences points between the left and right and as well to the scene points (Hartley and Zisserman, 2003).

The calibration function used in this work is based on OpenCV library[6].

5.2.4 Rectification algorithm

The disparity is the difference between the same points in the left and right images. The calibration process generates the parameters used to rectify the images, where the rectification process is the transformation of the left and right images to obtain the same horizontal epipolar lines. The rectification process used in this study is based on Bouguet's algorithm (Bradski and Kaehler, 2008).

The process starts by dividing the rotation matrix R that is responsible for rotating the right image into the left image into two rotating matrices, R_l and R_r, for each image. These two rotation matrices rotate the left and right images by a half rotation. This rotation aligns both image planes with the baseline, but the images are not aligned in the raw data. Therefore, we find a correction matrix to rotate the epipolar lines into infinity and align them horizontally with the baseline.

In the Stereoscopic model, it is assumed that the left camera was set as the origin of the system. Starting with the epipole point e_{1_l} in the left image and connecting to the epipole point e_{1_r} in the right image, the point is translated along the baseline that defines the translation vector T. This leads to Eq. (5.13):

$$e_1 = \frac{T}{||T||}. \tag{5.13}$$

Using the cross product of e_1 will generate e_2, which is orthogonal to the focal length ray. This results in e_2 being orthogonal to e_1. The result is shown in Eq. (5.14):

[6] OpenCV used (Zhang, 2000) to find the camera matric and used (Brown, 1971) to find the lens distortion.

$$e_2 = \frac{\left[-T_y \; T_x \; 0\right]^T}{\sqrt{T_x^2 + T_y^2}}. \tag{5.14}$$

The last vector is e_3, which is orthogonal to e_1 and e_2, and can be calculated via a cross product:

$$e_3 = e_1 \times e_2. \tag{5.15}$$

Now, we add these vectors into the correction matrix R_{corr}, which transforms the epipolar lines to be infinite and parallel with the baseline by rotating the image about the projection center.

$$R_{corr} = \begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix}. \tag{5.16}$$

R_{corr} is multiplied by the split rotation matrix to form correction rotation matrices for the left and right images.

$$R_{l_{corr}} = R_{corr} \, R_l \tag{5.17}$$

$$R_{r_{corr}} = R_{corr} \, R_r. \tag{5.18}$$

This leads to the importance of a given rotation matrix and translation matrix to rectify an image. The rotation and translation matrices are taken from the essential matrix, i.e., decomposing the essential matrix allows the rotation and translation matrices to be calculated.

5.2.5 Online geometry update

This subsection integrates the above discussion to generate a relationship between the image angle and the motor encoder angle. Mapping between motor space to image space lead to errors if we use the encoder angle direct to the image angle (Kyriakoulis et al., 2008).

As explained in the above section, the process is divided into two parts: an offline calibration process and an online geometry update. The offline calibration calculates the essential matrix and the internal parameters of the cameras. The essential matrix is decomposed to generate the rotation and translation matrices. The translation matrix is a pure translation from the left to right camera origins.

In theory, the rotation matrix should be equal to the pure rotation around the y-axis. However, in reality, this assumption is not valid because of the actual installation of the camera on the platform and the installation of the camera sensor. The calibration result in online calibration returns the rotation matrix, including these small values around the x- and z-axes. Therefore, the rotation matrix returns three angles. The complete rotation matrix is a product of multiplying the rotation matrices in XYZ order[7]:

$$R = R_x(\psi) \times \mathrm{R_y}(\theta) \times R_z(\emptyset). \tag{5.19}$$

The rotation matrix is solved to return the individual angle. These angles are recorded as the image space angles. The most important angle is θ_{img}, which changes the angle around the y-axis.

The calibration process is done 30 times with different configurations (different verge angles) and each time the encoder verge angle $\theta_{encoder}$ is recorded. The complete 30-configuration calibration set constituted one run, and 20 runs were performed. The data of the calibration process are used to generate a linear relationship between the encoder angle and the image angle:

$$\theta_{img} = e + \eta \times \theta_{encoder}, \tag{5.20}$$

where e refers to the error due to the mechanical misalignment and lens distortion and η is an estimated factor to correct the encoder angle.

[7] In OpenCV Library the product is follow Z-Y-X sequence (Bradski and Kaehler, 2008).

5.2.6 Disparity

After the rectification of the system, the generated left and right images are used to compute the disparity map. Correspondence is then established, following the extensive literature, for example (Scharstein and Szeliski, 2001). The primary junction of correspondence is to find the point in the right image to match the point in the left image and then calculate the differences in the x-axis. These differences are called the disparity.

The Semi-Global Block-Matching algorithm (SGM) (H. Hirschmüller, 2005) is used in this study to evaluate the disparity map of the rectified images. SGM is a global stereo matching algorithm using multiple direction searches (pixel-wise) to smoothen the output, where the matching cost used in SGM is Mutual Information to overcome issues in lighting, different time exposures, and reflection (Banz et al., 2010). The pixel-wise method calculates the final disparity by summing the total cost of the disparities at different angles from the scan line. This approach ensures that there is some smoothness in the disparity.

$$E(D) = \sum_{P}(C(p, D_p) + \sum_{q \epsilon N_p} P_1 T[|D_p - D_q| = 1] + \sum_{q \epsilon N_p} P_2 T[|D_p - D_q| > 1]). \quad (5.21)$$

Equation (5.21) represents the minimized cost function used by SGM, where p and q are the pixel indices in the image, $C(p, D_p)$ is the cost of disparity matching based on the intensity, N_p represents the neighbour of the pixel p, and P_1 and P_2 are constraints to penalize the change in the disparity, where P_1 represents the change equal to 1 and P_2 represents the change greater than 1 (Hirschmuller, 2008).

The disparity map is used to transform the pixel from the image coordinate in 2D into a world coordinate in 3D $[X\ Y\ Z]^T$ relative to the camera origin. This process is done using a triangulation approach in Eqs. (5.22)–(5.24). In Eq. (5.23), x and y represent the

modified coordinates of the object in the image frame, b is the baseline, d is the disparity, and f represents the focal length.

$$Z = f * \frac{b}{d} \tag{5.22}$$

$$X = f * \frac{x}{Z} \tag{5.23}$$

$$Y = f * \frac{y}{Z}. \tag{5.24}$$

5.3 Experiment

The setup of the experimental system was divided into two stages. The first stage collected the data for the calibration process to find the relationship between the image angle and the motor angle. The second stage evaluated the new calibration algorithm.

5.3.1 Collecting data

This section explains the process of obtaining the data to help extract the parameters of the active stereo vision system. Exploring the parameters of the system and comparing the image space to the motor space required generating data for different platform setups, which meant setting different verge angles and baselines; 30 configurations of varying verge angles and five configurations for the baseline were selected.

In each configuration, a calibration process was performed as explained in Section 5.2.3 to find the parameters of the system using a calibration board. The board consists of an 8 × 6 array of black and white squares with sizes of 34.5 mm in height and width. The algorithm used to find the corners on the checkerboard also detected the 48 internal corners on the board. For a robust calibration, 15 images were taken of the calibration board at various positions and orientations relative to the platform as recommended by (Bradski and Kaehler, 2008).

To accelerate and improve the collection of data, the calibration process was automated using a Baxter robot. Automating the calibration process reduced the time required to complete the calibration process by three times and improved the calibration result.

Figure 5-3 shows the data collection setup, where the platform was installed in front of Baxter at a distance of 2 m and the calibration board was fixed on the arm of the robot. A total of 40 positions and orientations of the board were pre-recorded using the Baxter teaching methods. A desktop PC was used to control Baxter, and a laptop was used to control the platform and perform the calibration process. A UDP connection was used to communicate between the PC and the laptop.

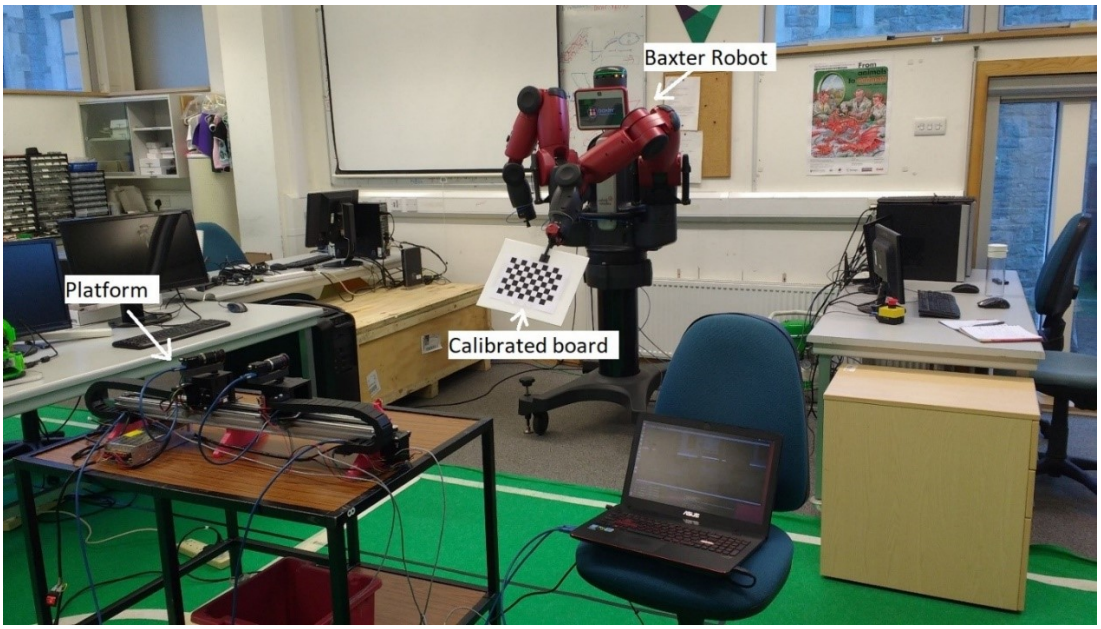

Figure 5-3: Baxter holding the checkerboard while the rig works on the calibration (in the lower left of the figure).

Figure 5-4 shows a flowchart of the calibration process, where the process starts by setting the verge angle. The second step is to find the corner and to move the arm to a new position. This step was repeated until 15 sets of images were taken successfully with the corners detected. Then, the calibration process is started at the same time as the evaluation of the quality of the calibration; when the output meets the requirement that the projection error is less than 0.1, the calibration process is a success, and the system moves to a new

configuration. If the projection error is larger than 0.1, the process repeats until it meets the requirement. This algorithm was repeated 20 times to generate data for the analysis.

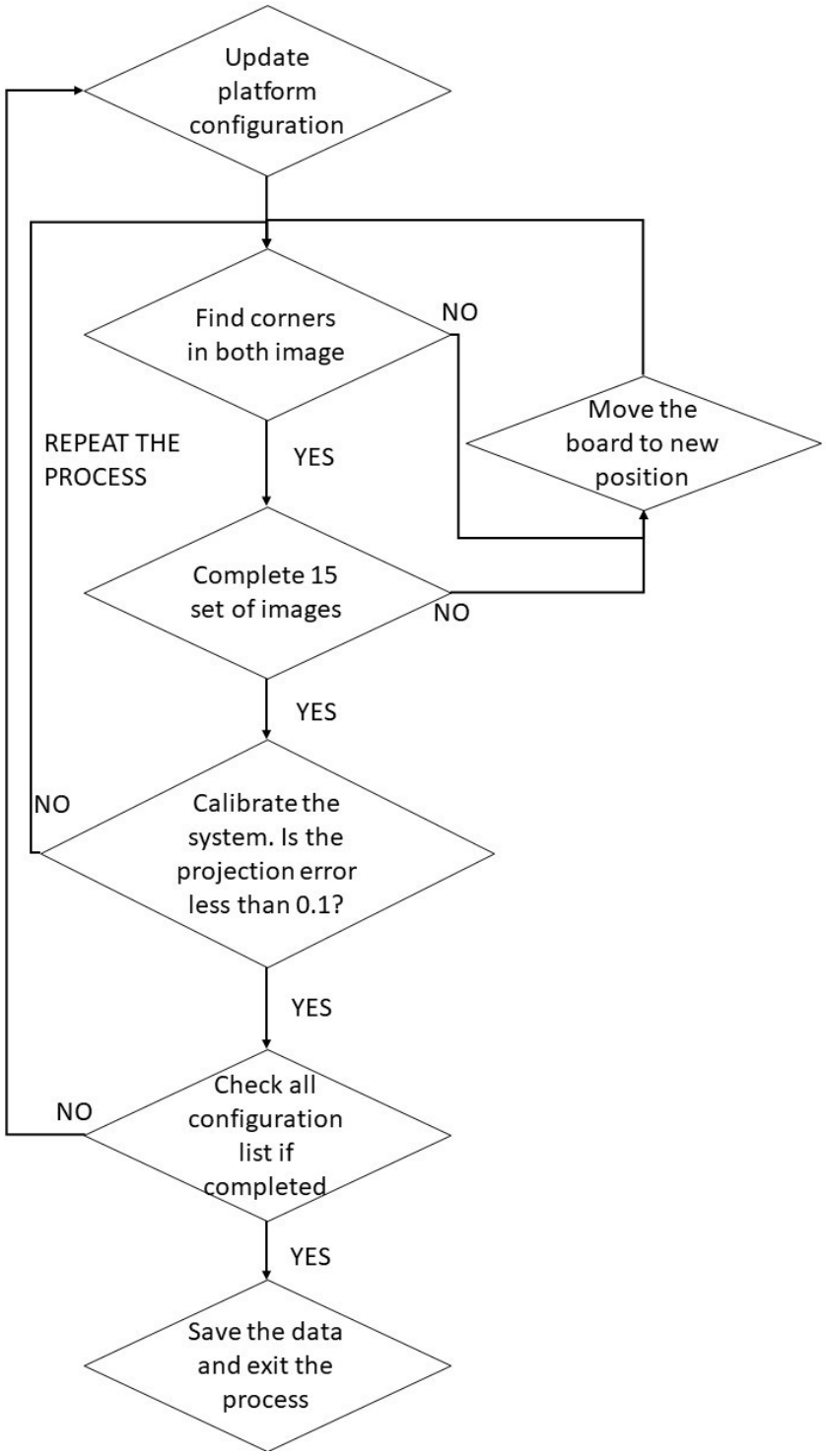

Figure 5-4: Flowchart of the automated calibration process.

5.3.2 Rectification

The calibration algorithm results in a rectified image where the epipolar lines of the left and right images become co-linear and parallel with the horizontal axis. To measure the performance of this rectification, a projection error measurement was used as described in (Hartley and Zisserman, 2003). The projection error is defined as the difference between the point y-axis in the left image and the point y-axis in the right image, as shown in Figure 5-5 (Forsyth and Ponce, 2012).

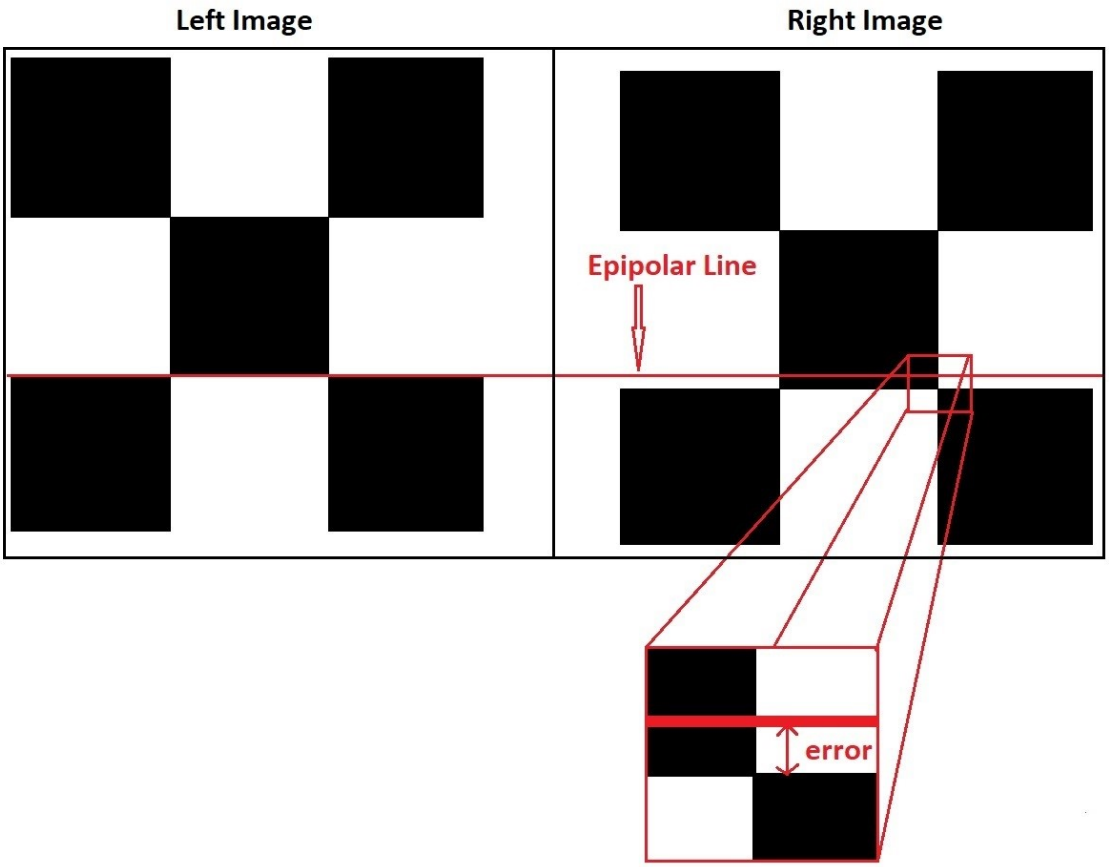

Figure 5-5: Definition of the error generated in the rectified images.

A calibration board was placed in different locations and orientations at distances between 1.5 and 2.5 m from the platform. This allowed us to obtain more data and to evaluate the calibration algorithm more accurately. The geometry of the system needs to be updated when the configuration of the platform changes, and rectified images should then be generated. The rectified images are the output of the calibration algorithm, and these two

images are used to evaluate the quality of the calibration. The evaluation algorithm uses the calibration board to detect the corners of the left and right images and then calculate the Root Mean Square error (RMS), i.e., Eq. (5.25). The output value is in units of pixels.

$$error = \sqrt{\left(y_{l_i} - y_{r_i}\right)^2} \tag{5.25}$$

5.3.3 Surface compression

The data generated from the disparity map are used to create a 3D point cloud related to the system origin, which is a physical dimension of the scene. These data are used to evaluate the quality of the system in generating the point cloud. A spherical object was placed in front of the system, and a 3D point cloud was generated for this sphere. These data were then compared to the ground truth of the sphere that was generated using a 3D model.

The iterative closest point (ICP) algorithm was used to translate and rotate the source of the point cloud to the reference by minimizing the differences (Rusinkiewicz and Levoy, 2001); that is, ICP was used to align the two point clouds. There are four steps that ICP uses in the alignment process, as described in the work of (Rusinkiewicz and Levoy, 2001).

1. Apply the correspondent to the points where the strategy starts by selecting a point with a uniform distribution.
2. Use singular value decomposition to compute the rotation and translation between the reference and source point clouds.
3. Apply rotation and translation to the registered point cloud.
4. Calculate the error between the corresponding points by applying SSD.

The above steps were repeated until the error reached the threshold value.

To evaluate the generated sample (S) point cloud of the platform, it was compared to the reference point cloud that was generated using a model, which we refer to as the ground truth (G). The Euclidean distance algorithm, Eq. (5.26), is used to compute the distance between each point in the source that lies near the point in the reference point cloud. The differences between the sample and the ground truth were calculated using the RMS using the Euclidean distance:

$$RMS_{error} = \sqrt{(S_x - G_x)^2 + \left(S_y - G_y\right)^2 + (S_z - G_z)^2}. \quad (5.26)$$

In the experiment, three spheres were used, with different diameters (80, 125, and 150 mm). CAD software was used to generate the ground truth, which was then converted to a point cloud. These point clouds were set to have a subsampling between points equal to 1 mm in all directions (Figure 5-6A).

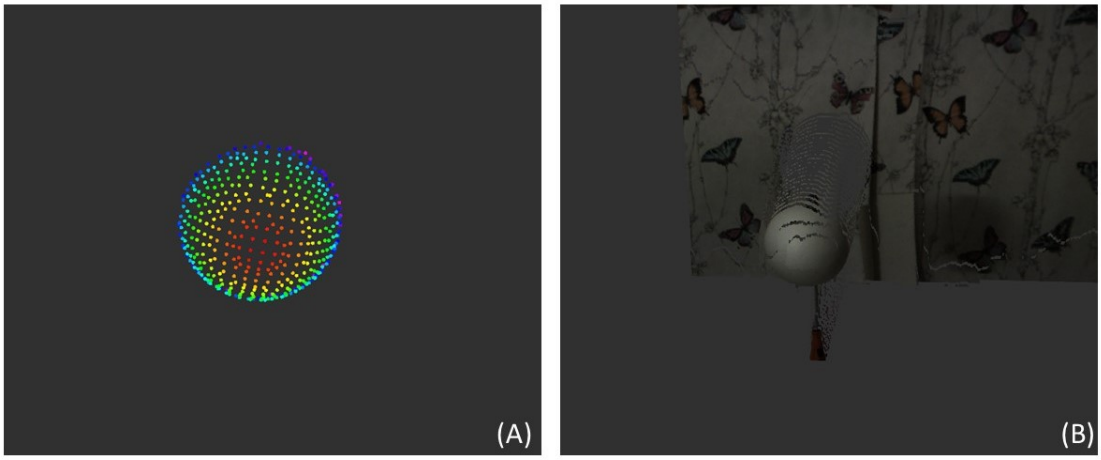

Figure 5-6: (A) Point cloud of the ground truth for a sphere with a diameter of 120 mm and (B) generated point cloud of a sphere with a diameter of 120 mm.

The generated point cloud from the platform is shown in Figure 5-6(B) prior to post-processing to remove the surrounding points that do not belong to the sphere; the post-processing was done using the Point Cloud Library[8] (Rusu and Cousins, 2011). The setup of the experiment is shown in Figure 5-7.

[8] http://pointclouds.org/

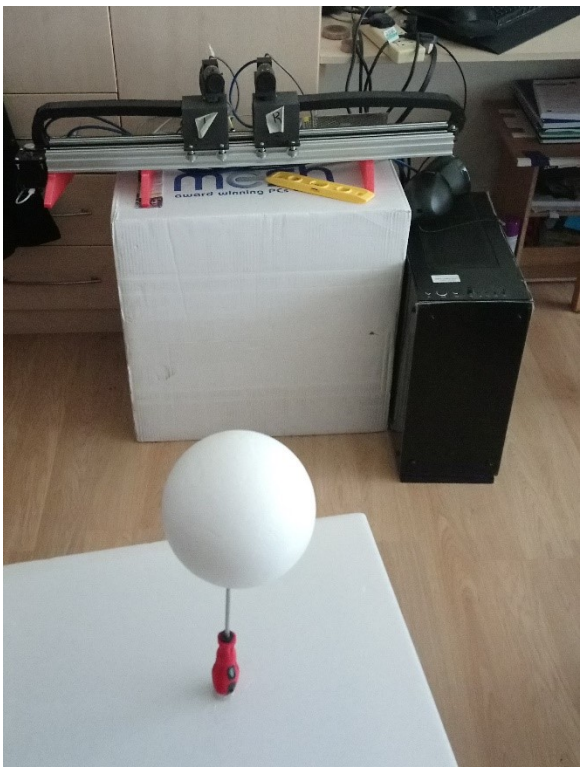

Figure 5-7: The setup for the shape reconstruction using a sphere with a diameter of 120 mm.

The data were collected at different configurations (verge angles from −6° to 12° and baselines from 55 to 250 mm) while the ball was placed at different positions between 1 and 2.5 m from the platform. A set of 10 samples was taken at each configuration.

5.4 Results and discussion

5.4.1 Offline calibration

The results of the offline calibration allow us to understand the geometry of the platform in depth; these data show the tolerance of the manufacturer and the repeatability of the motors. As explained in Section 5.2.5, the only variable axes are the verge angle (yaw) and the baseline (along with the Y-axis), whereas the other axes are fixed, i.e., the pitch and roll angles and translation along the Y- and Z-axes. These should be fixed in the different configurations. The value of the roll and pitch angles are shown in Figure 5-8, where the roll angle is 0.526°, with a margin of error of $\pm 0.047°$, and the pitch angle is $-0.433°$, with a margin of error of $\pm 0.015°$. These two values were generated as a result of the assemble misalignment in the platform and cameras; as a technical note, Flea3

Point Gray cameras (FL3-U3-120S3C-C) have an accuracy of ±0.5° of the sensor assembly. The same points apply to the result of the translation along the Y- and Z-axes (Figure 5-9). As shown in Figure 5-9, the Z-axis reading is 3.6 mm, with a large margin of error of ±2.3 mm, and this was the result of identifying the optical center of the cameras. This leads to an issue with measuring the distance if it is assumed to be fixed. To resolve the error in Z-axis, a relationship was computed from the calibration data to update the Z-axis during the changing of the configuration.

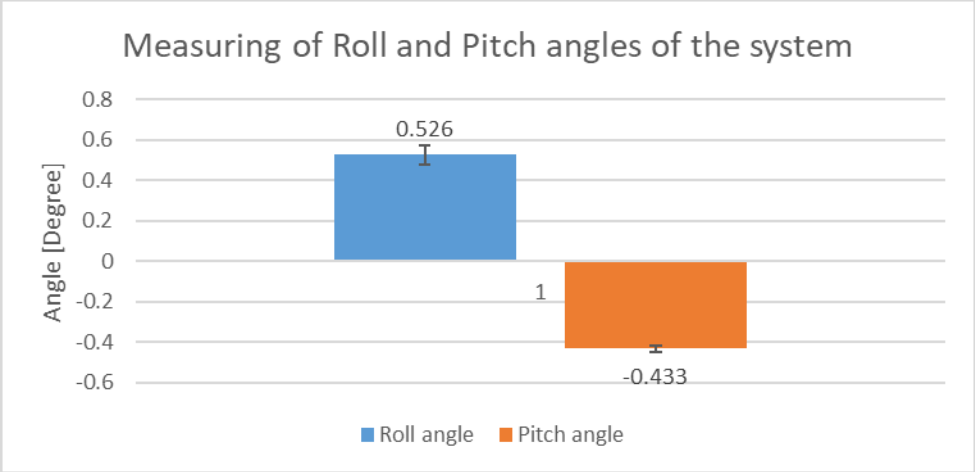

Figure 5-8: The result of the offline calibration process for the roll and pitch angles.

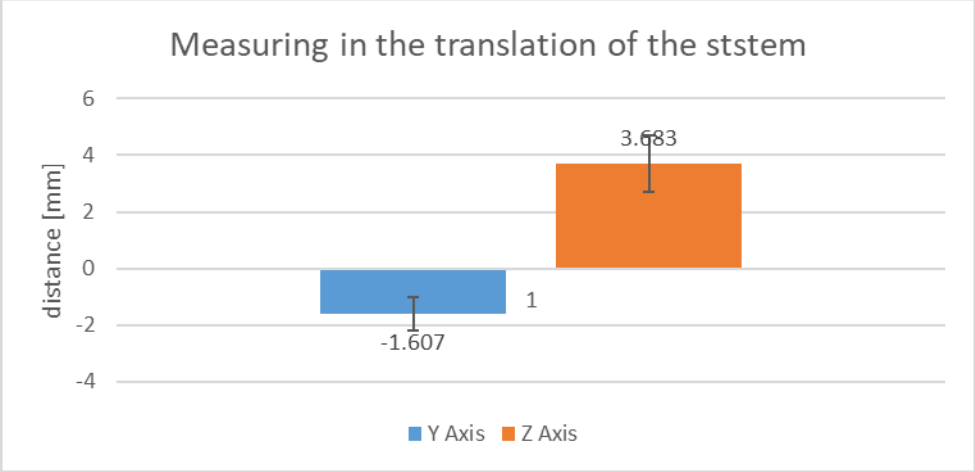

Figure 5-9: The result of the offline calibration process for the translation of the Y- and Z-axes.

Theoretically, the verge angle is directly correlated to the motor angle. After processing the data in the offline calibration, the raw data related to the verge angle were generated and plotted against the sum of the encoder angles (Figure 5-10). As shown in Figure 5-10, the image angle generated by the offline calibration and the encoder angles show a linear

relationship with a coefficient of determination equal to 99.93%. From the data, η is equal to 0.9641, and the error value e is equal to 0.5786. Inserting these values into Eq. (5.20) results in Eq. (5.27):

$$\theta_{img} = 0.5786 + 0.9641 \times \theta_{encoder}. \quad (5.27)$$

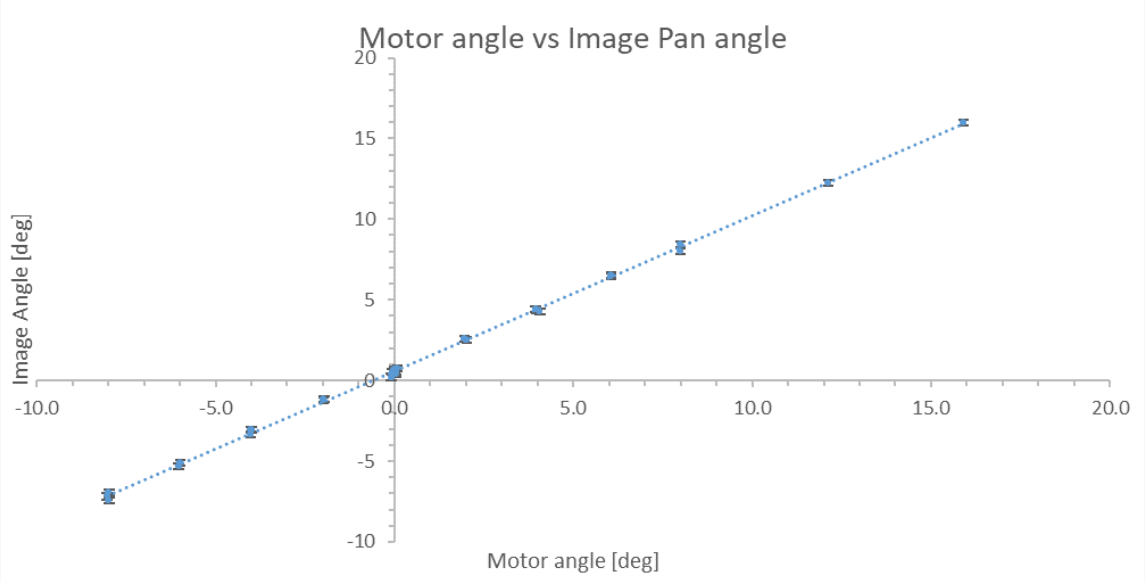

Figure 5-10: The image angle versus the motor angle. The image angle was calculated using the stereo calibration process, and the motor angle was measured using the encoders.

Accordingly, Eq. (5.27) was used to update the image angle by providing the encoder angle reading from the motor. This improved the updates of the geometry of the system. Comparing this result to that of Dankers et al., the epipolar geometry was updated in a more accurate process, which studied the platform in more detail before starting the online update (Dankers et al., 2004). This result will help improve the vision in humanoids, manipulator arms, and mobile robots that use active stereo vision and will extend the working volume of the binocular vision.

5.4.2 Online geometry update

Equation (5.27) was used to calculate the image angle based on the input of the encoder angle; the new image angle was then used to update the essential matrix. This process was done during the online running time, as described in Section 5.2.5. To evaluate the

new algorithm, the projection error was used as described in the experimental section. The result of the projection error is shown in Figure 5-11. This result was collected at different verge angles and baselines, and the experiment was repeated 20 times. In general, the result shows that the platform and the online calibration algorithm have repeatability with a marginal range of ±0.5 pixels, which gives us confidence in the ability of the platform for repeating tasks.

Figure 5-11 indicates that the projection error has a linear relationship with the verge angle when the baseline has a small value, e.g., a baseline of 50 or 100 mm. However, the projection error increases with increasing baseline size. This could be a result of the misalignment in the roll angle, which was set in the opposite direction, or the y displacement misalignment during the manufactures, which increases with the baseline. Moreover, the projection error increases by increasing the diverge angle, and drop when the platform start to verge, the error is not constant; this is due to the position of the target: the images started to overlap, which led to a drop in the error. Figure 5-11 shows that, when the verge angle starts to increase, the projection error starts to decrease, where the target gets close to the horopter. At an angle of 6°, the projection error drops because of the position of the target, which leads to zero disparity. The zero disparity reduces the disparity range and the error in the depth measurement.

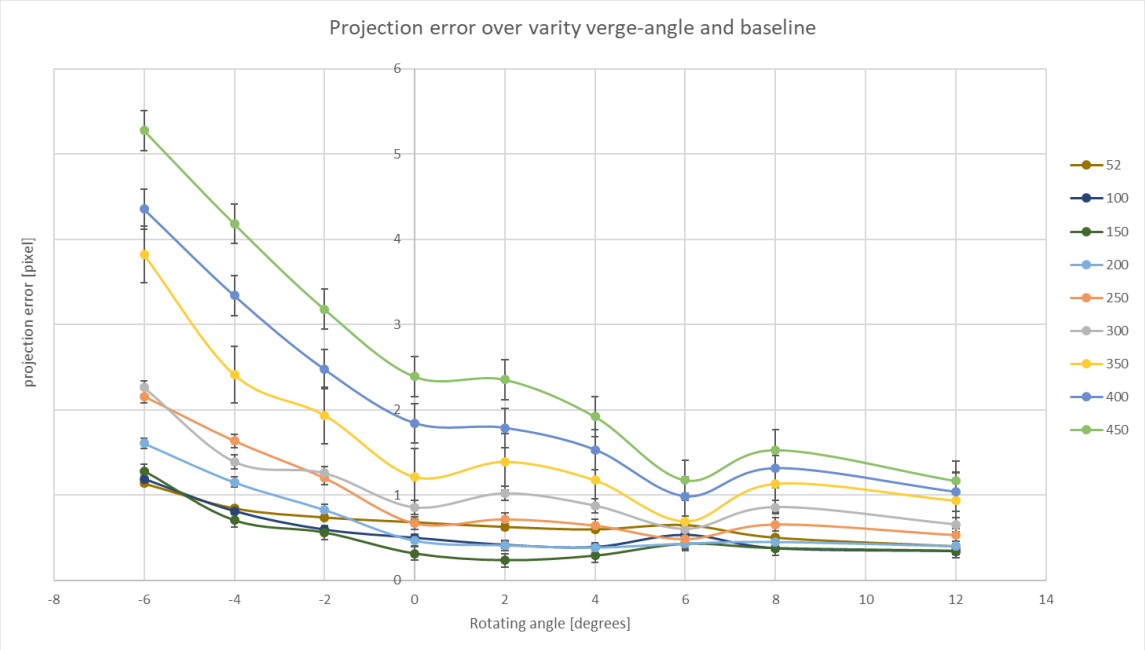

Figure 5-11: Projection error at different verge angles and baselines; the error in the points is ±0.233 pixels.

A list of rectified images captured at different verge angles is shown in Figure 5-12. The colored lines show the epipolar lines where the pixel in the left image is lying on the same line. Figure 5-12(A) was captured at the parallel focal axis, and the rest were taken in 2° increments. This shows that the image sizes decrease with increasing verge angle; the red square represents the image size after rectification.

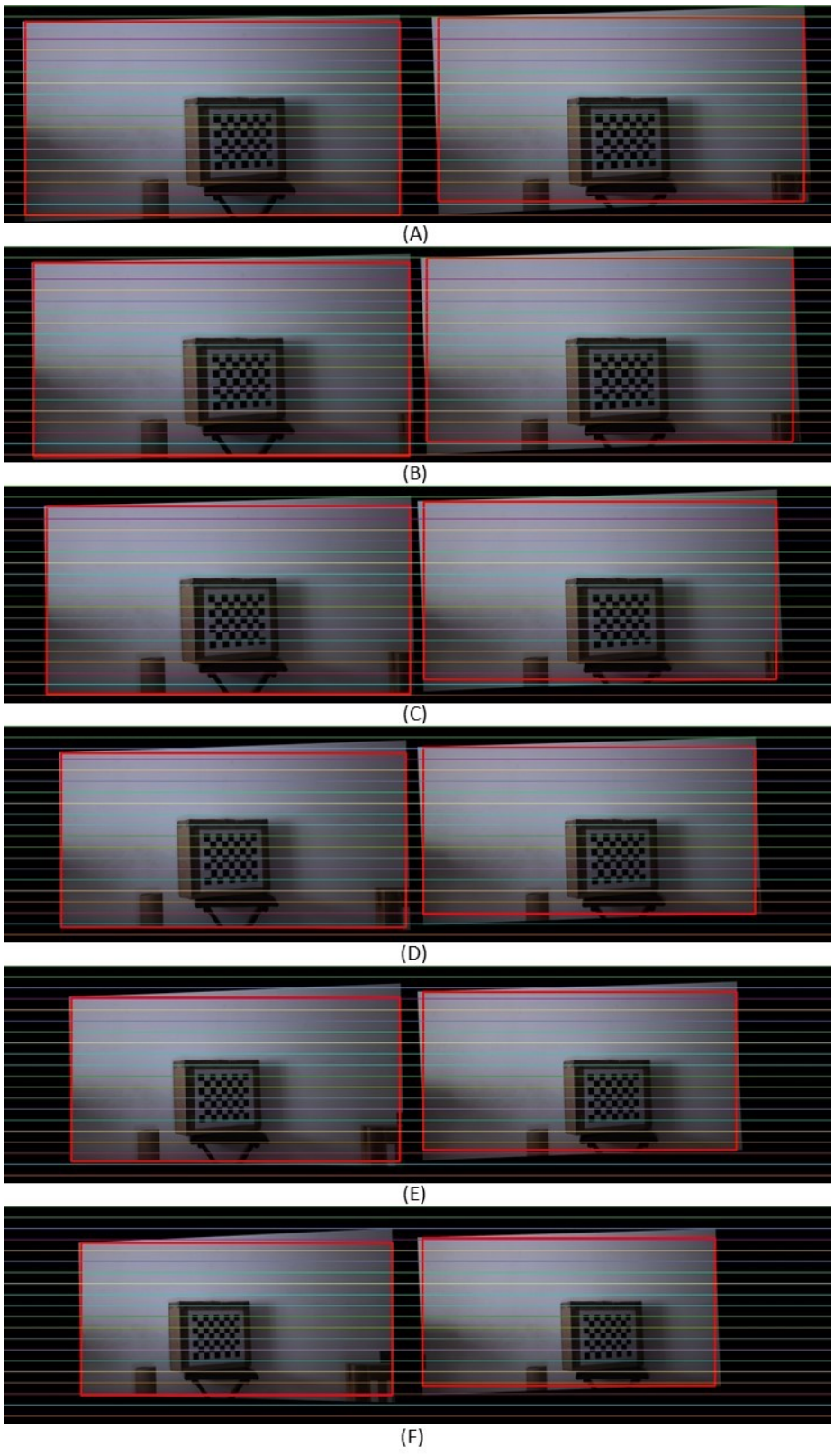

Figure 5-12: Rectified image using the online updated geometry. The lines represent the epipolar lines, and the red square shows the size of the image after rectification: (A) at the parallel focal length, (B) at an angle of 2°, (C) at an angle of 4°, (D) at an angle of 6°, (E) at an angle of 8°, and (F) at an angle of 10°.

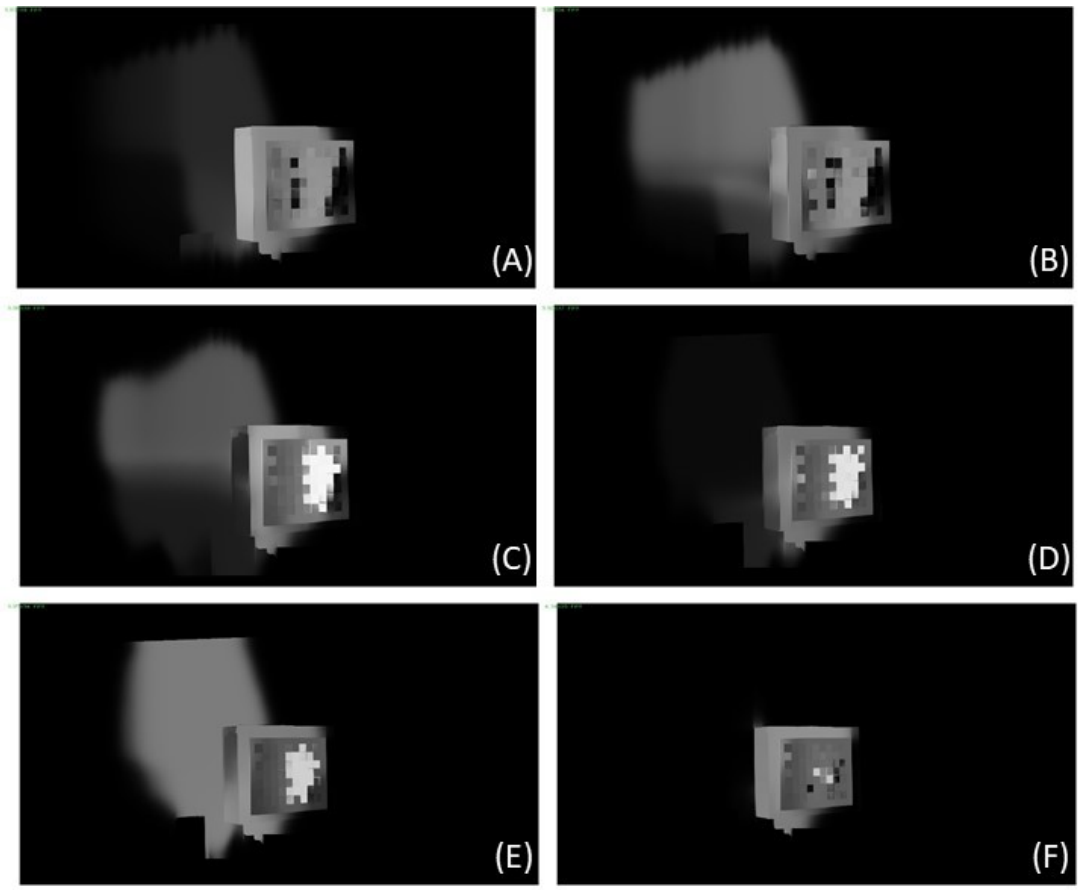

Figure 5-13: Disparity map of a box used to evaluate the projection error: (A) at the parallel focal length, (B) at an angle 2°, (C) at an angle of 4°, (D) at an angle of 6°, (E) at an angle of 8°, and (F) at an angle of 10°.

Figure 5-13 shows the disparity map of the rectified images at different verge angles. The disparity shows the box that was used to evaluate the process. The corresponding process was based on the SGM algorithm with a window size of 5×5 pixels and a disparity number of 256[9]. The size of the windows was selected based on the output of the projection error analysis (Figure 5-11) to cover the potential error in the rectified image. At the same time, windows at this size will sharpen features, as discussed in (Szeliski, 2009). As shown in Figure 5-13, the disparity map becomes more intense with increasing verge angle, where Figure 5-13(F) with an angle of 10° is due to the overlap of the images. Because the disparity map can only provide a visual analysis, the next section generates a point cloud to compare to the ground truth.

[9] Note that OpenCV was used to compute the disparity map. The parameter was tuned after the system update to new configuration. Moreover, OpenCV can deal with Zero Disparity.

5.4.3 Surface compression

To demonstrate the quality of the disparity map, the disparity was converted into a point cloud using the triangulation equations, as described in Section 5.3.3. The ground truth point cloud was generated using a CAD model. A sample of the data used in the comparison is shown in Figure 5-14.

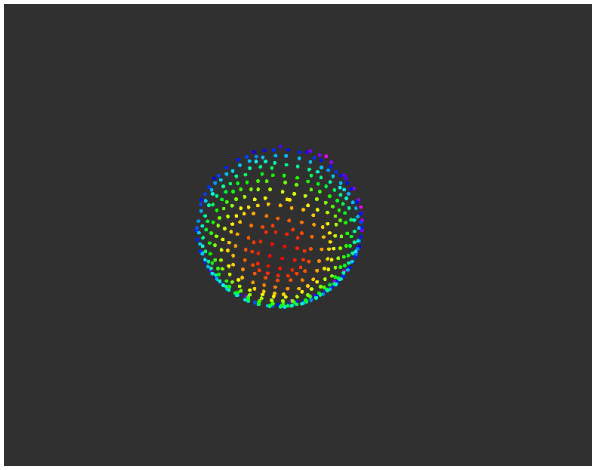

Figure 5-14: A sample of a post-processed point cloud used in the comparison for a 120mm diameter sphere

Figure 5-15–17 show the result of computing the RMS between the ground truth and the sample for three sizes of the sphere (80, 120, and 150 mm). The result describes the sum of the differences of the points from the ground truth. Five different baselines (55, 100, 150, 200, and 250 mm) were used to generate samples at different verge angles (from −6° to 12°) with steps of 2°. The overall result has the same shape as the result of the projection error (Figure 5-11) and shows that the result of the baseline with 100 mm has the lowest RMS and that an increase in the baseline led to an increase in the RMS. The RMS of the baseline with 55 mm has the highest RMS in the three cases because of the proportional error in measuring the depth in relation to the baseline, as described in (Dang et al., 2009).

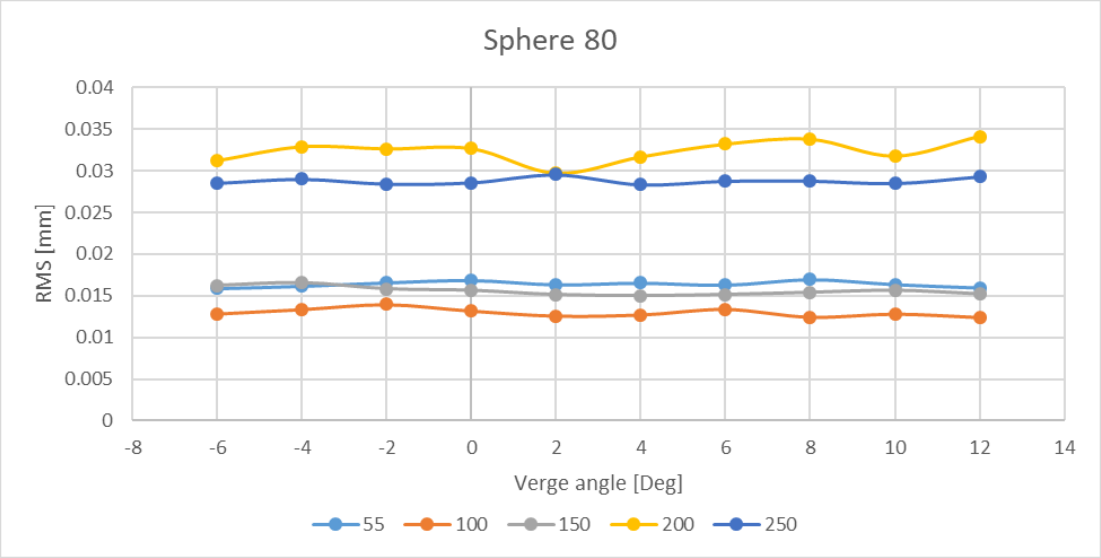

Figure 5-15: RMS error for a sphere with a diameter of 80 mm at different baselines and verge angles.

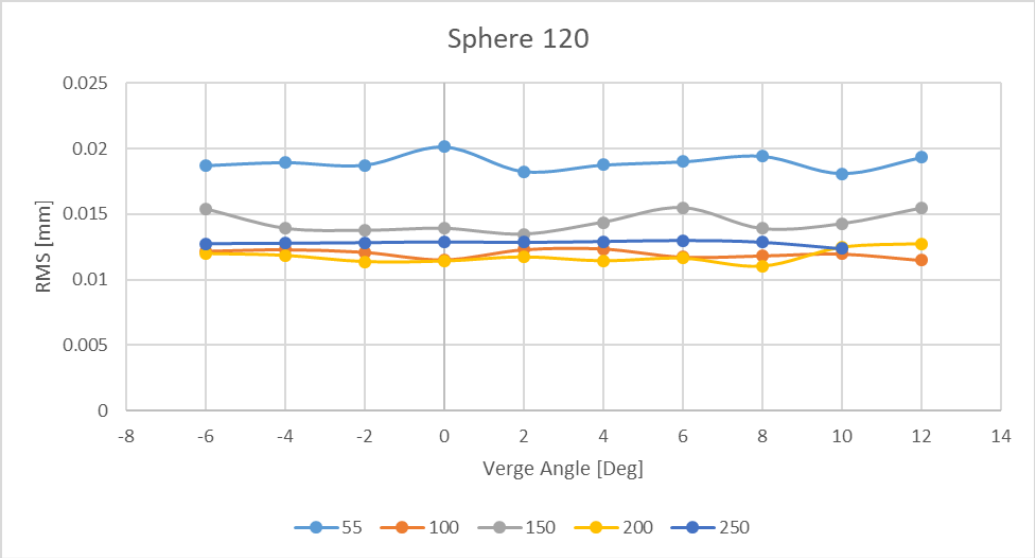

Figure 5-16: RMS error for a sphere with a diameter of 120 mm at different baselines and verge angles.

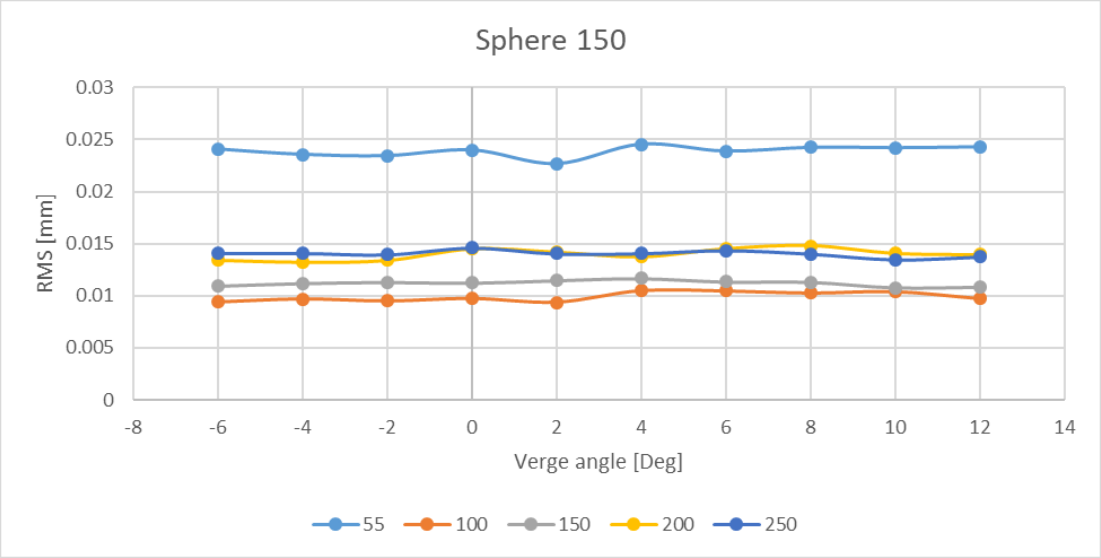

Figure 5-17: RMS error for a sphere with a diameter of 150 mm at different baselines and verge angles.

However, the RMS of the verge angle shows a linear result at different verge angles, with a slight drop in the result at larger verge angles; this is because the overall error in the projection was 4 pixels, and 5 pixels were used to compute the disparity windows to overcome mismatching at the scan line. This result may lead to a misunderstanding in the use of the variable verge angle in computing the disparity if the result shows an approximate equal RMS at different verge angles. However, to generate the sample, post-processing was performed on the sample to reduce the amount of RMS points computed, and as shown in Section 5.4.2, the disparity became smaller when the verge angle increases. Moreover, the measurement of the depth approached the origin of the system, where the parallel focal length of the minimum depth was 1 m, and for the verge angle, the depth converged to 0.5m. However, the size of the sphere does not affect the result of the object reconstruction; all results had an average RMS of approximately 0.02 mm and a margin of error of ±0.0039 mm at a confidence of 95%.

The drawback of this algorithm is that the size of the rectified image generated becomes smaller when the verge angle increases. This occures due to the behavior of epipolar lines at verge angle (Figure 5-18). Moreover, the rectification process makes this line parallel with the baseline; therefore, the new image becomes smaller.

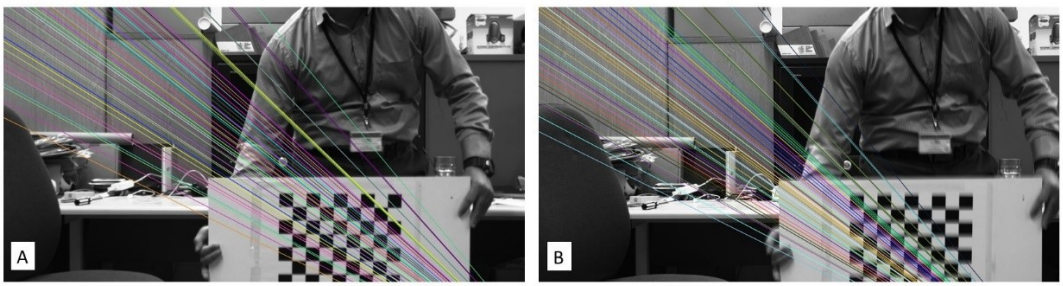

Figure 5-18: Epipolar line before rectification at a verge angle of 8°: (A) left image and (B) right image.

5.5 Conclusions

An active stereo vision platform with three degrees of freedom, providing individual camera pans with a shared variable baseline, was constructed and assessed for its depth resolution and repeatability. A study was performed using both traditional stereo disparity estimations and the camera verge angle to provide depth information.

The problem of computing the epipolar geometry of an active stereo vision system was studied to avoid traditional methods that use feature-based algorithms (Bjorkman and Eklundh, 2002). A relationship was found between the image angle and the encoder angle to update the epipolar geometry of the system directly from the encoder reading.

An offline calibration process was performed to find measurements in the image space of the platform, and then, these measurements were used to find the relationship between the image space and the encoder angle. A linear correlation was found between the image space and encoder angle with a shift of 0.5° in image space. The overall measurement of the epipolar geometry in image space was found using the offline calibration.

In order to evaluate the performance of the rectification algorithm, the projection error based on SSD (Hartley and Zisserman, 2003) was used. The maximum projection error that the platform generates at de-verge is ±5 pixels and when the platform starts to verge the error drop to ± 1.24 pixels at 12°. This compares to ±2.38 pixels in the work of (Hart et al., 2008). This result shows that increases in the baseline increase the projection error, and increases in the verge angle decrease the projection error and the effect of overlapping

between the two images. A drawback of this algorithm is that the size of the new rectified images becomes smaller when the verge angle increases. The maximum verge angle that allowed the image to work with is 20°.

The disparity map depends on the quality of the rectification algorithm which the better the rectification the better the disparity map; therefore, an experiments to evaluate the disparity map was conducted. The disparity maps show good results in different configurations. Point cloud compressions were made with ground truth datasets to evaluate the quality of the shapes. These compressions show that the quality of the shape has an average standard deviation of 0.0142 m and a margin of ±0.0039 m.

Overall, the system improves the quality of the disparity map by controlling the baseline and the verge angle. One of the main advantage of the system is the capability of focus on one target with reconstruct the 3d shape using a small disparity search area. As a result, the system extends the working volume space of robots. Future studies will automate the optimal baseline and verge angle based on the object position to reduce the error.

Chapter 6

Pyramid Normalized Cross-Correlation-based Algorithm

Different from such fixed stereo vision, this chapter proposes a depth estimation method that uses a vergence controller. In the proposed method, a matching feature-based correlation technique is integrated with a Gaussian pyramid to control the slave camera's gaze and integrated with exciting platform (version 2). Here, depth is estimated considering target position by analyzing the external parameters of the system such as the baseline, and the pan angle of the left and right camera. The point where both focal rays intersect is referred to as the fixation point. An algorithm was developed to ensure that both rays intersect at the fixation point. In addition, a fast control system was developed to keep both cameras focused on the same point. We performed experiments to evaluate the proposed method, and the results are compared to those reported by Zhang and Tay (2011).

In addition, the proposed depth estimation method is compared to two stereo vision cameras, i.e., the ZED, which uses traditional stereo correspondence to estimate depth, and the Intel RealSense D415, which uses an infrared projector to estimate depth.

The remainder of this paper is organized as follows. Verge depth mathematics is introduced in Section 6.1. Section 6.2 describes the experimental setup used to evaluate the performance of the proposed algorithm and the implemented platform. Experimental results are presented and discussed in Section 6.36.4, and conclusions and suggestions for future work are given in Section 5.4.

6.1 Background: Verge-based depth vision

Vergence cue methods focus on a point in space where the centers of two images (I_{c_l} and I_{c_r}) are aligned with the interested point of the target, i.e., the fixation point. The angle between the two cameras is the verge angle. Figure 6-1 shows the coordinate frames of the system. The fixed frame, i.e., the of the system, $O_P O_i i = r i = l$-axis, and the distance between the$Y_p Y_p b Y_r Y_l$,..

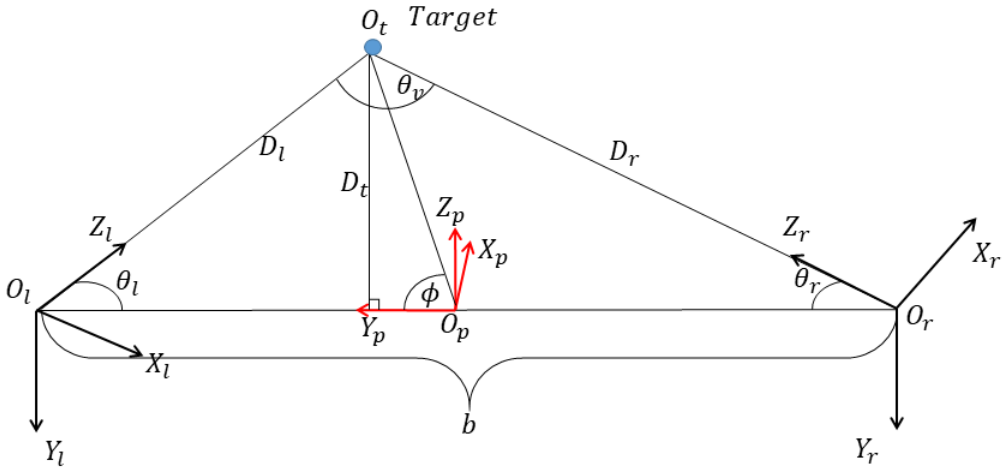

Figure 6-1: Two-and-a-half depth coordinate system.

The geometric rotation between the left and right frames is referred to as an essential matrix E_{lr}, which is a (4×4) homogenous transformation matrix, as described by Hartley and Zisserman (2003). E_{lr} describes the rotation and translation between the left and right frames expressed as follows:

$$E_{lr} = [R \,|T] \tag{6.1}$$

Here, R is the Euler rotation angle of the right frame relative to the left frame, and $T = [\,X\; Y\; Z]^T$ is the translation of the right frame relative to the left frame. Each frame rotates independently; thus, E_{lr} is the product of E_l and E_r calculated from O_p. Note that the left and right frames only rotate around the Y_i-axis; therefore, the only variable rotation matrix used is $R(Y)$, where θ_i is the angle of rotation ($i = r$ or l). $R(Y)$ is expressed as follows:

$$R(Y)_i = \begin{bmatrix} \cos(\theta_i) & 0 & \sin(\theta_i) \\ 0 & 1 & 0 \\ -\sin(\theta_i) & 0 & \cos(\theta_i) \end{bmatrix} \quad (6.2)$$

The essential matrix E_{lr} contains the information required to calculate the distance to the target, i.e., the angles of rotation and the distance between the frames (Cyganek and Siebert, 2009). Assuming that the plane (O_r, O_l, O_t) intersects the target origin, the left origin and right origin, all internal angles (including that of the baseline) can be derived trigonometrically. Therefore, the internal angles can be expressed as follows:

$$\theta_l = 90 - \theta_{external_l} \quad (6.3)$$

$$\theta_r = 90 + \theta_{external_r} \quad (6.4)$$

$$\theta_v = 180 - (\theta_l + \theta_r) \quad (6.5)$$

The sine rule is used to calculate D_l and D_r as follows:

$$D_l = \frac{b \times \sin(\theta_r)}{\sin(\theta_{verge})} \quad (6.6)$$

$$D_r = \frac{b \times \sin(\theta_l)}{\sin(\theta_{verge})} \quad (6.7)$$

Therefore, to calculate the depth of the target on plane (O_p O_l O_r), a line D_t is drawn from the target O_t perpendicular to X_p (Figure 6-1) to form two right angles (left: (O_l O_t O_p); right: (O_r O_t O_p)). Then use Pythagorean theorem, D_t is calculated using Eq. (6.8) where $i = r\,, l.$

$$D_t = D_i \times \sin(\theta_i) \quad (6.8)$$

Now use Pythagorean theorem to calculate Y_t as well, as shown in Eq. (6.9). Here, baseline b is negative when θ_l is used in the calculation and positive when θ_r is used.

$$Y_t = \frac{\pm b}{2} \pm \left(D_i \times \cos(\theta_i)\right) \tag{6.9}$$

Eqs. (6.8) and (6.9) give the position of the target relative to the platform's frame. Note that the sign of the baseline depends on which side is considered to compute Y_t (e.g., for the left angle, $Y_\mathrm{t} = \frac{b}{2} - \left(D_i \times \cos(\theta_i)\right)$).

Note that the above discussion assumes that the target lies on the same plane as the image center. Therefore, using the rotations angle of a tilting motor, the final coordinates of the object can be calculated. In standard coordinates, D_t is the X-axis and $Y_\mathrm{t}$is the Y-axis; therefore, the Z-axis is computed using the rotation around the tilting axis $R_y(\theta)$ (Eq. (6.10)).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos(\theta_\mathrm{y}) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \begin{bmatrix} D \\ Y_t \\ 0 \end{bmatrix} \tag{6.10}$$

6.1.1 Fixation object

The platform includes a master and slave system to keep both cameras' centres focused on the target centroid using coarse-to-fine template matching based on a Gaussian pyramid where the slave tracks the center of the master camera.

Figure 6-2: ArUco pattern refers to number 1.

Typically, the master camera tracks the target using an object detection algorithm or colour threshold technique. In this study, the ArUco detection algorithm, which tracks a specific target (Figure 6-2), is used. ArUco is an OpenCV library for camera pose estimation using squared markers (Garrido-Jurado et al., 2014). The ArUco algorithm was selected to evaluate the system because it demonstrates high precision and fast pattern detection.

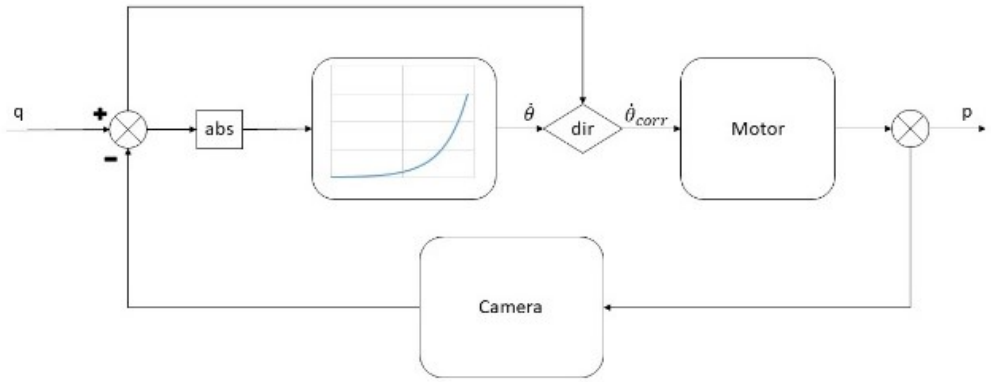

Figure 6-3. Motor controller based on the exponential function.

The vergence controller uses an exponential function controller to align the centroid of the target to the center of the image (Figure 6-3). The algorithm returns the position of the target in image coordinate where the center of the image is where the fixation point is required to be on. This difference is describing how far the object from the center of the image. This difference is the input to the exponential function and the output is angular velocity. The input and the output of the exponential function is always positive therefore, a direction correction function is used to correct the direction of the angular velocity.

$$\dot{\theta} = \exp^{(q \times \lambda)} \times \beta \tag{6.11}$$

Where $\dot{\theta}$ is the angular velocity in rpm, β is control constant that control the range of the output to meet the range of the motor. While λ is the constant describe the shape of the output. q is the input to the exponential function where this has to be always positive. The

direction of the output velocity $\dot{\theta}$ is corrected using the sign of the q before taking the absolute of this value eq.(6.12).

$$\dot{\theta}_{corr} = \frac{q}{|q|} \times \dot{\theta} \tag{6.12}$$

Where $|q|$ is an absolute q always positive.

6.1.2 Coarse-to-fine template-matching algorithm

A template-matching algorithm searches a large image I using a small image template T, where the template represents the target information in the image. Here, the search type is classified as 2D, i.e., the template passes over an image in two directions u and v that represent the x- and y-axes, respectively. The similarity between the template and the region in the image is then computed. Therefore, to determine the position of the template in image I, a cost function is used to estimate similarity between the template and the position of the template in the image, which is stored in matrix M. The size of matrix M is determined by subtracting the size of template T from image I ($I - T$). Depending on the cost function, the location of the template in the image is determined by the smallest or largest value in M. Eq. (6.13) shows a Normalized Cross-Correlation (NCC) differences cost function where the large image is (u, v) and the size of the template is (m, n). Using this cost function, the best location for the template is determined by the largest value in M.

$$NCC(u, v) = \frac{\sum_{m,n}(T(m,n) \times I(u+m, v+n))}{\sqrt{\sum_{m,n} T(m,n)^2 \times \sum_{m,n} I(u+m, v+n)^2}} \tag{6.13}$$

NCC methods were selected to compute matching because NCC removes large differences between the template and image around a large brightness (Bradski and Kaehler, 2008).

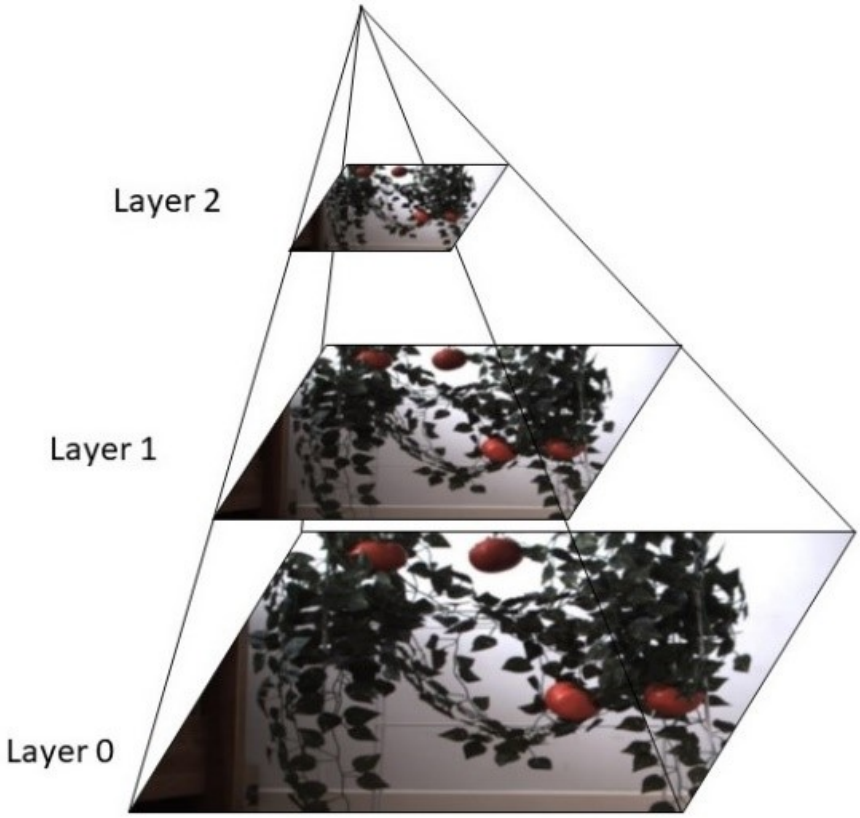

Figure 6-4. A three-level Gaussian pyramid.

In the proposed platform, the template is a window with a small number of pixels cropped from the center of the master image (100x100). A coarse-to-fine Gaussian pyramid is constructed to improve tracking quality and increase the speed of the template-matching algorithm by reducing the number of unnecessary computations (Figure 6-4). The Gaussian pyramid algorithm produces a sequence of images by down-sampling, where the image size and resolution are reduced at every level (I^{Lth}) (Fouda and Ragab, 2013). Here, the Gaussian pyramid produces images that are half the size of the level under the current level ($I^{L+i} = \frac{I^L}{2}$).

Note that Gaussian pyramids are calculated for both the template and image. The cross-correlation algorithm implemented at the top level I^L (coarse) uses both the template and image. A threshold is applied to the output M^L. M^L is up-sampled by a factor of 2, and a counter is applied to find the region with large differences $C^L = (x,\ y,\ width,\ height)$.

The search in level I^{L-1} is constrained to region C^L, which covers the neighborhood pixels around the maximum likeliness found in the previous level. In each level, the search is reduced to the highest matching feature until the search reaches the base of the pyramid I^0,which represents the finest resolution image.

6.1.3 Sensitivity of depth measurement to erroneous system assembly and calibration

In practical applications, various parameters must be considered to generate accurate output. These parameters can have significant effect on the system's output depending on their contribution to the overall error. Kanatani (Kanatani, 2005) identified four factors that influence the system's final output, i.e., (1) accuracy bound, (2) reliability evaluation, (3) computational efficiency and (4) model plausibility.

In a binocular vision system, different parameters contribute to depth measurement errors. A detailed discussion of these parameters can be found in the literature (Dang et al., 2009; Kanatani, 2005). Such parameters vary relatively to the extent to which they increase the number of errors in the depth measurement. In practical vergence vision applications, the verge angle depends on the fixation point and the gaze of slave camera having the same fixation point as the master camera. Computing the depth depends on the verge angle, which, in turn, depends on encoder reading, baseline measurement and pixel size. Note that these values relate to the geometry of the platform and its manufacturing process. The accuracy of computing the fixation point depends on the computational efficiency and the algorithm used to compute the fixation point.

In the following, we provide a geometrical analysis of the system to evaluate parameters contribution to error. By solving Eq. (6.8) using the left angle is as follows:

$$D_t = \frac{b \sin(2\theta_l)}{2 \sin(\theta_v)} \tag{6.14}$$

The error in b are related to the error in various measurement quantities, thus allowing for an error in b as $b = \ \mathrm{b}_0 \ \pm \delta \mathrm{b}$

$$D_t = \frac{(b_0 \pm \delta b) \times \sin(2\theta_l)}{2\sin(\theta_v)} \tag{6.15}$$

The result of depth D_t will be affected owing to the errors in Equation (5.15). Thus, we write $D_t = \ D_{t_0} \ \pm \delta D_t$. A Taylor expansion is applied to solve the depth error δD_t in Equation (5.15):

$$\delta D_t = \frac{b_0^2 \times \sin(2\theta_l)}{2\sin(\theta_v)} \ \delta b \tag{6.16}$$

Now solving Eq. (6.14) to b, then replacing the output with b in Eq.(5.16) we derive

$$\delta D_t = \frac{D_t^2 \sin(2\theta_l)^3}{8\sin(\theta_v)^3} \ \delta b \tag{6.17}$$

Eq. (6.17) shows that depth error is proportional to the square of the depth. However, compared to orthogonal stereo vision, the depth error in a verge system is multiplied by eight. The same applies to Eq. (6.9), which shows that the size of the baseline does not affect error in the Y_t measurement. Here, the verge angle is inversely proportional to the error in measuring Y_t.

To clarify the effect when a verge angle error occurs in the system, we differentiate Eq. (6.14) relative to verge angle θ_v. The output is shown in Eq. (6.18).

$$\delta D_t = \frac{b}{2}\sin(2\theta_l)\,(-\csc(\theta_v))\cot(\theta_v)\,\delta\theta_v \tag{6.18}$$

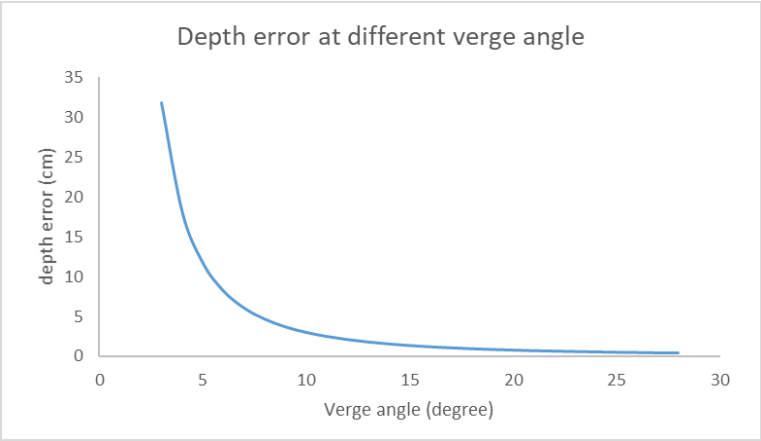

Figure 6-5. Depth error at different verge angles (smaller angles give greater depth).

Increasing the verge angle reduces the depth error δD_t where the verge angle is inversely proportional to the square depth error. The verge angle is the sum of θ_r and θ_l; therefore, both angles contribute to the accuracy of the final depth output. Figure 6-5 shows the theoretical error generated by the verge angle in a worst-case scenario for the proposed system where the encoder resolution is 12 bits (equal to 0.08°).

The depth error δD_t shown in Figure 6-5 was computed using the maximum error generated using the given encoders. The profile of the error curve exactly follows the curve generated by the analytical error (Eq. (6.18)) (Sahabi and Basu, 1996). The results of both verge angle and depth relative to depth error demonstrate that the size of the error increases significantly as the measured depth increases, which is consistent with the result obtained when a disparity calculation is used in an orthogonal stereo vision.

Disparity maps employ a multi-stage technique that introduces errors into the process, including the pixel matching process. A fixed stereo vision system introduces errors from different parameters that significantly affect the final output, such as misalignment of the y-axis of both cameras, camera rotation and the size of the pixels. Based on a 3D reconstruction of the target, Kanatani (2005) discussed errors in stereo vision systems and how they affect performance. A fixed stereo vision system depends on the quality of the calibration output, where internal and external parameters are computed for use in a

rectification process. Error analysis of standard stereo vision systems can be classified as the effect on normalized image coordinates, the effect on pixel coordinates and the effect on 3D reconstruction (Dang et al., 2009) (see appendix D). In vergence stereo systems, depth errors are generated by the external geometry (i.e., the assembly and encoder quality) and the effectiveness of the tracking algorithm.

The source of errors in vergence and orthogonal stereo vision systems are compared in Table 6-1. As can be seen, most of the parameters of orthogonal systems do not contribute to errors in vergence systems. Appendix D contains a brief error analysis for an orthogonal stereo vision.

Table 6-1: Error sources in orthogonal stereo vision vergence vision systems

Error Source	Orthogonal Stereo Vision	Vergence Vision
Baseline	$\delta D \propto \frac{D^2}{f * b}\delta d$	$\delta D_t \propto \frac{D_t^2 \times \sin(2 \times \theta_l)^3}{8 \times \sin(\theta_v)^3}\ \delta b$
Verge angle	$\delta D\ \alpha \frac{D^2}{b}\tilde{x}_L\ \tilde{y}_L \delta\theta_v$	$\delta D_t\ \alpha\ -\csc(\theta_v)\cot(\theta_v)\,\delta\theta_v$
Tilting angle	$\frac{\delta Z}{\delta \Theta_l} \approx \frac{Z^2}{b}\tilde{x}_l\tilde{y}_l$	None
Focal length	$\frac{\delta Z}{\delta f} \approx \frac{Z}{f}$	None

Pixel size also contributes to the performance of the vergence controller, where smaller pixel size results in more accurate target centroid detection. In fixed-camera stereo vision systems, pixel size contributes to the size of the depth error when computing disparity. While active stereo vision depends on the precise of location of the centroid of the target.

6.1.4 Motor controller-based exponential function

The control system used to move the pan and tilt motors are based on an exponential function. The control system comprises a motor and a camera that is used as a feedback sensor.

$$\dot{\theta} = \exp^{(q \times \lambda)} \times \beta \tag{6.19}$$

A block diagram of the motor controller is shown in Figure 6-3. Here, the camera generates the position of the target in pixel. The input to the motor is the angular velocity $\dot{\theta}$ in rpm, which is computed using the exponential function (Eq. (6.19)). The motor controller and the selecting of the parameter is presented in chapter 4.

6.2 Experiments

Quantitative and qualitative experiments using a five degree of freedom active stereo vision platform (Figure 6-6) were conducted to evaluate the proposed algorithm.

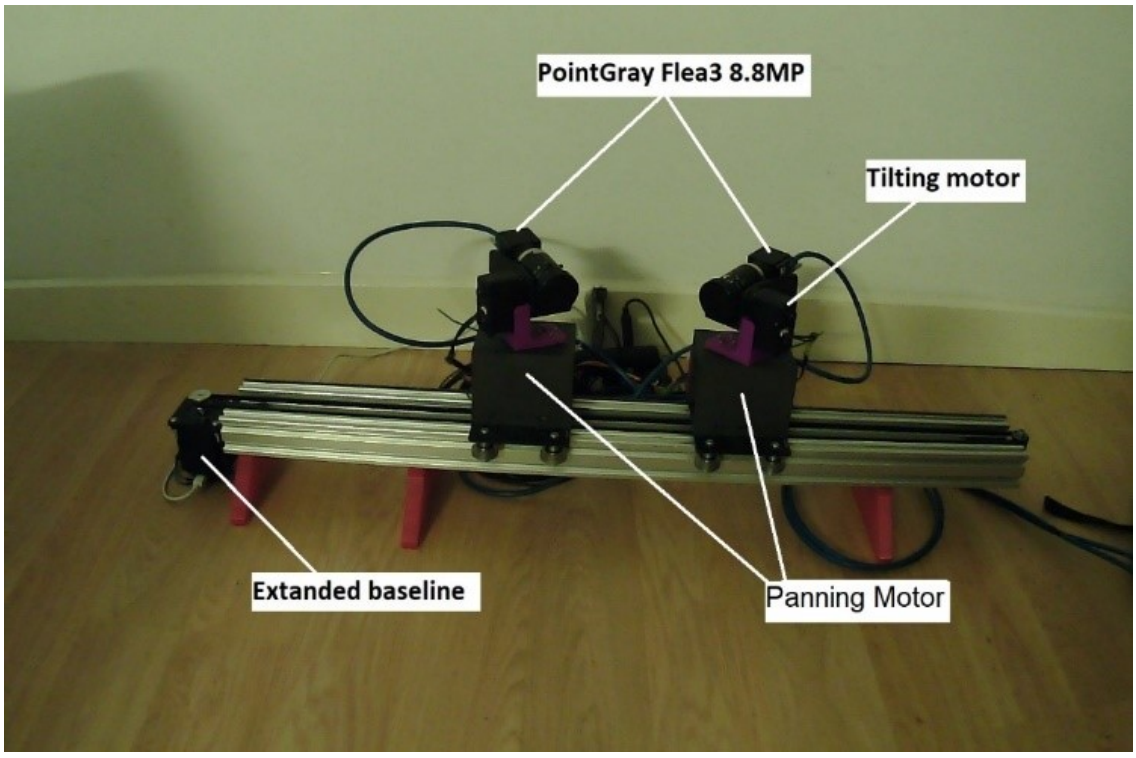

Figure 6-6. A stereo vision platform.

6.2.1 Pyramid NCC experiment

Figure 6-7. A template for in-depth measurement with 16 targets at different heights.

The first experiment involved a qualitative test where a 40×40 cm template comprising 16 targets at different heights and positions with a center-to-center distance of 10 ± 0.1 cm was constructed (Figure 6-7). The targets were numbered from 1 to 16 according to the ArUco numbering system. The numbering started at the top left corner and ended at the bottom right corner. The origin of the template was marked as number 18. It is critical to allocate the center of the template because the targets' coordinates can only be obtained if the origin is known. As shown in Figure 6-7, the 16 targets are somewhat similar in terms of their intensity, and this similarity confuses the algorithm, which justifies using the vergence controller to track each target individually to determine how accurately the system verges toward the targets.

6.2.2 Unbalance brightness Conditions

Figure 6-8: A tomato setup used in evaluating the performance of the platform.

The second experiment was a qualitative and quantitative test that used an artificial setup for tomato fruit detection to evaluate the pyramid NCC (PNCC) algorithm and the platform (Figure 6-8) to test the performance of the system under different lighting inputs between the master and slave cameras and depth measurements. This experiment involved three different configurations, where the slave camera's IRIS status was fully open in all configurations. Here, the difference was only in the state of the master camera's IRIS, i.e., fully opened, one-quarter opened and half opened. There were four targets in these configurations with different depths and positions, where the master camera was manually fixed on the target. Then, the vergence controller algorithm was executed for the slave camera to verge on the target, whose depth was recorded. The position of the tomato was set within the calibrated range of the platform (minimum to maximum range: 40–200 cm). Note that the position of the tomato was determined using a measuring tape.

6.2.3 Depth estimation experiment

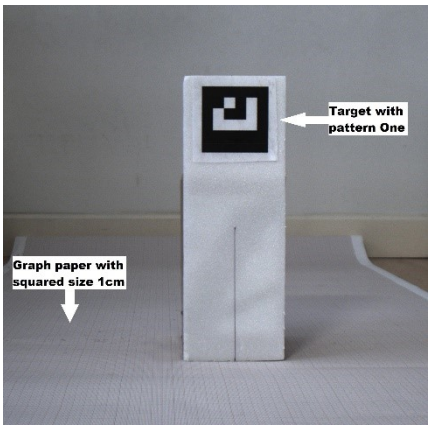

Figure 6-9. ArUco pattern on a calibrated paper.

The third experiment was designed to evaluate the accuracy of the depth estimation of the algorithm implemented in the active stereo vision platform by setting the baseline to four values (10, 20, 30 and 40 cm) to evaluate the effect of baseline length on the depth measurement. Figure 6-9 shows the pattern attached to a stand. The pattern was placed at various depths (40–200 cm at 20 cm intervals). The pattern was placed on various X-axes to justify the accuracy of the platform. The measurement was repeated 10 times for each interval, where the starting point of the slave camera was adjusted to the zero position during each measurement. The purpose of resetting the starting point of the camera to the zero position was to evaluate the performance and repeatability of the vergence controlller to verge on the fixation point. Consequently, the experiment was repeated 360 times to accommodate all baselines and interval conditions.

6.2.4 Performance comparison

The experimental system was compared to the work of Zhang and Tay (2011), in which the image resolution is 200×200 pixels with a three-level pyramid (baseline: 24 cm). Note that we used the same configuration for our system to compare the algorithms.

The accuracy and repeatability of the platform was tested by comparing the compression of the proposed system, the ZED stereo camera (Stereolabs Inc. San Francisco, CA, USA)

and the Intel RealSense Depth Camera D415 (Intel Corp., Santa Clara, CA, USA). In this evaluation, the same depth configuration is used as the setup described for the depth estimation experiment.

6.2.5 Small object depth detection

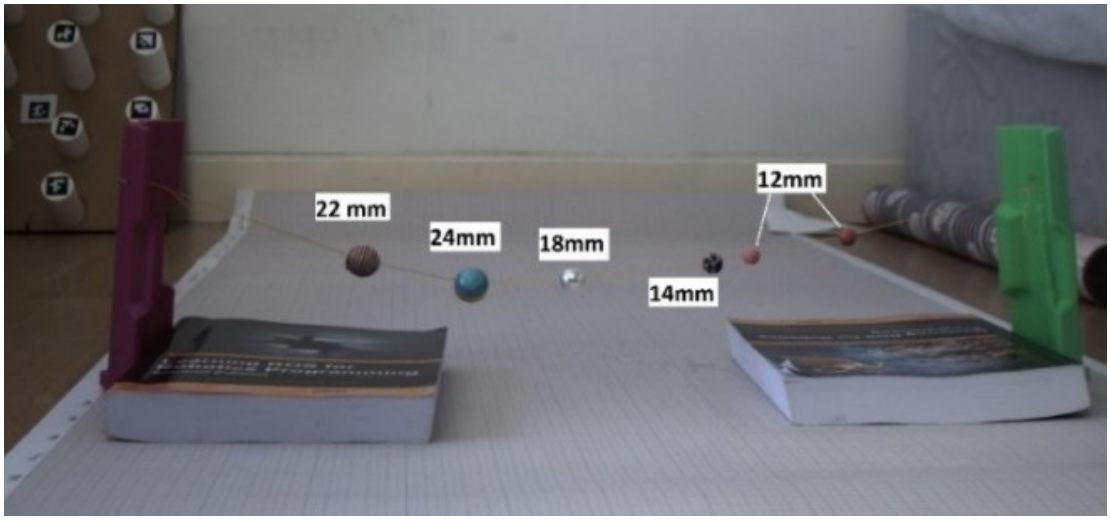

Figure 6-10. Estimating depth of a small object (distance: 150 cm).

Figure 6-10 shows the configuration of the fourth experiment. The main purpose of this experiment was to compare the performance of the three sensors (ZED, Intel RealSense D415, and our system) by detecting the depth of six small targets (1.2–2.4 cm). These objects were placed 150 cm away from the sensors. Then, the estimated depth and ability to measure the depth were recorded.

6.2.6 Field experiments

Figure 6-11: The greenhouse experiment setup.

Finally, the system is tested in the field with a real tomato in a greenhouse. Where the system place in front of the tomato trees (Figure 6-11). Two scenes were set to test the vergence controller the first one with four target at distance between 100 and 105 cm and the second scene with six targets at distance between 85cm and 95 cm that make the scene more crowded (Figure 6-12). Gaze on the tomato was done manually, by selecting the target in the screen then the gaze controller fixes the centroid of the target to the fixation point of the system. After the master camera fully gazed on the target the vergence controller turn on to verge on the targets. The setup of the vergence controller is full resolution image (2080×1040), the template size was set to 150×150 and the pyramid layer set to seven layers.

Figure 6-12: two scenes used in testing the vergence controller and platform. (a) Four tomatoes setup (100 – 105cm) (b) Six tomatoes setup (80 – 95 cm).

6.3 Results and Discussion

6.3.1 PNCC results

Here, the results of the vergence controller algorithm are presented and discussed. Using the PNCC algorithm to verge the slave camera to the fixation point of the master camera has many advantages, where the correlation between both images became more robust and accurate under different lighting conditions and similar repeatable features in the scene.

The template shown in Figure 6-7 was used to evaluate the reliability of the algorithm. Note that the template has a pattern that can deceive the algorithm. Moreover, some template-matching algorithms, such as SAD and SSD, are sensitive to illumination changes (Gräßl el at., 2003); therefore, the experiment included changing the illumination input to the master camera by controlling the IRIS.

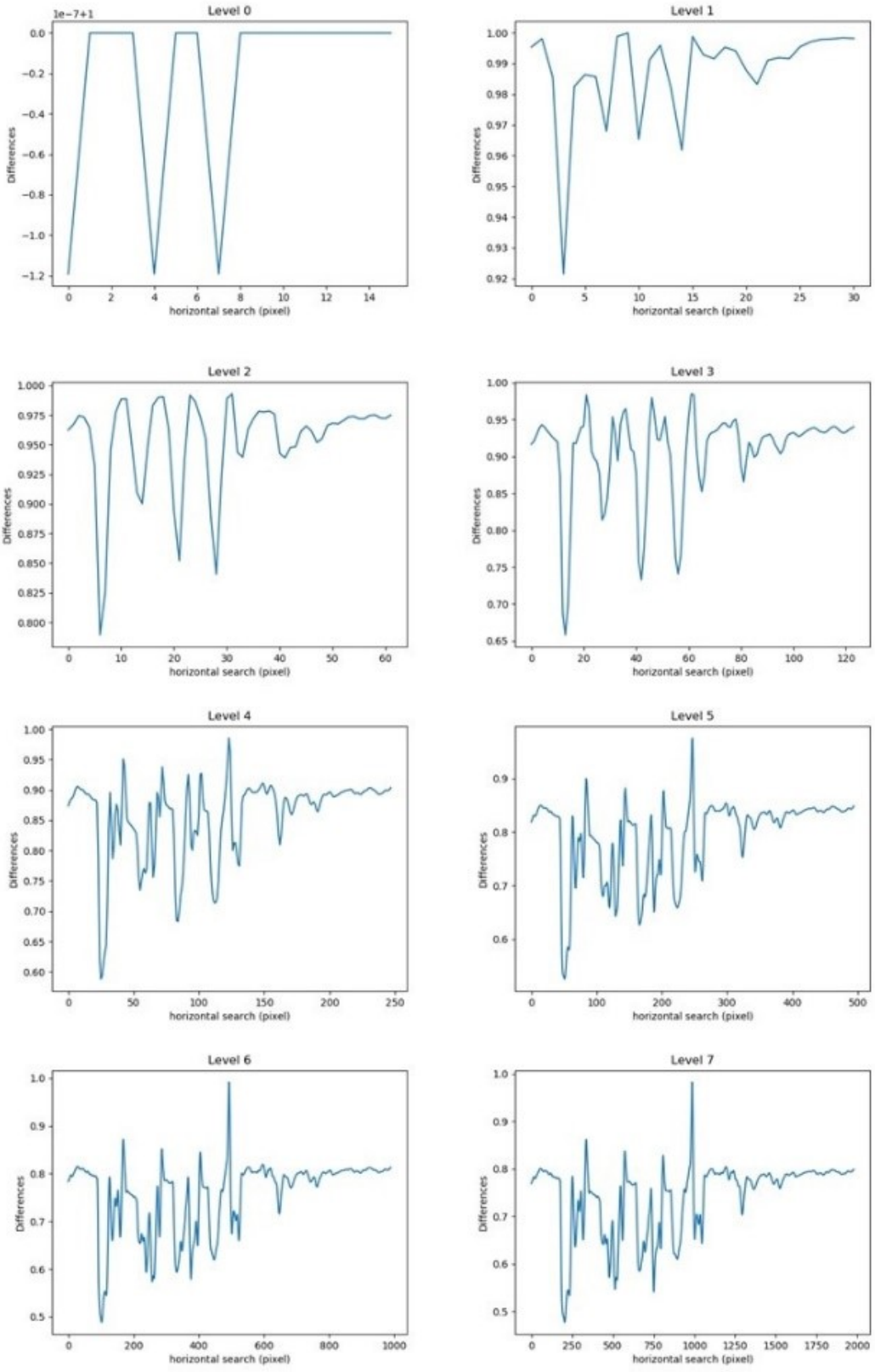

Figure 6-13. Output of PNCC algorithm at target 4 in the template (target depth: 1.5 m).

The results shown in Figure 6-13 are the output of the verged slave camera on target four obtained using the PNCC algorithm, where the process began at level zero and increased

to level seven (image sizes of 16×8 and 2048×1080). Note that the output levels are smaller than the actual image due to spatial convolution windowing. The results show how the pyramid algorithm helps fix the slave camera's gaze by referencing the master camera's fixation point. The pyramid algorithm was used to focus the correlation on the target at coarser levels (levels 0–4) and control the precision of the output using finer levels (levels 5–7). The results are shown in Figure 6-14. The template has multiple targets that are similar in terms of shape, which confuses standard NCC relative to determining the target. The final verge on the fixation point has an error of ±10 pixels due to the controller's behavior. This error may occur due to the resolution of the motors, which is 12-bit (±0.088°) for pan and 10-bit (±0.29°) for tilt. Consequently, this may affect depth estimation performance slightly despite the fact that the cameras have a high resolution (4.4 MP) that help in improving the align the center of the target.

As shown in Figure 6-13, there are four peaks that demonstrate sharp rises through the levels, and, in level 7, these peaks are (300, 0.86), (572, 0.83), (807, 0.82) and (986, 0.98) from left to right, respectively. These peaks are the four patterns in the template with nearly equal intensity, and these patterns lie on the same horizontal line. Note that this occurs for nearly all patterns in the template. In some cases, the system verges on the pattern with high error in the final fixation point, and this is due to differences in the perspective view of the master and slave cameras at large angles. This error depends on the shape of the target, e.g., a cube target will have different shadows at different side, which consequently leads to increase the error during verge process.

Another issue that occurs when using the PNCC algorithm is the selection of the template size. The size of the template determines the precision of the target's position in the slave camera. A large template leads to a slight shift in the position due to extra neighbor pixels, which results in over-computation of the target's features. This also occurs if the template

is small as there is insufficient information to consider during the search, which leads to poor verge on the target. The target size in the image varies depending on its position in front of the camera, i.e., distant targets are smaller and vice versa. Generally, to realize accurate vergence controller, the window size should be adaptive to the size of the target, which is determined by the number of the pixels. Therefore, controlling window sizes will be the focus of future research.

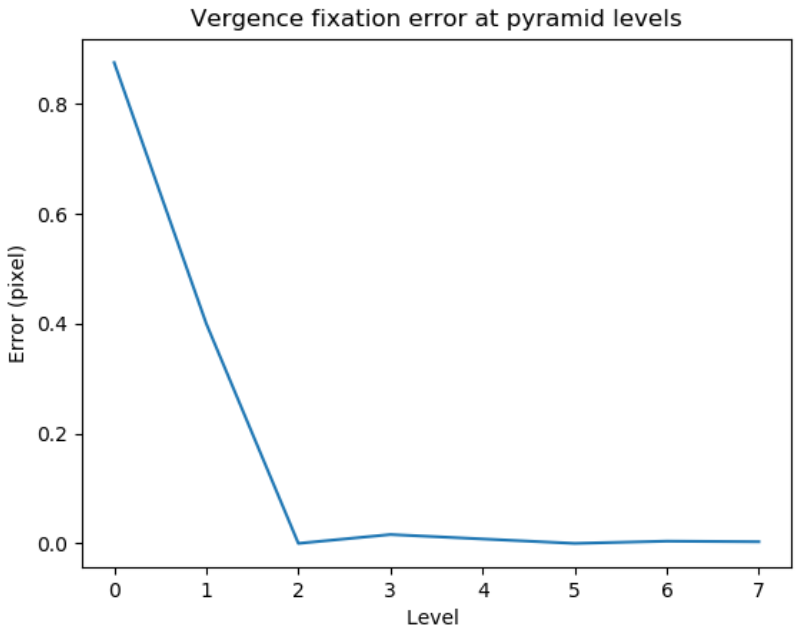

Figure 6-14. Error in pyramid levels when the system is fully verged on the fixation point (pattern number 4).

One disadvantage of using a pyramid in template matching is that the target search process depends on the course levels of the pyramid (i.e., the top), where, if the output location of the target at the course level is in the wrong position, the rest of the search in subsequent layers will be in this incorrect region. Typically, this occurs when testing the system in a busy environment. Furthermore, the system fails when both cameras rotate more than 35°, where the perspective of both views become large which leads to failure in vergence controller. In some cases, the vergence controller tracking is overcome if the object is spherical.

6.3.2 Unbalance lightning conditions for Master and Slave

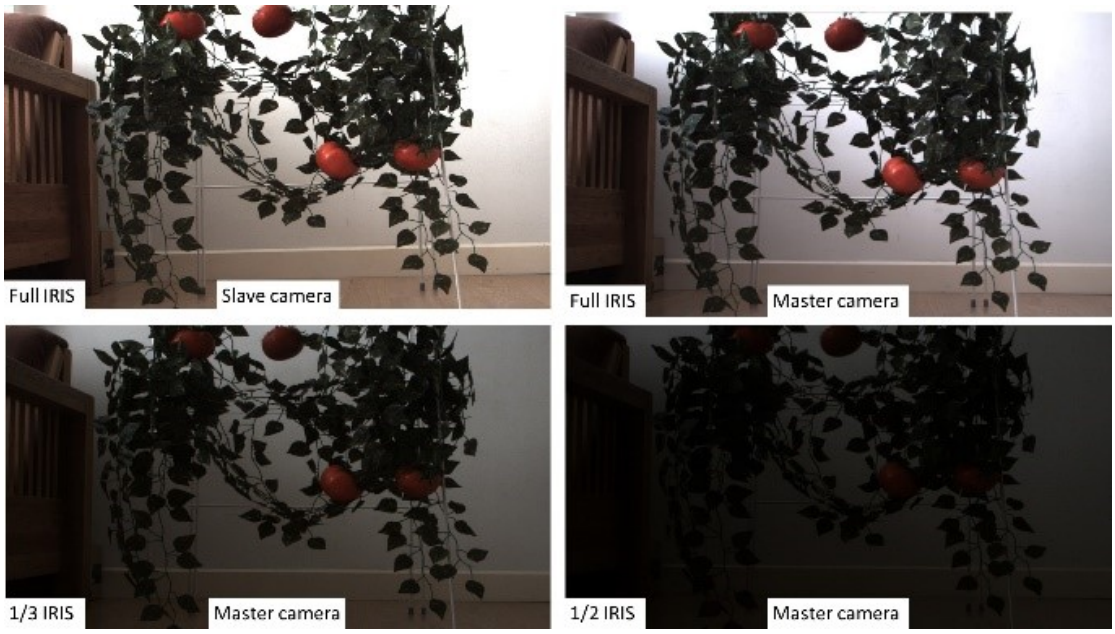

Figure 6-15. Different Brightness between the master and slave images.

In this experiment, the PNCC algorithm was tested against changing brightness levels between the master and slave cameras. Figure 6-15 shows the three lighting conditions input to the master camera (lighting conditions for the slave camera were unchanged).

This setup helps identify the robustness of the algorithm under different lighting condition, e.g., if one camera faces the light source and the other camera light source is behind the other camera, the camera facing the light source will have a darker image due to limitations in camera dynamic range.

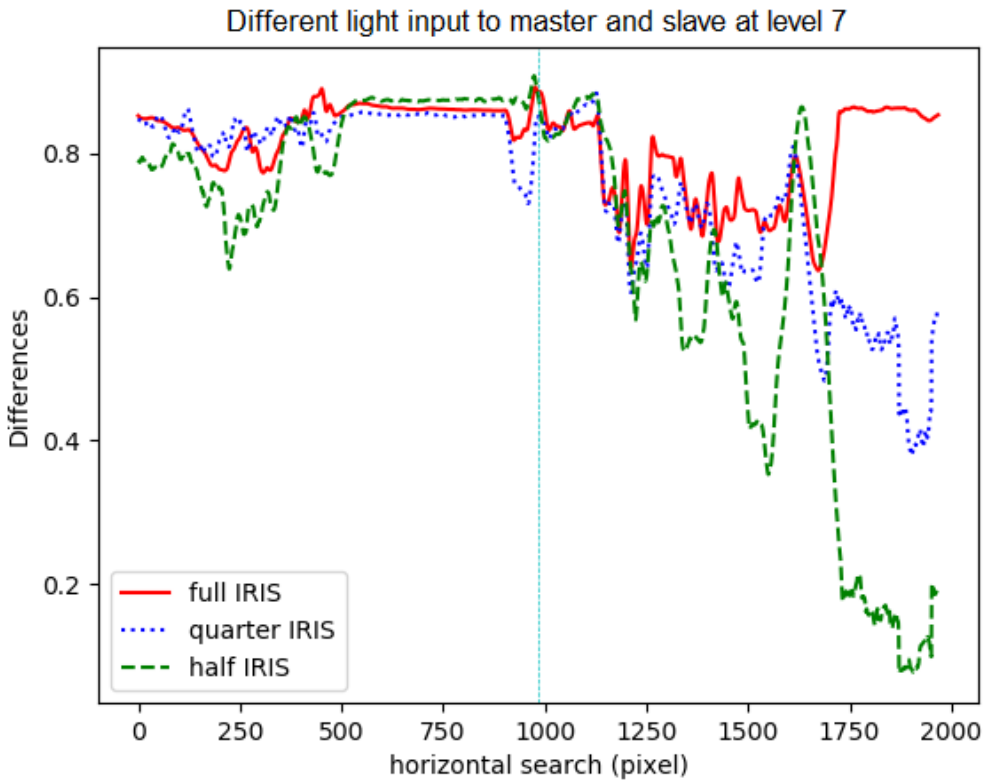

Figure 6-16. Verge at target under three lighting conditions by controlling camera IRIS.

Figure 6-16 shows the output of level 7 when the system verged on the target under different lighting conditions for the master camera. Here, the vertical dashed line represents the center of the image (where the target should be located). The error was calculated by measuring the difference between the centers to the peak of the PNCC output. The error was 0.71%, 1.3% and 1.9% for the equal, one-quarter and one-half IRIS configurations between the master and slave cameras, respectively. These errors could be due to the arithmetic precision of the PNCC calculations ($\pm$ 1 pixel), wherein some steps are required to use an integer rather than a float. Alternatively, these errors could be due to the controller margin error ($\pm$10 pixel) that was set to stop the isolating. Considering the camera used in this work (i.e. 12-bit which has over 4,000 tones) the image processing algorithm will not lead to large error.

These results demonstrate promising performance in a practical application. The results of this experiment are significant for computer vision and robotics fields because, in

practical application, many noise sources exist, and it is difficult and expensive to control all of them, especially in computer vision implementations.

6.3.3 Depth estimation results

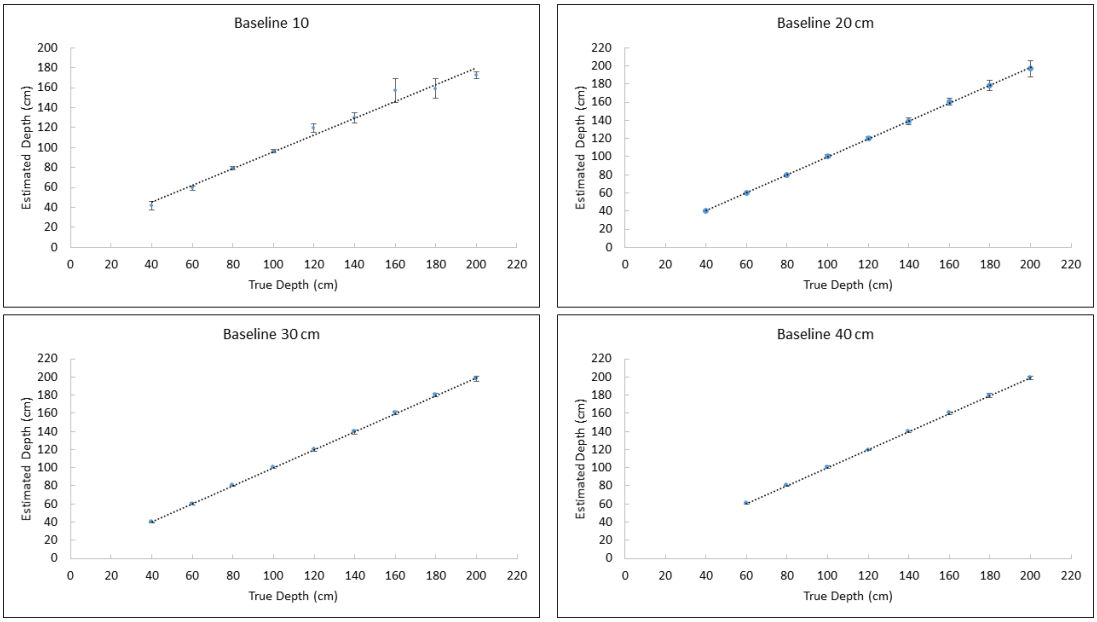

Figure 6-17. Vergedepth versus true depth for four baselines.

The accuracy of the vergence controller depth is influenced by the accuracy of the mechanical joints and encoder resolution of the platform. A quantitative measurement was performed to evaluate the accuracy of the system. Figure 6-17 shows the depth measurements results with the standard deviation (error bar) for various baselines. In Section 6.1.3, the behaviour of the measurement was explained theoretically, and the experimental results validate the stated theory. The results show that the error and estimated depth are directly proportional to each other, where a rise of either of these factors results in an increase in the other one. Here, the depth is inversely proportional to the verge angle, where an increase in-depth reduces the verge angle. On the other hand, the baseline length is directly proportional to the verge angle, where any expansion in the baseline results in a larger verge angle. Overall, these observations are summarized in Figure 6-18. This behavior is clearly shown in Figure 6-17, e.g., the estimated depth of the target at 200 cm is improved dramatically by increasing the baseline (10, 20, 30 and

40 cm) because the estimated depth is 172.22 ± 3.45 cm, 196.83 ± 9.1 cm, 198.02 ± 2.7 cm and 198.64 ± 2.06 cm, respectively. The overall results show that baseline size yields limited improvement relative to depth estimation due to the relationship between the verge angle and the depth estimation.

The large standard deviation formed in baseline 30 cm and baseline 40 cm is partially related to the controller margin error (±10 pixels) and the mechanical assembly, such as error in the position of the camera origin. In chapter 5, error in the mechanical joints was measured and analysed, and the overall error was ±0.01 cm. Moreover, depth estimation is related to the resolution, where finer resolution results in a more precise vergence controller and finer depth measurement. For example, if the target at 100 cm and moves to a new position that is 120cm perpendicular to the baseline the depth measure at travel of 1.5 cm. For high-resolution images, the system measures the target in millimetres (100, 101.5, 103.0, .., 120 ± 1.0 cm); however, for low-resolution images, the system measures the target in centimetres (i.e. 101, 102, .., 120 ±1 cm).

Variables / Relation	Percentage of Error (E)	Depth (D)	Verge Angle (V)	Baseline (B)
$\delta E/\delta D$	↑	↑	↓	–
$\delta D/\delta V$	↓	↓	↑	–
$\delta V/\delta B$	↓	–	↑	↑

Figure 6-18. Error relationship between verge angle, depth estimation and baseline.

Table 6-2 summarizes the depth estimation results with four baselines, i.e., 10–40 cm with an increment of 10 cm. The table shows the average depth, standard deviation in centimetres and the Mean Absolute Error (MAE) in percentage. For the 10 cm and 20 cm baselines, the MAE increased sharply as the depth increased. For the 30 cm and 40 cm

baselines, the MAE increased slightly with increased depth. However, for baselines greater than 30 cm, the minimum depth should be set to 60 cm because, at most positions less than 60 cm, the system failed to verge on the target due to the large perspective view between both images.

Table 6-2: Depth estimation summary of four baselines

True Depth	Baseline 10 cm		Baseline 20 cm		Baseline 30 cm		Baseline 40 cm	
	Avg. depth ± std (cm)	MAE (%)	Avg. depth ± std (cm)	MAE (%)	Avg. depth ± std (cm)	MAE (%)	Avg. depth ± std (cm)	MAE (%)
40	41.93 ± 4.1	2.36	39.96 ± 0.64	0.38	39.89 ± 0.37	0.31	Fail	0.00
60	59.5 ± 2.01	1.55	60.14 ± 0.65	0.48	60.22 ± 1.02	0.77	60.35 ± 0.89	0.69
80	79.68 ± 1.69	1.33	79.89 ± 0.82	0.69	80.02 ± 0.83	0.64	79.99 ± 0.89	0.55
100	96.47 ± 2	3.67	100.49 ± 1.96	1.57	100.19 ± 0.85	0.67	100.35 ± 0.95	0.71
120	119.37 ± 4.04	2.77	120.08 ± 1.71	1.43	119.49 ± 1.33	1.24	119.29 ± 0.76	0.88
140	129.82 ± 5.22	10.18	139.06 ± 3.51	3.03	139.43 ± 2.14	1.52	139.43 ± 1.00	0.84
160	157.07 ± 11.57	8.24	160.66 ± 3.55	2.76	160.39 ± 1.96	1.50	159.99 ± 1.06	0.86
180	158.99 ± 9.99	21.00	178.66 ± 5.56	4.98	180.21 ± 2.12	1.73	179.56 ± 2.14	1.60
200	172.22 ± 3.45	27.77	196.83 ± 9.1	8.10	198.02 ± 2.7	2.98	198.64 ± 2.06	2.12

Another depth estimation experiment was performed with artificial tomato setup (Figure 6-8) to evaluate the reliability and repeatability of tracking a target. Here, the master camera was fixed on the target, and the slave camera had a different starting point. The baseline was set to 20 cm, and the measurement was repeated 10 times for each target.

Table 6-3: Depth estimation with artificial tomato setup (baseline: 20 cm).

Target	True depth (cm)	Avg ± std (cm)	MAE (%)
1	150 ± 0.2	148.91 ± 2.13	2.15
2	140 ± 0.2	139.81 ± 1.25	1.11
3	135 ± 0.2	132.24 ± 1.73	2.76
4	126 ± 0.2	125.97 ± 0.86	0.68

Table 6-3 shows the experimental results of the tomato depth estimation. As can be seen, the results are nearly the same as the results (e.g. average and standard derivation) of the previous experiment (Table 6-2). This experiment illustrates how the platform can perform under different configurations. Moreover, in this experiment, the target was fixed; therefore, the results demonstrate the reliability (almost equal to $\pm$ 3 cm) of the PNCC algorithm and the platform is.

Figure 6-19. The output of fully verged on two different targets.

Thus, we conclude that the overall performance of the system depends on the size of the baseline. When the baseline is less than 20 cm, the error in-depth estimation increases, leading to poor results. However, when the baseline is set to a value greater than 20 cm, depth estimation is improved to up to ±2.1 cm at a depth of 200 cm.

Comparison of experimental platform to previous work

Table 6-4 compares the depth estimation outputs of our system and Zhang and Tay's system. The system configuration, such as baseline, image size and pyramid levels, was set according to Zhang and Tay's system (baseline: 24 cm, image size: 200×200, three pyramid levels). The results show that our system demonstrates slightly better improvement in terms of depth estimation because the average absolute error is better. In terms of standard deviation, our system outperformed Zhang and Tay's system in all three runs, where our system achieved an average error of ±1.83% compared to ±3% for Zhang and Tay's system. It is apparent that Zhang's system shows a large standard deviation when it comes to measuring depths closer to the system. As a result, distortion of the perspective view of the object increases in the log-polar space.

Table 6-4: Depth estimation results of Zhang and Tay's and proposed systems.

True Depth (cm)	Zhang and Tay's system		Our system		Average error ± std dev differences (%)
	Average estimated depth (cm)	Average error ± std dev (%)	Average estimated depth (cm)	Average error ± std dev (%)	
80	84	5 ± 2	80.66	1.13 ± 1.74	3.87 ± 0.26
100	106	6 ± 3	101.17	1.57 ± 1.73	4.43 ± 1.27
180	182	3 ± 3	179.87	2.78 ± 1.83	0.22 ± 1.17

Comparison of experimental platform with exciting systems

Table 6-5 shows the depth computation results obtained by the three sensors (ZED, Intel Realsense D415, and our system). The results show that our system performs comparably to the other two sensors. At a depth of 200 cm, the ZED camera demonstrates the largest std (±4.37 cm), followed by our system (±2.06 cm) and the Intel D415 (±1.63 cm). Note

that the results of the ZED camera and Intel D415 meet the specifications given in their respective user manuals. The D415 consistently overestimated distance, and our system underestimated distance. Our system provided the lowest MAE (2.12%), followed by the ZED camera (3.5%) and D415 (2.7%). However, our system's performance is considered poor compared to both systems because the depth is less than 100 cm, which could be due to the distortion in the perspective view that occurs when both cameras rotate greater than 35°.

Table 6-5: Depth estimation of ZED camera, Intel D415, and our system at baseline 40 cm.

True Depth (cm)	ZED camera		RealSense D415		Vergence cue (40 cm)	
	Avg. depth ± std (cm)	Mean Absolute Error (%)	Avg. depth ± std (cm)	Mean Absolute Error (%)	Avg. depth ± std (cm)	Mean Absolute Error (%)
100	100 ±1.37	0.9%	100 ± 0	0.00%	100.35 ± 1.26	0.94%
160	160 ± 4.11	3.5%	161.4 ± 0.93	1.40%	159.99 ± 1.06	0.86%
200	200 ± 4.37	3.5%	202.7 ± 1.63	2.7%	198.64 ± 2.06	2.12%

The three sensors adopt different approaches to compute depth. In our system, the depth estimation depends on mechanical joints to position the cameras, which may be considered a disadvantage when depth estimation of multiple targets must be performed repeatedly for each individual target. In contrast, the other sensors estimate the depth of all targets in a single run. Overall, our system demonstrates depth measurement performance that is comparable to that of traditional stereo vision systems.

Small object depth estimation

In this experiment, the depth of small objects was measured using the three sensors (Figure 6-10). Table 6-6 shows the depth measurements. Here, the ZED camera failed to detect the objects and the Intel D415 managed to detect the two largest objects (diameters greater than 2 cm) with a reliable measurement of std ±0.60cm and ±1.4 cm for objects with diameters of 2.4 cm and 2.2 cm, respectively. Note that our system detected all targets. The output of our system shows that the MAE and std reduced as the diameter of

the objects increased, where the MAE and std values dropped from ±2.58% to ±1.06% and ±2.49 cm to ±1.12 cm, respectively. This variation was due to the size of the windows and controller error. Our system estimated the depth of all targets because the search process was performed for one target at a time, which depends on the features of the target directly from the image rather than computing the disparity of the entire scene, as is the case with the Intel D415.

Table 6-6: Depth estimation of ZED camera, Intel D415, and our system at baseline 40 cm.

True Depth (cm)	ZED camera		Intel D415		Vergence cue (40 cm)	
	Avg. depth ± std (cm)	Mean Absolute Error (%)	Avg. depth ± std (cm)	Mean Absolute Error (%)	Avg. depth ± std (cm)	Mean Absolute Error (%)
100	100 ±1.37	0.9%	100 ± 0	0.00%	100.35 ± 1.26	0.94%
160	160 ± 4.11	3.5%	161.4 ± 0.93	1.40%	159.99 ± 1.06	0.86%
200	200 ± 4.37	3.5%	202.7 ± 1.63	2.7%	198.64 ± 2.06	2.12%

6.3.4 Field experiment results

Two types of results are evaluated in this experiment: (I) the depth estimation of each target and (II) the verge on the target by computing the disparity. Figure 6-20 and Figure 6-21 show the output of the vergence controller experiment when the system verge on the fixation point. Figure 6-20 shows scene one where there are four tomato fruits while Figure 6-21 shows scene two with six tomato fruits. The difference between scene one and scene two is that scene two contains more targets that are close to each other which challenge the system to verge correctly on the fixation point of the gaze. In both the scenes, the system manages to verge on the target with an accuracy of ±10 pixels. This range is due to the control margin set in the controller to stop the isolation and also supported by the lab evaluation.

Figure 6-20: The output of vergence controller on scene 1.

Figure 6-21: The output of vergence controller on scene 2

The vergence controller performs reliably well when the master camera fully gazed on the target and stay still, but when the master camera moves from one target to another, the vergence controller lose the tracking of the master camera particularly when the master camera on the leafs. The reason is that the background shares a common intensity value. However, this is not a big issue as long as the vergence controller verged correctly

on the final fixation point, or this could be avoided by suspending the vergence controller when the master camera changes the targets.

The second evaluation is the depth estimation, Table 6-7 shows the result. The platform was set at a baseline of 20 cm, and the targets were between 85cm and 105cm. The depth estimation in the greenhouse has larger std value compared to the lab result, which is due to the noise present in outdoor environments. For example, for the target at scene two, the std is ± 1.32 cm at a depth of 85 cm which is 1.5 times bigger than the result of the lab. The increase in the std is due to the presents of noise such as lighting intensity, the effects of the dynamic range that the camera has a low dynamic range.

Table 6-7: Depth estimation for field experiments

Scene 1		
Target	True depth (cm)	Avg ± std (cm)
a	96.8	±1.02
b	96.0	±0.71
c	101.8	±0.84
d	104.6	±1.14
Scene 2		
Target	True depth (cm)	Avg ± std (cm)
a	85.6	±1.32
b	91.2	±1.30
c	86.8	±1.30
d	94.8	±0.45
e	85.0	±0.71

6.4 Conclusion

In this chapter, the PNCC algorithm for vergence controller has been studied. The proposed model was integrated with a binocular platform that has five degrees of freedom. The model was designed to overcome the problems in the traditional NCC algorithm by introducing a Gaussian pyramid method. Improvements were observed with respect to the accuracy and reliability of the controller and stability of fixation on the

target in respect to the work of Zhang and Tay (2011). The verge on the target in a complex environment and presence of similar shapes of the proposed algorithm is more robust than conventional NCC and other template-matching techniques where the maximum difference is ± 10 pixels. Our experimental system demonstrates good verge on the target under different lighting conditions between the master and slave cameras while maintaining the same output when both cameras have the same IRIS configuration (i.e. the error increase to 1.9% when the master camera IRIS half opened).

Through multiple quantitative and qualitative experiments, the experimental system demonstrated good depth estimation with standard deviation of ± 2.06 cm at a worst-case depth of 200 cm, where the depth estimation improved when the baseline was greater than 20 cm MAE equal to 2.21% (Table 6-2). In addition, the system showed improvement compared to an existing method in terms of depth estimation and reliability, where the overall highest percentage of error at three different depths was 4.43% at 100 cm. The experimental system was also compared to other stereo vision sensors, and better depth estimation results were obtained despite a minor disadvantage of the system, i.e., it can only obtain the depth of one target at a time, thereby making it slower than orthogonal stereo vision systems.

The experiments conducted showed that the system requires certain improvements to ensure that the cameras perform more precisely when rotates more than 35 degrees. Therefore, a future research question can be posed which is how to minimize the perspective distortion error.

However, another experiment was conducted to test the capability of estimating the depth of a small objects (e.g., 1.2 and 1.4 cm objects), which are difficult to estimate with most stereo vision systems. The results indicate that our experimental system demonstrates

reliable and robust depth estimation for such targets at a standard deviation of ± 1.12 cm at 150 cm.

The algorithm and the platform was tested in outdoor environment with a tomato bush. The result shows that the system operates outdoors, in natural lighting, with a std ± 1.32 cm at 85 cm with a verge error of ± 10 pixels.

In future, the proposed PNCC algorithm could be improved by implementing an adaptive template size, which will allow verge on the fixation point to be more precise when it changes based on the target's size and position. In addition, the experimental system will be tested in a practical environment for estimating the position of tomatoes fruit harvesting process.

Chapter 7

Visual Attention Model Based Active Binocular System for Harvesting

In this chapter, we propose a cognitive model for an active binocular platform with five degrees of freedom that is integrated with a manipulator arm. The model is a combination of a visual attention models that generates a saliency map, a gaze control system, a vergence control system chapter 5 (Mohamed et al., 2018a), and a 3D reconstruction algorithm chapter 4 (Mohamed et al., 2018b). The model is designed to harvest tomatoes; therefore, it generates a cognitive map with information concerning the probability of the target being a tomato, its 2D position, the shape of the target, the 3D location of the target, using the verge algorithm, and information concerning the grasping affordance using the corresponding stereo images. As far the authors are aware, we are the first to implement a cognitive model for harvesting tomatoes using active binocular vision. The advantage of using visual attention in harvesting is that the visual attention is focussed on a special target in the scene; the background is uniform, with leaves, ground, and sky, while the fruit is distinctive from the rest of the scene. Moreover, the saliency map focuses on the salient region despite the enormous variations in the outdoor illumination (e.g. sunny, partially cloudy, overcast or shadowy). This process quickly provides information concerning the possible target that can be used in further processes (e.g. classifying the maturity of the tomato). One motivation for using a visual attention model is to help save computational power and time compared to the traditional process of sliding windows over an image. Moreover, using an active binocular vision system allows the surrounding environment to be explored quickly.

The rest of the chapter is divided as follows. First, the proposed model is presented in detail; then an evaluation and experiment are presented; and, finally, a discussion and conclusion are given.

7.1 Tomato colour maturity

Tomato maturity can be distinguished based on the level of redness. Tomatoes pass through different stages to be ready for picking (Agrawal et al., 2016). Figure 7-1 shows the stages of readiness that tomatoes goes through until being ready for picking. According to Kaur and Guptata (2017) and farmer in tomato business[10], the tomato fruits being picked is shown in Figure 7-1 (d), (e) and (f); therefore, the main task conducted in this part was to detect the tomato fruits that are ready for picking as show in the Figure 7-1 (d), (e) and (f).

Figure 7-1: Tomato maturity stages.

[10] Farmers and expertise are those who monitor the farming labs in Plymouth university.

7.2 Attention based vision for grasping tomato

The visual attention model is integrated with an active binocular platform. Two cameras are used in the algorithms: the master and slave cameras. The master camera is used to explore the scene and identify targets, while the slave camera is used to compute the 3D information regarding the targets. The controller of these two cameras depends on the visual attention model. The target information is used to update the cognitive map; this information describes the target, including the probability that the target is a tomato, the 3D position of the target, and information pertinent to grasping. The target information is computed by processing different maps, i.e., (1) the saliency map to process the master image and compute the 2D position of the target, (2) the gaze and vergence maps to determine the 3D positions of the targets generated by the saliency map, and (3) the point cloud map, which uses stereo vision to compute the 3D target information. The cognitive map is linked to the manipulator arm, which is used to grasp the target based on the provided information. Appendix C explained the integration between the platform and GummiArm.

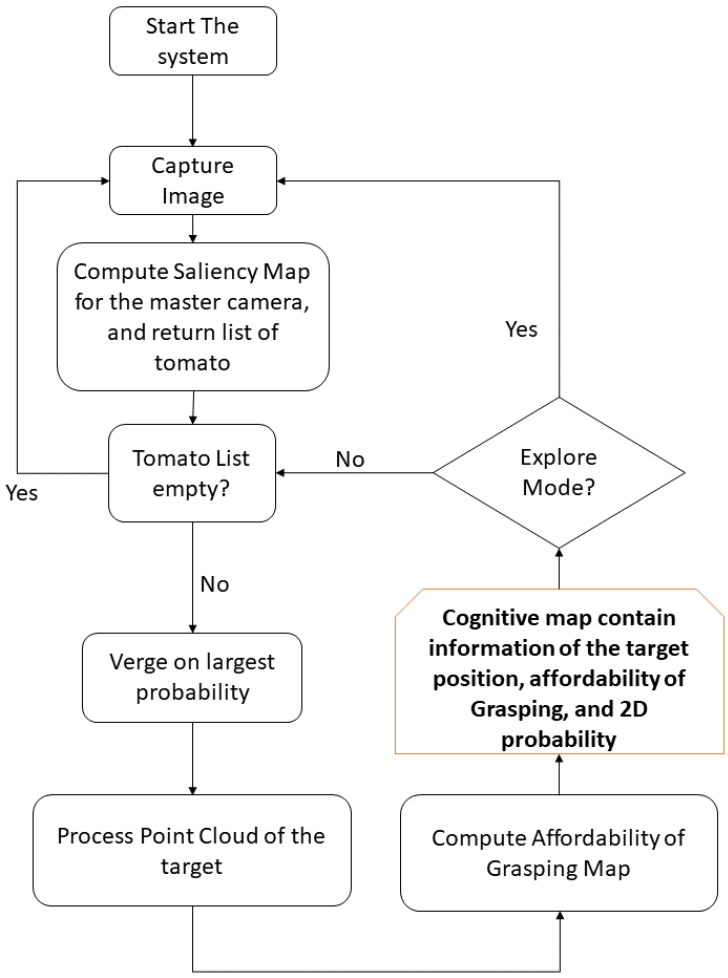

Figure 7-2: The proposed visual attention model. The entire information about the target is store in cognitive map.

Figure 7-2 shows the layout of the visual attention model, starting from capturing an image using the master camera. A saliency map is used to compute the salient features in the captured imaged; then a probability model is used to check if the targets are tomatoes. Later, the system verge on the target with the highest probability is used to compute the 3D position, and a point cloud process is used to compute the grasping affordance (Song et al., 2016). Finally, this information is updated in the cognitive map. The system has two modes where when explore mode is on the system compute the saliency map every time it verge on target. When the explore mode is off the system first verge on all targets before it captures new image from the master image.

7.2.1 2D Saliency map

The saliency map is the output of the visual attention model and contains the salient features of the master image. The model used in this study is a bottom-up model based on the work of Itti et al. (1998). The image input to the model is a colour image decomposed into different representations, such as colour, intensity, and edges, which are processed in parallel (Figure 7-3). A feature map is computed from each representation.

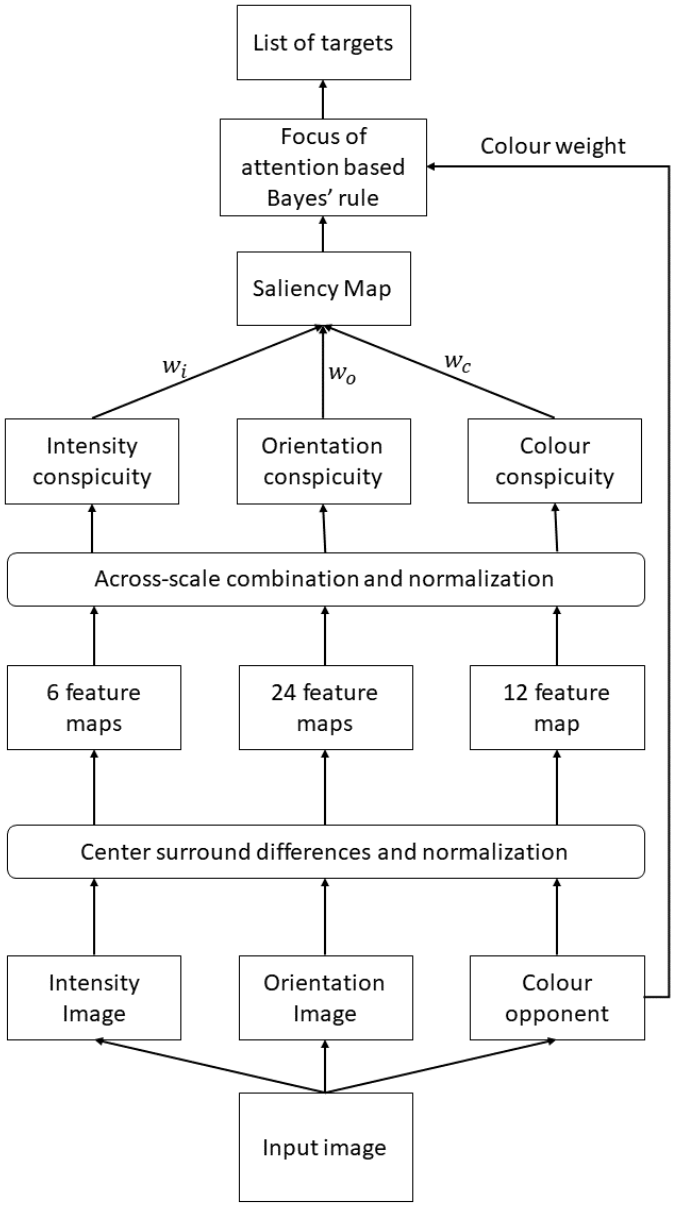

Figure 7-3: Proposed Saliency map architecture.

Colour Feature maps

There are many colour spaces that are excited in computer vision (e.g., red–green–blue, cyan–magenta–yellow, and hue–luminance–saturation). These colour spaces are used for different applications depending on the information required to process an image. The best-known colour space is RGB; this colour space contains three channels representing the values of red, green, and blue (Figure 7-4(a)) that describe the wavelength of the colour. Figure 7-4(b) shows an image from a greenhouse where the output channels are affected by the surrounding lighting. The HLS colour space is widely used in computer vision because it separates the luminance (L) and the chrominance hue (H) and saturation (S); HLS makes descriptions of the colour easier (Dawson-Howe, 2014). Another colour space that attempts to match the psychometrics of human vision is LAB; this colour space has a channel to store the values of the lightness (L) and the colour opponents green–red in channel A and yellow–blue in channel B.

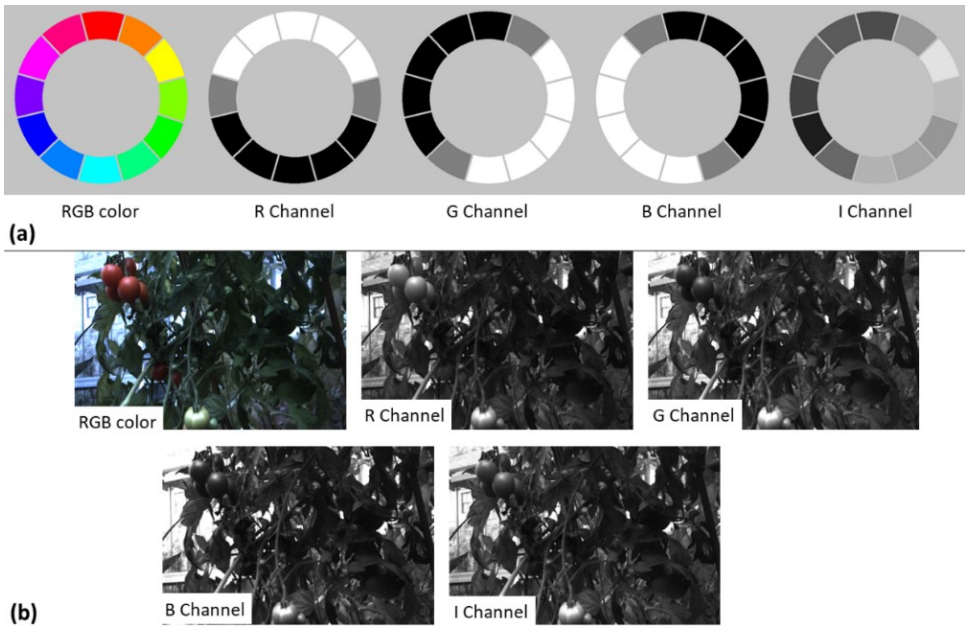

Figure 7-4: RGB image with range of colour, and split channel for red, green and blue. I represent the intensity image. (a) show a ground truth of the colour. (b) Image from the dataset shows the separate channels with the effect of lighting.

In this study, the colour space used to produce the colour feature maps is RGB. Starting with the RGB colour space, the three channels are split into r, g and b, which represent

red, green and blue, respectively. First, the three channels are processed to compute the maximum value per pixel location, where the output of a single channel RGB_{max} has the same size as the original image and any pixel less than or equal to zero is replaced by 0.0001 to avoid dividing by zero in subsequent processes.

Applying the colour opponent, as done in the LAB colour space, two feature maps are computed: red–green and blue–yellow (RG and BY, respectively). As can be seen in Figure 7-4, the red channel indicates that the red colour has a maximum value of 255 (white), and the green colour has the lowest value 0 (black); the opposite is true in the green channel. By computing the differences between the red and green channels, any information about the colour green is removed. The output RG is normalised by RGB_{max} to ensure that the output is between 0 and 1 (Eq. (7.1)). Figure 7-5 shows the output of Eq. (7.1) .

In the same colour space, the yellow (y) channel is computed by taking the minimum value between the red and green channels in an element-wise operation. Then, the blue–yellow opponent is computed and normalised (Eq. (7.2)) (Engel et al., 1997; Krantz, 1975). Figure 7-5 shows the colour opponent between blue and yellow.

$$RG = \frac{r - g}{RGB_{\max}} \tag{7.1}$$

$$BY = \frac{b - y}{RGB_{\max}} \tag{7.2}$$

The above colour analysis is used to distinguish between each colour wavelength; for example, when cancelling the green light, the redness can be weighed, and the opposite is also true (Krantz, 1975). The same methodology applies to the blue and yellow colours.

The intensity image is a grey scalar image showing the average of the three channels (Eq. (7.3)). The intensity image is used to extract the brightness features of objects because

previous studies have shown that attention is related to intensity (Miau et al., 2001; VanRullen, 2003).

$$I = \frac{r + g + b}{3} \tag{7.3}$$

Figure 7-5: Colour opponent where the green subtract from the green and the same for the yellow subtract from the blue channel.

Orientation feature map

The orientation feature map adds extra details to the saliency map to help focus on the shapes and the contrast. Processing the edge as a feature gives the saliency map the ability to draw a boundary around the feature extracted using the colour and intensity.

Based on the work of Itti et al. (1998), the edge detection in our study was based on a Gabor filter combined with a Gaussian pyramid. The Gabor filter is used to process the complicated textures in an image to detect and segmentation features (Kolekar, 2002). The implementation of the Gabor filter in this study is used for the intensity feature map I and generates four orientation maps, $O(\theta)$ $(\theta \epsilon \{0°, 45°, 90°, 135°\})$. A Gaussian pyramid is used to construct nine layers, $\sigma \epsilon [0,1, \ldots, 8]$. The final output of the edge feature map is $O(\sigma, \theta)$. Figure 7-6 shows the output of the orientation feature map process.

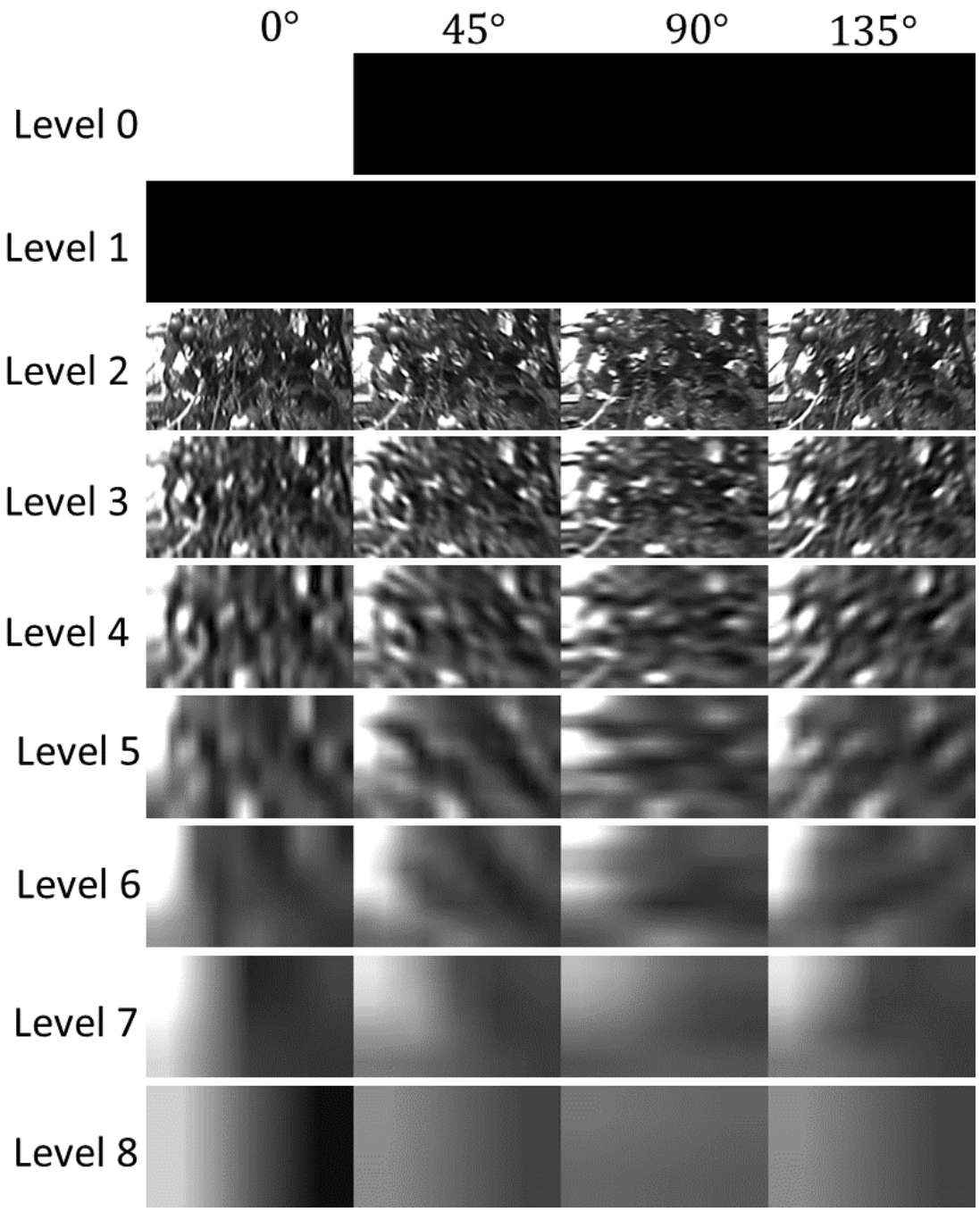

Figure 7-6: Orientation Feature Map outputs. Four orientations with right levels outputs. Note that the images size are reduces with the increase of the level, but to make it more clear the size keeps the same.

Centre-surround operator

```
FeatureMapCenterSurroundDiffirences( featureMapList ):
    outputList = []
    finerLevel = {2,3,4}
    courseLevel = {3,4}
    for s in finerLevel:
      for c in courseLevel:
            courseImage = featureMapList [s+c]
            courseImage = resize (courseImage, featureMapList[s].shape)
            differenceImage = abs(GaussianMaps[s] — courseImage)
            outputList.append(differenceImage)
    return outputList
```

Figure 7-7: Center-surround for feature enhance algorithm

The area of contrast in the feature maps (f) is computed by applying the centre-surrounding model (Driscoll et al., 1998). The centre-surrounding model is an algorithm mimicking the visual cortex in locating the salience region in a scene, where the visual cortex depends on the stimuli and the region surrounding the stimuli (Driscoll et al., 1998). That is, the centre-surrounding process boosts small features that are surrounded by different intensities. Gaussian pyramids (Levitt and Lund, 1997) are used to process these features by initially creating nine levels of the feature map, where each level is half the size of the layer below. These layers are divided into fine layers and coarse layers ($c \in \{2,3,4\}$ and $s = c + \delta$ with $\delta \in \{3,4\}$, respectively (Itti et al., 1998)).

The centre-surrounding differences (d) is computed by taking the difference between layers, $d = f_c \ominus f_s$. The operation $\ominus$ is a process that subtracts two matrices element by element. Note that the size of f_s is resized to the size of f_c. Figure 7-7 shows the algorithm for the centre-surrounding operation.

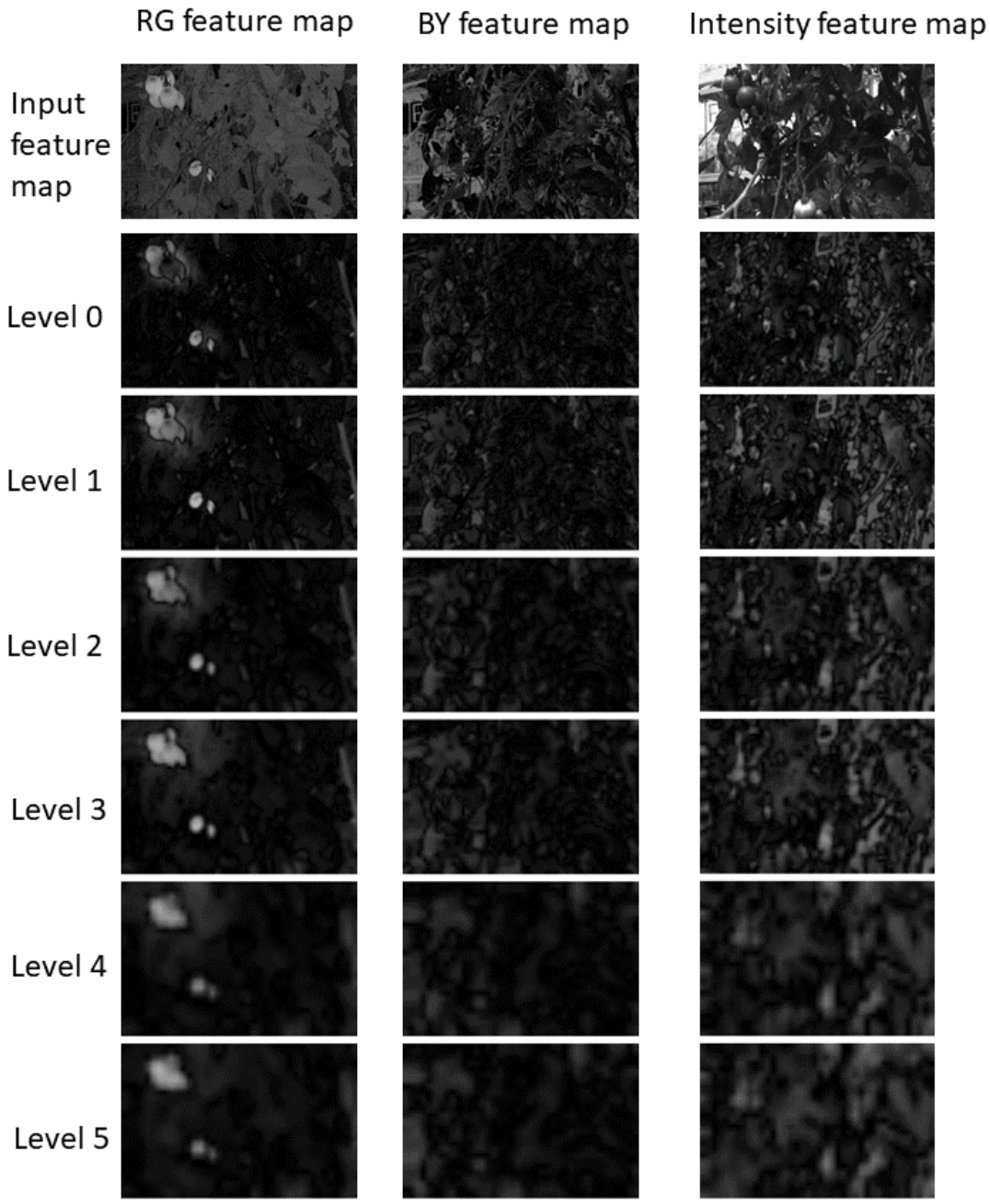

Figure 7-8: Center surrounding operation output. Six feature maps for each input with different six sizes. (Note that the increase of the level the size of the output is drop but in case the clarity a constant size is set for all levels)

The outputs from the centre-surrounding operation are six feature maps with different sizes from the input feature maps. Figure 7-8 shows the output of the centre-surrounding operation for three feature maps: the (I) red–green and (II) blue–yellow colour opponents and (III) the intensity feature map. The centre-surrounding operation boosts the features in the feature maps; this is clearly shown in the RG feature map, where the tomato pixels (close to 255) are surrounded by lower pixel values and the tomato pixels have been

increased, while the surrounding pixels have been reduced to darker pixels (zero). Because the centre-surrounding operation boosts the small surrounding features, the tomato has been converted to a bigger pixel, and the surrounding pixels have become darker in the BY feature map. This output is shown at Level 0.

Combine feature maps (saliency map)

Before combining the centre-surrounding difference feature maps f, preprocessing needs to be done to normalise, resize and combine f into a conspicuity map, C_{map}. The normalisation process uses min–max normalisation (Cao et al., 2016) and linearly scales the feature maps to ranges of $[-1,1]$ or $[0,1]$. This operation is used to lower the noise in the feature maps, and in our case, the range is set to $[0,1]$. After normalisation, the average local maximum $\bar{m}$ is computed by segmenting f into small predefined regions, and the global maximum value M is computed. The normalised feature map is multiplied by $C_{map} = f(N) \times (M_N - \bar{m}_N)^2$ (Itti et al., 1998). This operation is referred to as a uniqueness weight (Frintrop, 2006) and is used to enhance the feature map with the most salient feature while minimising feature maps with many salient features—for example, if the salient feature depends on the orientation instead of the colour. Finally, all feature maps are resized to the size of the original image and summed to generate C_{map}.

The uniqueness weight operation has a problem that has been pointed out by Itti el at. (1998). Taking the differences between the global maximum and the local maximum will only work if there is one maximum peak; however, if there are two peaks with equal maxima, the operation will return zero and ignore the entire feature map (Frintrop, 2006). There are a few workaround solutions to this problem that have been proposed by Itti el at. (1998) and Frintrop (2006). The proposed method by Frintrop (2006) is to divide the feature map by the square root of the local maximum instead of using the global

maximum. Applying this method requires a tuning process to set a minimum threshold for the local maximum. In Frintrop (2006), the threshold was set to 50% of the global threshold. In our study, we used the operation proposed by Itti el at. (1998); however, instead of computing the global maximum, we set it equal to one.

$$C = f(N) \times (1 - \bar{m}_N)^2 \qquad (7.4)$$

Figure 7-9: Conspicuity maps of three input feature map. (a) colour, (b) intensity, and (c) orientation.

The output of the combine and normalize operation is three conspicuity maps (the RGB colour feature map, an intensity feature map, and an orientation feature map) (Figure 7-9). These features are multiplied by assigned weights (the sum of the total assigned weight is equal to one); then, the final saliency map (S_{map}) is computed by taking the sum of the four weighted conspicuity maps. The weights are used to control the contribution of each feature map to the final saliency map (Eq. (7.5)). The sum of the total weights is one.

$$S_{map} = C(RGB) * w_{RGB} + C(HSV) \times w_{HSV} + C(I) \times w_I + C(O) \times w_O \qquad (7.5)$$

Focus of attention

The focus of attention (FOA) is the process used to determine the most salient region (MSR) in the saliency map. There are different approaches to finding the FOA. In the saliency map, the MSR is determined by finding the maximum value. The majority of published studies focus on this point by using a fixed shape, such as a circle or rectangle, for the FOA (Itti et al., 1998). Another algorithm for FOA is the flooding algorithm, which uses a segmentation process after the most salient point in the saliency map (Walther et al., 2002). This algorithm finds the shape of the region by searching through the pixels

neighbouring the most salient pixel; this produces a good result in the saliency map (Frintrop, 2006). This algorithm requires tuning the threshold parameter of the flooding algorithm to stop.

Draper and Lionelle (2005) proposed another approach using a watershed algorithm to find the FOA. This algorithm does not require a tuning parameter (Frintrop, 2006). All the above methods can be used to detect a single SMR in the saliency map or to find multiple FOAs; this is referred to as inhibition of return (IOR) (Frintrop, 2006).

The watershed algorithm was implemented in this study because it provides information concerning the target, i.e., the contour, radius, and position, and does not require parameter tuning. To use the watershed algorithm, the saliency map is scaled to be between 0 and 255 instead of between 0 and 1. The starting point of the watershed algorithm is the maximum pixel in the saliency map; then the algorithm searches for the region belonging to that pixel.

The output of the FOA process is the location of the MSR (x_s, y_s), the contour of the region C_s and the radius of the region r_s. C_s is used to cover the region in the saliency map to select the second maximum point in the IOR process. The location and radius are used to crop the target for use in the gaze and vergence stage. The target information computed during the IOR is used in the vergence control system, as explained in chapter 5, to improve the verge on the target; using pyramid normalised cross-correlation (PNCC), the size of the target is used to determine the size of the template. However, because the IOR is used to find the pixel with the maximum value in the saliency map and then to cover those pixels with C_s, the limit for the IOR was set to 40% of the maximum salient region to stop the process from looping through all the pixels in the image.

A training model can be used to check if the output is a tomato using the RG and BY feature maps; the region of the saliency map can be checked in the feature map and used to support the saliency map output.

Compute the probability of tomato using feature map

In this study, a saliency map is computed to locate the salient region in a scene. The saliency map is used to detect the fruit due to its ability to detect the salient region in scenes where the most frequent occurrence is plants and leaves with a background of sky and ground. In this case, the fruit is counted as the salient feature in the scene due to its colour, shape, and smaller number of pixels. However, in many cases, there are other features that are in the scene and count as salient features (Figure 7-10 (c)). Such incorrect predictions can be due to the algorithm used to determine the FOA, where the search is for the pixel with the highest value. Another cause could be that the scene being processed has a non-regular background due to the distribution of leaves and the gaps between them allowing the sunlight to pass; this is, indeed, a salient feature, but an unwanted feature in our case. Figure 7-10 (b) shows the saliency map of Figure 7-10(a); as can be seen, the saliency map finds the most salient feature, which is the high-intensity light from the sun.

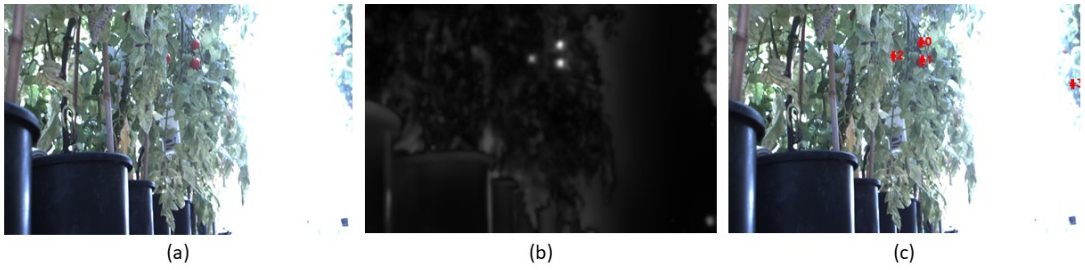

Figure 7-10: Output of the saliency map in scene the attention of number three is not a tomato fruit but it's on leaves with high brightness. (a) the scene image (b) Saliency map. (c) Focus of attention.

In general, the target of interest in this study is a ripe tomato, which tends to be a reddish colour; therefore, we use the RGB channel information to compute the probability of the region of interest selected by the saliency map being a tomato, $w \epsilon \{0,1\}$ (0 indicates non-tomato and 1 indicates tomato). Using the information of the pixel to be checked, the

RGB measurement is $x = [x^R, x^G, x^B]$. We use Bayes' rule (Eq. (7.6)) to classify the pixel x as tomato or non-tomato (Prince, 2012).

$$\Pr(w = 1|\, x) = \frac{\Pr(x|w = 1)\Pr(w = 1)}{\sum_{k=0}^{1} \Pr(x|w = k)\Pr(w = k)} \tag{7.6}$$

The likelihood and the prior probability are computed as shown in eq.(7.7) and eq.(7.8) respectively (Prince, 2012).

$$\Pr(x|w = k) = \mathrm{Norm}_{\mathrm{x}}[\mu_k\,, \Sigma_k] \tag{7.7}$$

$$Pr(w) = \mathrm{Bern}_w[\lambda] \tag{7.8}$$

A linear regression is used to estimate the parameters μ_k, Σ_k and λ ($k \in \{0,1\}$) using pre-labelled data during training.

The result is classified as a tomato if the probability is greater than 0.5 ($\Pr(w = 1|x) > 0.5$) and as non-tomato if the probability is less than 0.5. This model has a drawback in that there is an overlap between the pixels that could lead to misclassifications; therefore, the average of the probability is computed around the chosen salient pixel.

7.2.2 Gaze and vergence feature map

The next maps in the visual attention model are used to compute the 3D position of the target using the active binocular system. In this process, called attention-based vision, the target is selected based on the visual attention model, and then the eye verge is based on that target (Solé Puig et al., 2013). In this process, two algorithms run in parallel: the gaze and vergence controllers. The gaze controller controls the master camera to align the fixation point with the target provided by the saliency map (i.e., it moves the master camera focal ray to match the centroid of the target). The vergence controller controls the

slave camera to track the master camera's fixation point (i.e., it moves the slave camera focal ray to match the fixation point of the master camera).

Both controllers use exponential functions to control the cameras based on a PNCC algorithm (for more details, see chapter 5). In chapter 5, one of the parameters used to improve the fixation on the centroid of the target in the PNCC algorithm was the template size, where a larger or smaller size leads to increases in the error. Therefore, in this work, the target size from the saliency map is computed to update the template size for both the gaze and vergence controllers. When the vergence controller verges correctly on the target, the 3D position of the target is computed.

The saliency map provides a list of targets with the 2D location, contour, and radius of the target as well as its probability of being a tomato. The gaze controller chooses the target with the maximum probability, updates the template size, and moves the camera to the fixation point, which is the centroid of the target. The vergence controller runs in parallel to the gaze controller. The 3D position and the information from the saliency map of the target are updated onto the cognitive map. This process is repeated until all targets from the saliency map are visited.

7.2.3 Affordance to grasping feature map

When the system verges on the target, a stereo image is captured to compute the depth map and then convert the depth map into a point cloud. This process has been studied and evaluated in chapter 4. In this step, the generated point cloud is used to compute the 3D structure of the target and the probability of the grasping affordance. Because the 3D position of the target is known and the 2D position is known from the vergence controller, the point cloud is segmented based on these data to only retain the target of interest.

The iterative closest point (ICP) algorithm (Rusinkiewicz and Levoy, 2001) is used to compute the differences between the target point cloud and the ground truth point cloud. The ground truth point cloud is a 3D model of a tomato (note that the average tomato size was considered when generating the ground truth). The ICP algorithm computes the transformation between the target and the actual shape. The root mean square (RMS) is used to compute the difference between the two point clouds. The RMS describes how well the target matches an actual tomato. A smaller RMS indicates a higher similarity between the two objects.

The size of the target and the total surface area of the target that is visible can be computed using the point cloud. This information is used to determine the grasping affordance and the actual position of the target by adding the radius of the target to the 3D position, because the vergence controller depth is estimated to the surface of the target, not to the centre of the target.

7.2.4 Cognitive map

The cognitive map is a dynamic map used to store the target data found by the feature maps. The cognitive map is updated with a target at every loop, where a loop starts with saliency map and ends when the system successfully computes the grasping affordance. Every target in the cognitive map contains three primary pieces of information: (I) the 2D probability computed by the saliency map, (II) the 3D position computed by the gaze and vergence controllers and (III) the grasping affordance computed from the point cloud.

The target in the cognitive map has a lifetime for each specific cycle of the process. The lifetime is set based on the needs of the process. For example, the lifetimes of the targets in the cognitive map are sufficiently long for the manipulator arm to pick all the targets. The cognitive map is dsesigned to act as a brain for the robot by providing the required

information to reach the targets; this information helps the robot determine which target should be picked first, based on its location and probability of being a tomato. The robot limits are set within the cognitive map to process the targets and arrange them in order, from the easiest to pick to the hardest. For example, the workspace of the robot is defined to help determine whether the target is within reach.

7.3 Experiment and evaluation of the system

In general, there is no standard approach to evaluate these types of systems because there is no standard dataset. Frintrop (2006) identified four approaches to evaluate a model: (1) using human perception to compare the results, (2) comparing the results with other visual attention models, (3) testing the system under image transformations such as rotation and resizing, and (4) evaluating the performance in an application. Two approaches are used in this study to evaluate the performance of the system which (1) and (2).

7.3.1 Saliency map evaluation

The saliency map was evaluated by embedding it in the application; this is the best way to test the system. The saliency map is designed to detect tomatoes in a greenhouse. The input image size is 640×420. A dataset was collected consisting of 133 pictures taken from different views. The saliency map and FOA were computed for each picture. Figure 7-11 shows a few examples from the dataset. These images were selected to show the performance of the system when detecting tomatoes under different light intensities and in different positions. As can be seen, the system manages to detect the tomatoes despite the intensity of the light, which changes the colour of the tomatoes (Figure 7-11(f) and (g)). Target detection based on colour is a challenging problem in computer vision and robotics due to the large variety in the colour spectrum and the illumination intensity (Ilea and Whelan, 2011); however, the saliency map algorithm manages to detect the tomatoes at different light levels.

In the saliency map, the light region (pixel value = 255) represents the MSR in the scene. Visual attention is a good method to use to track an object in a scene without specifying the features of the target, as long as the target can be differentiated from the rest of the scene. This is due to the mechanism used in visual attention to detect the MSR in the scene. Tracking the target of interest without specifying the feature to be tracked, or narrowing the search of the target in the scene to a few regions, is one of the advantages of using a saliency map. The saliency map saves time when searching for a target under different backgrounds and reduces the necessary tuning of parameters (e.g., the threshold values for RGB or HSV).

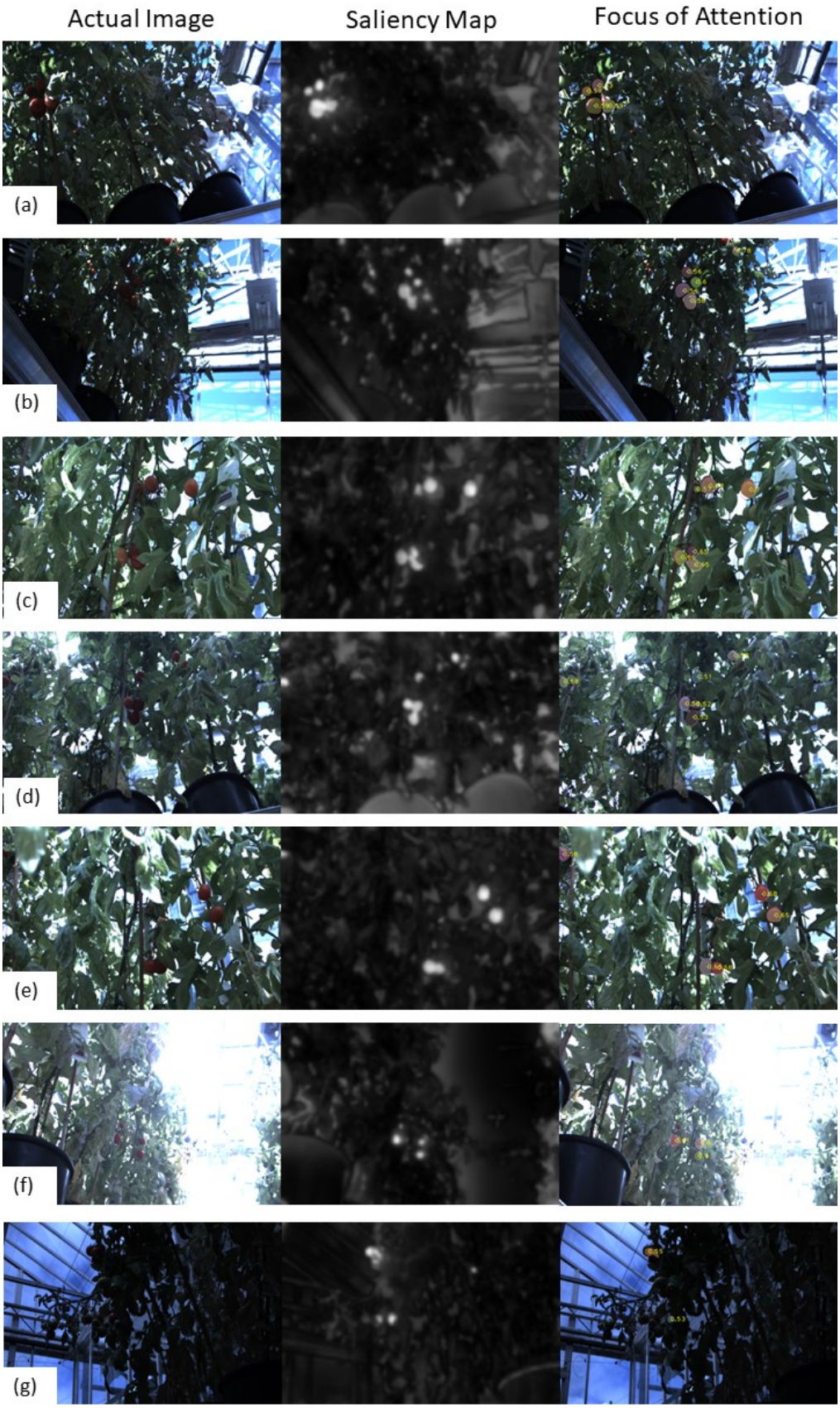

Figure 7-11: The result of the Saliency map with detecting tomato. Left input image, center Saliency map and right Focus of attention with probability the selected target is tomato.

In this study, the target is a mature tomato with a red colour. The saliency map manages to detect most of the tomatoes present in a scene that have a mature red colour (Figure 7-11). However, in some cases, the system could not detect immature tomatoes (green–red colours); this is due the nearly equal values of the red and green wavelengths, which leads to lower values in the red–green opponent feature map, because the saliency map in this study depends highly on the colour feature (the weight of the colour map is $w_{RGB} = 60\%$, while those of the intensity and orientation maps are $w_i = 20\%$ and $w_o = 20\%$, respectively). The immature tomatoes, therefore, do not appear as salient in the saliency map. However, this is not important because the system is designed to detect ripe tomatoes.

Figure 7-12 shows an example where the effect of the environment produces a salient region due to the complexity of the scene, with some openings between the leaves leading to salient features. The output of the saliency map matches actual human perception behaviour; however, in this study, tomatoes are the targets of interest, and the pre-training model is designed to measure the probability of a target being a tomato or non-tomato using the colour feature map. Figure 7-12 (c) shows the FOA when activating the IOR and using the trained model to determine the presence of a tomato. The figure shows the region of interest with its probability, where higher probabilities are closer to the tomato.

Figure 7-12: Example of saliency map with salient region but not tomato. (a) input Image, (b) Saliency map with salient region that has equal value to the tomato but are not belong to tomato and (c) Focus of attention.

The system manages to detect the tomatoes in clusters (Figure 7-12(c)), where the saliency map takes into account the edge detection and combines it with the colour feature to provide a stop line for the shape of the object; this result is combined with the watershed algorithm. This is advantageous for a system harvesting tomatoes because tomatoes grow in clusters, not separately. In some of the results, the saliency map identifies a cluster of tomatoes as one object; however, the watershed algorithm works to separate them. However, the implementation of the watershed algorithm in the saliency map adds an extra parameter to tune the output; unlike the work of Draper and Lionelle (Draper and Lionelle, 2005), which eliminated a tuning threshold, in this study, a threshold was set to improve the tracking of the region of interest. This difference is due to the embedded application, which is outside and affected by sunlight, therefore making the problem more complex than the indoor problem.

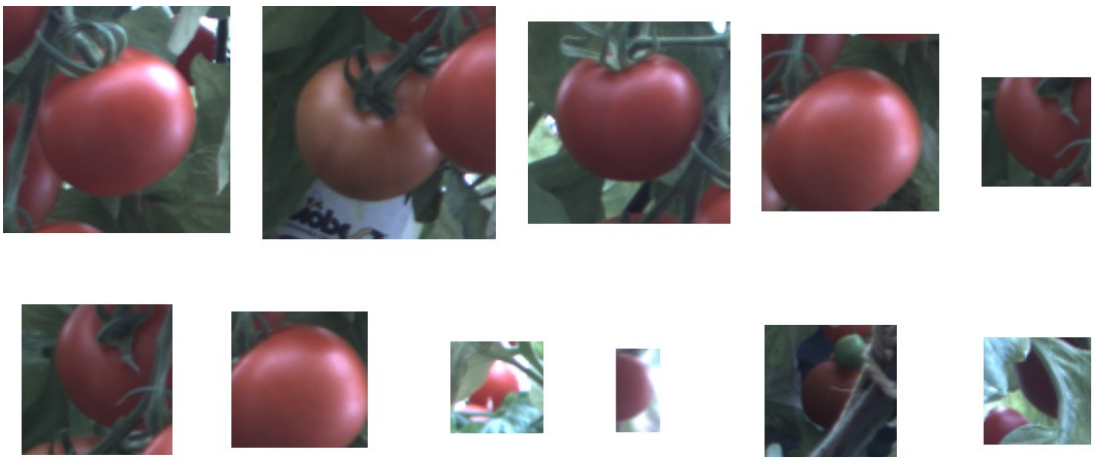

Figure 7-13: List of FOA from different images from the dataset. FOA is depend on the size of the tomato and the surrounding.

The saliency map generates a list of targets in the images along with their probability of being a tomato; this is used in the subsequent feature maps. The cropped images are cropped by adding 10 pixels to the overall size provided by the saliency map. Figure 7-13 shows the output used in the gaze and vergence controllers.

The saliency map was run on 120 images of tomato plants with different quantities of tomatoes. Each image was manually checked to assess the detected tomatoes and the number of the tomatoes in each image. The accuracy of the saliency map is 82.94%.

7.3.2 Cognitive map experiment and result

The visual attention model integrated with the platform was evaluated in a greenhouse. Different scenes were set up for the evaluation process (Figure 7-14). The scenes differ in the quantity of tomatoes and the position of the platform inside the greenhouse (e.g., the position relative to the sunlight). The distance between the platform and the tomato plants was between 80 cm and 120 cm.

Figure 7-14: Visual attention model evaluation experiment in a greenhouse.

Table 7-1 shows the cognitive map output for the evaluation experiment. The table shows the number of the tomatoes in each scene that are ripe or unripe. The output of the system is shown in the table as well, indicating the detected tomatoes, the false detections, and, in the last column, the corrected verge on the tomato.

Table 7-2 and Table 7-3 shows the individual target data for scene 1 and scene 2 (Appendix B contains the complete list of all 8 scenes output). The data is the 2D probability computed by the saliency map, the affordances of grasping in RMS and the successful of the vergence controller to verge on the target.

Table 7-1: Cognitive map experiment output.

Scene	Ripe	Unripe	Detection	False detection	verged
1	5	5	5	0	5
2	3	1	3	1	2
3	6	5	5	0	4
4	5	5	5	0	3
5	6	4	5	0	3
6	5	4	4	0	3
7	5	4	5	0	3
8	6	3	5	1	5

Ripe: red tomato in the scene ready to be pick.
Unripe: green-red tomato in the scene still required some time to get mature.
Detection: the detected tomato by the system.
False detection: the wrong detection by the system that is not tomato.
Verged: describe the successful verge on the detected targets.

Table 7-2: Scene 1 individual targets output.

Target	2D probability	Affordance RMS	Verge success
0	73.2%	10.07	success
1	63.3%	0.08	success
2	75.0%	0.05	success
3	89.5%	0.15	success
4	90.1%	0.23	success

Table 7-3: Scene 2 individual targets output.

Target	2D probability	Affordance RMS	Verge success
0	78.9%	0.62	fail
1	84.6%	0.47	success
2	75.9%	5.23	fail
3	47.6%	1.02	success

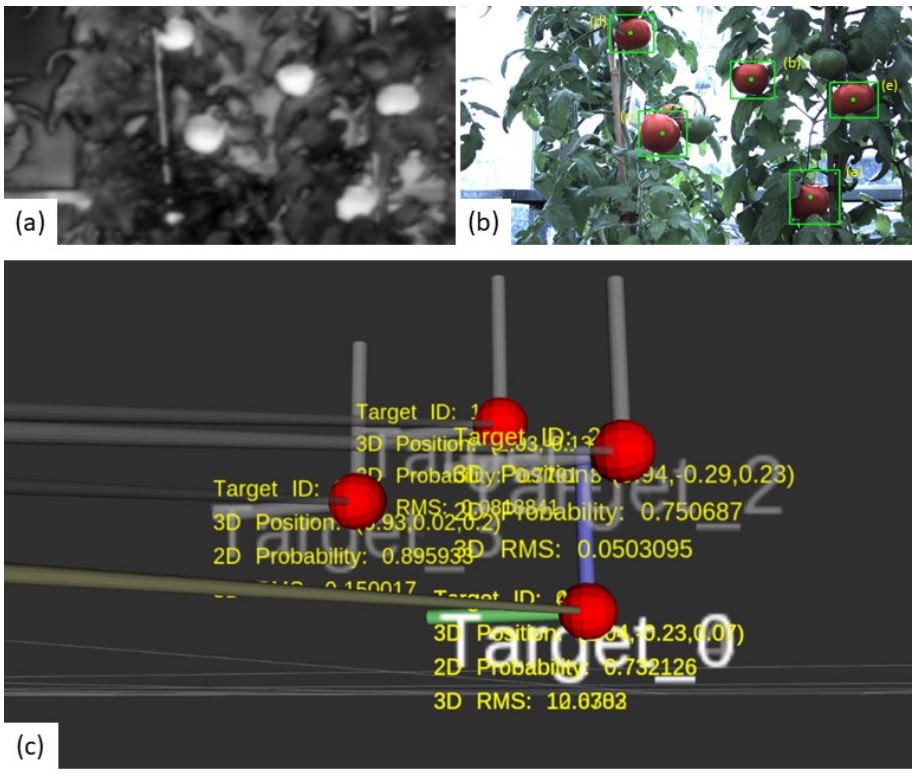

Figure 7-15: The output of the cognitive map for a full cycle of the saliency map for Scene 1: (a) the computed saliency map, (b) the focus of attention, where the green box is the template size, and (c) a visualization of the targets and their data.

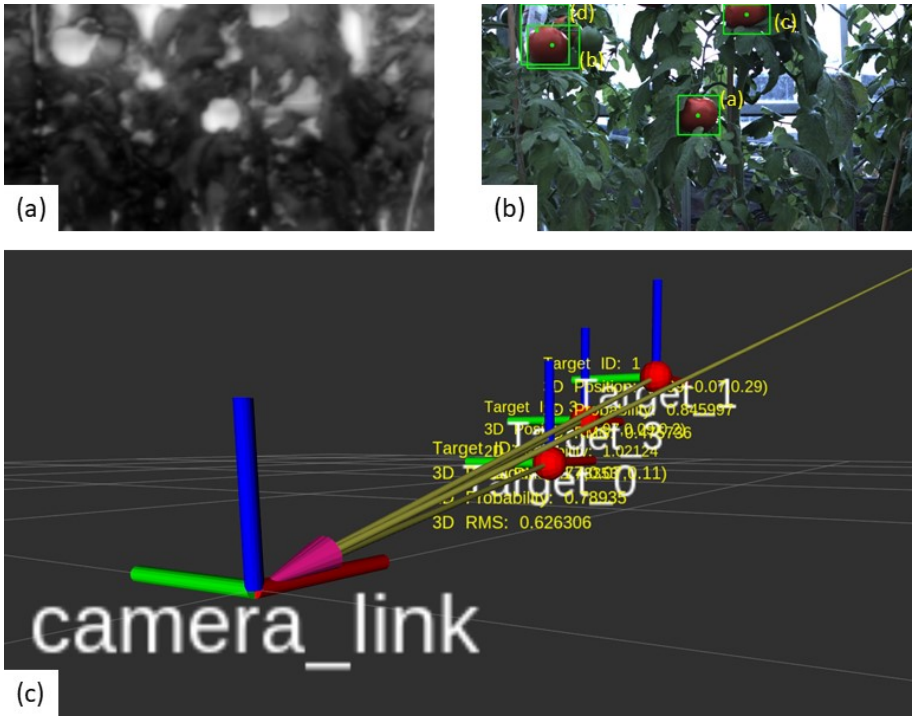

Figure 7-16: The output of the cognitive map for a full cycle of the saliency map for Scene 2: (a) the computed saliency map, (b) the focus of attention, where the green box is the template size, and (c) a visualization of the targets and their data.

7.4 Discussion

Figure 7-15 and Figure 7-16 show a completed cycle of the cognitive map for two different scenes. The two scenes differ in the position of the platform. In Scene 1 (Figure 7-15), five tomatoes were detected by the system, while four tomatoes were identified in Scene 2 (Figure 7-16). A summary of these scenes is shown in Table 7-1 (note that the correspond to the run number is the scene number).

Table 7-2 and Table 7-3 shows the data if individual target in the scene.

The FOA shows the tomatoes with their centroids and the size of the template used in the gaze and vergence maps (Figure 7-15 (b) and Figure 7-16 (b)). However, when computing the FOA of Tomato (a) in Scene 1, a larger template size than the actual tomato was used; this is because the saliency map also detects the small opening above the tomato as a salient area and identifies both the tomato and the opening as one region (Figure 7-15 (a)). Furthermore, the watershed algorithm counts both the tomato and the opening as one object. This frequently occurs, especially if the plant has few leaves. However, a closer look at the output of the FOA confirms that the centroid of the template is always on the tomato. This is due to the methods used to choose the centroid of the target, i.e. selecting the maximum pixel in the scene and then applying the watershed algorithm to detect the contour of the target. To avoid this problem, the orientation feature map weight can be increased to produce more edges. Conversely, an erosion operation can be applied to the saliency map to erode the boundaries of the regions and separate them. However, this will cause another issue, where the contour of the target will become smaller.

In Scene 2 (Figure 7-16 (b)), the FOA shows that there are two tomatoes (Tomatoes (b) and (d)) within the same region, which should not occur when applying the watershed algorithm. However, this is due to noise, which affects the watershed algorithm during

the selection of the area, and the method used to produce the rectangle of the target by measuring the maximum height and width. Indeed, Tomato (d) is the thin line beside Tomato (b).

Figure 7-15 (c) and Figure 7-16 (c) show a visualisation of the cognitive map, where the targets are shown in the red spheres relative to the platform. This information is also plotted on the side of the tomatoes in rviz. The cognitive map is used to represent the information that is computed based on the saliency map, and the 3D data are used to enable the robot to decide which targets should be picked first (

Table 7-2 and Table 7-3). Therefore, the cognitive map contains the fundamental data concerning the targets, i.e., the probability of being a tomato in 2D and the 3D appearance and position relative to the platform.

The grasping affordance describes the distance to the target based on shape matching. In other words, the generated point cloud is compared to a ground truth tomato, where first, the point cloud is transformed to match the ground truth position, and then the Euclidean distance is computed between the ground truth and the point cloud. Calculating the probability of the target being a tomato in this step has drawbacks due to the modifications made on the platform. The tilting joints for each motor contribute to the epipolar error. The original algorithm for the online epipolar geometry update is performed on the first version of the platform (a platform with a fixed tilt), where the error from the tilt angle is constant; however, in the second version, the tilt of the motor changes based on the target position. The tilt motor has an 8-bit encoder (0.29°). However, in this study, the ground truth is a 3D model in which the size is set based on an average, which affects the accuracy. In future studies, a more robust algorithm needs to be used for comparisons to actual tomatoes.

Computing the grasping affordance provides valuable data about the targets that help determine if they are tomatoes. For example, in Scene 1, Tomato (a) is hidden by leaves and only a small portion of the tomato is visible. The result of the grasping affordance for Tomato (a) (in the visualisation this tomato has ID 0) is 10.03, which is large compared to those of the other tomatoes (below 1.0). This large value is due to the presence of the leaves surrounding the tomato and is an effect of the corresponding process. Note that Tomato (a) has a large template, but the point cloud process detects the tomato using the contour found in the saliency map.

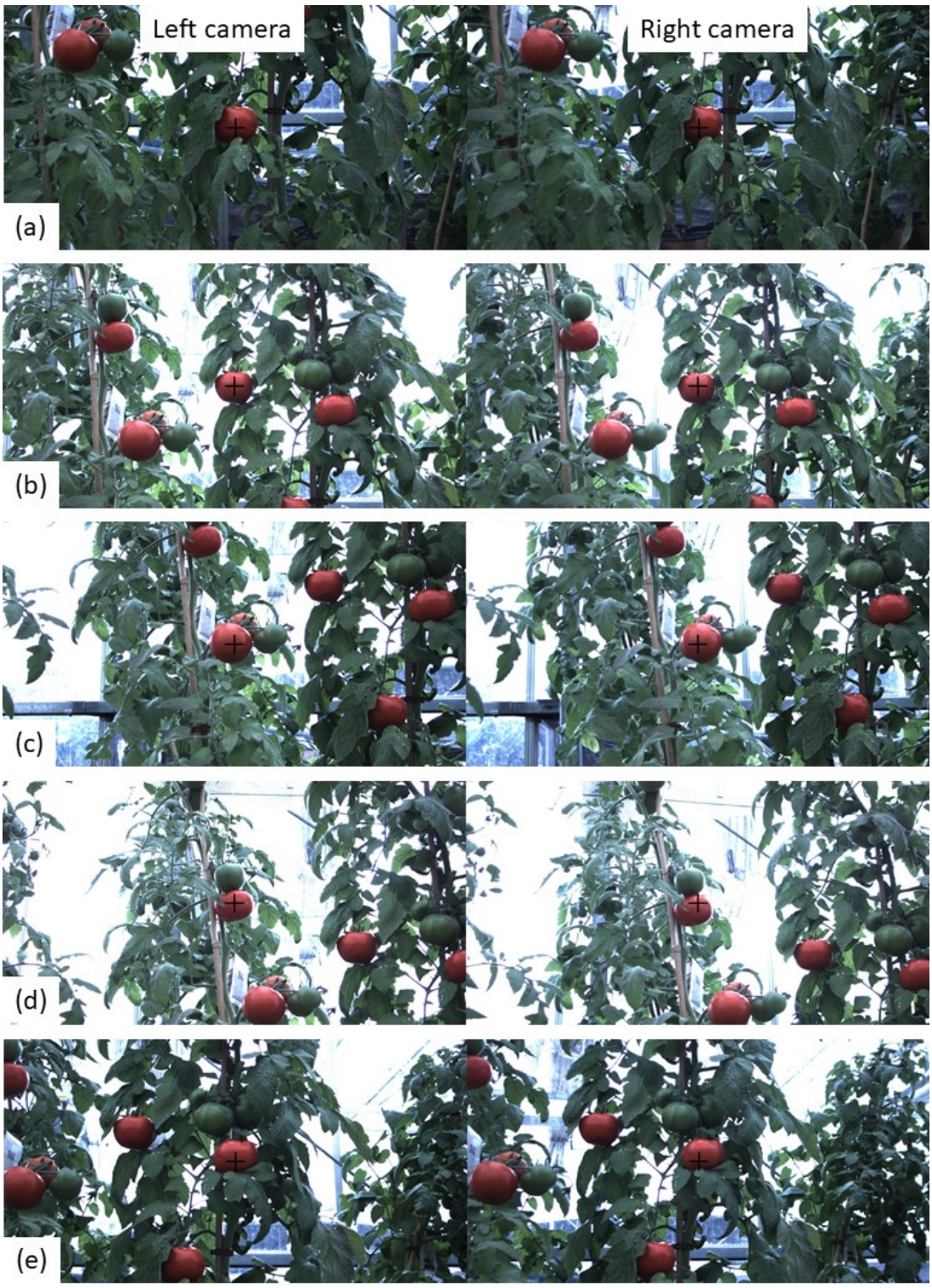

Figure 7-17: The output of the completed cycle of the cognitive map by verging on all targets of scene one.

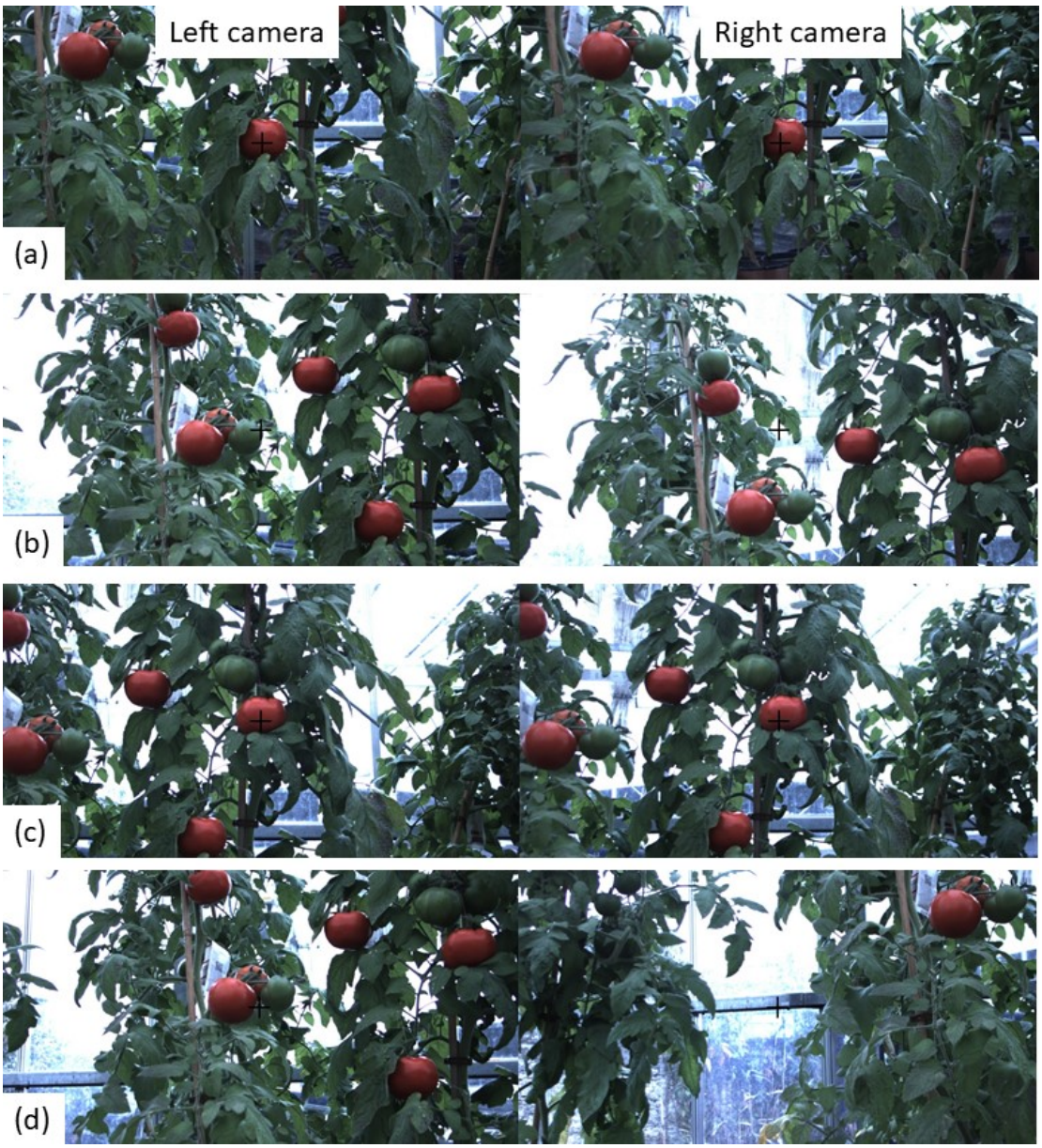

Figure 7-18: The output of the completed cycle of the cognitive map by verging on all targets of scene two.

The output of the saliency map is a list of the tomatoes (i.e., their 2D positions and sizes in the images). The system starts with the target with the largest probability using the PNCC algorithm. The PNCC controller requires the correct size of the target to verge on it precisely, as explained in chapter 5. Figure 7-17 and Figure 7-18 show the system completely verged on all targets detected by the saliency map. As can be seen, the system successfully verges on most of the targets in both scenes. Table 7-1 shows the result of the experiment done in the greenhouse using the cognitive map.

However, in some cases, the system fails to verge on the targets. For example, Figure 7-18 shows the result of the complete verge for Scene 2 (Table 7-3), where Figure 7-18 (b) and Figure 7-18 (d) show where the system fails to verge on the target and leads to an incorrect 3D position estimation; this is shown in Figure 7-16 (c), where the visualisation of the cognitive map is shown. There are two targets that have incorrect positions: one is out of the scene and the other (the target with ID 1) is farther from the two identified tomatoes and closer to the platform. Note that the target with ID 1 still has a 3D probability; this is because the system is using an independent algorithm to compute the point cloud and the master camera already has a correct gaze for the target.

Running the gaze and vergence controllers in parallel causes a major issue for the performance of the system. This issue occurs when the gaze controller moves from one target to another. This leads to confusion in the vergence controller, which tracks the centre of the master camera. The vergence controller sweeps left and right without tracking the actual centre of the master camera until the gaze controller completes the gaze process. To prevent this, the vergence controller was run in serial after the gaze controller.

The cognitive map can be extended with extra data using the three main feature maps. For example, extra information can be computed using the point cloud. These data tell the robot from which direction it is the best to approach the fruit based on the intensity and the direction of the point cloud. The generated point cloud represents the tomato and removes all other objects (e.g. leaves). The centre of the tomato is computed by taking the average of these points. Using the centre of the points and assuming that the points represent vectors from the centre of the tomato, these vectors are summed and the direction is computed. As another example, using the saliency map output as extra data,

we can compute, via an analysis, contour features, such as how the contours close to form a circle.

One additional concern that affects the performance of the system is that the cameras have a low dynamic range; this can be seen in Figure 7-17 and Figure 7-18, when the system verges on the highest tomato in the scene, the sun faces the camera, and the images darken. This also could be one of the reasons why the system could not verge on some tomatoes in the scenes. However, this limitation shows the strength of the propose detection algorithm.

7.5 Conclusion

This chapter presents a proposed visual-based attention model for the harvesting process. This is the first attempt to apply a visual attention model to the harvesting process. The model was divided into three main feature maps: (I) the saliency, (II) the gaze and vergence controllers, and (III) the grasping affordance. The saliency map is a bottom-up model based on the work of Itti et al. (1998). The model was modified to focus on detecting tomatoes in outdoor environments. The FOA was based on a watershed algorithm to detect the most salient region and determine the contour of that region. A Bayes' model was integrated with the saliency map to determine the output of the FOA region from the tomato and non-tomato regions. The other two feature maps have been studied in detail in the previous works (Mohamed et al., 2018b, 2018a). The vergence controller estimates the position of a target and the epipolar online geometry that generates a point cloud.

The saliency map was tested and evaluated on a dataset of 120 images taken of tomato plants and shows an accuracy of 82.94%. The saliency map shows potential for detecting

tomatoes and computing their contours, especially for clustering tomatoes. Moreover, the model performed very well under different lighting conditions.

The three feature maps were integrated into a cognitive map containing the data of the tomatoes. These data represent the basic data of every target, including the 3D position and the probability of that target being a tomato based on 2D image processing and the point cloud. Table 7-1 shows the output of an experiment performed in a greenhouse. The affordance of grasping in this system was not as reliable as the result in chapter 4 (i.e. the projection error is ± 10 pixels comparing to platform version 1 ± 2 pixels). This due to the platform upgrade.

In some situations, the system failed and detected non-tomato objects due to the noise in a scene, especially when the presence of only a few leaves on the plant led to an increase in the number of salient regions in the saliency map. Moreover, the vergence controller failed to track the master camera during target changes where the background is most cover by leaves. To avoid this issue, the gaze and vergence controllers needed to be run in series. In the future, one improvement to the saliency map will be to classify the maturity of the tomato by implementing a top-down model in the system. The system can be used in detecting different fruits/vegetable with a minor modification in the system.

Chapter 8

Conclusion

8.1 Summary

In this thesis, a new active binocular vision platform has been researched, developed and tested, for efficient and robust detection and 3D localization a tomato fruit outdoor. The approach regards tomato fruit detection as a three step process: (I) the visual attention subsystem detects the most salient region in the image which is the mature tomato (II) localize the interest tomato fruit using vergence vision and (III) compute the affordance of grasping using depth map. The final design was a five degree of freedom (DOF) system: four DOFs to control the pan and the tilt of each camera independently, and one DOF to control the size of the baseline. A variety of experiments were performed to assess the performance of the platform. The error associated with various components of the platform was deduced and improvements made accordingly.

The system has been tested *in situ* in a greenhouse where the system detects and estimates the position of the tomato fruit. The strength of the system is the robustness of the detection algorithm under variety of lighting condition outdoor using a saliency map with just few training data. In other words, the detection system is mimicking human behaviour in the sense that it does not require a special parameter tuning to operate on a specific application like the other machine learning or vision systems applications. Where the detection algorithm operates in many different scenes, for example, indoor, outdoor, greenhouse etc. without the need to re-tune the parameters. Another strength in the detection system, is that the detection system is not only work to detect tomato fruit, it can be used to detect different fruits that has a salient feature in the scene with a slightly change in the saliency map few training data.

However, human has the ability to notice many salient regions in the scene which the system inherits this capability from human. This is a limitation in the system where many salient features in the scene is detected that are not belong to the target which increase the time of the focus of attention process to determine the tomato only.

Furthermore, the accuracy of the system in estimating the 3D position of the target using vergence controller showed a good performance, experiments show no more than ± 3 cm within the range of 2 meter . The thesis has shown that the system is applicable to harvesting application. Where the result in the body of the thesis shows that the system has a good performance in computing the position of the detected tomato fruits. However, the system is computing the position of the target one by one which make the system is not suitable for some applications, but in general for a manipulator robot with one arm is enough to work at a commensurate speed to the gummi-arm, allowing fruit to be picked at full actuator speed, with no slow-down due to the camera system controller and vergence control.

Finally, the system used an updated epipolar geometry to compute the point cloud of the tomato. In the initial experiment (with version 1) the system shows a reliable rectification process that leads to a good depth map. The projection error in the system increase after using version 2 (i.e. the projection error increase from ± 1.24 to ± 9.3 pixels). However, to minimise the error in the depth map a large window size (15×15 pixels) and a post-filter was used which was suffusion to produce a good result in the final application of the system. The advantage of using a correspondence algorithm in the platform is minimise the disparity range and focus on the actual target especially in a harvesting environment where the scene is highly complex scene.

8.2 Contribution to knowledge

The author's knowledge, the problem of using an active binocular vision in harvesting is unique and no prior published work exists. The gaps in the literature are twofold: on one hand, the problem of integrating an active stereo vision system for depth estimation in harvesting has not been explored. On the other hand, using a visual attention based saliency map to identify fruits/vegetables. Therefore, the system would potentially be of interest to the research community in the fields of both computer vision and robotics. More specific contribution is present in the following subsections. A list of publication was published from this work (Mohamed et al., 2018c, 2018a, 2018b, 2016).

8.2.1 Online epipolar geometry to active stereo vision

In orthogonal stereo vision, the calibration done once to compute the epipolar geometry. In this research, the epipolar geometry of the system is updated based on the change in the motor encoder. This problem was taken from two published work Dankers et al. (2004) and Sapienza et al. (2013). Both ideas have been combined into one. The method use in this work is split into two part an offline process (Sapienza et al., 2013) and online process (Dankers et al., 2004). In the offline process, a linear equation between the motor angle and image angle was found by calibrating the system over varying geometry configurations. For each change in the verge angle a projection matrix was computed using the calibration process. The projection matrix contains the camera matrix, the rotation and translation matrix between the left and right camera. By decompose the projection matrix the raw data of the geometry is produces (e.g. the baseline size, rolls, pitch, and yaw rotating angle).

The propose algorithm, was evaluated by measuring the projection error between both images. The projection error in the rectified images was ±1.24 pixels when the system verge on the target which compare to ± 2.38 pixel in Hart et al. (2008). The platform and

the algorithm were tested further by computing the point cloud of three diameter spheres and comparing it with the actual shapes. The Iterative closest point (ICP) algorithm was used to compute the differences between the computed point cloud and the ground truth point cloud. The result was an average standard deviation of 0.0142 m and a margin of ±0.0039 m. This result can be compared to the orthogonal stereo vision system like Intel D415 with accuracy of ± 0.005 m and the ZED camera ± 0.01 m There is one issue in this algorithm which is the size of the image getting smaller when the verge angle increase.

Importantly, the raw data generated in the offline calibration provides an information about the accuracy and the repeatability of the platform's hardware. For example, the pitch and roll angles miss-alignment is $-0.433° \pm 0.015°$ and $0.526° \pm 0.047°$ respectively. This information is essential for improvements made to the algorithm in subsequent chapters and also lays the foundation for studying the error in the platform and how the result based on the pixels (image) is different to the actual hardware. based on this work two paper was published Mohamed et al. (2018b) and Mohamed et al. (2017).

8.2.2 Vergence controller

In this part of the system, depth estimation based on the vergence controller was studied. A Gaussian pyramid normalized cross-correlation was applied to control the verge on the fixation point. The controller implemented was based on the exponential function where to allow smooth movement of the camera. The error associated with the vergence controller was analysed. Depth estimation was computed via simple triangulation using the motor angle and baseline distance when the system fully verged on the fixation point. The result shows that the size of the baseline has a direct effect on the quality of the depth estimation, with the depth error being directly proportional to the baseline. Overall, the

proposed system produces a robust result for depth estimation with a standard deviation of ± 2.06 cm accuracy at a depth of 200 cm. The actual working distance is between 80cm to 140cm where the standard deviation is ± 1.5 cm with baseline equal to 200 cm.

Moreover, for the large objects, the system result compares favourably with exciting stereo vision system such as ZED and Intel Realsense D415, lying between the two systems in term of the accuracy of the measurement. However, the three systems have been tested to estimate the depth of small targets (diameter from 1.2 to 2.5 cm at 150 cm). The major advantage of the system is the stability to detect small object at distance where the propose system out preform the other two systems, for example ZED cannot measure the depth of these objects at 150cm whereas the propose system manage to measure the depth of all these objects.

The system has been tested in the field where this is the first attempt to use a vergence controller in the harvesting process. The field experiments showed similar trends in depth estimation and verge on the fixation point to those conducted in the lab, where the standard deviation ± 1.32 cm at a depth of 85 cm which is 1.5 times bigger than the result of the lab. This part of thesis is publish as journal paper Mohamed et al. (2018a).

8.2.3 Visual attention model

In this part of the dissertation, a cognitive map was designed based on a visual attention model to detect the tomato fruit. The model used three main features maps: (I) saliency map, (II) gaze and vergence map, and (III) affordance of grasping map. Each feature map yields certain information regarding the target that help the robot to decided which target should go for. For example, check the target position if it within the reach of the gripper. The system was tested and evaluated in a greenhouse with natural light that challenged computer vision.

The proposed model is based on bottom-up visual attention model, where the saliency map estimates the tomato using the master camera (i.e. the saliency map finds the interest region by applying a local contrasts and determine the unique features). A Bayes' approach was integrated with the focal of attention in order to determine the peak probability that corresponding tomato fruits in the scene. The result of the saliency map model shows the robustness of detecting the tomato fruit in outdoor environment (e.g. in a complex lighting condition) using a few datasets. Where comparing this to other machine learning that required a hundreds of images and a complex algorithm. The accuracy of the saliency map in detecting the tomato is 82.94%. Applying saliency map to tomato detection help to fast detection the tomato as the tomato is a salient in the scene compare to the ground and the trees. One of the limitation in the saliency map is that when the system at close distance to the tomato trees and there are many tomatoes in the scene which leads to fail in the detection as a result of that the tomato taken a large amount that make them not salient.

The gaze and vergence controller is used to compute the location of the fruit and finally, the afforces of the grasping is computed using point cloud (see Table 6-1, Table 6-2 and Table 6-3). The system generates a cognitive map that contain the data of the targets in the scene that link to the robot. However, the vergence controller has two type of fails, where when the master camera center is not within the field of view of the slave camera where the system verge on the wrong target if there was a high percentage of similarity (e.g. the minimum value of matching was set to 80%) or the system stay without moving due to there is no matching to the target. The second case is when the visual attention system run both the gaze and the vergence controller in parallel where when the gaze moves from one target to other the vergence controller get loss until the gaze controller focus on the target. This is due to the high similarity in the tree leaves where the PNCC

fail to determine the actual focus point. This part of thesis was published in Mohamed et al. (2018c).

8.3 Future improvement and development

One of the main advantages in the propose system is the ability to compute the depth of small objects that the traditional stereo vision could not detect (chapter 5). The system should be extending and examining to detect small targets such as cherry and raspberry. Moreover, the propose algorithm is based on bottom- up visual attention model which can be extended into top-down model that detect the maturity of the fruit. The model can also be uses to detect different type of fruits/vegetable as long as the fruit/vegetable are salient in the scene. Furthermore, visual attention model has been under research for many years and there are many models that can be integrated with the propose detection system that will improve the use in harvesting.

There are two aspects required for further improvement that were identified during the experiments and analysis of the result. The improvements are required in both (I) hardware and (II) software. The results of the experiment done on both epipolar geometry update and vergence controller depend on the resolution of the encoder. Where the higher resolution encoder will improve the depth estimating based vergence controller (chapter 5) and improve the online epipolar geometry (chapter 4).

One of the main part that required to upgrade in the new platform is adding a neck pan joint. The neck pan joints will extend the working space and in the same time will help to avoid the larger perspective distortion in the images, where the result shows that when the motor rotate more than 40° the PNCC algorithm (chapter 5) the error on the fixation point is increase. Also in the online epipolar geometry update (chapter 5) will the rectified image size is drop to small size. The neck pan joints will help maintain the error as low as possible in vergence controller and large image size in the rectification process.

The second improvement in the platform is replacing the motor with a DC motor and more accurate encoder 14-16 bits. Where this encoder will improve the depth estimating based vergence controller (chapter 5) and improve the online epipolar geometry (chapter 4). The vergence depth will be smoother using this encoder where it will be smoother. The DC motor is replaced in order to match the new encoders.

One of the major issue in the proposed platform was the cameras. Where the selection of the cameras to meet the requirement of the initial aim of the PhD project where it was indoor. The new cameras should have a high dynamic range (HDR) and wide field of view enough to cover the working space of the platform. Finally, there are methods that can be used such as FPGA or GPU to accelerate the computation speed of the system.

Based on the software development, the software developed in this thesis was based on two programming languages which are C++ and python. The main reason of using these two languages was due to the computation speed of C++ and the easy debugging with python. The code was written in python, is to accelerate the prototyping and meet the deadline of the PhD. Due to this the speed of the system was scarify in order to provide a working prototype at the end of this projects. Therefore, one of the main improvement in this system is to re-write the code in C++ to accelerate the speed of the system. Or re-write the code to work with a FPGA. Furthermore, the code still is not yet optimized.

Based on the software development, the software developed in this thesis was based on two programming languages which are C++ and python. The main reason of using these two languages was due to the computation speed of C++ and the easy debugging with python. The code was written in python, is to accelerate the prototyping and meet the deadline of the PhD. Due to this the speed of the system was scarify in order to provide a working prototype at the end of this projects. Therefore, one of the main improvement in

this system is to re-write the code in C++ to accelerate the speed of the system. Or re-write the code to work with a FPGA.

Nowadays, deep learning is one of the future use in all sort of work. In future the platform will be developed using deep learning approach where the verge controller can be improving to verge on the target with high accuracy even improve the speed of the system. This can be implemented using reinforcement learning approach where the training will be offline and then run outdoor with more real views.

Appendix A

Camera Specification

The camera used in this work is PointGrey Flea3 8.8MP (FL3-U3-120S3C-C) The specification is listed below. This was taken from https://www.ptgrey.com/flea3-88-mp-color-usb3-vision-sony-imx121-camera .

Table A- 1: Camera configuration.

Resolution	4096 x 2160
Frame Rate	21 FPS
Megapixels	8.8 MP
Chroma	Colour
Sensor Name	Sony IMX121
Sensor Type	CMOS
Readout Method	Rolling shutter with global reset
Sensor Format	1/2.5"
Pixel Size	1.55 µm
Lens Mount	C-mount
ADC	12-bit
Quantum Efficiency Blue (% at 470 nm)	63
Quantum Efficiency Green (% at 525 nm)	73
Quantum Efficiency Red (% at 640 nm)	49
Temporal Dark Noise (e-)	3.06
Saturation Capacity (e-)	5966
Dynamic Range (dB)	64.49
Gain Range	0 dB to 24 dB
Exposure Range	0.021 ms to 1 second
Trigger Modes	Standard, bulb, multi-shot
Partial Image Modes	Pixel binning, ROI
Image Processing	Gamma, lookup table, hue, saturation, and sharpness

Image Buffer	32 MB
User Sets	2 memory channels for custom camera settings
Flash Memory	1 MB non-volatile memory
Opto-isolated I/O Ports	1 input, 1 output
Non-isolated I/O Ports	2 bi-directional
Serial Port	1 (over non-isolated I/O)
Auxiliary Output	3.3 V, 150 mA maximum
Interface	USB 3.1 Gen 1
Power Requirements	5-24 V via GPIO or 5 V via USB3
Power Consumption (Maximum)	<3 W
Dimensions	29 mm x 29 mm x 30 mm
Mass	41 grams
Machine Vision Standard	USB3 Vision v1.0
Compliance	CE, FCC, KCC, RoHS. The ECCN for this product is: EAR099.
Temperature (Operating)	0° to 45°C
Temperature (Storage)	-30° to 60°C
Humidity (Operating)	20 to 80% (no condensation)
Humidity (Storage)	20 to 95% (no condensation)

Appendix B

Cognitive Map Output Tables

This appendix contains the output tables of the cognitive map experiment. There are 8 table for 8 scenes. The scenes vary by the position of the platform in relate to the trees.

Table B- 1: Cognitive map for Scene 1.

Target	2D probability	Affordance RMS	Verge success
0	73.2%	10.07	S
1	63.3%	0.08	S
2	75.0%	0.05	S
3	89.5%	0.15	S
4	90.1%	0.23	S

Table B- 2: Cognitive map for Scene 2.

Target	2D probability	Affordance RMS	Verge success
0	78.9%	0.62	F
1	84.6%	0.47	S
2	75.9%	5.23	F
3	47.6%	1.02	S

Table B- 3: Cognitive map for Scene 3.

Target	2D probability	Affordance RMS	Verge success
0	67.0%	1.03	s
1	71.9%	0.25	s
2	72.4%	0.016	s
3	77.3%	0.22	s

Table B- 4: Cognitive map for Scene 4.

Target	2D probability	Affordance RMS	Verge success
0	68.1%	0.036	f
1	70.7%	0.069	s
2	-	-	f
3	86.1%	0.308	s
4	68.6%	0.212	s

Table B- 5: Cognitive map for Scene 5.

Target	2D probability	Affordance RMS	Verge success
0	86.2%	1.17	s
1	87.4%	0.137	s
2	-	-	f
3	74.2%	0.15	s
4	57.8%	-	f

Table B- 6: Cognitive map for Scene 6.

Target	2D probability	Affordance RMS	Verge success
0	77.4%	0.251	s
1	70.3%	0.062	s
2	81.4%	0.023	s
3	86.1%	0.39	f

Table B- 7: Cognitive map for Scene 7.

Target	2D probability	Affordance RMS	Verge success
0	91.0%	0.019	s
1	93.4%	0.928	s
2	100.0%	0.055	f
3	80.2%	1.83	s

Table B- 8: Cognitive map for Scene 8.

Target	2D probability	Affordance RMS	Verge success
0	89.3%	0.32	s
1	100.0%	0.15	s
2	92.3%	0.2	s
3	88.6%	1.8	s
4	-	-	f

Appendix C

Integration with Manipulator Robot

In this appendix, the integration between the platform and a manipulator arm is presented. This integration is suit any manipulator arm as long as the robot work based on ROS.

GummiArm is a robot arm with seven DOFs based on the mechanisms of human and animal sensor motor systems (Stoelen et al., 2016)[11]. GummiArm can be considered soft robotics owing to the joint mechanisms that use agonist–antagonist actuators connected with a flexible tendon (artificial muscle) that controls the stiffness of the joints during operation (Figure C-1). GummiArm is a 3D-printed arm and uses the Dynamixel digital servo.

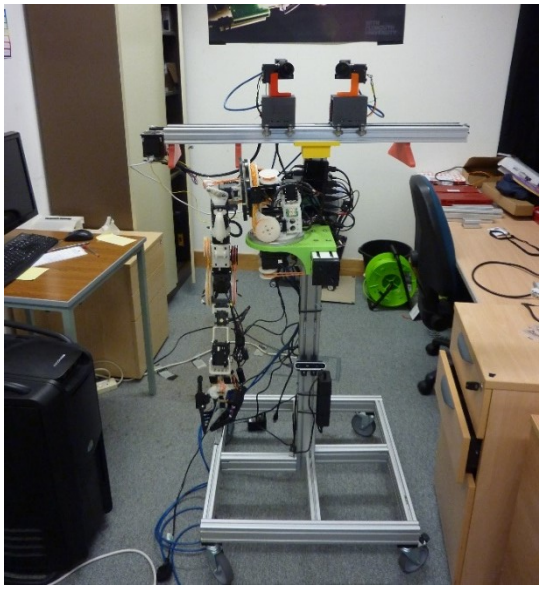

Figure C- 1: GummiArm robot

The environment during harvest can be unpredictable, and detecting all hazards in the environment using perception sensors can be difficult (e.g. in most fruit plants, a thin

[11] The GummiArm system is entirely open source and can be found in http://mstoelen.github.io/GummiArm/.

string is used to support the plants). Hence, the GummiArm is a good option to employ in various types of environment owing to its flexibility and robustness when it encounters hazards. GummiArm has different types of gripper for fruit harvesting, including a gripper for tomato fruits. Its software is based on ROS. In this work, the stereo vision platform was integrated with the robot by linking the cognitive map to the robot to perform the picking process.

Appendix C.1 Integrate the platform with a robot

The cognitive map is linked to the GummiArm in order to pick the target. The GummiArm is based on Robot Operating System ROS and has its own state machine[12] for picking process. In general, the state machine came with GummiArm is a cycle of different processes as shown in figure where the cycle starts by (1) locating the 3D position of the fruit, (2) command the arm to move closer to the target, (3) a visual servoing to close the gripper to the fruits, (4) close the gripper and pull the fruits to separate it from the tree, and (5) place the fruit in a tray (Figure C- 2). In this thesis, the exciting software provided by GummiArm is used with a minor modification in the process (1) to work with the proposed system.

[12] GummiArm is designed for harvesting different fruits and vegetables. GummiArm already came with its own software for grasping cycle which is based on a state machine that was design in order to simplify the modification in the harvesting steps or integrate different sensors. note that this work used the exciting state machine came with the robot and modified the detection node to work with the platform.

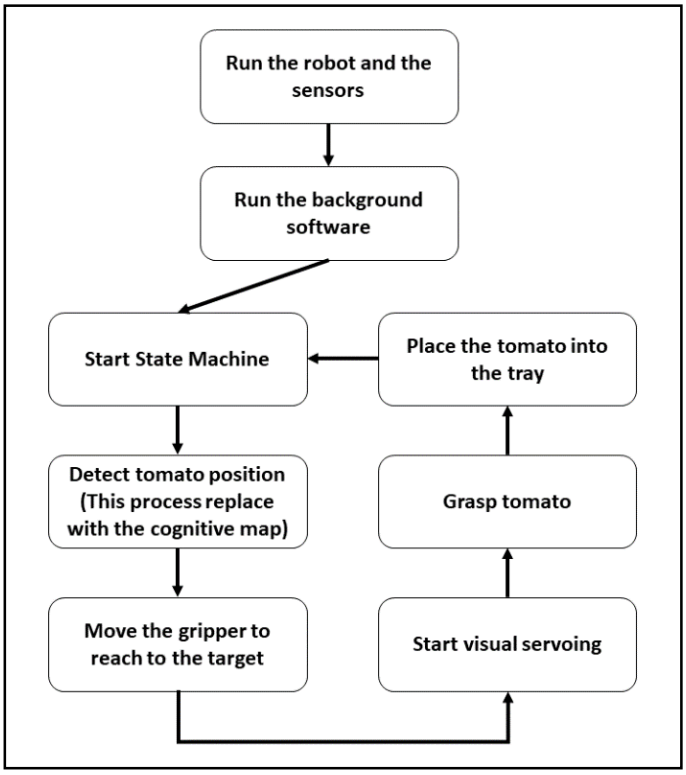

Figure C- 2: GummiArm state machine flowchart. The detection process was replaced with the propose cognitive map.

Figure C- 1 shows the setup between the platform and the GummiArm where the platform fixed on the top of GummiArm. A calibration process was done between the GummiArm and the stereo vision platform to ensure the correct transformation when the GummiArm process to pick the targets. The calibration process is based on the work Tsai and Lenz (1989)[13]. The output of the calibration process is a transformation matrix.

The cognitive map is an independed node that store targets data and list them depend on the priority. The cognitive map required parameters to priorities the data. These parameters are (I) the transformation matrix between the robot link and the platform link (i.e. targets in the cognitive map transform to robot base_link) (II) Robot workspace (i.e. define the limits of the robot by define the work space), and (III) priorities the parameters (i.e. pick the closest target first or pick the target with large probability etc.). Finally, the cognitive map is publishing the target that most suitable for the input parameters.

[13] The work of Tsai and Lenz (1989) was re-written into ROS packages with a generic structure to work with any manipulator arm and camera.

To sum up, the propose platform and system follow the standard procedure to integrate any vision system with a manipulator arm. In this work, the integration uses an exciting package to calibrate the vision system with the manipulator arm.

Appendix D

Computer Vision

This appendix presents the analysis of a fixed stereo vision system that consist of two cameras observing the same scene. First, a single pinhole camera model is discussed then the stereo model is presented.

Appendix D.1 Single Camera Model

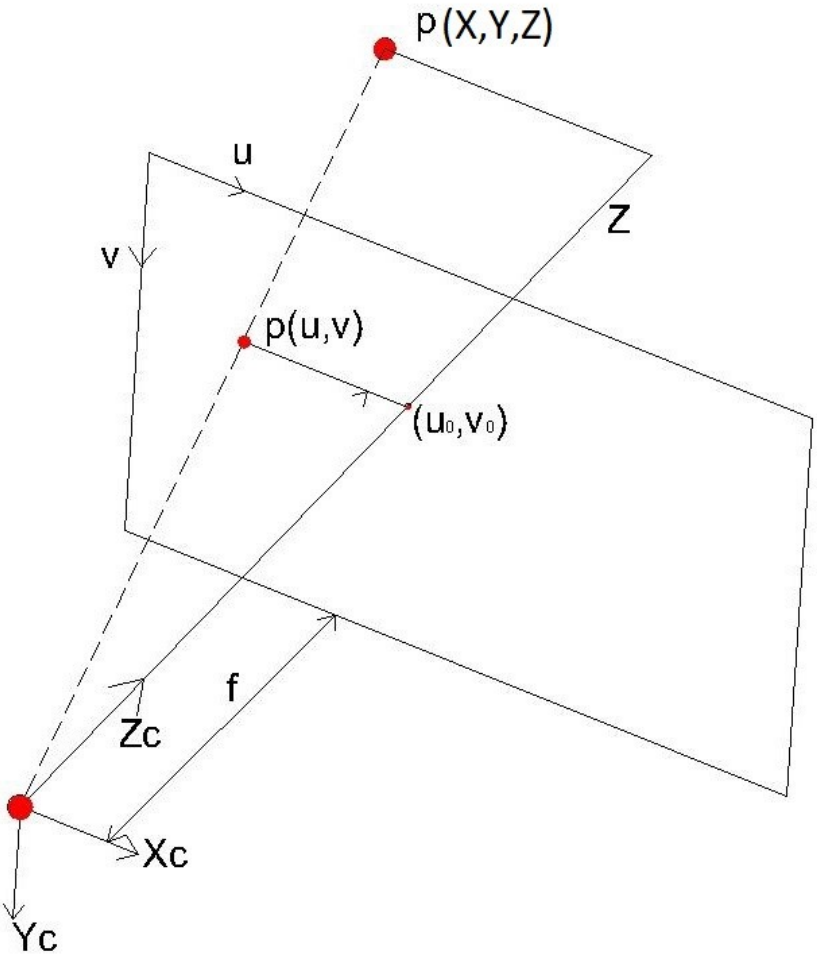

Figure D- 1: Pinhole camera model.

Figure D- 1shows the simple camera model, where the camera coordinate (0,0,0) C is the center of the camera and the image plane has its own coordinate (u, v). (u_0, v_0) is the

point intersects with the Z axis of the coordinate of the camera and the image plane. f is the distance from the center of camera to the image plan which called the focal length. p is a point in front of the camera plane with attached coordinate (X, Y, Z). In the image plane, this point p is present in 2D coordinate (u_p, v_p). Applying a right triangle trigonometry analysis, the following Eq. (D.1) and (D.2) are established.

$$\frac{u_p}{\mathrm{X}} = \frac{f}{Z} \tag{D.1}$$

$$\frac{v_p}{\mathrm{Y}} = \frac{f}{Z} \tag{D.2}$$

Let us re-write Eq. (D.1) and (D.2)to represent the equations in the image coordinate. Note that, the image coordinate start from the top left of the image plane for Adrian and Gary (2008). The results are in (D.3) and (D.4)

$$\frac{u_p - u}{\mathrm{X}} = \frac{f}{Z} \tag{D.3}$$

$$\frac{v_p - v}{\mathrm{Y}} = \frac{f}{Z} \tag{D.4}$$

The above equations show that only X and Y coordinates can be estimated as a scalar while Z cannot be calculated. Eq. (D.3) and (D.4) can be written in matrix form eq.(D.5).

$$\begin{bmatrix} u_p \\ v_p \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{D.5}$$

Or

$$p = MP \tag{D.6}$$

M contain the internal parameters of the camera, p is the point in image coordinate and P is the world coordinate of the point.

Appendix D.2 Stereo camera Model

Understanding the sources of error in the stereo vision system is important to implement improvements. In this section, the sources of error are presented, starting with the geometrical errors and then the errors generated by the image processing derived from the stereo model (Figure D- 2). The errors are classified into two categories, namely, geometry coordinate and image coordinate. The geometry coordinate error is the error that leads to differences in the geometric size of the system, such as the baseline or yaw angle. Meanwhile, an image coordinate error is the error generated within the pixel dimensions, such as the pixel size and focal length. Generally, these two types of error directly affect the output of the system with regard to the 3D reconstruction of objects' shapes and the measurement distance of the targeted objects (Kanatani, 2005).

Figure D- 2 shows the model of the stereo system cameras. Point P refers to an object that is located in front of the system and has a coordinate of (X, Y, Z), which is related to the origin of the system. Each camera has an origin of O_r and O_l ; these origins lie on the same axis (i.e. x-axis), separated by distance b. The plane in front of the z-axis of each camera is where the image projection is formed, and the distance from the cameras' origin to this plane is referred to as the focal length f. The right and left planes have their own coordinates (x_r, y_r) and (x_l, y_l), respectively.

In Figure D- 2, two triangles (o_l, o_r, P) and (x_l, x_r, P) are formed, which are similar triangles,

$$\frac{x_l - x_r}{b} = \frac{f}{Z} \ . \tag{D.7}$$

Hence,

$$Z = f \times \left(\frac{b}{x_l - x_r}\right). \tag{D.8}$$

Equation (D.8) shows how to calculate the depth of point P based on the focal length and the baseline, where $(x_l - x_r)$ refers to the disparity d, which is the difference between the left and right points with regard to image coordinates. Replacing $(x_l - x_r)$ with d yields

$$Z = f \times \frac{b}{d}. \tag{D.9}$$

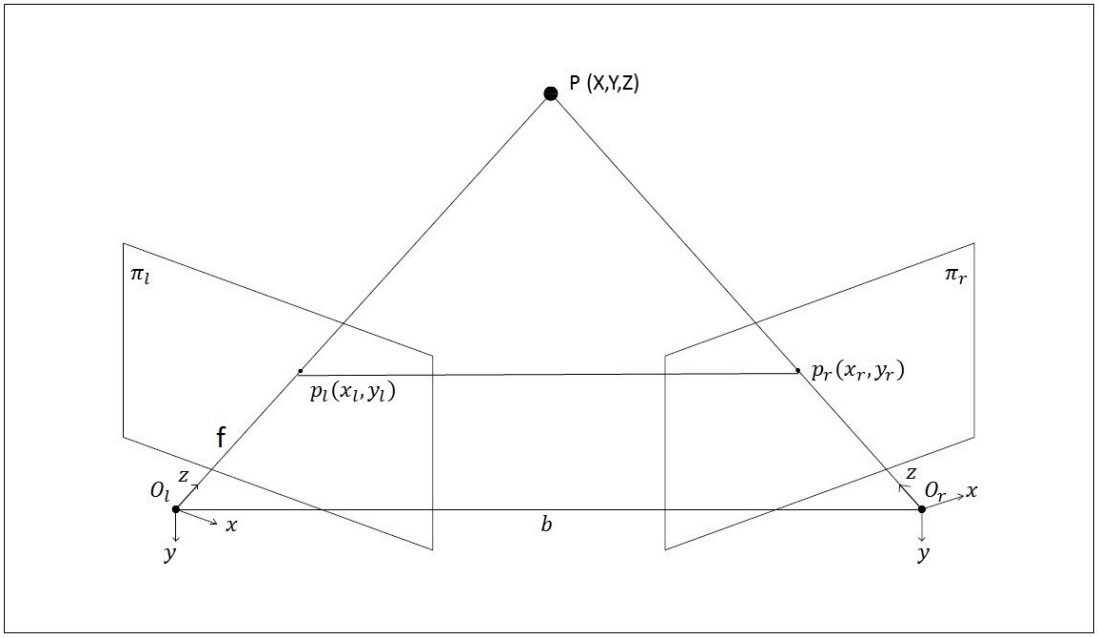

Figure D- 2: Stereo vision mode of the two cameras.

The errors in d are related to the errors in various measurement quantities, thus allowing for an error in d as

$$d = d_o \pm \delta d. \tag{D.10}$$

Hence, Equation (D.9) can be written as

$$Z = f \times \frac{b}{d_o \pm \delta d}. \tag{D.11}$$

The result of depth Z will be affected owing to the errors in Equation (D.11). Thus, we write

$$Z = Z_o \pm \delta Z. \tag{D.12}$$

A Taylor expansion is applied to solve the depth error δZ in Equation (D.11):

$$\delta Z = f \times \frac{b}{d_o^2}\, \delta d \tag{D.13}$$

or equivalently

$$\delta Z = \frac{Z^2}{f \times b} \times \delta d. \tag{D.14}$$

Equation (D.14) represents the error in measuring the depth of the object. The error in estimating the distance to the object is the formation of a quadratic relation to the depth error, but the depth error is inversely proportional to the baseline and focal length. As shown in Figure D- 3, each camera can be associated with two imaginary projection lines that are generated from one pixel. These lines intersect with that from the other camera in the intersection area, thereby forming a diamond shape. The height of the diamond represents δZ. Generally, as shown in Figure D-2, δZ increases as the baseline decreases.

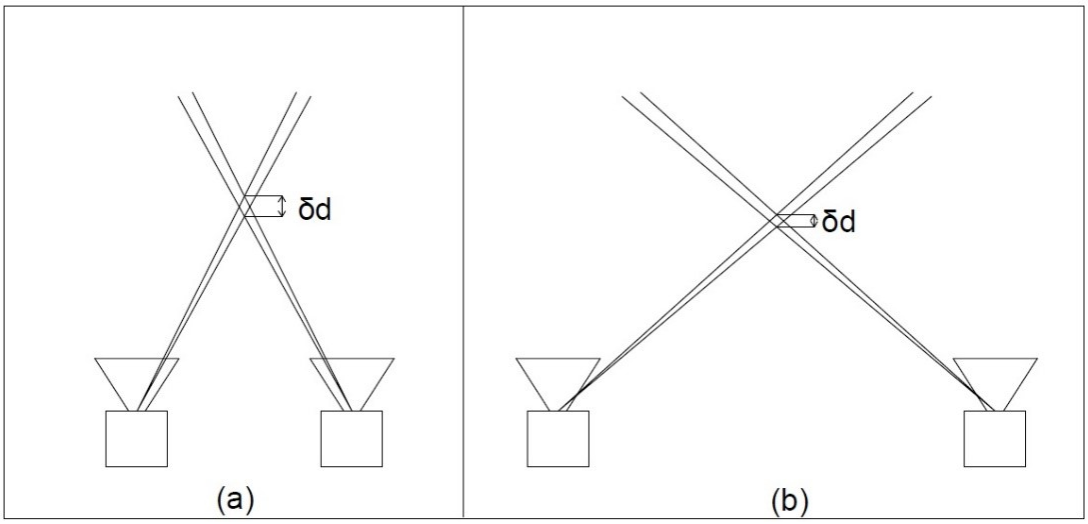

Figure D- 3:Depth error based on the baseline, where the diamond height is equal to the error in depth. (a) Short baseline with large error and (b) twice the baseline in (a) with smaller depth error

The above analysis has shown that error in depth measures can be related to the geometry of the platform.

The error analysis in this section was extended to identify the errors in different variables of the stereo system. Table D- 1 shows the error source of different variables in the system and how these errors can affect the geometrical and image coordinate and the 3D reconstruction (Thao Dang et al., 2009).

If the yaw angle is determined as the source of error, then the depth error will increase proportional to the depth in the scene. The yaw angle has large negative effects on the wider field of view of the cameras rather than on the narrower field of view; these effects cause an increment in the vertical misalignment (Thao Dang et al., 2009).

Table D- 1: Error source and its effects on the stereo vision system [source: Thao Dang et al. (2009)].

error source	**effect on normalized image coordinates** ("$\sim$" denotes normalized coordinates)	**effect on pixel coordinates** $\Delta\mathbf{x} = \begin{pmatrix}\Delta d \\ \Delta y\end{pmatrix}$ **disparity error scan line error**	**linear sensitivity of 3D reconstruction**
yaw error $\Delta\Psi_L$	$\boldsymbol{\Delta}\tilde{\mathbf{x}}_L \approx \begin{pmatrix}1+\tilde{x}_L^2 \\ \tilde{x}_L\tilde{y}_L\end{pmatrix}\Delta\Psi_L$	$\Delta d \approx f_L(1+\tilde{x}_L^2)\Delta\Psi_L$ $\Delta y \approx f_L\tilde{x}_L\tilde{y}_L\Delta\Psi_L$	$\frac{\Delta Z}{\Delta\Psi} \approx -\frac{Z^2}{b}(1+\tilde{x}_L^2)$
pitch error $\Delta\Phi_L$	$\boldsymbol{\Delta}\tilde{\mathbf{x}}_L \approx -\begin{pmatrix}\tilde{x}_L\tilde{y}_L \\ 1+\tilde{y}_L^2\end{pmatrix}\Delta\Phi_L$	$\Delta d \approx -f\tilde{x}_L\tilde{y}_L\Delta\Phi_L$ $\Delta y \approx -f(1+\tilde{y}_L^2)\Delta\Phi_L$	$\frac{\Delta Z}{\Delta\Phi_L} \approx \frac{Z^2}{b}\tilde{x}_L\tilde{y}_L$
roll error $\Delta\Theta_L$	$\boldsymbol{\Delta}\tilde{\mathbf{x}}_L \approx \begin{pmatrix}-\tilde{y}_L \\ \tilde{x}_L\end{pmatrix}\Delta\Theta_L$	$\Delta d \approx (y_L - c_y)\Delta\Theta_L$ $\Delta y \approx (x_L - c_x)\Delta\Theta_L$	$\frac{\Delta Z}{\Delta\Theta_L} \approx \frac{Z^2}{b}\tilde{y}_L$
baseline error Δb	$\boldsymbol{\Delta}\tilde{\mathbf{x}}_{L/R} \approx \pm\begin{pmatrix}\frac{d_N}{2b} \\ 0\end{pmatrix}\Delta b$	$\Delta d \approx \frac{\Delta b}{b}d$ $\Delta y \approx 0$	$\frac{\Delta Z}{\Delta b} \approx -\frac{Z}{b}$
center offset $\boldsymbol{\Delta}\mathbf{c}_L$	$\boldsymbol{\Delta}\mathbf{x}_L \approx \boldsymbol{\Delta}\mathbf{c}_L$	$\Delta d \approx \Delta c_{L,x}$ $\Delta y \approx \Delta c_{L,y}$	$\frac{\Delta Z}{\Delta c_{L,x}} \approx -\frac{Z^2}{bf}$
focal length error Δf_L (one camera only)	$\boldsymbol{\Delta}\mathbf{x}_L \approx (\mathbf{x}_L - \mathbf{c})\frac{\Delta f_L}{f}$	$\Delta d \approx (x_L - c_x)\frac{\Delta f_L}{f}$ $\Delta y \approx (y_L - c_y)\frac{\Delta f_L}{f}$	$\frac{\Delta Z}{\Delta f_L} \approx -\frac{Z^2}{bf^2}(x_L - c_x)$
focal length error Δf (both cameras)	$\boldsymbol{\Delta}\mathbf{x}_{L/R} \approx (\mathbf{x}_{L/R} - \mathbf{c})\frac{\Delta f}{f}$	$\Delta d \approx \frac{\Delta f}{f}d$ $\Delta y \approx 0$	$\frac{\Delta Z}{\Delta f} \approx -\frac{Z}{f}$

References

Agrawal, S., Jha, S., Dewangan, C., 2016. Grading of Tomatoes using Digital Image Processing on the Basis of Color. Int. J. Res. Eng. Technol. 5, 138–140.

Aragon-Camarasa, G., Fattah, H., Paul Siebert, J., 2010. Towards a unified visual framework in a binocular active robot vision system. Rob. Auton. Syst. 58, 276–286. https://doi.org/10.1016/j.robot.2009.08.005

Banz, C., Hesselbarth, S., Flatt, H., Blume, H., Pirsch, P., 2010. Real-time stereo vision system using semi-global matching disparity estimation: Architecture and FPGA-implementation, in: 2010 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation. IEEE, pp. 93–101. https://doi.org/10.1109/ICSAMOS.2010.5642077

Baqersad, J., Poozesh, P., Niezrecki, C., Avitabile, P., 2017. Photogrammetry and optical methods in structural dynamics – A review. Mech. Syst. Signal Process. 86, 17–34. https://doi.org/https://doi.org/10.1016/j.ymssp.2016.02.011

Barnich, O., Van Droogenbroeck, M., 2011. ViBe: A Universal Background Subtraction Algorithm for Video Sequences. IEEE Trans. Image Process. 20, 1709–1724. https://doi.org/10.1109/TIP.2010.2101613

Ben-Tzvi, P., Xu, X., 2010. An embedded feature-based stereo vision system for autonomous mobile robots. ROSE 2010 - 2010 IEEE Int. Work. Robot. Sensors Environ. Proc. 176–181. https://doi.org/10.1109/ROSE.2010.5675303

Bichot, N.P., 2001. Attention, Eye Movements, and Neurons: Linking Physiology and Behavior, in: Vision and Attention. Springer New York, New York, NY, pp. 209–232. https://doi.org/10.1007/978-0-387-21591-4_11

BISKUP, B., SCHARR, H., SCHURR, U., RASCHER, U.W.E., 2007. A stereo imaging system for measuring structural parameters of plant canopies. Plant. Cell Environ. 30, 1299–1308. https://doi.org/10.1111/j.1365-3040.2007.01702.x

Bjorkman, M., Eklundh, J.-O., 2002. Real-time epipolar geometry estimation of binocular stereo heads. IEEE Trans. Pattern Anal. Mach. Intell. 24, 425–432. https://doi.org/10.1109/34.990147

Bota, S., Nedevschi, S., 2011. Tracking multiple objects in urban traffic environments using dense stereo and optical flow, in: 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 791–796. https://doi.org/10.1109/ITSC.2011.6082960

Bradski, G.R., Kaehler, A., 2008. Learning OpenCV : computer vision with the OpenCV library. O'Reilly.

Brown, D.C., 1971. Close-range camera calibration. Photogramm. Eng. 37, 855–866.

Brown, M.Z., Burschka, D., Hager, G.D., 2003. Advances in computational stereo. IEEE Trans. Pattern Anal. Mach. Intell. 25, 993–1008. https://doi.org/10.1109/TPAMI.2003.1217603

Brown, Myron Z., Burschka, D., Hager, G.D., 2003. Advances in computational stereo. IEEE Trans. Pattern Anal. Mach. Intell. 25, 993–1008. https://doi.org/10.1109/TPAMI.2003.1217603

Brunton, A., Chang, S., Roth, G., 2006. Belief propagation on the GPU for stereo vision. Third Can. Conf. Comput. Robot Vision, CRV 2006 2006. https://doi.org/10.1109/CRV.2006.19

Cao, X.H., Stojkovic, I., Obradovic, Z., 2016. A robust data scaling algorithm to improve classification accuracies in biomedical data. BMC Bioinformatics 17, 359. https://doi.org/10.1186/s12859-016-1236-x

Cave, K.R., Wolfe, J.M., 1990. Modeling the role of parallel processing in visual search. Cogn. Psychol. 22, 225–271. https://doi.org/https://doi.org/10.1016/0010-0285(90)90017-X

Chen, Y., Zhang, R. hua, Shang, L., 2014. A Novel Method of Object Detection from a Moving Camera Based on Image Matching and Frame Coupling. PLoS One 9, e109809. https://doi.org/10.1371/journal.pone.0109809

Choi, S.-B., Jung, B.-S., Ban, S.-W., Lee, M., 2004. Trainable attention model based vergence control for active stereo vision system. Intell. Sensors, Sens. Networks Inf. Process. Conf. 2004. Proc. 2004 519–524. https://doi.org/10.1109/ISSNIP.2004.1417515

Connor, C.E., Egeth, H.E., Yantis, S., 2004. Visual Attention: Bottom-Up Versus Top-Down. Curr. Biol. 14, R850–R852. https://doi.org/10.1016/J.CUB.2004.09.041

Constante, P., Gordon, A., Chang, O., Pruna, E., Acuna, F., Escobar, I., 2016. Artificial Vision Techniques to Optimize Strawberry's Industrial Classification. IEEE Lat. Am. Trans. 14, 2576–2581. https://doi.org/10.1109/TLA.2016.7555221

Cyganek, B., Siebert, J.P., 2009. An introduction to 3D computer vision techniques and algorithms. John Wiley & Sons.

Dang, T., Hoffmann, C., Stiller, C., 2009. Continuous stereo self-calibration by camera parameter tracking. IEEE Trans. Image Process. 18, 1536–1550. https://doi.org/10.1109/TIP.2009.2017824

Dankers, A., Barnes, N., Zelinsky, A., 2007. MAP ZDF segmentation and tracking using active stereo vision: Hand tracking case study. Comput. Vis. Image Underst. 108, 74–86. https://doi.org/10.1016/j.cviu.2006.10.013

Dankers, A., Barnes, N., Zelinsky, A., 2004. Active Vision – Rectification and Depth Mapping. Aust. Conf. Robot. Autom.

Dankers, A., Zelinsky, A., 2004. CeDAR: A real-world vision system: Mechanism, control and visual processing, in: Machine Vision and Applications. Springer-Verlag New York, Inc., pp. 47–58. https://doi.org/10.1007/s00138-004-0156-3

Das, S., Ahuja, N., 1995. Performance analysis of stereo, vergence, and focus as depth cues for active vision. IEEE Trans. Pattern Anal. Mach. Intell. 17, 1213–1219. https://doi.org/10.1109/34.476513

Dawson-Howe, K., 2014. A Practical Introduction to Computer Vision with OpenCV, 1st ed. Wiley Publishing.

Denman, S., Fookes, C., Sridharan, S., 2010. Group Segmentation During Object Tracking Using Optical Flow Discontinuities, in: 2010 Fourth Pacific-Rim Symposium on Image and Video Technology. IEEE, pp. 270–275. https://doi.org/10.1109/PSIVT.2010.52

Doyle, F.J., 1964. The Historical Development of Analytical Photogrammetry *. New York, NY.

Draper, B.A., Lionelle, A., 2005. Evaluation of selective attention under similarity transformations. Comput. Vis. Image Underst. 100, 152–171. https://doi.org/https://doi.org/10.1016/j.cviu.2004.08.006

Drath, R., Horch, A., 2014. Industrie 4.0: Hit or Hype? [Industry Forum]. IEEE Ind. Electron. Mag. 8, 56–58. https://doi.org/10.1109/MIE.2014.2312079

Driscoll, J.A., Peters, R.A., Cave, K.R., 1998. A visual attention network for a humanoid robot, in: Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190). IEEE, Victoria, pp. 1968–1974. https://doi.org/10.1109/IROS.1998.724894

Du, G., Chen, M., Liu, C., Zhang, B., Zhang, P., 2018. Online Robot Teaching With Natural Human–Robot Interaction. IEEE Trans. Ind. Electron. 65, 9571–9581. https://doi.org/10.1109/TIE.2018.2823667

Engel, S., Zhang, X., Wandell, B., 1997. Colour tuning in human visual cortex measured with functional magnetic resonance imaging. Nature 388, 68.

Facciolo, G., 2015. MGM : A Significantly More Global Matching for Stereovision. Bmvc2015 5. https://doi.org/10.5244/C.29.90

Fang, Z., Xu, D., Tan, M., 2011. A Vision-Based Self-Tuning Fuzzy Controller for Fillet Weld Seam Tracking. IEEE/ASME Trans. Mechatronics 16, 540–550. https://doi.org/10.1109/TMECH.2010.2045766

Forsyth, D., Ponce, J., 2012. Computer vision : a modern approach. Pearson.

Fouda, Y., Ragab, K., 2013. An efficient implementation of normalized cross-correlation image matching based on pyramid, in: 2013 International Joint Conference on Awareness Science and Technology & Ubi-Media Computing (ICAST 2013 & UMEDIA 2013). IEEE, pp. 98–103. https://doi.org/10.1109/ICAwST.2013.6765416

Frintrop, S., 2006. VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/11682110

Frintrop, S., Rome, E., Christensen, H.I., 2010. Computational Visual Attention Systems and Their Cognitive Foundations: A Survey. ACM Trans. Appl. Percept. 7, 6:1--6:39. https://doi.org/10.1145/1658349.1658355

Fua, P., 1993. A parallel stereo algorithm that produces dense depth maps and preserves image features. Mach. Vis. Appl. 6, 35–49. https://doi.org/10.1007/BF01212430

Fusiello, A., Fusiello, A., Trucco, E., Trucco, E., Verri, A., Verri, A., 2000. A compact algorithm for recti cation of stereo pairs. Mach. Vis. Appl. 16–22. https://doi.org/10.1007/s001380050120

Gao, H., Liu, Y., Li, D., Yu, Y., 2017. Vision localization algorithms for apple bagging robot, in: 2017 29th Chinese Control And Decision Conference (CCDC). pp. 135–140. https://doi.org/10.1109/CCDC.2017.7978080

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F.J., Marín-Jiménez, M.J., 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognit. 47, 2280–2292. https://doi.org/https://doi.org/10.1016/j.patcog.2014.01.005

Georgoulas, C., Andreadis, I., 2010. FPGA based disparity map computation with vergence control. Microprocess. Microsyst. 34, 259–273. https://doi.org/10.1016/j.micpro.2010.05.003

Gibaldi, A., Canessa, A., Sabatini, S.P., 2015. Vergence Control Learning through Real V1 Disparity Tuning Curves 22–24. https://doi.org/10.1109/NER.2015.7146627

Gibaldi, A., Vanegas, M., Canessa, A., Sabatini, S.P., 2017. A Portable Bio-Inspired Architecture for Efficient Robotic Vergence Control. Int. J. Comput. Vis. 121, 281–302. https://doi.org/10.1007/s11263-016-0936-z

Grady, L., 2006. Random walks for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 28, 1768–1783. https://doi.org/10.1109/TPAMI.2006.233

Gräßl, C., Zinßer, T., Niemann, H., 2003. Illumination Insensitive Template Matching with Hyperplanes, in: Michaelis, B., Krell, G. (Eds.), Pattern Recognition. Springer

Berlin Heidelberg, Berlin, Heidelberg, pp. 273–280.

Gruner, H., 1976. Photogrammetry: 1776-1976.

H. Hirschmüller, H., 2005. Accurate and efficient stereo processing by semi-global matching and mutual information. IEEE Int. Conf. Comput. Vis. Pattern Recognit. 2, 807–814.

Häni, N., Isler, V., 2016. Visual servoing in orchard settings, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2946–2953. https://doi.org/10.1109/IROS.2016.7759456

Hart, J., Scassellati, B., Zucker, S.W., 2008. Epipolar Geometry for Humanoid Robotic Heads, in: Caputo, B., Vincze, M. (Eds.), Cognitive Vision. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 24–36. https://doi.org/10.1007/978-3-540-92781-5_3

Hartley, R., Zisserman, A., 2003. Multiple view geometry in computer vision, 2nd ed. cambridge university press, cambridge.

Hayashu, S., Ganno, K., Ishii, Y., Tanaka, I., 2002. Robotic Harvesting System for Eggplants. Japan Agric. Res. Q. JARQ 36, 163–168. https://doi.org/10.6090/jarq.36.163

Heinke, D., Humphreys, G.W., Centre, B.S., 2004. Computational Models of Visual Selective Attention. A Review, in: In Connectionist Models in Psychology. Psychology Press, pp. 273–312.

Hermann, S., Klette, R., 2013. Iterative Semi-Global Matching for Robust Driver Assistance Systems. Springer Berlin Heidelberg, pp. 465–478. https://doi.org/10.1007/978-3-642-37431-9_36

Hirschmuller, H., 2008. Stereo Processing by Semiglobal Matching and Mutual Information. IEEE Trans. Pattern Anal. Mach. Intell. 30, 328–341. https://doi.org/10.1109/TPAMI.2007.1166

Hirschmüller, H., 2011. Semi-Global Matching Motivation, Developments and Applications. Photogramm. Week 173–184.

Hirschmüller, H., Scharstein, D., 2007. Evaluation of Cost Functions for Stereo Matching 1–8.

Hu, W.C., Chen, C.H., Chen, T.Y., Huang, D.Y., Wu, Z.C., 2015. Moving object detection and tracking from video captured by moving camera. J. Vis. Commun. Image Represent. 30, 164–180. https://doi.org/10.1016/j.jvcir.2015.03.003

Ilea, D.E., Whelan, P.F., 2011. Image segmentation based on the integration of colour–texture descriptors—A review. Pattern Recognit. 44, 2479–2501.

https://doi.org/https://doi.org/10.1016/j.patcog.2011.03.005

Itti, L., Koch, C., Niebur, E., 1998. A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. 20, 1254–1259. https://doi.org/10.1109/34.730558

Jensen, J.R., 2007. Remote sensing of the environment : an earth resource perspective, 2nd ed. Pearson Prentice Hall, NJ.

Jiang, R., Jáuregui, D., White, K., 2008. Close-range photogrammetry applications in bridge measurement: Literature review. Measurement 41, 823–834. https://doi.org/https://doi.org/10.1016/j.measurement.2007.12.005

Kale, K., Pawar, S., Dhulekar, P., 2015. Moving object tracking using optical flow and motion vector estimation, in: 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions). IEEE, pp. 1–6. https://doi.org/10.1109/ICRITO.2015.7359323

Kanatani, K., 2005. Statistical Optimization for Geometric Computation: Theory and Practice, 1st ed. Dover Publications, New York.

Kapach, K., Barnea, E., Mairon, R., Edan, Y., Ben-Shahar, O., 2012. Computer Vision for Fruit Harvesting Robots – State of the Art and Challenges Ahead. Int. J. Comput. Vis. Robot. 3, 4–34. https://doi.org/10.1504/IJCVR.2012.046419

Kaur, K., Guptata, O.P., 2017. A Machine Learning Approach to Determine Maturity Stages of Tomatoes. Orient. J. Comput. Sci. Technol. 10, 683–690. https://doi.org/10.13005/ojcst/10.03.19

Kazmi, W., Foix, S., Alenyà, G., Andersen, H.J., 2014. Indoor and outdoor depth imaging of leaves with time-of-flight and stereo vision sensors: Analysis and comparison. ISPRS J. Photogramm. Remote Sens. 88, 128–146. https://doi.org/https://doi.org/10.1016/j.isprsjprs.2013.11.012

Khan, A., Aragon-Camarasa, G., Siebert, J.P., 2016. A Portable Active Binocular Robot Vision Architecture for Scene Exploration BT - Towards Autonomous Robotic Systems, in: Alboul, L., Damian, D., Aitken, J.M. (Eds.), . Springer International Publishing, Cham, pp. 214–225.

Kise, M., Zhang, Q., 2008. Development of a stereovision sensing system for 3D crop row structure mapping and tractor guidance. Biosyst. Eng. 101, 191–198. https://doi.org/https://doi.org/10.1016/j.biosystemseng.2008.08.001

Klarquist, W., Bovik, A., 1997. Adaptive variable baseline stereo for vergence control. 1997 Ieee Int. Conf. Robot. Autom. - Proceedings, Vols 1-4 1952–1959. https://doi.org/10.1109/ROBOT.1997.619074

Koch, C., Ullman, S., 1985. Shifts in selective visual attention: towards the underlying

neural circuitry. Hum. Neurobiol. 4, 219–27.

Kolekar, M.H., 2002. An Algorithm for Designing Optimal Gabor Filter for Segmenting Multi-Textured Images. IETE J. Res. 48, 181–187. https://doi.org/10.1080/03772063.2002.11416274

Krantz, D.H., 1975. Color measurement and color theory: II. Opponent-colors theory. J. Math. Psychol. 12, 304–327. https://doi.org/10.1016/0022-2496(75)90027-9

Krotkov, E., Henriksen, K., Kories, R., 1990. Stereo ranging with verging cameras. IEEE Trans. Pattern Anal. Mach. Intell. 12, 1200–1205. https://doi.org/10.1109/34.62610

Kwon, H., Park, J., Kak, A.C., 2007. A New Approach for Active Stereo Camera Calibration, in: Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE, pp. 3180–3185. https://doi.org/10.1109/ROBOT.2007.363963

Kyriakoulis, N., Gasteratos, A., Mouroutsos, S.G., 2008. Fuzzy vergence control for an active binocular vision system, in: IEEE (Ed.), 2008 7th IEEE International Conference on Cybernetic Intelligent Systems, CIS 2008. pp. 1–5. https://doi.org/10.1109/UKRICIS.2008.4798931

Levitt, J.B., Lund, J.S., 1997. Contrast dependence of contextual effects in primate visual cortex. Nature 387, 73.

Li, Y., Li, Y.F., Wang, Q.L., Xu, D., Tan, M., 2010. Measurement and Defect Detection of the Weld Bead Based on Online Vision Inspection. IEEE Trans. Instrum. Meas. 59, 1841–1849. https://doi.org/10.1109/TIM.2009.2028222

Lin, Y., Tong, Y., Cao, Y., Zhou, Y., Wang, S., 2017. Visual-Attention-Based Background Modeling for Detecting Infrequently Moving Objects. IEEE Trans. Circuits Syst. Video Technol. 27, 1208–1221. https://doi.org/10.1109/TCSVT.2016.2527258

Loan, T.T.K., Pham, X.-Q., Nguyen, H.-Q., Tri, N.D.T., Thai, N.Q., Huh, E.-N., 2015. Homography-Based Motion Detection in Screen Content, in: Park, D.-S., Chao, H.-C., Jeong, Y.-S., Park, J.J. (Jong H. (Eds.), Advances in Computer Science and Ubiquitous Computing. Springer Singapore, Singapore, pp. 875–881.

Lowe, D.G., 1999. Object Recognition from Local Scale-Invariant Features, in: Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2, ICCV '99. IEEE Computer Society, Washington, DC, USA, pp. 1150--.

Luong, Q.-T., Faugeras, O.D., 1997. Self-Calibration of a Moving Camera from Point Correspondences and Fundamental Matrices. Int. J. Comput. Vis. 22, 261–289. https://doi.org/10.1023/A:1007982716991

Maeda, Y., Nakamura, T., 2015. View-based teaching/playback for robotic manipulation. ROBOMECH J. 2, 2. https://doi.org/10.1186/s40648-014-0025-4

Marefat, M.M., Wu, L., Yang, C.C., 1997. Gaze stabilization in active vision—I. Vergence error extraction. Pattern Recognit. 30, 1829–1842. https://doi.org/10.1016/S0031-3203(97)00066-6

Miau, F., Papageorgiou, C., Itti, L., 2001. Neuromorphic algorithms for computer vision and attention, in: Bosacchi, B., Fogel, D.B., Bezdek, J.C. (Eds.), Proc. SPIE 46 Annual International Symposium on Optical Science and Technology. SPIE Press, Bellingham, WA, pp. 12–23.

Milanese, R., 1993. Detecting Salient Regions in an Image : From Biological Evidence to Computer Implementation. Ph.D Theses, Univ. Geneva.

Mohamed, A., Culverhouse, P.F., Cangelosi, A., Yang, C., 2018a. Depth Estimation Based on Pyramid Normalized Cross-correlation Algorithm for Vergence Control. IEEE ACCESS.

Mohamed, A., Culverhouse, P.F., Cangelosi, A., Yang, C., 2018b. Active Stereo Platform: Online Epipolar Geometry Update. EURASIP J. Image Video Process. 2018:54, 16. https://doi.org/10.1186/s13640-018-0292-8

Mohamed, A., Culverhouse, P.F., Cangelosi, A., Yang, C., 2018c. Integrate a Visual Attention Model with a Binocular Platform for Harvesting Tomatoes. IEEE ACCESS.

Mohamed, A., Culverhouse, P.F., De Azambuja, R., Cangelosi, A., Yang, C., 2017. Automating Active Stereo Vision Calibration Process with Cobots. IFAC-PapersOnLine 50, 163–168. https://doi.org/10.1016/j.ifacol.2017.12.030

Mohamed, A., Yang, C., Cangelosi, A., 2016. Stereo Vision based Object Tracking Control for a Movable Robot Head. IFAC-PapersOnLine 49, 155–162. https://doi.org/10.1016/J.IFACOL.2016.07.106

Muhlmann, K., Maier, D., Hesser, R., Manner, R., 2001. Calculating dense disparity maps from color stereo images, an efficient implementation, in: Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001). IEEE Comput. Soc, pp. 30–36. https://doi.org/10.1109/SMBV.2001.988760

Nakabo, Y., Mukai, T., Hattori, Y., Takeuchi, Y., Ohnishi, N., 2005. Variable Baseline Stereo Tracking Vision System Using High-Speed Linear Slider, in: Proceedings of the 2005 IEEE International Conference on Robotics and Automation. IEEE, pp. 1567–1572. https://doi.org/10.1109/ROBOT.2005.1570337

Nations, U., 2017. World Population Prospects The 2017 Revision. New York.

Patil, S., Nadar, J.S., Gada, J., Motghare, S., Nair, S.S., 2013. Comparison of Various Stereo Vision Cost Aggregation Methods. Int. J. Eng. Innov. Technol. 2, 222–226.

PEARSON, K., 1905. The Problem of the Random Walk. Nature 72, 294–294.

https://doi.org/10.1038/072294b0

Pfeiffer, S., 2017. The Vision of "Industrie 4.0" in the Making-a Case of Future Told, Tamed, and Traded. Nanoethics 11, 107–121. https://doi.org/10.1007/s11569-016-0280-3

Prince, S., 2012. Computer Vision: Models Learning and Inference. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511996504

Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A., 2009. ROS: an open-source Robot Operating System, in: ICRA Workshop on Open Source Software.

R. Jiménez, A., Ceres, R., L. Pons, J., 2000. A SURVEY OF COMPUTER VISION METHODS FOR LOCATING FRUIT ON TREES. Trans. ASAE 43, 1911. https://doi.org/https://doi.org/10.13031/2013.3096

Rakun, J., Stajnko, D., Zazula, D., 2011. Detecting fruits in natural scenes by using spatial-frequency based texture analysis and multiview geometry. Comput. Electron. Agric. 76, 80–88. https://doi.org/10.1016/J.COMPAG.2011.01.007

Reddy, N.V., Vishnu, A. V, Reddy, V., Pranavadithya, S., Jagadesh Kumar, J., 2016. A CRITICAL REVIEW ON AGRICULTURAL ROBOTS. A Crit. Rev. Agric. Robot. Int. J. Mech. Eng. Technol. 7, 183–188.

Ren, X., Wang, Y., 2016. Design of a FPGA hardware architecture to detect real-time moving objects using the background subtraction algorithm, in: 2016 5th International Conference on Computer Science and Network Technology (ICCSNT). IEEE, pp. 428–433. https://doi.org/10.1109/ICCSNT.2016.8070194

Rosenblum, L.D., 2010. See what I'm saying : the extraordinary powers of our five senses. W.W. Norton.

Rougeaux, S., Kita, N., Kuniyoshi, Y., Sakane, S., Section, A.S., 1993. Tracking A Moving Object With A Stereo Camera Head. n Proc. 11th Annu. Conf. Robot. Soc. Japan 1–4.

Rusinkiewicz, S., Levoy, M., 2001. Efficient variants of the ICP algorithm, in: Proceedings Third International Conference on 3-D Digital Imaging and Modeling. IEEE Comput. Soc, Quebec, pp. 145–152. https://doi.org/10.1109/IM.2001.924423

Rusu, R.B., Cousins, S., 2011. 3D is here: Point Cloud Library (PCL), in: 2011 IEEE International Conference on Robotics and Automation. IEEE, pp. 1–4. https://doi.org/10.1109/ICRA.2011.5980567

Sabater, N., Morel, J.-M., Almansa, A., 2011. How Accurate Can Block Matches Be in Stereo Vision? SIAM J. Imaging Sci. 4, 472–500. https://doi.org/10.1137/100797849

Sahabi, H., Basu, A., 1996. Analysis of error in depth perception with vergence and spatially varying sensing. Comput. Vis. Image Underst. 63, 447–461. https://doi.org/10.1006/cviu.1996.0034

Salmane, H., Ruichek, Y., Khoudour, L., 2011. Object tracking using Harris corner points based optical flow propagation and Kalman filter, in: 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 67–73. https://doi.org/10.1109/ITSC.2011.6083031

Sang De Ma, 1996. A self-calibration technique for active vision systems. IEEE Trans. Robot. Autom. 12, 114–120. https://doi.org/10.1109/70.481755

Sapienza, M., Hansard, M., Horaud, R., 2013. Real-time visuomotor update of an active binocular head. Auton. Robots 34, 35–45. https://doi.org/10.1007/s10514-012-9311-2

Scharstein, D., Szeliski, R., 2001. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms 47, 7–42. https://doi.org/10.1023/A:1014573219977

Shibata, M., Honma, T., 2002. 3D object tracking on active stereo vision robot, in: 7th International Workshop on Advanced Motion Control. Proceedings (Cat. No.02TH8623). IEEE, pp. 567–572. https://doi.org/10.1109/AMC.2002.1026983

Shields, C., 2016. Aristotle's Psychology, in: Zalta, E.N. (Ed.), The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University.

Sicard, P., Levine, M.D., 1989. Joint recognition and tracking for robotic arc welding. IEEE Trans. Syst. Man. Cybern. 19, 714–728. https://doi.org/10.1109/21.35336

Solé Puig, M., Pérez Zapata, L., Aznar-Casanova, J.A., Supèr, H., 2013. A Role of Eye Vergence in Covert Attention. PLoS One 8, e52955. https://doi.org/10.1371/journal.pone.0052955

Song, H.O., Fritz, M., Goehring, D., Darrell, T., 2016. Learning to Detect Visual Grasp Affordance. IEEE Trans. Autom. Sci. Eng. 13, 798–809. https://doi.org/10.1109/TASE.2015.2396014

Stefano, L. Di, Marchionni, M., Mattoccia, S., 2004. A fast area-based stereo matching algorithm. Image Vis. Comput. 22, 983–1005. https://doi.org/10.1016/j.imavis.2004.03.009

Stoelen, M.F., Bonsignorio, F., Cangelosi, A., 2016. Co-exploring Actuator Antagonism and Bio-inspired Control in a Printable Robot Arm, in: Tuci, E., Giagkos, A., Wilson, M., Hallam, J. (Eds.), From Animals to Animats 14. Springer International Publishing, Cham, pp. 244–255.

Szeliski, R., 2009. Computer Vision: Algorithms and Applications, Computer. Springer.

Tanaka, M., Maru, N., Miyazaki, F., 1994. 3-D tracking of a moving object by an active stereo vision system. Proc. IECON'94 - 20th Annu. Conf. IEEE Ind. Electron. 2, 816–820. https://doi.org/10.1109/IECON.1994.397891

Tejada, V.F., Stoelen, M.F., Kusnierek, K., Heiberg, N., Korsaeth, A., 2017. Proof-of-concept robot platform for exploring automated harvesting of sugar snap peas. Precis. Agric. 18, 952–972. https://doi.org/10.1007/s11119-017-9538-1

Thacker, N., Mayhew, J., 1991a. Optimal combination of stereo camera calibration from arbitrary stereo images. Image Vis. Comput. 9, 27–32. https://doi.org/10.1016/0262-8856(91)90045-Q

Thacker, N., Mayhew, J., 1991b. Optimal combination of stereo camera calibration from arbitrary stereo images. Image Vis. Comput. 9, 27–32. https://doi.org/10.1016/0262-8856(91)90045-Q

Timings, T., 2008. Fabrication and welding engineering. Newnes.

Tippetts, B.J., Lee, D.J., Archibald, J.K., Lillywhite, K.D., 2011. Dense disparity real-time stereo vision algorithm for resource-limited systems. IEEE Trans. Circuits Syst. Video Technol. 21, 1547–1555. https://doi.org/10.1109/TCSVT.2011.2163444

Treisman, A.M., Gelade, G., 1980. A feature-integration theory of attention. Cogn. Psychol. 12, 97–136. https://doi.org/https://doi.org/10.1016/0010-0285(80)90005-5

Treue, S., 2003. Visual attention: The where, what, how and why of saliency. Curr. Opin. Neurobiol. https://doi.org/10.1016/S0959-4388(03)00105-3

Trucco, E., Verri, A., 1998. Introductory techniques for 3-D computer vision. Prentice Hall PTR, Upper Saddle River, NJ, USA.

Tsai, R.Y., Lenz, R.K., 1989. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. IEEE Trans. Robot. Autom. 5, 345–358. https://doi.org/10.1109/70.34770

Tsang, E.K.C., Shi, B.E., 2006. Active Binocular Gaze Control Inspired by Superior Colliculus, in: The 2006 IEEE International Joint Conference on Neural Network Proceedings. IEEE, pp. 7–14. https://doi.org/10.1109/IJCNN.2006.246652

Uliano, K.C., 1992. Operator / equipment Performance Measures : Results Of Literature Search Operator / 1: quipment Performall1ce Measures : Results of Literature Search.

Underwood, J.P., Hung, C., Whelan, B., Sukkarieh, S., 2016. Mapping almond orchard canopy volume, flowers, fruit and yield using lidar and vision sensors. Comput. Electron. Agric. 130, 83–96. https://doi.org/https://doi.org/10.1016/j.compag.2016.09.014

Unger, C., Ilic, S., 2014. A Stochastic Cost Function for Stereo Vision. Bmvc 1–11.

VanRullen, R., 2003. Visual saliency and spike timing in the ventral visual pathway. J. Physiol. 97, 365–377. https://doi.org/10.1016/J.JPHYSPARIS.2003.09.010

Veksler, O., 2003. Fast variable window for stereo correspondence using integral images. 2003 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, 2003. Proceedings. I-556-I–561. https://doi.org/10.1109/CVPR.2003.1211403

Walther, D., Itti, L., Riesenhuber, M., Poggio, T., Koch, C., 2002. Attentional Selection for Object Recognition – A Gentle Way, in: Proceedings of the Second International Workshop on Biologically Motivated Computer Vision, BMCV '02. Springer-Verlag, London, UK, UK, pp. 472–479.

Wang, L., Gong, Mingwei, Gong, Minglun, Yang, R., 2006. How far can we go with local optimization in real-time stereo matching. Third Int. Symp. 3D Data Process. Vis. Transm. 129–136.

Wang, L., Liao, M., Gong, M., Yang, R., Nister, D., 2007. High-quality real-time stereo using adaptive cost aggregation and dynamic programming. Proc. - Third Int. Symp. 3D Data Process. Vis. Transm. 3DPVT 2006 798–805. https://doi.org/10.1109/3DPVT.2006.75

Wang, Z., Guo, L., Wang, S., Chen, L., Wang, H., 2017. Review of Random Walk in Image Processing. Arch. Comput. Methods Eng. 1–18. https://doi.org/10.1007/s11831-017-9225-4

Westheimer, G., 2004. Center-surround antagonism in spatial vision: Retinal or cortical locus? Vision Res. 44, 2457–2465. https://doi.org/https://doi.org/10.1016/j.visres.2004.05.014

Won Jin Kim, In-So Kweon, 2011. Moving object detection and tracking from moving camera, in: 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). IEEE, pp. 758–759. https://doi.org/10.1109/URAI.2011.6146005

Wu, J., Smith, J.S., Lucas, J., 1996. Weld bead placement system for multipass welding [using transputer-based laser triangulation vision system]. IEE Proc. - Sci. Meas. Technol. 143, 85–90. https://doi.org/10.1049/ip-smt:19960163

Xiang, R., Jiang, H., Ying, Y., 2014. Recognition of clustered tomatoes based on binocular stereo vision. Comput. Electron. Agric. 106, 75–90. https://doi.org/10.1016/J.COMPAG.2014.05.006

Xiong, Y., Matthies, L., 1997. Error analysis of a real-time stereo system, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 1087–1093. https://doi.org/10.1109/CVPR.1997.609465

Xu, Y., Dong, J., Zhang, B., Xu, D., 2016. Background modeling methods in video analysis: A review and comparative evaluation. CAAI Trans. Intell. Technol. 1, 43–60. https://doi.org/10.1016/J.TRIT.2016.03.005

Yim, C., Bovik, A.C., 1994. Using a Hierarchical Image Structure Pb ' rbl. Control 0–5.

Yu, H., Baozong, Y., 1996. Zero disparity filter based on wavelet representation in the active vision system. Proc. Third Int. Conf. Signal Process. 1, 279–282. https://doi.org/10.1109/ICSIGP.1996.567163

Zhang, X., Tay, L.P., 2011. A spatial variant approach for vergence control in complex scenes. Image Vis. Comput. 29, 64–77. https://doi.org/10.1016/J.IMAVIS.2010.08.005

Zhang, Z., 2000. A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. 22, 1330–1334. https://doi.org/10.1109/34.888718

Zhao, Y., Gong, L., Huang, Y., Liu, C., 2016a. A review of key techniques of vision-based control for harvesting robot. Comput. Electron. Agric. 127, 311–323. https://doi.org/https://doi.org/10.1016/j.compag.2016.06.022

Zhao, Y., Gong, L., Zhou, B., Huang, Y., Liu, C., 2016b. Detecting tomatoes in greenhouse scenes by combining AdaBoost classifier and colour analysis. Biosyst. Eng. https://doi.org/10.1016/j.biosystemseng.2016.05.001