uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Alain Boyer
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.ASc. (Electrical and Computer Engineering)
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Adaptive Structured Light Imaging with Exposure and Focus Fusion for 3D Reconstruction and Autonomous Robotic Exploration

TITRE DE LA THÈSE / TITLE OF THESIS

P. Payeur
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

V. Aitken

J. Lang

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Adaptive Structured Light Imaging with Exposure and Focus Fusion for 3D Reconstruction and Autonomous Robotic Exploration

Alain Boyer

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfilment of the requirements
For the Master of Applied Science degree in Electrical and Computer Engineering

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

Your file Votre référence
ISBN: 978-0-494-79704-4
Our file Notre référence
ISBN: 978-0-494-79704-4

Abstract

This thesis research proposes several significant improvements to a previously developed structured light range sensor in order to enhance its robustness. By applying modern methods to classical structured light techniques, the improved sensor is capable of adapting to many different environments and generating 3D surface reconstructions of more general and unconstrained scenes. This is achieved by combining several algorithms in parallel, which permits the sensor to adapt in a reliable and autonomous manner to multiple colours, reflective characteristics and depths of field of the scene. The main motivation of this research is to ultimately mount the range sensor on a mobile platform, and perform autonomous navigation, mapping, modelling and exploration of complex environments. This thesis presents enhancements to the processing stage of the sensor, a complete overhaul of the acquisition stage, as well as a comprehensive set of results that demonstrate how the sensor adapts to the environment. Also, a complete prototype for a robotic mobile exploration system is presented and tested, validating the methods and techniques presented in this work.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

Scene reconstruction, a fundamental task in computer vision research, has received much attention over the years. The problem of reconstruction is to design a vision system capable of measuring a scene and representing the surfaces it detects by generating a 3D model. These models can be used for various applications such as modelling, recognition and mapping.

Passive and active vision are two complementary categories of techniques that can be used to perform scene reconstruction. The main difference is that passive vision simply observes the scene, while active vision projects a pattern, usually using light, and analyzes how the pattern interacts with the scene to make measurements. Passive vision has a high dependence on the presence of features in the scene and, at best, can only generate sparse 3D information. On the other hand, active vision can perform precise measurements and generate high resolution 3D reconstructions. Active vision has been widely researched and is still the preferred method for dense 3D range sensing and reconstruction.

Two active vision technologies have gathered much interest over the years. Laser range sensors are accepted as the state-of-the-art standard for 3D measurement since they can achieve very high accuracy with very low computation time. However, they require specialized hardware that is usually expensive and in general simply not feasible for many smaller scale applications. Structured light range sensors, on the other hand, are regarded as an affordable solution for 3D range sensing since they can be assembled using common off-the-shelf digital cameras and liquid crystal display (LCD) projectors. Although the precision and resolution of structured light sensors may not be as high as their laser-based counterparts, they produce accurate and dense range scans that are not achievable with passive vision.

This thesis research builds upon the previous version of a structured light range sensor developed by Desjardins [1] and presents many significant improvements. The objective is to make the active vision system more robust while improving its ability to adapt to the scene. As a result, the new sensor can not only perform more reliable object measurement but also reconstruct more general and unconstrained scenes containing many objects, colours, reflectance characteristics and depths of field.

1.1 Motivations

The main motivation of this research is to further drive the development of classical structured light sensors by applying modern methods, such as dynamic range imaging, adaptive computer vision and intelligent image processing to significantly enhance the performance of 3D range sensors. Not only is the resulting range sensor a low-cost solution made of components that are readily available, but the framework and technology places much of the complexity within software. This makes it easy to integrate more complex algorithms and advanced processing in order to increase the adaptability of the sensor.

The long-term goal is to mount the new and improved structured light range sensor onto a robotic mobile platform and perform autonomous navigation, mapping, modelling and exploration of complex and uncontrolled environments. Using the new sensor is beneficial as it provides a complete low-cost solution to generate precise 3D range data with rich colour information to be analyzed by higher level algorithms. Since the acquisition system is designed with autonomous robotics in mind, it has the ability to scale well as it can change from mapping and navigation to high-resolution reconstruction and modelling when approaching objects of interest.

1.2 Objectives

As stated above, the main objective of this thesis research is to build upon a previously developed structured light sensor prototype and improve its effectiveness by making it flexible, robust and adaptable. The central goal is to refine the sensor such that it is a black box device used to acquire 3D range data, for many different applications, with the press of a single button. This is primarily achieved by addressing the current system's limitations in both its acquisition and processing stages. The results is that the existing sensor is transformed from a functional prototype to a robust and usable system.

Most of the processing stage is incrementally improved and refined to allow for better performance and robustness. The main objectives are as follows.

- Improve the calibration of extrinsic parameters to allow for robust calibration target detection and more accurate metric parameter estimation.
- Enhance the colour segmentation of the structured light pattern in order to remove unnecessary algorithm parameters and unreliable post processing.

- Develop a more flexible structured light code detection algorithm to handle angled and curved surfaces that produce skewed codes on the image plane.
- Add the functionality to interpolate a surface from the generated 3D point cloud to more easily visualize and interpret the data produced by the sensor.

The acquisition stage is totally redesigned to allow for three major improvements that enable an automated acquisition procedure. These enhancements address the primary goal of this research, which is to design a structured light sensor that can automatically adapt to the scene. The key objectives are as follows.

- Eliminate the problems that arise when projecting a colour coded pattern onto a scene with multiple colours and significant colour variation.
- Compensate for multiple object brightness and scene reflectance characteristics during each camera capture.
- Increase the depth of field of the sensor in order to capture data from multiple focus planes and extend the range of the sensor.

Throughout this thesis, relevant solutions, in the form of improvements to the existing sensor prototype, are presented to address each of the above objectives. The combination of these improvements enables an entirely autonomous acquisition procedure that can be used to acquire as much 3D information as possible from a wide variety of unknown and unconstrained scenes and environments.

1.3 Thesis Organization

This thesis consists of seven chapters. After the introduction, a review of scholarly literature pertaining to the technologies used within this work is presented in Chapter 2. A detailed description of the former range sensor implementation is given followed by a comparison of techniques and algorithms relating to the sensor improvements. Chapter 3 discusses the various enhancements to the algorithms of the processing stage. The precise calibration, robust code segmentation, flexible code detection and better mesh visualization are presented along with results demonstrating the improvements. Chapter 4 presents the major changes made to the acquisition stage that make the sensor adaptable to many scenes. The new time-multiplexed acquisition, exposure fusion image capture and focus plane analysis are treated and validated with concrete results that demonstrate the new

adaptability of the system. Chapter 5 discusses how all of the proposed enhancements are integrated into a complete 3D measurement and reconstruction system. The high-level acquisition and processing stage procedures are summarized and the operating characteristic of the sensor are discussed. Chapter 6 presents a prototype scene reconstruction system using the improved range sensor. The high-level framework is explained and results are given showing the robustness of the sensor and validating the current work. Finally, Chapter 7 concludes the thesis by summarizing the key contributions and listing possible directions for future work.

Chapter 2 Literature Review

Range sensors, leveraging structured light technology, have been widely researched over the years. However, the recurring theme is that most sensors tend to be designed for controlled environments. This is present in many forms such as constraints on the colour of objects being imaged, the imaging of a single object at once and the need to maintain a constant distance between the sensor and the object of interest for example. The goal of this research is to eliminate these constraints and produce a range sensor that can adapt by itself and operate in many different environments.

This chapter presents a review of certain key technologies that work together in order to produce a range sensor capable of acquiring information from unknown and unconstrained scenes and environments. The first section describes the former prototype of a structured light range sensor that is the basis for this work. A general overview is given and some limitations are discussed. The second and third sections respectively present exposure and focus fusion. These technologies form the basis by which the sensor is enhanced in order to render it adaptable to its environment. The fourth and fifth sections provide reviews related to registration and mesh generation, which are used for the development of a modelling system based on the range sensor.

2.1 Structured Light Range Sensor

The structured light range sensor, initially developed by Desjardins *et al.* in [1], [2] and [3], is a data acquisition module capable of measuring an object or scene and producing a 3D point cloud reconstruction. The sensor relies on a robust active vision technique using structured light to generate features that are matched within a stereo pair of images and then triangulated to generate 3D data. The main advantages of this setup are that the sensor is still capable of acquiring 3D data on featureless objects and remains low cost since it is composed of readily available consumer-grade electronic components. The physical system is composed of two cameras mounted on a rigid bracket as a stereo pair above an LCD projector as shown in Figure 2.1. The cameras and projector are driven by a conventional personal computer where all the controls and algorithms are implemented in software.

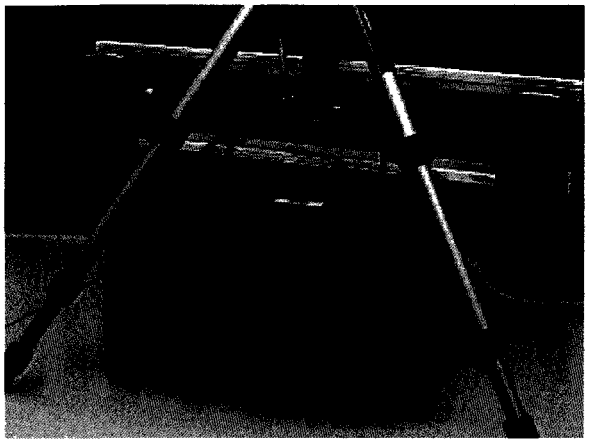


Figure 2.1. Stereoscopic structured light range sensor.

2.1.1 Calibration

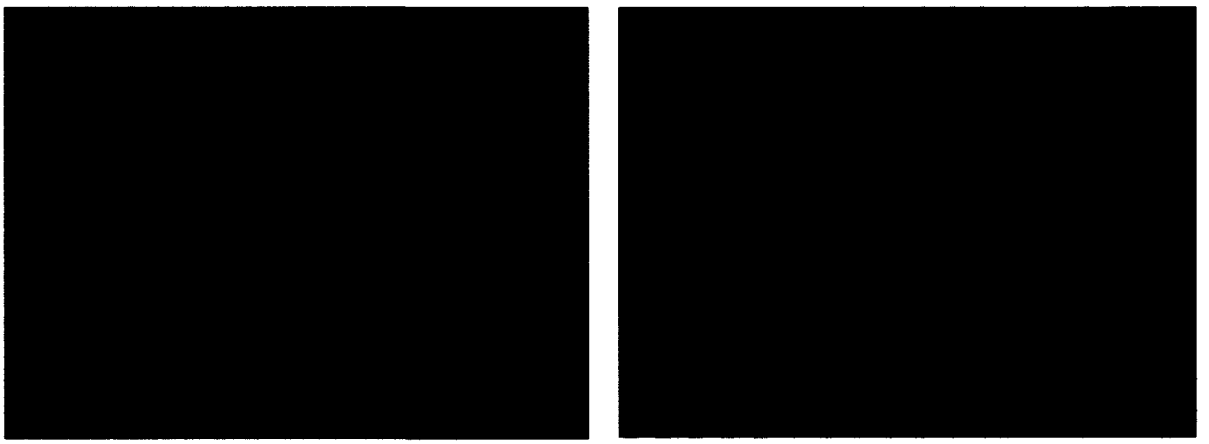
The only calibration required is of the intrinsic and extrinsic camera parameters, leaving the projector uncalibrated with the rest of the system. This provides maximum flexibility for the sensor to be moved between acquisitions when measuring objects from different points of view. Also, the focus and zoom parameters of the projector can be adjusted in accordance with the depth of the scene so that the projected pattern always appears sharp and in focus.

The intrinsic parameters of each camera are computed separately since it is required to place the calibration target such that it occupies as much of the field of view as possible. The calibration target consists of a chessboard pattern mounted on a flat surface. It is waved in front of the camera to be calibrated and several images are acquired. Some simple image processing is performed to find the image coordinates of the corners, which are then passed to Zhang's algorithm [4] to compute the intrinsic parameters. The parameters include the focal length of the lens, the size of CCD pixels and the principle point of the CCD array, as well as the radial distortion of the image.

The extrinsic parameters consist of the translation T and rotation R between the cameras, both of which are critical for accurate 3D reconstruction. Again, the same chessboard calibration target is waved before the cameras, ensuring that it is fully visible in both fields of view, and images are acquired. The corners are detected and Zhang's algorithm is used to compute the translation and rotation between each camera and the calibration target. These transformations are easily combined and an estimation of T and R between the cameras is derived for each pair of images. The translation and rotation vectors are averaged over all pairs of images to obtain a global estimate of T and R between the cameras.

2.1.2 Pattern

To achieve a dense 3D reconstruction on featureless objects, the structured light technique is applied using a bi-dimensional pseudo-random (PR) pattern. The PR pattern consists of a two dimensional array of coloured squares, shown in Figure 2.2, that has been generated offline using the pseudo-random iterative approach of Morano *et al.* [5]. The pattern is defined such that each 3x3 neighbourhood of squares is a unique code word of length 9 composed with 3 square symbols encoded using the colours red, green and blue. The uniqueness of each code word ensures accurate matching between the stereo pair of images.



(a) (b)

Figure 2.2. (a) Bi-dimensional pseudo-random pattern and (b) a zoomed in view.

2.1.3 Acquisition Stage

To perform structured light, the projection of the pattern and acquisition of stereo images are synchronized within an acquisition stage. Before the acquisition can begin, several parameters are set, such as the gains for the colour channels and the exposure time of the cameras. This ensures that the captured colours are similar to those projected and that the images are not under- or over-exposed. First, left and right reference images are acquired with no projected pattern. Second, the pattern is projected and a second pair of images is acquired. Finally, background subtraction is performed and the resulting left and right difference images are saved to disk for further processing.

This spatial-neighbouring pattern is combined with a time-multiplexed [6] approach which iteratively marches the entire pattern horizontally and vertically during the acquisition stage.

As the pattern is marched, images are acquired and difference images are computed and stored to disk. This provides a dynamic way of increasing the spatial density of artificial features and subsequent 3D points. The marching, or shifts, can be adjusted depending on the desired reconstruction density.

2.1.4 Processing Stage

Once the images are acquired, the processing stage begins where the imaged pattern is analyzed and a 3D point cloud is generated. The image processing starts by converting the images to the HSV colour space and performing a histogram analysis to determine the dominant colours in the images [7]. These colours are used to segment the image into three masks representing the red, green and blue squares. Next, the masks are labelled using a connected component analysis to identify all of the individual coloured squares. Finally, the size of each region is computed and this data is used to perform a statistical analysis of the masks. Small regions are dropped and large regions are segmented once more using a higher threshold and then re-labelled. The last step consists of computing the centroid of all regions to use as a coordinate reference for the corresponding square.

The 3x3 codes are now recovered from both left and right images by considering each square region and finding its 8 closest neighbours. The neighbours are sorted horizontally and vertically and a look-up in the projected pattern is performed to validate the code. The codes are further validated by computing their confidence based on valid neighbouring codes. Duplicate and low confidence codes are dropped from further processing.

The correspondence problem becomes trivial as the codes that are present in both left and right images are matched using a brute force approach. Outliers are removed by using a RANSAC algorithm to drop point matches characterized by a large distance from their respective epipolar lines. Next, the optimal triangulation of Hartley *et al.* [8] is applied to the resulting point matches and a reconstruction of 3D points is produced. This triangulation technique first adjusts the 2D matches based on best fitting epipolar lines and then performs the reconstruction using the direct linear transform method. Finally, colour information from the reference images is mapped to the 3D coordinates, producing a richer dataset.

2.1.5 Limitations

Since the previously developed sensor is a first prototype, it has many limitations that are

now outlined. First, the extrinsic calibration is weak by design as it relies on multiple extrinsic parameter estimations that are then averaged. This leads to an overall average error of 4mm when comparing the point cloud data with real world measurements. For this type of sensor, sub millimetre accuracy is desired and feasible with proper calibration.

Second, the exposure time and colour gain parameters of the cameras need to be manually selected by the operator for every different scene. This imposes human intervention, in the form of parameter tweaking, before each acquisition. Since one of the objectives consists of mounting the sensor atop a mobile robotic platform, these parameters should be determined automatically.

Third, the colour square segmentation relies on several hard coded thresholds and a histogram analysis that must be post processed. This leads to the inability of handling other pattern colours and no flexibility to handle colour response and reflectance characteristics. By selecting a slightly different approach with different algorithms, the segmentation can be made more reliable and flexible.

Fourth, the code detection assumes that the 3x3 codes are not skewed in the captured images as a result of being projected onto surfaces not perpendicular to the projector's principle axis. This imposes a constraint that objects to be imaged must be mostly planar and positioned perpendicular to the sensor. To allow an arbitrary positioning of the sensor during mobile exploration, the code detection must be able to deal with skewed data and corner cases.

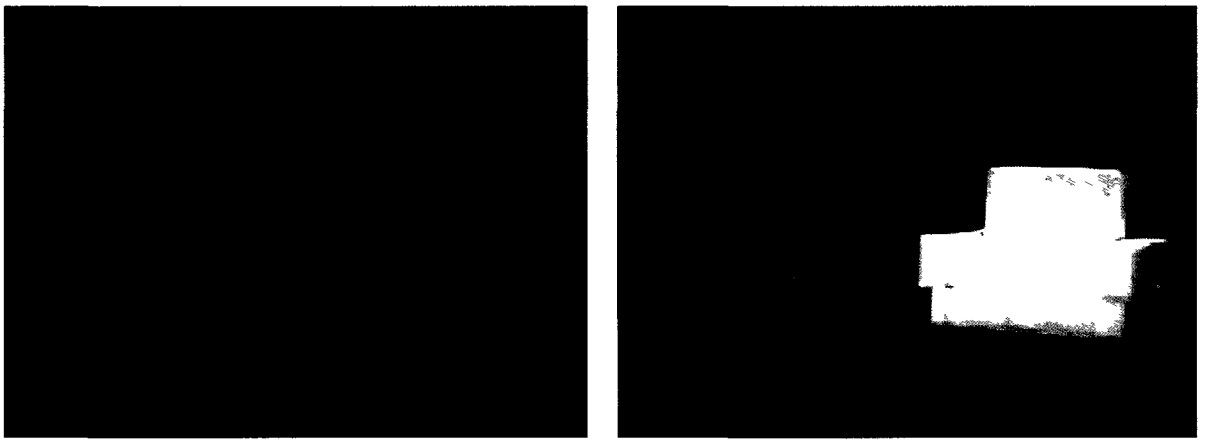
Finally, when an acquisition is launched, it is assumed that the projector is properly focused on the scene. Not only does this require more human intervention before the acquisition, but it also prevents the imaging of several objects located at different depths. To obtain a truly adaptive sensor, the latter must not only determine the focus automatically but also adjust its focus to different depths of field.

All of the constraints and limitations presented above prevent the range sensor from operating in an automated fashion. Since automatic operation is critical in a robotic application, these limitations are further discussed and addressed with flexible solutions throughout this thesis.

2.2 Exposure Fusion

The most important parameter to adjust when acquiring images from a digital camera is the

exposure time, as most other parameters can be set for a broad operational range. Depending on the scene to be imaged, with its multiple colours and reflectance properties, the exposure time will directly affect the saturation of the pixels in the resulting image. Although digital cameras allow for the adjustment of the exposure time, the parameter is global and applies to the entire imaging sensor. Thus, a trade off between different regions in the scene must take place and an optimal global exposure must be selected either manually or automatically. This means that it is common to obtain images with under- and over-exposed regions. Figure 2.3.a shows an example scene captured using a low exposure time of 15ms where the white chair is properly exposed but the dartboard is not very visible. On the other hand, Figure 2.3.b shows the same scene captured using a higher exposure of 35ms where the dartboard is properly exposed but the chair is saturated. The problem is compounded when dealing with the black background as well as when attempting to detect the reflection of a projected pattern.



(a) (b)

Figure 2.3. An example scene captured using (a) a low exposure of 15ms and (b) a high exposure of 35ms.

Although the results may be good enough for human interpretation, computer vision algorithms are usually much more sensitive to exposure and saturation. When dealing with complex scenes, a local exposure time is required to ensure that each region of the image is properly exposed, allowing for correct image processing. The concept of exposure fusion, which is inspired by high dynamic range imaging, is to capture multiple images of the same scene at different exposures and then fuse those images together to produce a single properly exposed image. Different approaches have been proposed and the following summarizes some of the more popular techniques.

The central idea of exposure fusion is to analyze a set of images with varying exposure, determine the properly exposed and poorly exposed regions of the images and then produce a composite image which only includes the properly exposed regions. The approach used to define image regions categorizes the techniques into two groups, block-based techniques and pixel-based techniques.

2.2.1 Block-Based Techniques

Block-based techniques do not take the geometry of the scene into account and blindly partition the image space into blocks. The blocks that form the composite image are selected from the source image where a certain metric is highest among all other source images. Goshtasby [9] assumes that properly exposed regions are associated with high entropy and therefore selects the block with the highest entropy. Instead of computing the entropy using image intensities, as is generally the case, image colours from the CIELab colour space are considered. The entropy is computed by first clustering the 256 dominant colours of the image using [10], approximating each colour by the closest dominant cluster and estimating the entropy based on a ratio between the numbers of pixels in the cluster versus the entire image. Zafar *et al.* [11] assume that properly exposed regions have the most detail and therefore more high frequency components. The Discrete Cosine Transform (DCT) is applied on the blocks and the block with the highest L2 norm of AC coefficients is selected.

Once the blocks are selected and tiled together to form the composite image, visible edges between the blocks will remain. The blocks must be blended together using a blending function such as the one presented in [9]. Despite the fact that Goshtasby describes an algorithm to iteratively compute the optimal block size, these block-based algorithms do not lend themselves well to structured light applications. For the algorithms to be useful, the block size must be large, which introduces constraints on the resolution of object colour and brightness variability as well as the size of the structured light pattern.

2.2.2 Pixel-Based Techniques

Pixel-based techniques offer a finer resolution of local exposure and thus are more easily able to adapt to complex scenes. The composite image is usually formed by weighting the pixels of the source images. Burt *et al.* [12] use the pyramid image decomposition technique

to fuse multiple images and were first to propose this algorithm for images of multiple exposure. The source images undergo pyramid decomposition, multiple weight maps at each pyramid level are computed using a measure of local variance and finally the composite image is formed by applying the weight maps and performing the inverse pyramid transform. Debevec *et al.* [13] use conventional high dynamic range imaging theory to fuse images of multiple exposures. The camera response curve is modelled during a calibration stage to obtain the relation between pixel values and exposure. This curve is then used to fuse the multiple images together, producing a radiance map with no saturated regions and a maximum amount of detail.

The recent work of Mertens *et al.* [14] proposes a technique to perform exposure fusion on a set of N images based on the work of Burt *et al.* [12]. The difference is that the weight maps are computed from quality measures (QM) of the N input images and not the images of each pyramid level. This separates the weight maps from the pyramid levels and allows for the computation of multiple QMs such as contrast, saturation and well-exposedness that are more meaningful measures at the pixel level.

The contrast QM consists of converting the input images to grayscale, applying a Laplacian filter and taking the absolute value of the response as shown in Equation 2.1. The saturation QM involves computing the standard deviation between the colour channels at each pixel of the input images using Equation 2.2. The well-exposedness QM determines how close the intensities of each pixel are to the centre of the range by weighing the intensities using a Gaussian curve as in Equation 2.3. All measures are computed for each pixel and are expressed as 2D weight maps where i and j are the pixel indices and k is the image index.

$$C_k = \left| I_k^G * L \right| \tag{2.1}$$

$$S_k(i,j) = \sqrt{\frac{1}{3}\sum_{C=\{R,G,B\}} \left(I_k^C(i,j) - \mathrm{m}(i,j,I_k)\right)^2}, \text{where } \mathrm{m}(i,j,I_k) = \frac{1}{3}\sum_{C=\{R,G,B\}} \left(I_k^C(i,j)\right) \tag{2.2}$$

$$E_k(i,j) = \exp\left(-\frac{(I_k(i,j)-0.5)^2}{2\sigma^2}\right), \text{where } \sigma = 0.2 \tag{2.3}$$

The quality measures are combined to produce weight maps for each image in the set via a multiplication as detailed in Equation 2.4. The w_C, w_S and w_E exponents provide a mechanism to weight the QMs accordingly and adjust their influence on the exposure fusion process. Since the weight maps are used to compute a weighted average over the set of images, the former are normalized using Equation 2.5 to ensure that each pixel sums to

one.

$$W_k(i,j)=(C_k(i,j))^{w_C}\times(S_k(i,j))^{w_S}\times(E_k(i,j))^{w_E} \tag{2.4}$$

$$W_k(i,j)=\left[\sum_{k'=1}^{N} W_{k'}(i,j)\right]^{-1} W_k(i,j) \tag{2.5}$$

Next, the N original images are decomposed into a Laplacian pyramid via the L{}^{l} operator and the weight maps are decomposed into a Gaussian pyramid via the G{}^{l} operator, where l indicates the level. The images are blended at each level by computing a weighted average of the Laplacian decomposition of the images using the Gaussian decomposition of the weight maps as expressed in Equation 2.6.

$$L\{R\}_{ij}^{l}=\sum_{k=1}^{N} G\{W\}_{ij,k}^{l} L\{I\}_{ij,k}^{l} \tag{2.6}$$

Finally, the resulting pyramid is collapsed, by applying the inverse Laplacian pyramid transform, to produce the final composite image R that is locally exposed. This exposure fusion technique is very straightforward to implement and is flexible in the sense that other quality measures can be computed and easily integrated into the existing algorithm.

2.3 Focus Fusion

Most literature on structured light range sensors assumes that the object or scene being imaged is at a relatively constant distance from the sensor and has a small depth of field. This ensures that the cameras and projector are always in focus and that the entire object or scene along with the projected pattern is clear and sharp. When building a flexible range sensor that must adapt to any scene, this assumption cannot be made. Moreover, when imaging general environments in the context of robotic exploration, the focus problem must be considered in order to detect objects regardless of their distance to the sensor. Following the same approach of exposure fusion, the concept of focus fusion is to capture multiple images of the same scene while varying the focus and then fuse those images together to produce a single image where all elements are properly focused. Several techniques have been proposed and are summarized here.

2.3.1 Block-Based Techniques

To perform focus fusion, the most common approach is to reuse the notion of dividing the

image space into blocks and generate a composite image using the blocks of the source images that are properly focused. The algorithm proposed by Zafar *et al.* [11], as described to perform exposure fusion, can also be applied to source images of varying focus. Larger AC coefficients of a block imply that the block has a sharper focus than the equivalent blocks in the others images and is used to form the composite image.

Another technique relies on multiresolution signal decomposition to analyze the focus of images. De and Chanda [15] propose a new non-linear wavelet constructed from morphological operators in order to perform the focus fusion at multiple resolutions. First, they use their analysis operator to decompose the input images into signal and detail spaces and recursively repeat this at multiple levels. Second, the signal and detail blocks with maximum absolute values are selected as representatives for the fused image. Third, their synthesis operators are used to reconstruct the composite image from the selected signal and detail blocks. This method of focus fusion is similar to other multiresolution analysis and relies heavily on the defined analysis and synthesis wavelet operators. Blocking effects, which are common with multiresolution techniques, can sometimes appear in regions where all input images are blurry. However, their technique uses integer computations and simple arithmetic, rendering it efficient and fast. The non-linear nature of their defined wavelet ensures that edges in the input images are preserved and not blurred in the final composite image.

2.3.2 Region-Based Techniques

A new and emerging category of techniques consists of detecting regions of images that are in focus and mapping those regions directly to the composite image. The central idea proposed by Hariharan *et al.* [16], [17] is to intelligently detect and segment focally connected regions in a set of N input images as opposed to segmenting physically connected regions. This proves more robust when objects span across multiple focus planes. First, horizontal and vertical Sobel masks are applied to the input images resulting in image gradients that represent sharpness in the images. Second, the horizontal and vertical gradients are combined to generate a sharpness mask for each input image as expressed in Equation 2.7, where I^X and I^Y represent the horizontal and vertical gradients respectively and k represents the image index. These sharpness masks are low passed filtered to reduce noise and increase neighbourhood relevance.

$$S_k(i,j)=\sqrt{(I_k^X(i,j))^2+(I_k^Y(i,j))^2} \tag{2.7}$$

Third, a comparison of the sharpness masks at each focus plane is performed to identify maximum values, which correspond to regions of high focus. These regions are expressed as partition masks, which are binary masks that represent focally connected regions, using Equation 2.8.

$$P_k(i,j)=S_k(i,j), \text{if } S_k(i,j)>S_l(i,j), \text{where } k\neq l \tag{2.8}$$

The partition masks are then used in the final step to merge the set of N input images into a locally-focused composite image R. This is achieved by multiplying the partition masks with their respective original image and forming the union set of partitions, as expressed in Equation 2.9.

$$R(i,j)=\bigcup_{k=1}^{N} R_k(i,j), \text{where } R_k(i,j)=P_k(i,j)\times I_k(i,j) \tag{2.9}$$

This method ensures the highest possible focus in the composite image since it directly maps entire sections of input images as opposed to a multiresolution technique.

2.4 Registration

The motivation of this work is to improve the range sensor in order to mount it atop a mobile robotic platform, acquire data from multiple points of view and integrate the resulting datasets together. A major difficulty in achieving this is the registration of the 3D point clouds between two or more viewpoints. There are two different approaches to accomplish 3D registration, both of which are detailed below.

2.4.1 Structured Techniques

The first method relies on feature matching to find corresponding feature points between two datasets and directly compute the translation and rotation parameters using existing closed-form solutions [18]. A minimum of three feature points are necessary. However, to reduce error, many points are used with a least squares minimization. The main difficulty with this technique is reliably detecting features and then matching them between the two datasets.

2.4.2 Unstructured Techniques

The second method is to perform the registration without analyzing the structure of the datasets. The classic technique in this category is the iterative closest point (ICP) algorithm of Besl and McKay [19], which operates in the space domain. It has been widely researched and refined. The algorithm attempts to compute the closest points between two point clouds, estimate the translation and rotation parameters, transform one of the point clouds and finally compute the mean squared error between both point clouds. The process is iterated and the registration estimation is further refined until the error is within an acceptable threshold. Although the algorithm is simple and precise, it tends to converge to a local minimum solution and therefore requires a good initial estimation of the transformation parameters.

Another unstructured method, such as the one presented by Curtis *et al.* [20], [21], consists of transforming the space domain point cloud to the frequency domain in order to decouple the estimation of the translation T and the rotation R. The first step is to voxelize the point clouds and express them as $P_1[\vec{n}]$ and $P_2[\vec{n}]$ where $\vec{n}$ is the space domain index vector. Since the datasets are of the same object or scene but from different points of view, the relation of Equation 2.10 can be defined. The Fourier transform of Equation 2.10 is computed, with $\vec{k}$ as the frequency domain index vector, M as the dimensional scale factor and results in Equation 2.11.

$$P_1[\vec{n}]=P_2[R\vec{n}+T] \tag{2.10}$$

$$F_{P_1}[R\vec{k}]=F_{P_2}[\vec{k}]e^{-j2\pi(R\vec{k})^T MT} \tag{2.11}$$

By considering the amplitude and phase of Equation 2.11 independently, the estimation of rotation and translation can effectively be decoupled as defined in Equations 2.12 and 2.13.

$$\left|F_{P_1}[R\vec{k}]\right|=\left|F_{P_2}[\vec{k}]\right| \tag{2.12}$$

$$\angle F_{P_1}[R\vec{k}]=\angle F_{P_2}[\vec{k}]-2\pi(R\vec{k})^T MT \tag{2.13}$$

Since the points that lie on an arbitrary axis in the space domain do not change under rotation, the axis of rotation is found by taking the absolute difference between the magnitude components and searching for a line of minimal energy that passes through the origin. This is achieved using a neighbourhood path searching algorithm that starts at the origin and follows the minimal energy path toward the edges. The angle of rotation is found

by using a subset of 3D frequency points, iterating over the possible rotations and selecting the angle that produces a minimal sum of absolute difference. Because of the Hermitian symmetry in the frequency domain, two possible solutions for the rotation angle are found. The proper solution is selected with the estimation of the translation parameter. The translation is found by rotating $P_2[\vec{n}]$ by each of the solutions, projecting the points onto the three cardinal axes and cross correlating the result with the projection of $P_1[\vec{n}]$. The distance between the maximum peaks, in the respective cross correlations, corresponds to the translation and the proper rotation solution is identified by the highest peak with lowest noise energy.

Although this technique provides a good global estimate of translation and rotation, the accuracy of the results is not very high. This is due to the use of a low voxel space resolution since the latter is limited by memory. In many cases, such a technique is used to obtain an initial estimate, which is then refined using a more locally accurate technique such as ICP.

2.5 Mesh Generation

The output of the structured light sensor, detailed in Section 2.1, is simply an unorganized point cloud expressed as a list of three dimensional coordinates. Although this representation is more than adequate for high level processing such as obstacle avoidance or object recognition, it is difficult to interpret by human operators. In order to visualize, evaluate and possibly determine the quality of the data, the point cloud must be transformed into a surface. Normally, this surface is represented as a triangular mesh that consists of vertices and faces. Fabio [22] attempted to classify the many different algorithms that transform point clouds to surfaces. In the context of this structured light range sensor, it is important to note two different categories.

2.5.1 Surface Interpolation

The first category of algorithms attempts to directly interpolate the point cloud data and generate a precise surface. The captured 3D points are mapped to vertices in a mesh and triangular faces are extracted to complete the surface.

Point cloud interpolation algorithms can be further subdivided into volume oriented and surface oriented techniques. Volumetric approaches typically consist of computing a volume

tetrahedralization from the point cloud, which is usually done using the 3D Delaunay triangulation [23]. Next, the convex hull of the tetrahedralization is extracted using algorithms similar to marching cubes [24] to generate a closed surface.

Surface approaches typically follow an advancing front method that selects a point at random and iteratively adds neighbouring points to form triangles. If no point can be added, another randomly selected point is considered and the process is repeated until all points in the cloud have been added to the mesh. The advantage of these algorithms, over their volumetric counterparts, is that they generally execute faster and consume less memory. Moreover, it is not only possible to generate closed surfaces, but also opened surfaces that are more relevant to the type of unorganized data coming from the structured light sensor imaging unconstrained scenes as opposed to objects.

The ball pivoting algorithm of Bernardini *et al.* [25] is a surface oriented method that interpolates an unorganized point cloud and generates a surface mesh. The input is a list of 3D points representing a sample of a measured surface and a ball of radius R. It is assumed that the density of the points is known and that the average distance between points is smaller than the diameter 2R of the ball. The algorithm consists of randomly locating three points, P_1, P_2 and P_3 that form a seed triangle such that all points fit inside the ball. The triangle formed by P_1, P_2 and P_3 is appended to the mesh and the edges E_{12}, E_{23} and E_{13} are added to the list of edges that compose the advancing front [26]. An edge E_{12} is selected and the ball is placed in contact with the edge's two boundary points P_1 and P_2. While remaining in contact with the boundary points, the ball is pivoted around the common edge E_{12} until it touches another point P_4. The new triangle formed by P_1, P_2 and P_4 is appended to the mesh and a join operation is performed, which replaces E_{12} with E_{14} and E_{24} in the advancing front. At this time, glue operations are performed to ensure that the redundant inside edge pairs are removed from the advancing front. If the pivoting does not find a point, another edge from the advancing front is selected and the pivoting continues. Once all edges of the advancing front are treated, another seed triangle is found and the process continues until all points have been considered.

This category of surface interpolation is mostly used to visualize the output data of the range sensor as it uses the generated 3D data points. This surface can also be used to evaluate the performance and accuracy of the structured light sensor during quantitative analysis.

2.5.2 Surface Approximation

The second category of algorithms attempts to approximate the surface using mathematical models. These algorithms usually generate a second surface mesh S' that approximates the original surface S as defined by the acquired 3D points P_i. This is done by defining new vertices that are as close as possible to the surface S defined by the original point cloud.

One of the first and widely accepted technique for surface approximation was presented by Hoppe *et al.* [27]. First, a tangent plane T_i is estimated for each P_i using the latter's K nearest neighbours. Second, a signed distance function for arbitrary points R in the space is defined as the distance from R to the tangent plane of the closest 3D point P_i. Finally, the marching cubes [24] algorithm is used to extract a mesh that defines the zero set of the distance function.

The new trend in surface approximation is the use of point set surfaces (PSS) techniques as first defined by Alexa *et al.* [28]. These techniques originated in point-based graphics research and are considered state-of-the-art in surface approximation. The high-level algorithm is very similar to what was proposed by Hoppe [27] but makes use of the moving least squares (MLS) framework [29]. First, a point R in the sample space is selected and a plane H is fit to the local neighbourhood of P_i by minimizing a weighted sum of squared distances. The weights of P_i are computed as a function of the distance between P_i and the projection of R onto H. Second, a polynomial approximation of P_i with respect to H is computed using the heights of P_i over H. Finally, R is projected onto this polynomial, which defines S'. Again, the marching cubes algorithm is used to extract an isosurface mesh representing S'. The advantage here is that the vertices of the mesh can be projected once again onto the MLS surface for increased accuracy.

Algebraic point set surfaces (APSS) is a recent improvement proposed by Guennebaud and Gross [30]. The main difference is that the data is fit to a sphere instead of a simple plane. In order to achieve this, algebraic spheres are used as opposed to geometric spheres since the latter must be found using iterative techniques and perform poorly around planar regions. On the contrary, algebraic spheres are defined using algebraic distance functions, which make them behave like a plane when no curvature is present in the local neighbourhood. The advantage over PSS is that the algorithm is stable in areas of high curvature and performs better in undersampled regions. Also, since the technique makes use of algebraic fitting, it is more general than geometric fitting and can be easily extended

to higher order surfaces.

This category of surface approximation is used for modelling applications as it is capable of parameterizing the 3D point cloud to model the underlying surfaces. This is useful when multiple range data scans of the same scene are acquired and merged together since it is able to compensate for the errors in the scanning and the registration.

2.6 Summary

This chapter presented the relevant literature on which this thesis research is based. First, a review of an existing structured light range sensor was given and its limitations were discussed. Second, exposure and focus fusion algorithms, which are integrated into the sensor, were analyzed to explain the basis of how the sensor will adapt to its environment. Finally, point cloud operations such as registration and mesh generation were reviewed to provide a base for an application of scene reconstruction demonstrating the pertinence of the improvements made to the range sensor.

Chapter 3 Processing Stage Enhancements

The processing stage of the sensor software stack takes left and right images, analyzes them through a set of image processing steps and generates a 3D point cloud as detailed in Section 2.1.4. The main operations are the segmentation of coloured squares produced by the structured light pattern, a statistical analysis to drop small regions and re-segment large ones, the recovery of unique 3x3 codes, the matching of code correspondences between left and right images and finally, the 3D reconstruction via optimal triangulation.

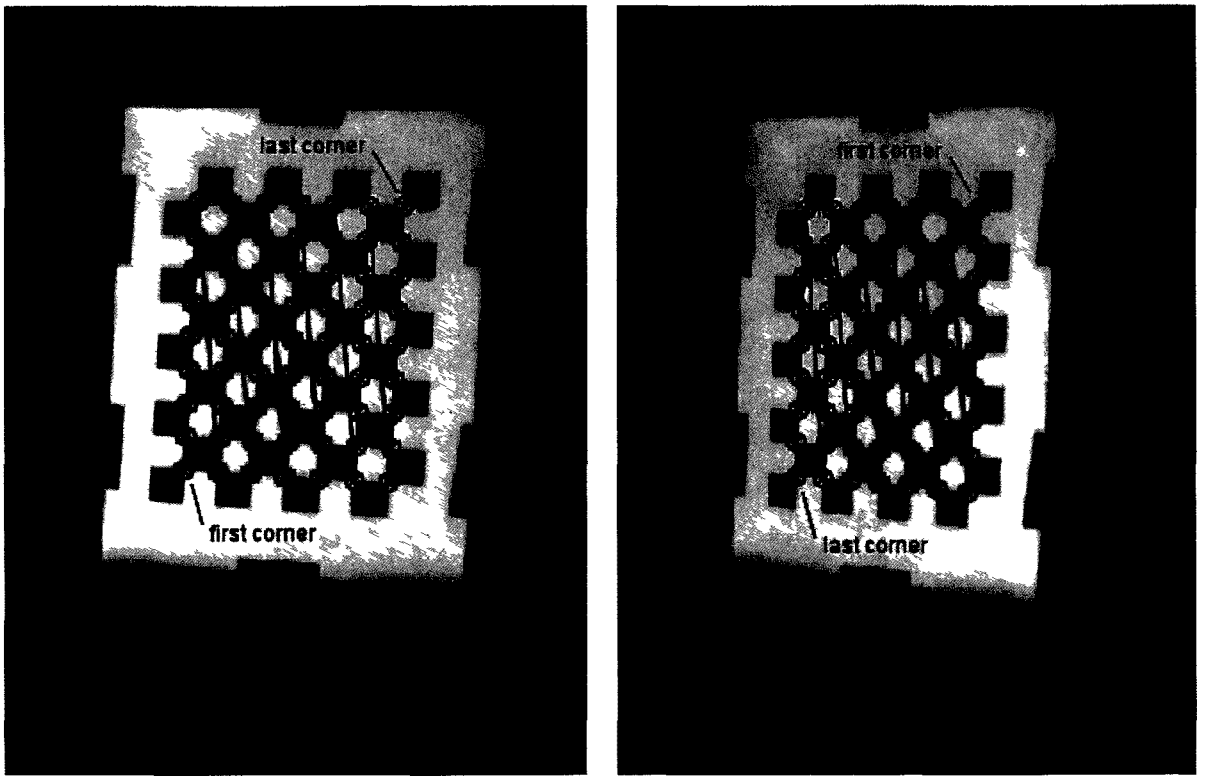
This chapter presents four enhancements to the processing stage that focus on key algorithms. The improvements address the initial extrinsic calibration, the segmentation of coloured squares, the 3x3 code detection and the final surface mesh generation. Each algorithm is presented with a discussion of its current limitations due to the chosen implementation. New and enhanced implementations are presented followed by concrete results demonstrating the improved performance and robustness.

3.1 Precise Extrinsic Calibration

Before beginning the processing stage, calibration of the cameras must be performed. Although the intrinsic calibration is acceptable, the previous method used to perform the extrinsic calibration between the left and right cameras was not optimal. First, the orientation of the chessboard calibration target was not detected robustly. Second, the estimation of the extrinsic parameters was not computed accurately. Finally, the scale factor was not computed at all, making it impossible to extract metric data.

3.1.1 Chessboard Orientation

When performing extrinsic calibration, a chessboard calibration target is waved within the field of view of both cameras while images are acquired. The chessboard corners are detected in both images and then used as matches between images to compute the extrinsic parameters. However, the order of the detected corners in both images is not guaranteed, as shown in Figure 3.1, where the first corner in the left image is located in the bottom-left and the first corner in the right image is located in the top-right. Such detection leads to mismatches between the left and right views, which compromise the correctness of the calibration parameters.



(a) (b)

Figure 3.1. (a) Left and (b) right images of chessboard with corners detected in opposite direction.

To detect this discrepancy, a blue sticker is affixed to the square of the chessboard that is located in the second row from the top and the second column from the left, as shown in Figure 3.2. This eliminates the symmetry of the calibration target and creates a reference for the orientation of the chessboard. With the detected chessboard corners, represented by circles in Figure 3.2, the location of the points of interest P_1^F, P_2^F, P_1^L and P_2^L are computed as the mid-point between corresponding corners. The super index refers to the first or last point groups and the sub index identifies the individual points in each group. The pixels within a 7x7 window about the points of interest are converted to the HSV colour space and the value component of each window is averaged giving V_1^F, V_2^F, V_1^L and V_2^L. Next, the value difference is computed between neighbouring points using Equations 3.1 and 3.2.

$$D^F = V_1^F - V_2^F \tag{3.1}$$

$$D^L = V_1^L - V_2^L \tag{3.2}$$

The results of Equations 3.1 and 3.2 are validated by ensuring that they differ by at least an

order of magnitude using Equation 3.3.

$$\frac{\min(D^F, D^L)}{\max(D^F, D^L)} < 0.1 \tag{3.3}$$

This is achievable since one of the differences will be between the same colour, which should have the same brightness. If the results are invalid, the images are rejected and new ones are acquired; otherwise, Equation 3.3 holds and the orientation detection becomes trivial. When D^F is larger than D^L, the blue sticker is at location P_1^F, which indicates that the corners were detected in the proper order. On the other hand, when D^F is smaller than D^L, the blue sticker is at location P_1^L and the order of the corners is flipped to ensure that the first corner is associated with the block containing the blue sticker.

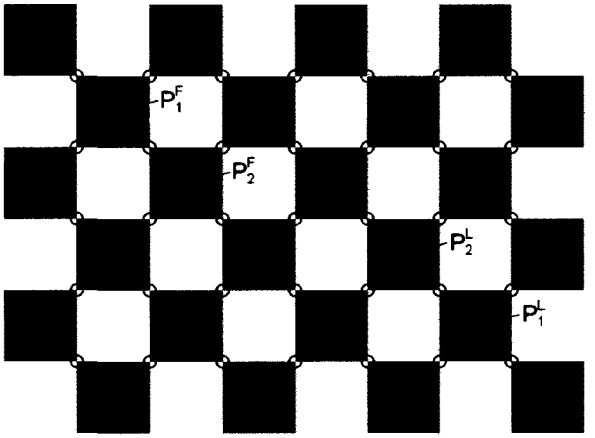


Figure 3.2. Chessboard with blue sticker reference and points of interest.

3.1.2 Epipolar Geometry

In the initial implementation, the actual calibration of extrinsic parameters was done by adapting Zhang's algorithm [4], as explained in Section 2.1.1. Although it is mathematically possible to derive the extrinsic parameters in this way, the results are not accurate since extrinsic calibration is not linear. Also, the process of computing extrinsic parameters for each of the different views and then averaging the translation and rotation vectors introduces error into the computations. To eliminate these problems, epipolar geometry is used to perform the calibration. The advantages are that the results are more precise and the entire dataset is processed at once when estimating the extrinsic parameters.

To compute the extrinsic parameters, the method proposed by Hartley and Zisserman [8] is

used and is briefly described here. The first step is to compute the fundamental matrix F, which is an algebraic representation of epipolar geometry between two image planes. The only information needed to compute F is a list of point correspondences between the left and right images, which are taken from the chessboard corners detected in the previous section. Secondly, the essential matrix E is computed by combining F and the left and right intrinsic matrices, I_L and I_R respectively. Finally, E is decomposed into a translation vector T and a rotation matrix R. This is performed using an SVD decomposition of E as well as some of its properties. Two possible translations and two possible rotations are found, thus giving four possible solutions. The ambiguity is solved by triangulating one correspondence point using all combinations of T and R, and selecting the solution that yields a 3D point in front of both cameras such that the Z coordinate is positive.

3.1.3 Scale Factor

The extrinsic calibration described in the previous section will only estimate the translation between the cameras up to a scale factor. In order to compute the physical translation between the cameras, the scale factor is computed by integrating several views of the chessboard calibration target and performing a least squares minimization between the ideal measurements of the target and the respective measurements using the triangulated 3D points.

The Euclidean distances between the first corner of the chessboard and all other corners, for all acquired images, are used to compute the least squares minimization. Considering N corner points detected per image, a subset of which are shown in Figure 3.3, a vector of ideal distances for one image is defined using Equation 3.4, where the points P_i are measured on the chessboard with the Z component set to zero and the operator d() computes the Euclidean distance.

$$X^{i}_{ideal}=\left[\mathrm{d}(P_1,P_2),\mathrm{d}(P_1,P_3),\mathrm{d}(P_1,P_4),\ldots,\mathrm{d}(P_1,P_N)\right]^T \tag{3.4}$$

Equation 3.4 is computed for each image and considering a total of M images, the global vector of ideal distances is constructed using Equation 3.5.

$$X_{ideal}=\left[\left[X^{1}_{ideal}\right]^T,\left[X^{2}_{ideal}\right]^T,\left[X^{3}_{ideal}\right]^T,\ldots,\left[X^{M}_{ideal}\right]^T\right]^T \tag{3.5}$$

The vector of measured distances is computed in much the same way and is expressed

using Equations 3.6 and 3.7. The only difference is that the points Q_i are taken from the projection of the chessboard corners into 3D space, which is possible since the extrinsic calibration is known.

$$X^i_{measured}=\left[d(Q_1,Q_2),d(Q_1,Q_3),d(Q_1,Q_4),\ldots,d(Q_1,Q_N)\right]^T \tag{3.6}$$

$$X_{measured}=\left[\left[X^1_{measured}\right]^T,\left[X^2_{measured}\right]^T,\left[X^3_{measured}\right]^T,\ldots,\left[X^M_{measured}\right]^T\right]^T \tag{3.7}$$

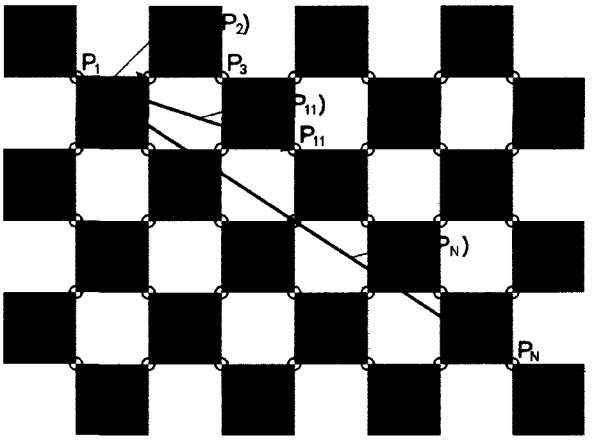


Figure 3.3. Distance measures used to determine the scale factor via least squares.

The scale factor S is computed by performing a standard least squares analysis of the ideal and measured distance vectors, as expressed in Equation 3.8. Since X_{ideal} and $X_{measured}$ are one dimension vectors, S is a scalar and it can be multiplied to each triangulated 3D point to obtain the latter's metric representation.

$$S=(X^T_{measured}X_{measured})^{-1}X^T_{measured}X_{ideal} \tag{3.8}$$

3.1.4 Results

The improvements made to the extrinsic calibration are verified by performing the calibration using the previous method and the new method. First, the intrinsic camera parameters are computed once and then used for both methods. Next, 30 pairs of images are captured by waving the chessboard such that it is in the field of view of both cameras. Finally, the extrinsic parameters are computed via both methods using the same pairs of images.

The above procedure is repeated five times and the results are shown in Table 3.1. The backprojection error (BPE) is computed by using the extrinsic parameters to triangulate the calibration points from the chessboard target into the 3D space and then projecting them

back to the image planes. The pixel distance between the original calibration points and the projected points is computed and averaged. The scale factor is computed as explained in Section 3.1.3 for both methods.

Dataset	Average BPE (pixel)		Maximum BPE (pixel)		Scale Factor (mm)	
	Previous	New	Previous	New	Previous	New
1	13.4323	5.4779	24.3745	8.0772	517.3023	389.0573
2	12.6311	3.9994	23.5815	6.3323	513.3073	391.2356
3	14.3453	2.9338	23.7126	5.1426	515.6287	390.1680
4	15.0204	3.1846	27.3125	5.3102	511.4069	390.5520
5	13.8491	5.7311	26.8343	8.1333	512.3041	389.6484

Table 3.1. Extrinsic calibration results comparing previous and new methods.

Although the backprojection errors give a good estimate for the precision of the two calibration methods, the numbers are biased since the same points that are used to compute the extrinsic parameters are also used to compute the backprojection errors. For a more objective comparison, a scene with simple elements, including a miniature foam model of a chair covered in white paper and a black curtain background, is imaged and metric reconstruction is performed using the extrinsic parameters of each method for each dataset. The scene with the entire projected pattern is shown in Figure 3.4 along with the position of key measurements used for comparison.

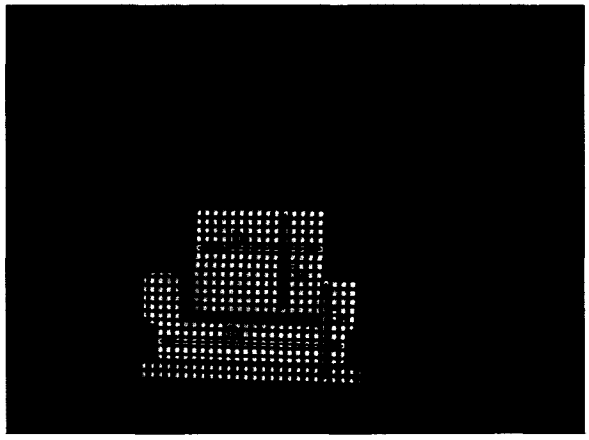


Figure 3.4. Scene with entire projected pattern and key measurement positions.

Measurements corresponding to the distances shown in Figure 3.4 are taken from the physical chair and background in order to provide a ground truth. The same distances are then computed from the point clouds triangulated using the respective extrinsic calibration.

The results are shown in Table 3.2 where the ground truth values are in square brackets and errors are in parenthesis.

Dataset	D_1 [961mm]		D_2 [712mm]		D_3 [186mm]	
	Previous	New	Previous	New	Previous	New
1	863.1 (97.9)	953.3 (7.7)	651.8 (60.2)	717.9 (5.9)	188.9 (2.9)	185.6 (0.4)
2	861.7 (99.3)	950.4 (10.6)	650.8 (61.2)	715.3 (3.3)	188.2 (2.2)	185.6 (0.4)
3	863.4 (97.6)	951.3 (9.7)	652.6 (59.4)	716.1 (4.1)	188.9 (2.9)	185.5 (0.5)
4	856.0 (105.0)	947.3 (13.7)	646.9 (65.1)	712.6 (0.6)	187.3 (1.3)	185.2 (0.8)
5	854.2 (106.8)	951.6 (9.4)	645.3 (66.7)	716.8 (4.8)	186.9 (0.9)	185.5 (0.5)
Dataset	**D_4 [145mm]**		**D_5 [259mm]**		**D_6 [135mm]**	
	Previous	**New**	**Previous**	**New**	**Previous**	**New**
1	146.9 (1.9)	145.5 (0.5)	268.1 (9.1)	258.3 (0.7)	139.7 (4.7)	135.7 (0.7)
2	146.4 (1.4)	145.5 (0.5)	266.9 (7.9)	258.4 (0.6)	139.1 (4.1)	135.7 (0.7)
3	146.9 (1.9)	145.4 (0.4)	268.1 (9.1)	258.3 (0.7)	139.7 (4.7)	135.7 (0.7)
4	145.7 (0.7)	145.1 (0.1)	265.7 (6.7)	258.1 (0.9)	138.5 (3.5)	135.4 (0.4)
5	145.4 (0.4)	145.4 (0.4)	265.2 (6.2)	258.2 (0.8)	138.2 (3.2)	135.6 (0.6)

Table 3.2. Metric reconstruction results along with the ground truth in square brackets and the error in parenthesis.

The metric reconstruction error, using the previous extrinsic calibration method, varies significantly over the different measurement positions and is particularly high at D_1 and D_2. On the other hand, the new calibration achieves a much more regular error and even sub-millimetre precision over D_3 through D_6. The reason for the greater error over D_1 and D_2 is that those measurements are taken farther away from the sensor and slightly outside of the calibration space. Finally, it should be noted that these errors arise from the estimation of the extrinsic parameters as well as the scale factor.

3.2 Robust Colour Segmentation

When using a coloured pattern, the first step of the image processing stage is to segment the coloured regions of the image from the background and then identify the colour of each region. Even though the images to process are the result of a background subtraction between images of the scene with and without the projected pattern, the procedure is not trivial. The previously implemented method lacked robustness and relied on user defined parameters. The goal is to make it fully automatic and capable of dealing with various

colours, which is required when the sensor images an unknown scene.

The first problem occurs when segmenting the coloured regions from the black background. Manually defined thresholds for each colour channel were defined and used to determine if pixels belonged to coloured regions or the background. This was possible since the input to the colour segmentation algorithm, referred as the input image, is a background subtraction of a scene; the result of subtracting an image without the projected pattern from one with the pattern. However, since the method relied on thresholds, the latter were optimized for a red, green and blue pattern only. This section of the algorithm performed quite poorly when used with other colours, for example, cyan, magenta and yellow, since the parameters had to be adjusted on a case by case basis.

Secondly, a hue histogram of the coloured pixels was created and analyzed to find the dominant peaks. The mode method [7] was used to detect the peaks; however, post processing was required since in most cases it returned more peaks than desired. The good peaks had to be differentiated from the bad peaks and a naive method of selecting the tallest peaks was implemented.

Finally, once the peaks were found, the thresholds for colour segmentation were determined simply by calculating the midpoint between peaks. Although straightforward, this threshold selection method contributed to noise along the border of the colour regions and had difficulty with washed out and low intensity colour regions. There was also a problem with red information bleeding into the blue mask because of improper handling of circular hue histograms.

3.2.1 Improved Colour Segmentation Algorithm

To address the problems stated above, a new algorithm for colour segmentation is proposed. The first step is to convert the input image to the HSV colour space. Since the input image is the result of a background subtraction, the background has a very small value component. The value channel is therefore thresholded, using an adaptive thresholding technique, resulting in a mask of the coloured areas of the image.

Next, the histogram of the hue channel is computed using only the coloured pixels that have been selected via the value channel mask. This step alone generates a much better and cleaner hue histogram since the background information is not present. The histogram peaks must be found, but since the number of colours used in the pattern is known a priori,

an algorithm that exploits this knowledge is selected. Tsai's algorithm [31] does not necessarily find peaks; rather, it provides a clustering method that finds dominant hills in the histogram representing the pattern colours. The main advantage of this algorithm is that it takes a parameter specifying the number of clusters to find. This assures that the proper number of peaks is found and no post processing is necessary to refine the detected peaks.

The last step is to find a pair of low and high thresholds for each histogram cluster in order to perform the final colour segmentation. Since the histogram is created using only the coloured pixels, the regions between the peaks are theoretically empty and normally contain 0 to 4 pixels per bin due to noise. This is advantageous since it simplifies the computation of low and high thresholds. First, the histogram is normalized such that the highest peak is set to 1. Second, by considering the three detected peaks P_i in Figure 3.5, the midpoints M_i are computed as the middle point between respective peaks, taking care to respect the circular nature of the histogram since the hue component is expressed in degrees. Third, to compute the low and high thresholds of P_2, the histogram is scanned from M_1 to P_2 and T_2^L is defined as the last point where the normalized histogram is smaller than 0.5%, which is roughly equivalent to 100 pixels with the current image resolution. If no such point is found, the range is scanned again and T_2^L is defined as the point between M_1 and P_2 with the smallest histogram value. A similar scan is performed from M_2 to P_2 to find T_2^H. Finally, the scanning process is repeated to find the low and high thresholds of the other two peaks.

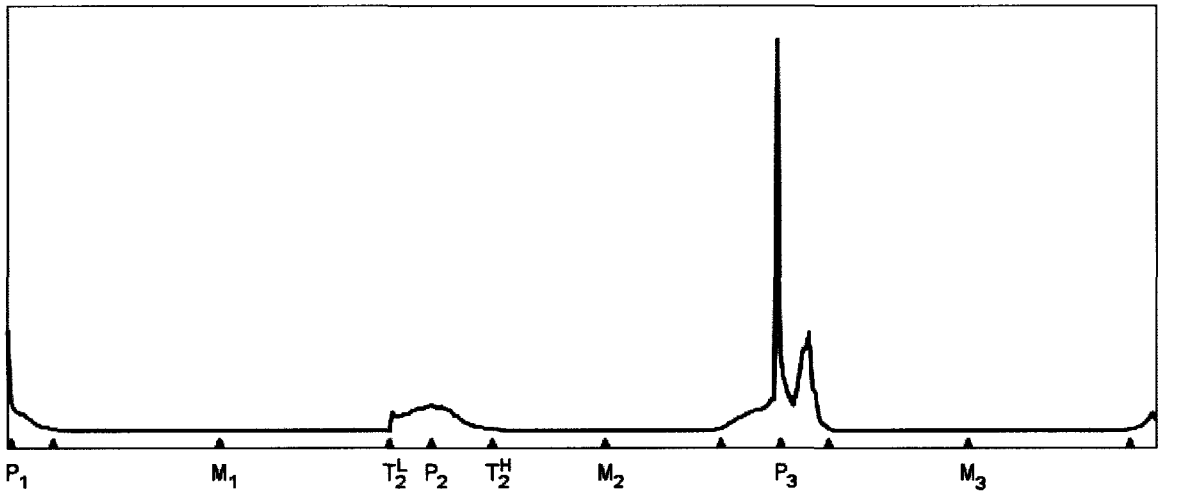


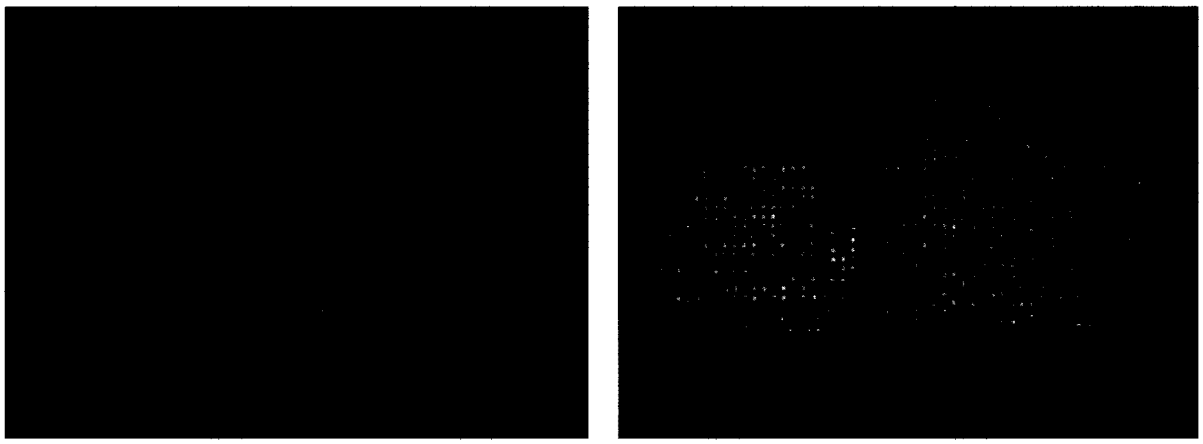
Figure 3.5. Normalized hue histogram and detected low and high thresholds.

The remaining operation is to perform the thresholding of the hue channel using the three pairs of low and high thresholds to produce three masks representing the different colour

regions. As a final precaution, the value mask is applied to the resulting colour masks in order to ensure that all background regions are eliminated if present.

3.2.2 Results

The new colour segmentation algorithm is tested against the old implementation to demonstrate its increased robustness and flexibility. A scene is imaged with two different coloured patterns. Figure 3.6 shows the red, green and blue pattern as well as the cyan, magenta and yellow pattern. The use of different colour patterns demonstrates the algorithms' flexibility in handling various colours.



(a) (b)

Figure 3.6. Scene imaged using (a) RGB pattern and (b) CMY pattern.

First, the RGB pattern is processed and the resulting red, green and blue colour masks for the computer monitor are shown in Figures 3.7.a and 3.7.b for the old and new algorithms respectively. Overall, the segmented squares are much more consistent in size with the new algorithm. Also, the square shapes are much smoother and better represent the actual projected light onto the scene. These are especially noticeable in the red mask of the monitor object in Figure 3.7.b.1.

The new algorithm is capable of segmenting regions where the pattern is faded or not as bright as the rest of the image. This normally occurs when the normal of the surface points away from the sensor as is the case on the rounded top region of the monitor. Because of adaptive thresholding, more squares are segmented in the upper right region of the green mask shown in Figure 3.7.b.2. These squares are completely ignored by the previous implementation as seen in Figure 3.7.a.2.

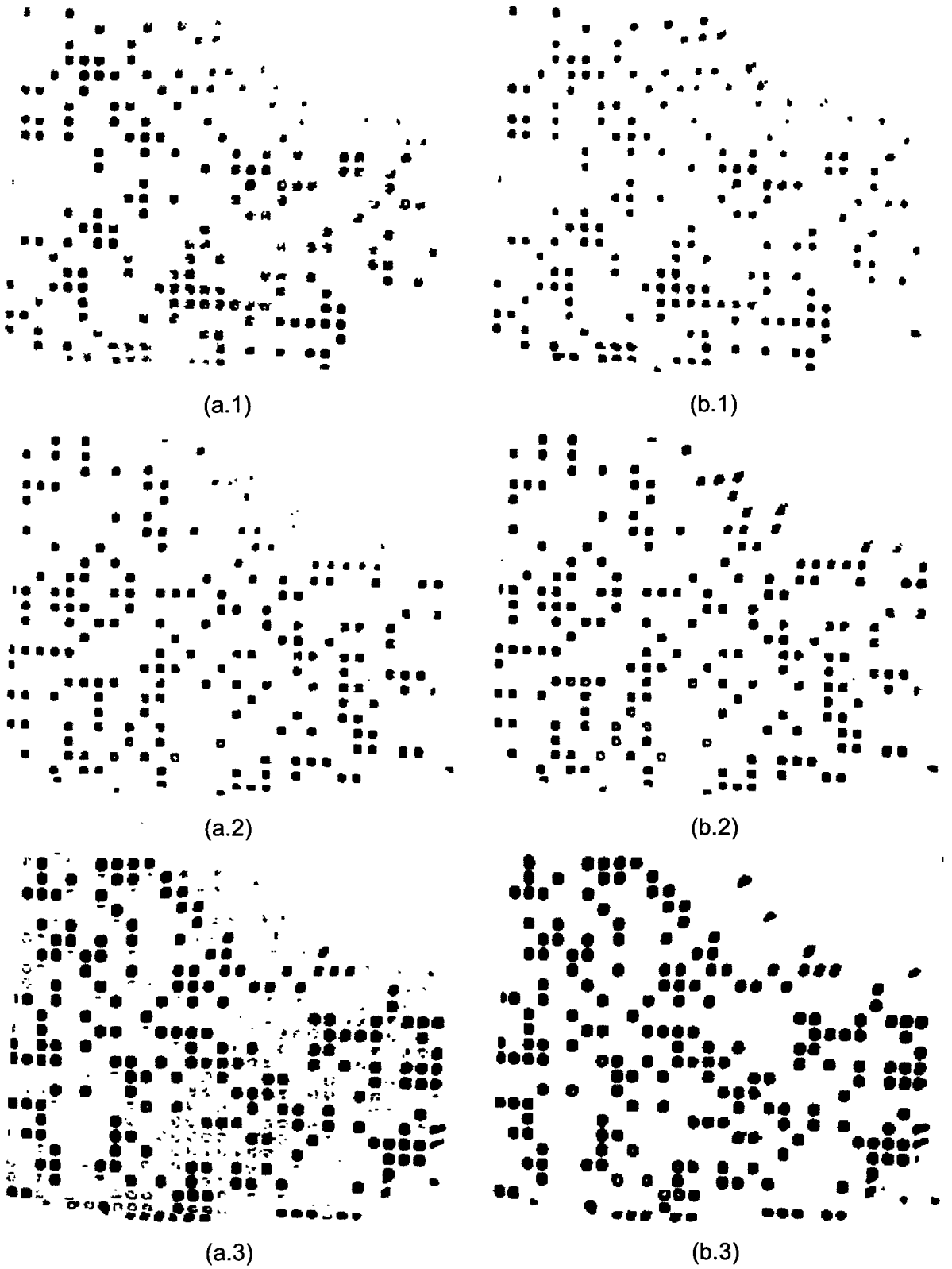


Figure 3.7. RGB colour masks of the monitor object using (a) the old algorithm and (b) the new algorithm for the colours (1) red, (2) green and (3) blue.

As for the blue mask, the old algorithm has difficulty differentiating between blue and red. Much of the noise in Figure 3.7.a.3 is generated by improperly identifying red information, at the end of the hue scale, as blue. With proper handling of the circular hue histogram, the red component is properly left out when using the proposed algorithm as shown in Figure 3.7.b.3.

To verify that the refined algorithm is capable of dealing with other colours, the CMY pattern is processed and the results are shown in Figure 3.8. Because of explicit colour channel parameters, the previous implementation cannot handle the CMY pattern reflected from the chair object. All of the squares are segmented into the magenta mask in Figure 3.8.a.

On the other hand, the proposed algorithm manages to segment the colours as seen in Figures 3.8.b.1 to 3.8.b.3. However, the masks contain a lot of noise, coming from other colours, due to non optimal segmentation thresholds. This is due to the fact that the projected CMY colours interact with the colours of the objects and are thus concentrated on a smaller region of the hue histogram. The problem is especially difficult with the cyan and yellow colours, shown in Figures 3.8.b.1 and 3.8.b.3 respectively, as there is no significant valley between them in the histogram.

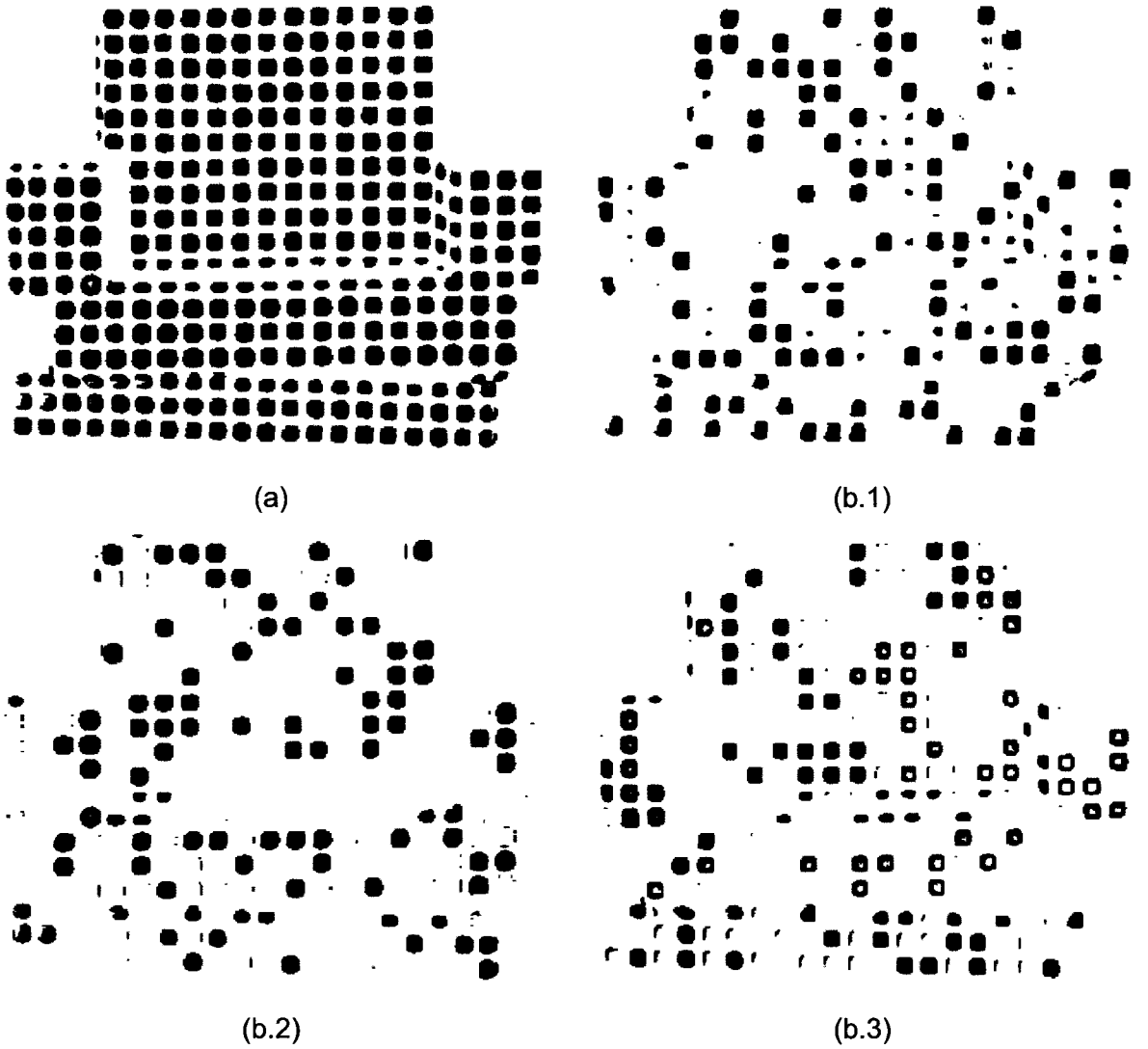


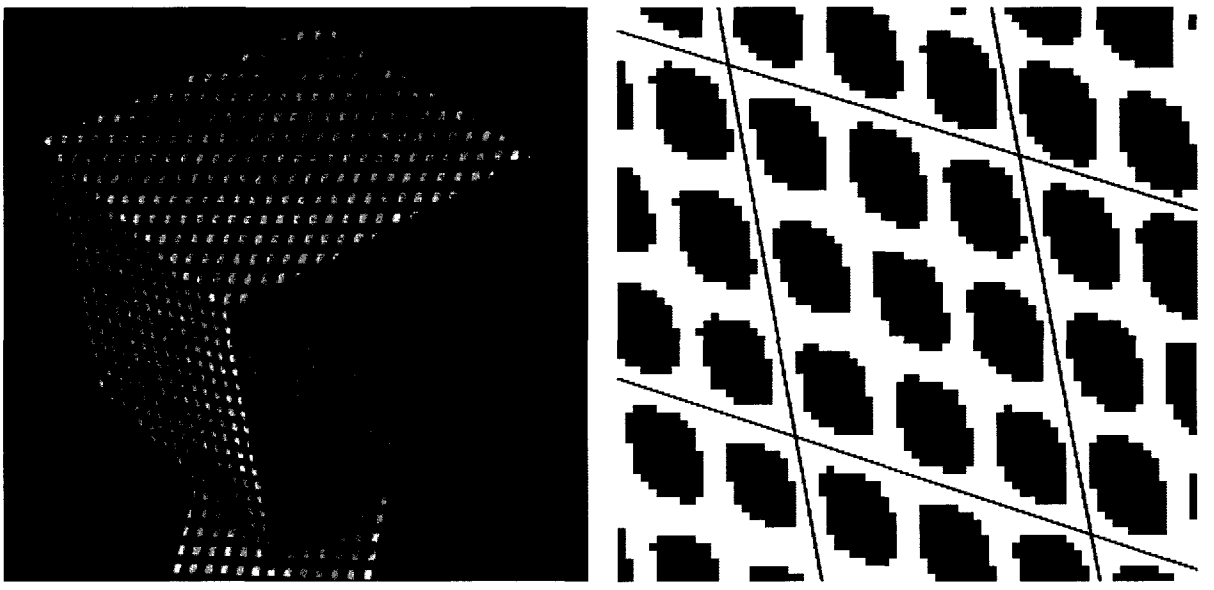
Figure 3.8. CMY colour masks of the chair object using (a) the old algorithm and (b) the new algorithm for the colours (1) cyan, (2) magenta and (3) yellow.

Finally, it is important to note that the improved robustness of the colour segmentation algorithm comes with only a negligible increase in execution time. The flexibility to deal with different colours and the improved handling of the RGB pattern far outweigh the slight execution time increase. Also, it is possible to perform more complex histogram analysis to further improve the results. However, this is not treated in the current work as the use of coloured patterns is abandoned to better handle object colours, as will be discussed in Chapter 4.

3.3 Flexible Code Detection

Once the coloured regions, also referred to as blobs, have been segmented, labelled and statistically analyzed to remove small blobs and re-segment large blobs, codes that consist of a spatial neighbouring of 3x3 blobs are recovered independently for each image. The previous method makes a naive assumption that prevents the detection of codes in certain conditions even though the blobs were properly segmented.

The main problem with the code detection is the assumption that a code is defined by a central blob and its eight closest neighbouring blobs. These are first sorted vertically and then horizontally to determine the spatial organization of the blobs that define the code. This assumption does not hold when the imaged surface generates a skewed alignment of blobs. For example, this is evident when imaging a cube where three faces are in the field of view, as shown in Figure 3.9.a. Figure 3.9.b depicts a section of the cube's left face where the pair of parallel lines indicates the true code of the centre blob B_c. The eight closest neighbours, labelled B_1 to B_8 and circled, do not form a valid code as blobs B_9 and B_{10} are not selected but rather B_7 and B_8. This test case is particularly important from a mobile robotics perspective since many scenes will have orthogonally aligned planes such as hallways, doorways and tables for example.



(a) (b)

Figure 3.9. (a) Original image of cube with pattern projected in white to better visualize the skewing, and (b) zoomed in section of acquired pattern showing eight closest neighbours not forming a valid code.

3.3.1 Improved Code Detection Algorithm

In order to allow for more flexible code detection, capable of dealing with skewed colour blobs, a new algorithm is presented. This algorithm is applied independently to every blob since each blob is a potential candidate for being the centre component of a code. The following explanation is given with respect to a centre blob, denoted as B_C. Most of the operations are performed in a 2D vector space with the origin defined as the centroid of B_C and all other blobs represented as vectors to their respective centroid. The algorithm is divided into three steps that are now detailed.

The first step consists of finding the closest neighbours of B_C. However, instead of only finding eight neighbours, the number is doubled, and the sixteen closest blobs are located. This is performed by computing the Euclidean distance between each blob and B_C, storing the information in a list, sorting the list and then taking the sixteen first blobs. The number of blobs to find was determined by trial and error; the distance of subsequent blobs is usually greater by an order of magnitude. Only the sixteen closest blobs are considered for the following steps.

The second step, shown in Figure 3.10, consists of determining opposing blob pairs. Due to the orthogonal nature of the projected pattern, a local section of a row of blobs will most likely appear in a straight line in the image, no matter what the rotation of the object. The same holds for columns of blobs, although the latter can be skewed with respect to the rows. Using this observation, a pair is defined as two adjacent blobs that are on the same row, column or diagonal as B_C. For example, valid pairs in Figure 3.10.a are B_1-B_2, B_3-B_4, B_5-B_6, B_7-B_8, B_9-B_{10} and B_{13}-B_{14}.

The pairs are found by using a vector projection computation. First, the closest blob B_1 is considered, marked as visited and selected as one of the blobs for the first pair. This defines the vector V_1 as shown in Figure 3.10.b. Next, the projection of all other blob vectors onto V_1 is computed using Equation 3.9 and the corresponding error vectors are computed using Equation 3.10.

$$V_{J,I}^{P}=\left(\frac{V_J \cdot V_I}{\|V_I\|^2}\right)V_I \tag{3.9}$$

$$V_{J,I}^{E}=V_J-V_{J,I}^{P} \tag{3.10}$$

For example, the projection of V_5 onto V_1 gives a projection vector $V_{5,1}^{P}$ and an error vector

$V^E_{5,1}$ as defined in Figure 3.10.b. Similar results for the projection of V_2 onto V_1 are also obtained by considering negative projections. The angles between all of the blob vectors and the line defined by V_1 are computed and only the blobs that have an absolute angle smaller than 15° are considered as pair candidates to B_1. This introduces some tolerance on the linearity of the alignment of the candidate blobs' centroid with the centre blob B_c. In the current example, this yields the blobs B_2, B_{11} and B_{12}, which are marked as visited. All of the candidates are assured to lie approximately along the same line defined by B_c and B_1. Finally, the blob selected to form a pair with B_1 is the blob with the smallest absolute projection vector length, in this case B_2. The process is repeated until all blobs are marked as visited. It should be noted that Figure 3.10.b is not to scale in order to clearly display the projection vectors.

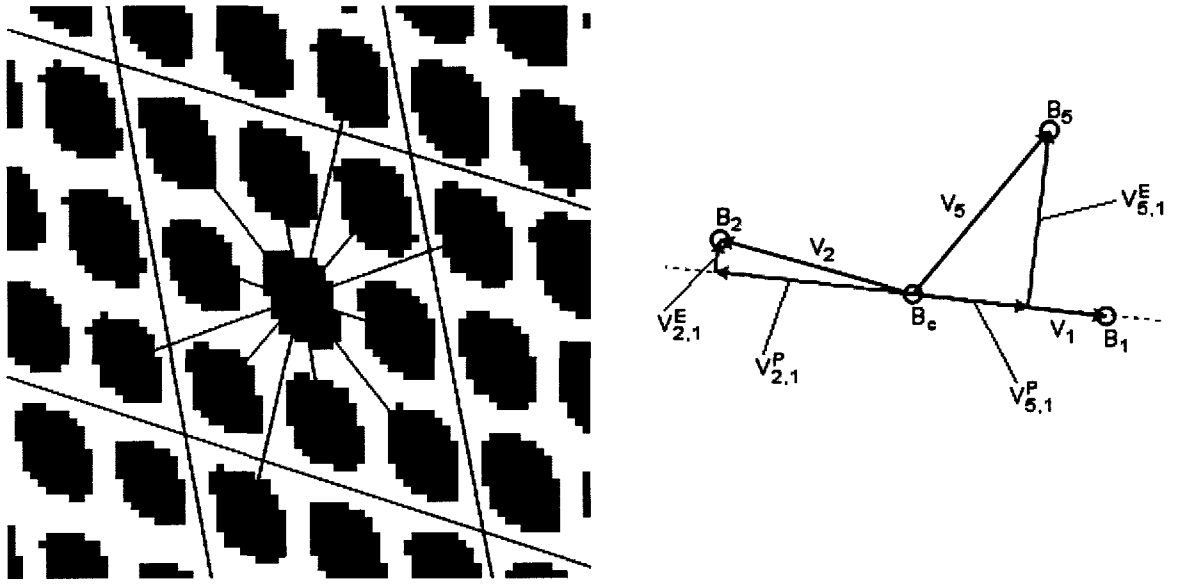


Figure 3.10. (a) Sixteen closest blobs and the resulting valid pairs. (b) Vector projection of V_5 and V_2 onto V_1.

The third and final step consists of identifying the relevant pairs, those that are part of the code, from the superfluous pairs. This is done using an ad-hock method that relies on the fact that the distance between blob borders, denoted as the empty distance D^E, is smaller for valid pairs when compared to invalid pairs, as shown by the bold line segments in Figure 3.11. First, the empty distances of each pair are computed using Equation 3.11.

$$D^E_{i,j} = D^E_i + D^E_j \tag{3.11}$$

Second, the four pairs with the smallest empty distance are selected as the pairs of blobs that compose the code. Finally, the selected blobs are sorted vertically and horizontally, with respect to their centroid, just like in the previous implementation. The sorting method is adequate since the projected pattern will not be rotated with respect to the cameras. The code is now fully detected since the colour has already been determined during the segmentation stage.

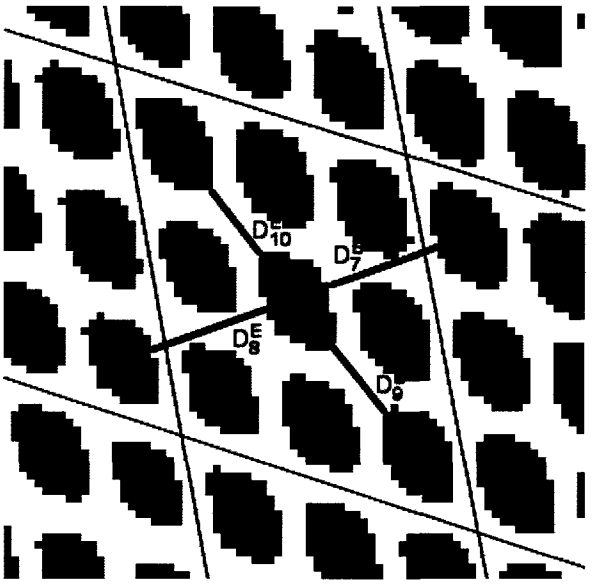


Figure 3.11. Empty distance for two key pairs.

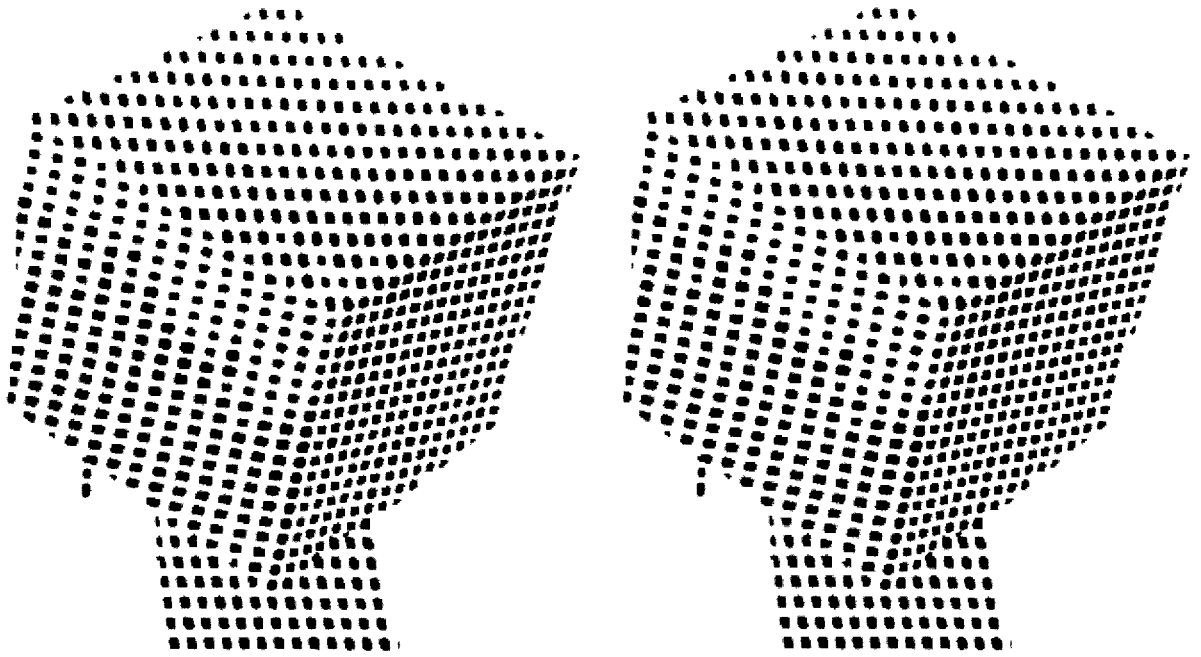
3.3.2 Results

The new vector based code detection algorithm is validated by comparing it to the previous implementation. The cube presented in Figure 3.9.a is imaged and both code detection algorithms are run on the same image dataset. In this case, the cube is a good subject for analysis as it guarantees that at least one side will contain skewed blobs. The latter is the use case of interest since the new algorithm is specifically designed to handle it, while still remaining functional when the blobs are arranged in a rectangular fashion, which is the general case.

Figures 3.12 and 3.13 respectively show the segmented blobs for the left and right images of the cube along with a side-by-side comparison of both algorithms. The superimposed black squares denote the centre blob of a detected and validated 3x3 code. The codes are deemed valid when the eight neighbours are properly found and matched with the same code in the opposite image, producing a correspondence between images.

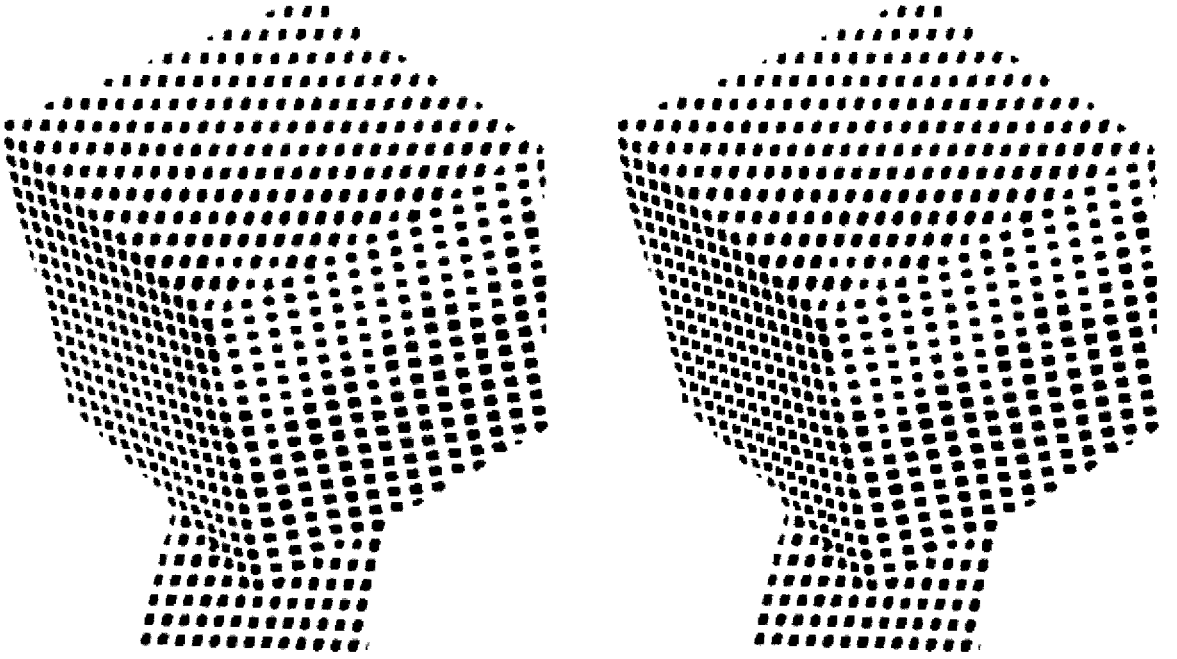
As is seen in Figures 3.12.a and 3.13.a, the previous algorithm is able to correctly detect codes on the upper face of the cube, since the blobs are arranged along a more rectangular lattice. On the other hand, the left and right faces of the cube prove to be more difficult to detect. The skewing of the blobs is particularly significant on the right face in Figure 3.12.a and on the left face in Figure 3.13.a. Although the codes of the left and right cube faces are properly detected in the respective right and left images, they ultimately must be detected in both images in order to produce valid codes and a correspondence.

The valid codes detected with the proposed algorithm are shown in Figures 3.12.b and 3.13.b. There is a significant improvement in detection as all the possible 3x3 blob clusters are found. It should be noted that the algorithm still cannot detect codes along the edges of the cube. For edges at the intersection of the top, left and right faces, this is due to the fact that the blobs are not evenly skewed and simply too distorted. For the other outside edges, there are simply no adjacent blobs to detect 3x3 clusters. This does not pose much of a problem however, as the unmarked blobs in Figures 3.12b and 3.13b are still part of valid codes and are also used as correspondences between images.



(a) (b)

Figure 3.12. Valid codes detected on the left image using (a) the previous implementation and (b) the new algorithm, and highlighted with black squares.



(a) (b)

Figure 3.13. Valid codes detected on the right image using (a) the previous implementation and (b) the new implementation, and highlighted with black squares.

Although the new algorithm is slightly more complex, much of the operations are simple vector mathematics, which do not incur much overhead. In fact, due to an unoptimized implementation of the previous algorithm, the proposed technique executes in slightly less time. This is achieved by efficiently managing the lists of centroids and neighbours as well as reducing the amount of iterations.

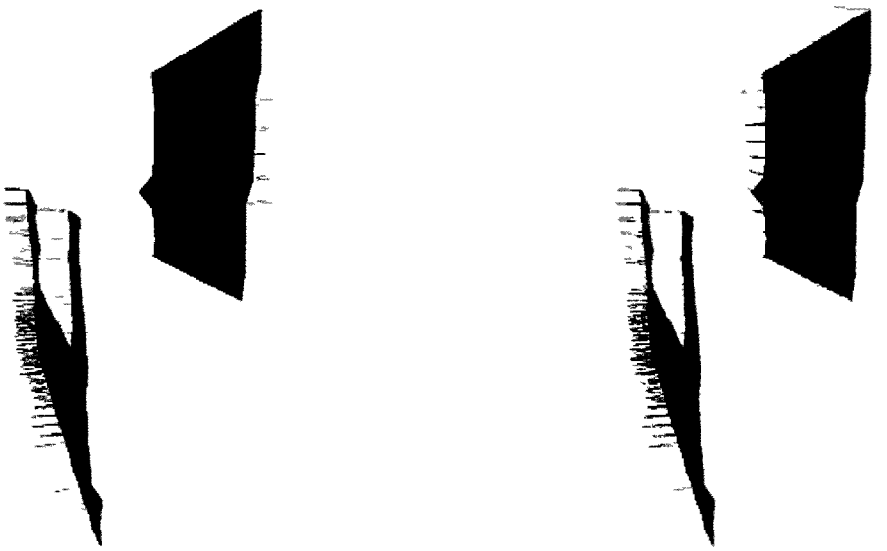
3.4 Mesh Generation and Visualization

The raw output of the structured light range sensor, a list of 3D points, can be interpreted by a high level algorithm but remains difficult to visualize and analyze for a human observer. Since the points represent a sampling of one or many surfaces, it is natural to interpolate the data and generate a surface that estimates the measured surface. Not only does the surface aid in visualizing the acquired data, it can also be used to perform a quantitative analysis of the sensor's performance.

3.4.1 Surface Interpolation

To interpolate a surface, the ball pivot algorithm [25] is applied to the 3D point cloud. This creates a mesh that is composed of vertices and triangular faces. Other than the list of 3D points, the algorithm has two main parameters, the radius and minimum distance, which must be defined prior to execution. The radius controls the size of the spherical neighbourhood that will be searched around an edge to find a vertex to be added to the mesh. The minimum distance is enforced when adding new faces to the mesh by ensuring that their edges are long enough.

The ball pivot algorithm was developed with object modelling in mind. The orientations of the faces are computed with respect to the centre of gravity of the point cloud such that the face normals point outward. This works well for closed surfaces representing objects, since the faces are properly oriented for 360° viewing. However, this is not optimal when interpolating points that are acquired using a structured light sensor from one point of view. Figure 3.14.a shows the result of applying the ball pivot algorithm to a single scan of a chair, as shown in Figure 3.4, along with the face normals. The bottom of the seat is correctly meshed as the faces point toward the viewer on the left. However, the faces of the seat back, in the upper region, are pointing away from the viewer. This is physically incorrect since the structured light sensor can only detect surfaces that reflect light back towards it. This problem is solved by manually setting the point cloud centre of reference to the origin of the reference frame, which is associated with the sensor. Next, the ball pivot algorithm is applied, which results in faces pointing away from the sensor, and the latter are inverted such that all of the face normals point towards the viewer. The proper face orientations are shown in Figure 3.14.b, where the viewer is located on the left.



(a) (b)

Figure 3.14. (a) Conventional ball pivot algorithm applied to a single scan of a chair, and (b) refined algorithm yielding proper surface orientations.

The ball pivot algorithm also assumes that the point cloud is regularly sampled such that the points are evenly distributed over the surface. This is a consequence of the ball radius parameter that cannot be adjusted during the execution of the algorithm. With the structured light sensor considered in this work, it is impossible to ensure regularity in the data set. For example, surface points that are closer to the sensor will have a higher spatial density than those that are farther away. This occurs since the projected squares' size and their neighbour distance increases as the distance from the sensor increases. Applying the algorithm with a small radius will interpolate the regions with high spatial density and ignore the regions with lower spatial density. Using a larger radius will allow the interpolation of all regions, however, some points in areas with higher spatial density may be ignored, since the search radius is larger and points farther away may be selected over closer points. These issues are addressed by applying the ball pivot algorithm iteratively while increasing the ball radius parameter. Small faces are interpolated first, and for each iteration larger faces are added to the mesh, producing a mesh that accurately interpolates regions of low and high spatial density.

Finally, since a colour measure is acquired for each 3D point, the latter is also transferred to each vertex of the surface mesh. Mesh viewers are then able to display the vertices with colour and interpolate the colours over the triangular faces giving the mesh a rich texture.

3.4.2 Automatic Parameter Detection

In order to apply the ball pivot algorithm to the point cloud automatically, the mean minimum distance between points must be computed since it is the basis for the automatic computation of the ball radius parameter. To accomplish this, the point cloud working volume is divided into a 3x3x3 voxel space and a point is selected arbitrarily from each voxel, where applicable, as a representative of that voxel. If the point cloud contains many disjoint sections, a higher voxel resolution can be selected to better cover the point space. For each representative point, the closest neighbour is found and the corresponding Euclidean distance is computed, giving a minimum distance for each voxel. In practice, it was found that computing the mean of the minimum distances was not robust. In certain data sets, erroneous points and outliers contributed to unproportionally small and large minimum distances respectively, which skew the dataset of minimum distances.

To address this, the list of minimum distances is sorted and the difference between adjacent distances is computed. Two scans are performed, one from the middle of the list to the beginning and another from the middle of the list to the end. While scanning, a running mean of the differences is computed. If the next difference is greater than 20 times the current mean, the scanning is stopped and further entries are discarded. This ensures that extreme values of minimum distance, located at the beginning and end of the list and the result of erroneous points and outliers, are not considered for the radius estimation.

Once the scans are complete, the mean of the remaining minimum distances is computed. Since the acquired 3D points are arranged in a rectangular lattice, Equation 3.12 is used to estimate the ball radius parameter, which is based on a simplified geometrical assumption shown in Figure 3.15.

$$R=\frac{\sqrt{D_{min}^2+D_{min}^2}}{2}=\frac{D_{min}}{\sqrt{2}} \tag{3.12}$$

This estimated radius is used for the first iteration of the ball pivot algorithm. Then, for each iteration, the minimum distance is increased by 10% and the corresponding radius, computed using Equation 3.12, is used. Once the minimum distance reaches 1.5 times its original value, the iterations are stopped. This criteria is used to ensure that the ball pivot algorithm does not generate faces with points on the second row or column away from the point of interest, as indicated in Figure 3.15. This normally translates to a maximum of 5 iterations.

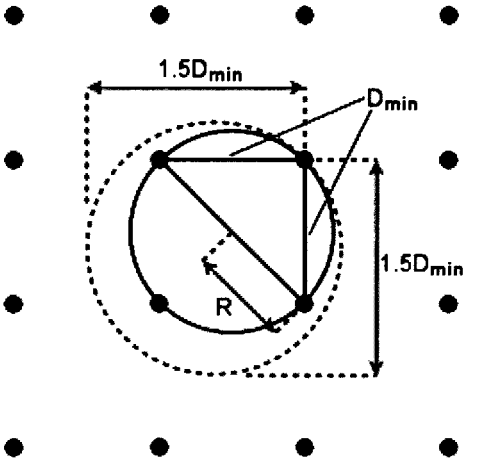
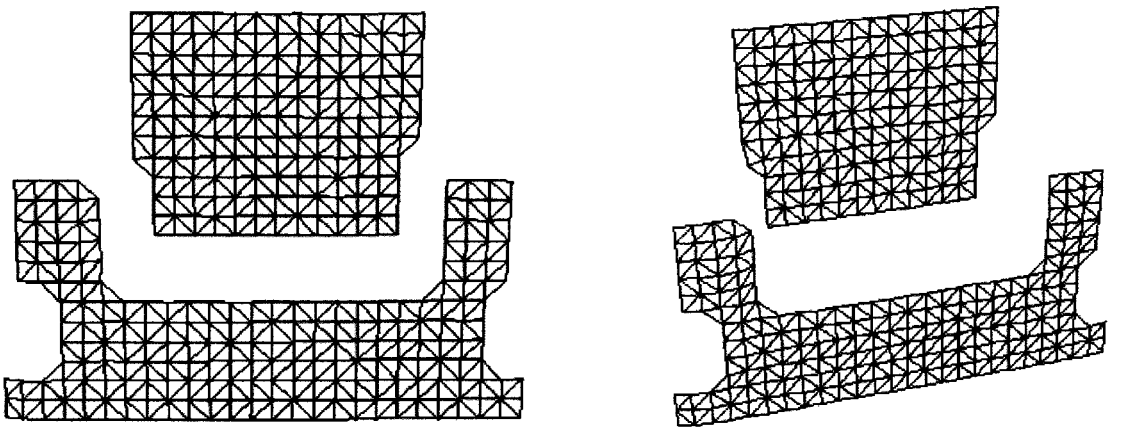


Figure 3.15. Geometrical assumption showing the estimates for D_{min} and R.

3.4.3 Results

The new surface generation step is tested and analyzed by applying the ball pivot algorithm to the datasets presented in Sections 3.1 to 3.3. The first example is that of the chair object with two views of the same mesh shown in Figure 3.16, to better observe the 3D nature of the results. The ball pivot algorithm simply takes a list of vertices and maps triangular faces to generate a surface. The raw output of the algorithm is a wireframe structure as shown.

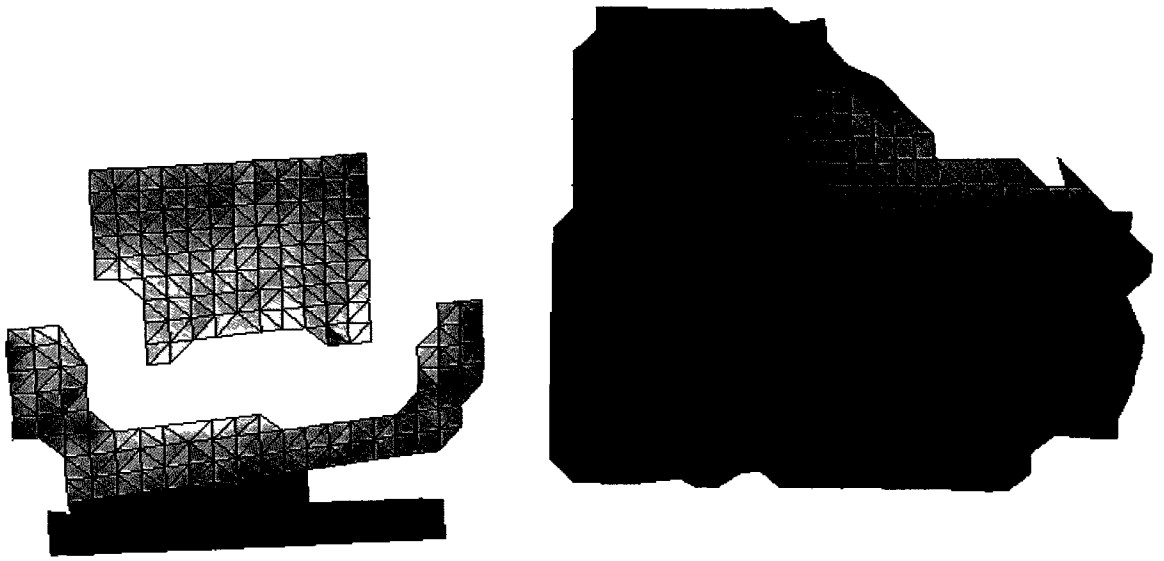


(a) (b)

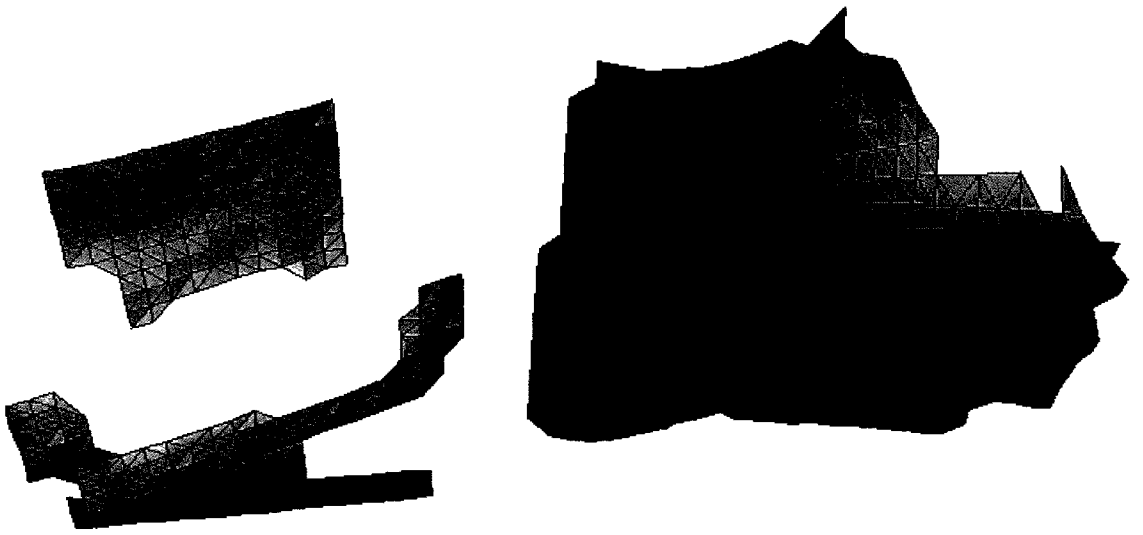
Figure 3.16. Two views of the chair object mesh.

Since the sensor is capable of acquiring colour information, a colour measure is assigned to each vertex by computing the average colour between the corresponding image points of

both left and right images. Most mesh viewers make use of this extra information in order to display the surface meshes in colour. Each triangular face is coloured by interpolating the colour values of its three vertices. Shown in Figure 3.17 is the chair and monitor scene of Figure 3.6.a, presented from two different viewpoints and displayed at low spatial density and in full colour along with the overlaid wireframe to better see the structure of the mesh. The darker regions on the right side of the monitor appear since there are physical holes on the side of the monitor and some 3x3 codes were centred directly over these holes.



(a)



(b)

Figure 3.17. Two views of the chair and monitor scene mesh.

The next object is the cube of Figure 3.9.a, where three of its orthogonal faces are simultaneously imaged. This test case clearly illustrates the relevance of applying the ball pivot algorithm iteratively while increasing the size of the ball radius. Since the top face of the cube is considerably angled with respect to the sensors principle axis, detected points that are farther away have a lower spatial density. As a result, not all faces are generated during the first iteration, with the initial radius size, as shown in Figure 3.18.a. By increasing the radius parameter, more and more faces are appended to the mesh as shown in Figures 3.18.b to 3.18.f. The main advantage is that the ball pivot algorithm is applied to the point set with a radius that is appropriate for the local distribution of points. Without this iterative procedure, a large ball radius parameter would have to be used globally and some points in areas of high spatial density would be ignored, generating an inaccurate mesh.

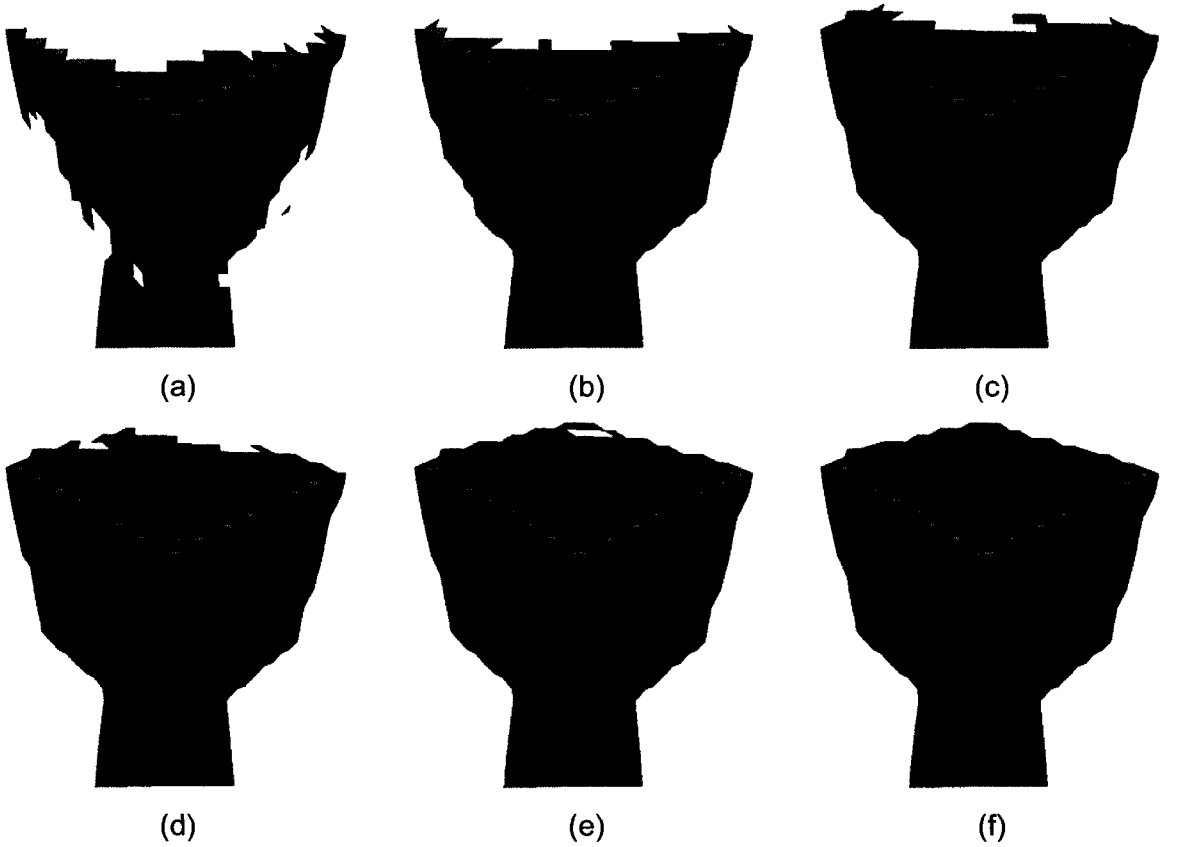
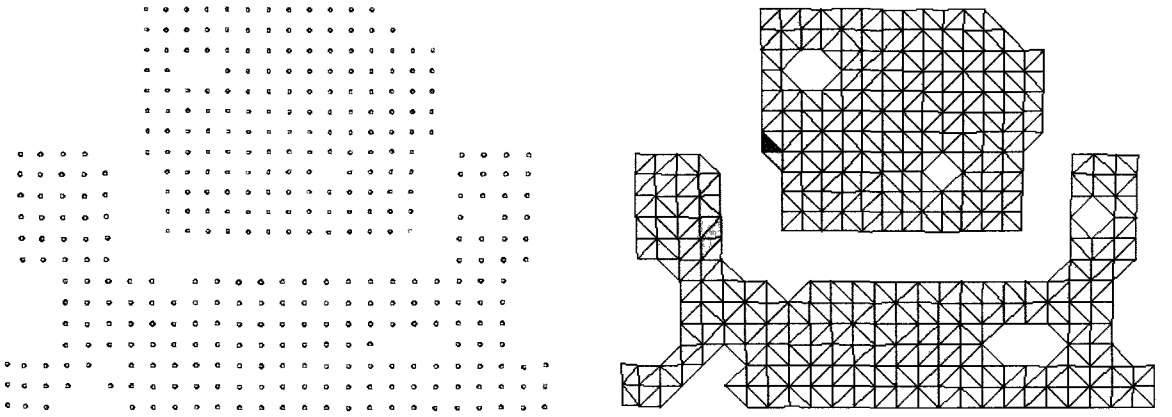


Figure 3.18. Growing mesh of cube object as the ball pivot algorithm is applied iteratively while increasing the radius parameter.

One important property of the ball pivot algorithm is that it inherently keeps track of the distance between vertices. It will not generate a face from vertices that are separated by a distance greater than the diameter of the pivoting ball. This may be a problem when

generating a mesh for modelling and visualization but is very useful when evaluating the performance of the structured light sensor. For example, Figure 3.19.a shows the same point dataset of the chair object but with certain points manually removed. The resulting mesh, generated by applying the ball pivot algorithm, is shown in Figure 3.19.b with visible holes. Although it is possible to increase the radius parameter in order to fill in the holes, this is not desirable since the sensor has effectively not detected points in those regions. When evaluating the effectiveness of the sensor, filing in the holes would lead to the assumption that the sensor was able to properly detect the entire surface, which is not the case.



(a) (b)

Figure 3.19. (a) Point cloud subset of the chair object and (b) the corresponding mesh with holes.

It should be noted that all of the meshing results shown in this section are generated using low density scans of the objects in question. Although it is possible to increase the spatial density of the sensor by marching the patterns horizontally and vertically, this leads to meshes with a large number of faces and it becomes difficult to show the details of the meshing algorithm and structure of the resulting meshes. Finally, throughout the remainder of this thesis, the output of the structured light sensor will be displayed and discussed using surface models generated using the modified ball pivot meshing technique.

3.5 Summary

This chapter presented enhancements to four key algorithms of the processing stage. First, the accuracy of the extrinsic calibration was increased by using epipolar geometry and the computation of the scale factor was added into the calibration routine. Second, the colour

blob segmentation was enhanced to better handle red, green and blue squares as well as other colours. Third, the 3x3 code detection algorithm was redesigned to use vector arithmetic in order to process skewed blobs. Fourth, the facility to automatically generate a surface mesh from the 3D points was added to aid in the visualization of the sensor's output. Finally, all enhancements were validated by results that show the increase in robustness and usefulness of the structured light sensor.

Chapter 4 Acquisition Stage Enhancements

The acquisition stage of the sensor handles the synchronization between the pseudo-random pattern projection and the image capture as detailed in Section 2.1.3. First, a pair of reference images is acquired without the pattern projection in order to obtain background images. Second, the pattern is projected and a second pair of images is acquired. The background images are subtracted from the latter resulting in difference images that are saved to disk for the subsequent processing stage. For an increase in spatial density, the above steps are repeated as the pattern is shifted horizontally and vertically.

Although this thesis work focuses on one particular structured light sensor, the previous implementation was developed with many of the same assumptions found in much of the recent literature. Similar systems that use structured light to acquire 3D range data, such as [32], [33], [34], [35] and [36], all have two major reoccurring limitations. First, the subject under analysis is usually a single object of small scale with uniform reflectance characteristics. Second, the setup is normally static, meaning that the projector focus is manually set and the subject is at a predetermined distance from the sensor.

This chapter aims to integrate modern methods to remove the above drawbacks from the prototype structured light sensor discussed so far. However, it is possible to apply these same techniques to other range sensors in order to improve their adaptability. First, the proposed 3D imaging approach is discussed and put in context with other adaptable sensors. Second, enhancements are presented that enable the sensor to adapt to scenes with varying colour, reflectance characteristics and depths of field. Also, detailed results that demonstrate the improvements are given for each enhancement.

4.1 Proposed 3D Imaging Approach

The previous version of the structured light range sensor put in place many of the core components, techniques and algorithms to perform 3D acquisition. Although good results were achieved on single objects, the range sensor's limitations become evident as more complex scenes are introduced. Some assumptions were also made that introduced limits on the sensor's flexibility and robustness when dealing with unconstrained scenes.

During the initial development, the coloured pattern was tested with objects of different colours and produced acceptable results. However, when tested with multicoloured objects

or entire scenes with a lot of colour variability, the colour of the projected pattern becomes distorted and causes the coloured square segmentation algorithm to fail. The conventional solution to this problem is the technique of adaptive structured light, which dynamically adjusts the projected pattern in response to the scene. This can be achieved by adapting the pixel colour [37] or pixel intensity [38] of the pattern using a feedback loop, to ensure that the coloured squares are detectable. The disadvantage is that extensive calibration between the cameras and the projector is necessary, which is incompatible and undesirable with the current structured light system. This would primarily impact the sensor's ability to adapt to the focus planes of the scene, which is explained subsequently. The proposed solution is to drop the use of colour and encode the square symbols using a time-multiplexed approach. The pseudo-colour pattern is therefore projected using three iterations of white squares and maximum projector intensity.

The above problem with colour is compounded with the fact that objects with multiple brightness and reflectance characteristics produce images with areas that are under- and over-exposed, contributing to a loss of 3D data in those areas. Figure 2.3.b shows an image of a scene where the black background is under-exposed with an average brightness of 5 and the white chair is over-exposed with a saturated brightness of 100 over a scale of 0 to 100. In the previous implementation, this was handled by manually setting the global exposure time of the cameras, via trial and error, prior to acquisition. Many disadvantages arose from this approach such as the need for parameter tweaking and more importantly the inability to acquire data from dark and bright regions simultaneously. A basic approach is to acquire multiple images while varying the global intensity of the projector and combine the images into a high dynamic range radiance map [39]. Although simple and effective, the problem of selecting a global exposure rate for the cameras still remains. This problem is further complicated when highly reflective objects lead to saturated areas in the image, regardless of the projector intensity. The proposed method is inspired by [39] but, instead of working at the projector level, it operates at the camera level. The pattern is projected at full intensity and multiple images are acquired while varying the exposure rate of the cameras. The images are then fused together using an exposure fusion algorithm, ultimately producing an image with a local exposure rate that compensates for the different reflectance characteristics in the scene.

Finally, the last weakness of the sensor is that it can only acquire data from one focus plane and ignores objects or areas of the scene where the pattern is out of focus and

unrecognizable by the segmentation algorithm. This is caused by the assumption that the object or scene of interest is located at a known and relatively constant distance from the sensor. In some cases, this can greatly reduce the amount of 3D information collected from a scene with a large depth of field. Also, the focus of the cameras and projector must be manually calibrated prior to acquisition, leading to yet another parameter to tweak. Most literature on structured light range sensors make this assumption, which assures that the cameras and projector are always in focus. When building a flexible sensor that must adapt to any scene, this assumption cannot be made and the focus problem must be considered. The proposed method is inspired by the solution to the exposure problem, mentioned above, and makes use of a focus fusion algorithm. The focus setting of the projector is automatically incremented as images are acquired and analyzed for their focal connectivity. Once the focus planes are detected, 3D range data is acquired for worthwhile planes and then fused together. With this method it is possible to obtain range data from a workspace that exceeds the focus capabilities of most structured light systems where focus is not adjusted.

4.2 Acquisition Modes

The pseudo-random (PR) pattern is defined using an alphabet of three symbols. The sensor's mode of acquisition determines how the symbols are encoded when projected onto the scene. The existing method is to encode the symbols using colour and assign a unique hue to each symbol. Although efficient, the reasons why this method is not optimal are discussed and a more robust method is proposed.

4.2.1 Colour Mode

A very important advantage of using colour to encode the spatial-neighbouring PR pattern is that the entire pattern can be projected and acquired at once. This is the most efficient form of structured light imaging since a maximum of one pattern is projected followed by one image acquisition step. The difficulty with this approach is to reliably detect and segment the coloured square symbols reflected from a wide variety of objects and scenes.

One of the recurring problems is with multicoloured objects and scenes with a lot of colour variability. In Figure 4.1, the high rate of colour change on the dartboard and the basket noticeably distorts the acquired square symbols. The red squares projected onto the red

areas of the dartboard appear in a low intensity dark green, as seen in Figure 4.1.c. Another problem is with dark objects that absorb most of the projected light. For example, the red and green squares projected onto the black background in Figure 4.1.d have a significantly lower intensity that their blue counterparts. This is further complicated when more objects and unknown colour combinations are added to the scene.

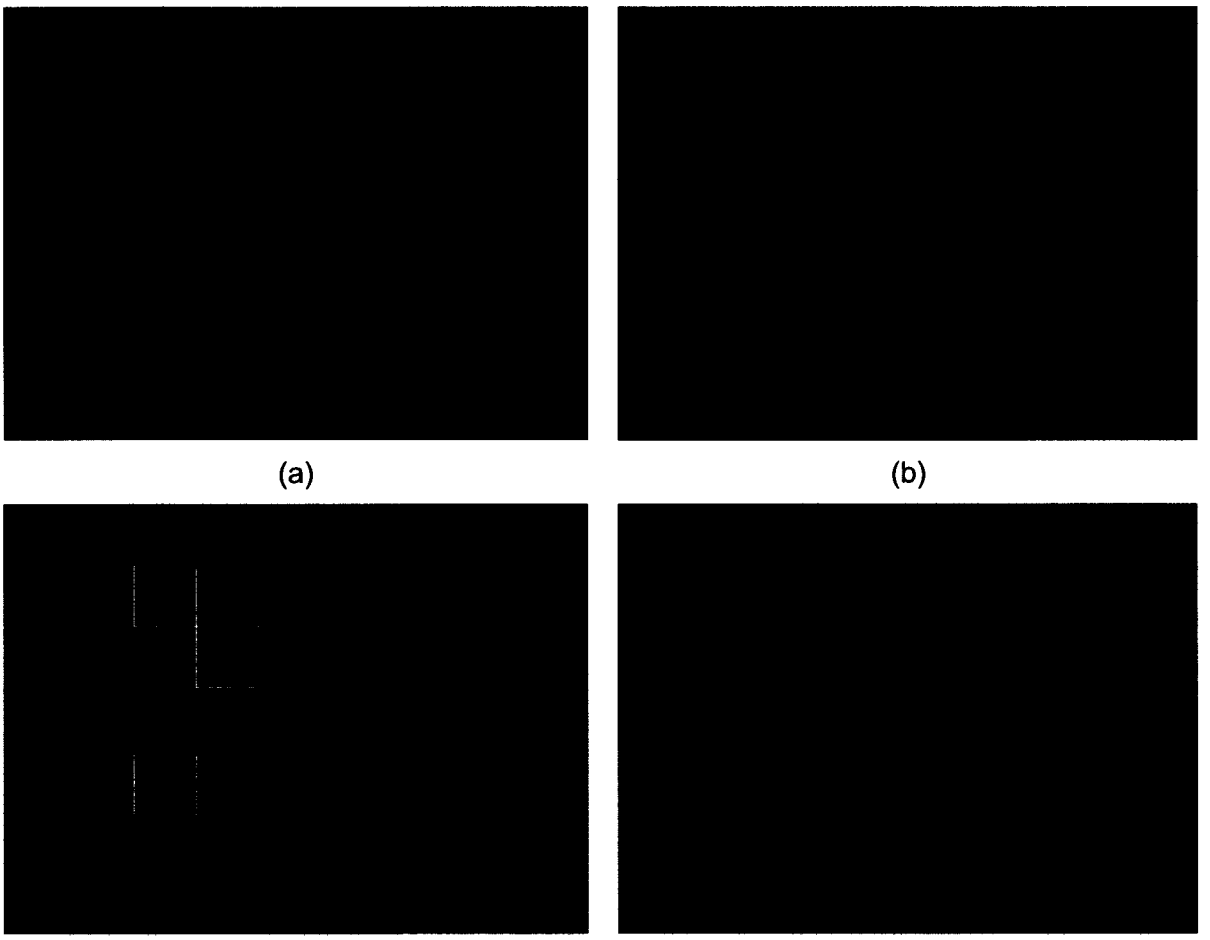


Figure 4.1. (a) Original image of the dartboard and basket scene and (b) the background subtraction of the colour encoded PR pattern. Zoomed in view of (c) the red squares appearing as green on the dartboard and (d) the low intensity of the red and green squares on the black background.

After much experimentation with coloured codes and because of the problems mentioned above, the use of a coloured spatial-neighbouring pattern is abandoned. Although a novel solution, it proves to work reliably only on uniform and lightly coloured objects that reflect light well. Also, with a focus adaptation in mind, it is undesirable to impose a calibration between cameras and projector to perform conventional adaptive structured light. It should

be noted that the colour mode is still available in the current implementation and can be manually enabled when performing an acquisition of a scene that permits it.

4.2.2 New Time-Multiplex Mode

Two factors account for much of the problems described in the previous section. First, coloured squares are distorted when projected onto an object of similar colour. Second, some colours are not projected with the same intensity as other colours. The proposed approach is to project the spatial-neighbourhood pattern using a single colour of white light at full intensity. The white colour ensures that the squares will be visible over any object colour and the full intensity ensures that the maximum amount of light is reflected from dark areas. The white pattern is even detectable on white objects since the added light provides enough contrast for the segmentation algorithm. Another positive side-effect is that the pattern has a higher chance of being detected from areas of the scene that are farther away from the projector, therefore increasing the range of the sensor.

Since the spatial-neighbouring PR pattern is composed of three symbols that were encoded using three different colour channels (red, green, blue), they are now encoded using a time-multiplexing approach and a single colour. All symbols with similar colour are grouped together producing a single channel of the pattern. In this case, three channels are constructed, each defining a unique pseudo-colour. Figure 4.2 shows the three individual pseudo-colour channels.

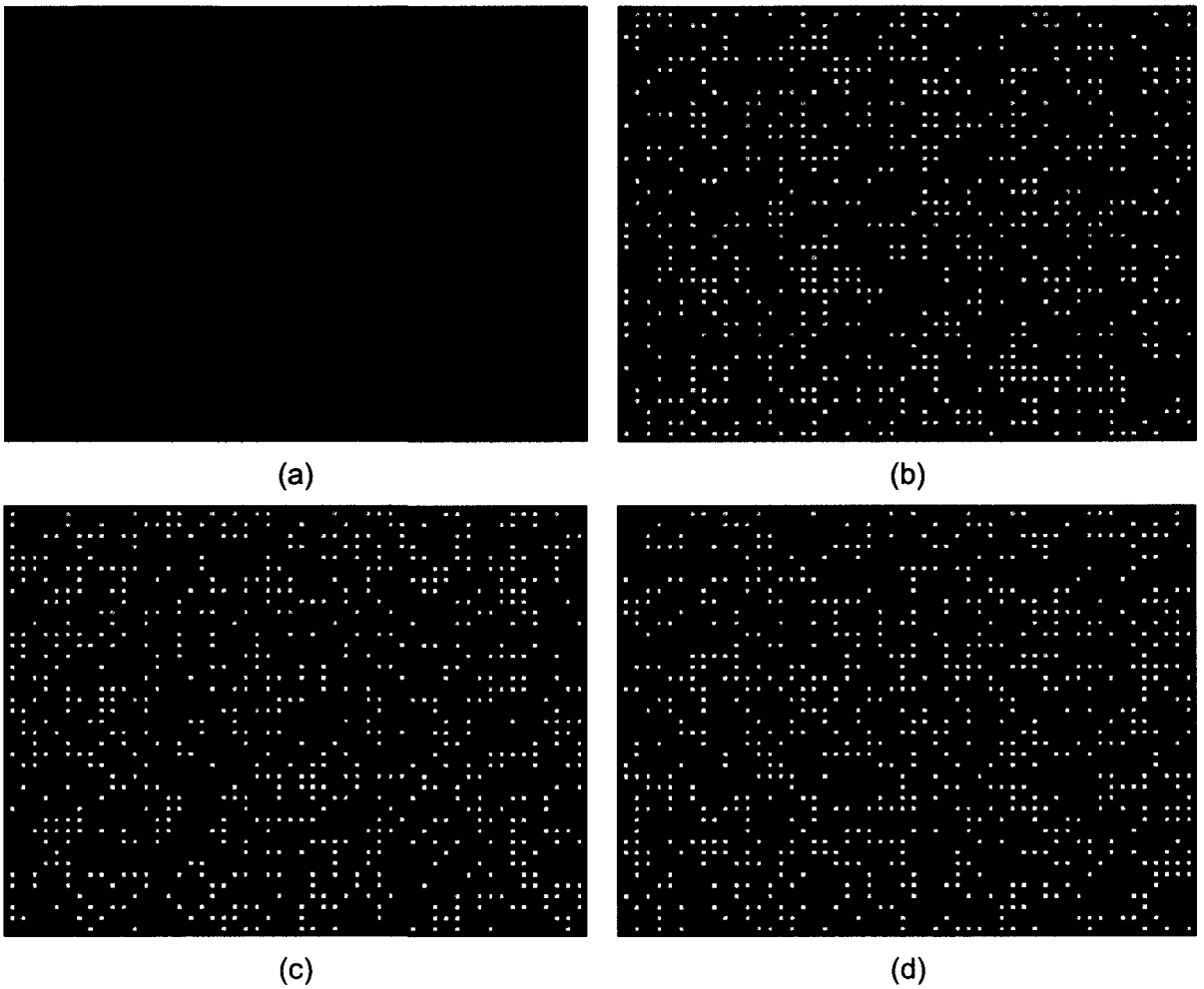


Figure 4.2. (a) Initial colour pattern and three time-multiplexed pseudo (b) -red, (c) -green and (d) -blue colour channels of the pattern.

The three individual pattern channels are successively projected in a time-multiplexing approach and three sets of images are respectively acquired. This simulates a simultaneous projection of the three colours. The square symbols in the acquired images are much more visible, which leads to a more reliable segmentation. The use of time-multiplexing does not introduce a significant increase in overhead since the colours do not represent unique code words but rather symbols, used to construct unique codes. In this case, only two extra projections and acquisitions are needed to obtain a complete pattern. However, this design imposes a constraint that limits the sensor to operate on static scenes only. It should also be noted that this is a separate time-multiplexing, not to be confused with the time-multiplexing at the entire pattern level, denoted as marching patterns, which increases the spatial density of the acquired range data.

With this new acquisition mode, the captured images do not contain any significant colour

information like the previous method. In practice, four pairs of colour images are acquired; a base pair with no projection and three pattern pairs, each with one pseudo-colour channel of the projected pattern. However, with the projection of white light and the difference between base and pattern images, only grayscale information is present. It is possible to leverage this result in order to significantly improve the segmentation part of the processing stage.

The existing method uses a global thresholding, based on hue thresholds, to segment the square colour regions of the pattern. With the colours no longer in use, a conventional intensity thresholding is applied to segment the square pseudo-colour regions. The acquired pattern image of a typical scene, shown in Figure 4.3, demonstrates the range of intensities possible. For example, the squares on the foreground objects are clearly visible and have a high intensity of grey. On the contrary, the squares on the background have a very low intensity since the black curtain does not reflect light as well as other colours.

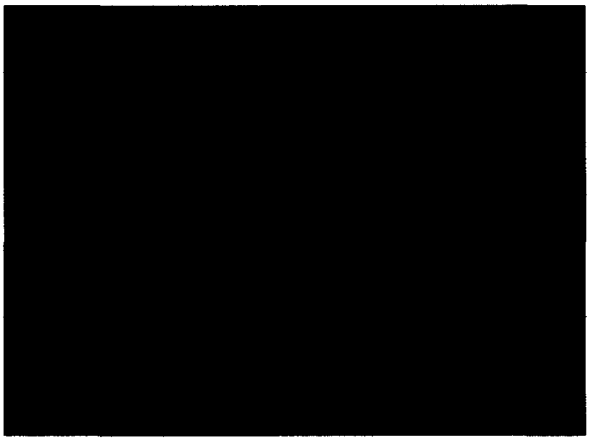


Figure 4.3. Difference in intensity between foreground and background objects.

Since the pattern channels are segmented using intensity, it is possible and relatively trivial to use an adaptive thresholding technique to perform the segmentation. The technique consists of determining a unique threshold for each pixel using the mean of pixel values around the point of interest. The thresholding is performed following Equation 4.1 where I is the original image, I' is the segmented image and T is the local threshold. The threshold T is computed over a block of size N where C represents a small constant used to eliminate the segmentation of noise in areas that contain no squares.

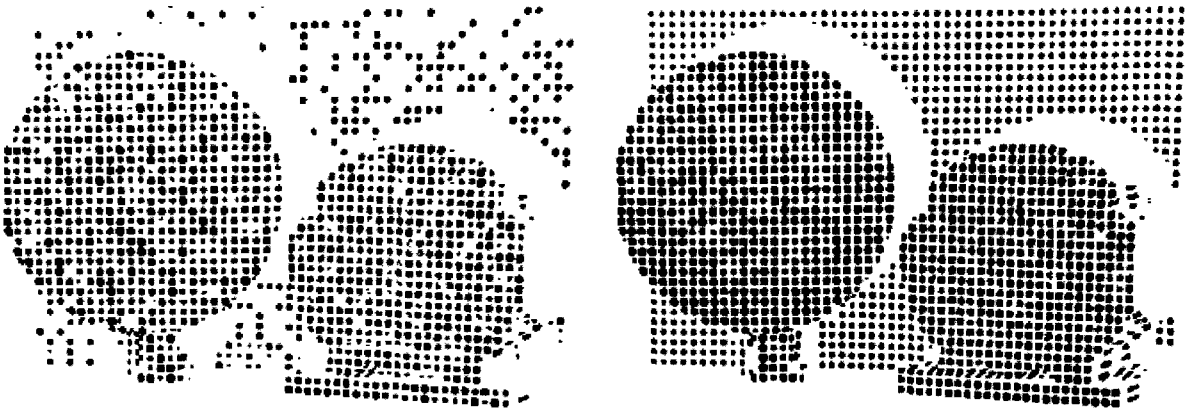
$$\begin{aligned} I'(i,j) &= 1, \textit{ if } I(i,j) > T(i,j) \\ &= 0, \textit{ otherwise} \\ T(i,j) &= \left[\frac{1}{N^2} \sum_{x=-N/2}^{N/2} \sum_{y=-N/2}^{N/2} I(i+x, j+y) \right] + C \end{aligned} \tag{4.1}$$

Since the new time-multiplexed acquisition mode does not use a coloured pattern, it is possible to implement the range sensor using black and white cameras. However, the main disadvantage with this approach is that it would be impossible to secure a colour measure for each 3D point in order to generate coloured models. Also, it is still possible to use the colour acquisition mode when the target scene does not have much colour variability and reflects the coloured pattern adequately. The benefit is a reduction of execution time to one half that of the new algorithm since only one pattern projection and image acquisition is necessary instead of three, assuming that the reference image acquisition is common to both methods.

4.2.3 Results

To evaluate the improved functionality of the new time-multiplexed acquisition mode, the dartboard and wicker basket scene is imaged using both modes and the results are compared. This scene is particularly interesting since the dartboard has a lot of colour variability, the basket consists of a textured surface with reflective strands and the black background does not reflect the projected pattern very well. Figure 4.4 shows the superimposed red, green and blue colour masks, from the projected squares segmentation, obtained using both acquisition modes.

The first noticeable improvement is that the time-multiplexed mode, shown in Figure 4.4.b, produces masks that are much more uniform. This is due to the fact that the same intensity of white light is projected for each pseudo-colour channel as opposed to the actual colours, which vary in overall intensity. As a result, the borders of the squares are more pronounced and consistent.



(a) (b)

Figure 4.4. Red, green and blue masks using (a) the colour and (b) the time-multiplexed acquisition modes.

As was previously shown in Figure 4.1.c, the red squares of the colour pattern tend to discolour when projected onto red surfaces. For example, the colour masks for the area containing the red number 7 target of the dartboard are shown in Figures 4.5.a and 4.5.b. When using the previous colour acquisition mode, the segmentation algorithm has difficulty detecting the discoloured squares. Most red squares are only partially detected and some are entirely missed. On the other hand, the masks computed using the new time-multiplexed mode are much more uniform and all of the pseudo-red squares are properly detected.

Next, the corresponding colour masks for a zoomed in section of the black background are shown in Figures 4.5.c and 4.5.d. Due to the low intensity of the red and green light emitted from the projector, as well as the poor reflectance characteristics of the black background, only the blue squares are detected when using the coloured pattern. The consequence is a complete loss of data for the background regions as it is impossible to detect 3x3 codes. Again, the new time-multiplexed acquisition mode is beneficial since the different pseudo-colour squares of the background are projected using the same white light and properly detected.

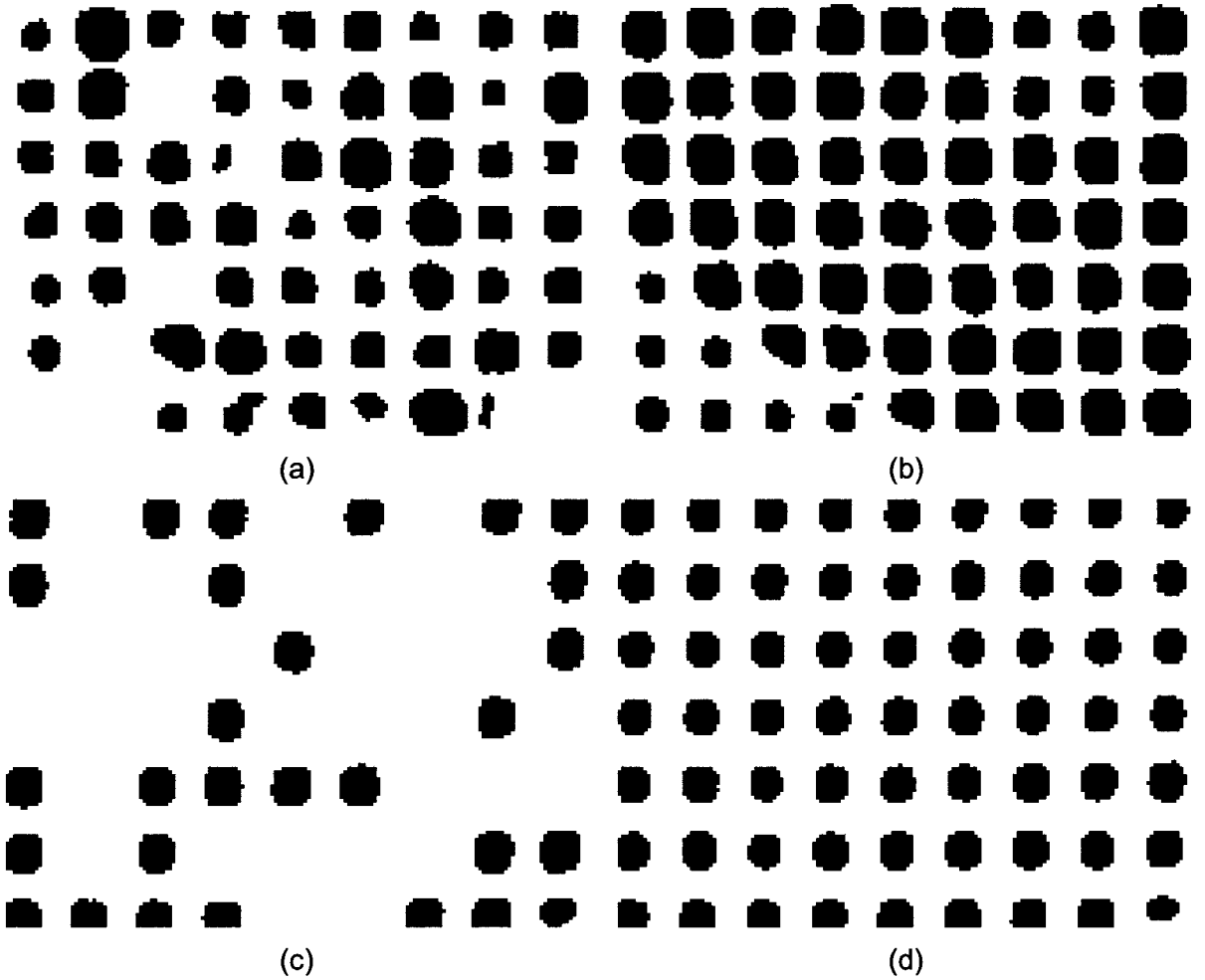


Figure 4.5. Zoomed in view of the masks for the dartboard using (a) the colour and (b) the time-multiplexed acquisition modes. Zoomed in view of the masks for the background using (c) the colour and (d) the time-multiplexed acquisition modes.

The benefits of using an adaptive threshold to perform the projected squares segmentation and generate the colour masks are shown in Figure 4.6. In this example, the time-multiplexed acquisition mode is used without and with adaptive thresholding. The main advantage, shown in Figures 4.6.a and 4.6.b, is that codes projected onto surfaces with low reflectance are detected. For instance, the squares projected onto the black background have a much lower intensity than those projected on the dartboard and basket. However, the squares on the background are detected when using adaptive thresholding, as shown by the colour masks, due to a lower local threshold used in that region.

The adaptive threshold is also beneficial in regions of high reflectance such as the reflective strands of the wicker basket woven in a crisscross pattern. Figures 4.6.c and 4.6.d show the colour masks of a zoomed in section of the basket. Using a global threshold for

segmentation, the squares that are projected onto the reflective strands are mostly split in two or completely undetected, because of a very high intensity in the centre. On the other hand, the adaptive threshold method handles the high intensity by increasing the local threshold, which results in a better segmentation. The resulting squares are not perfectly segmented but clearly present and useful for the detection of 3x3 codes.

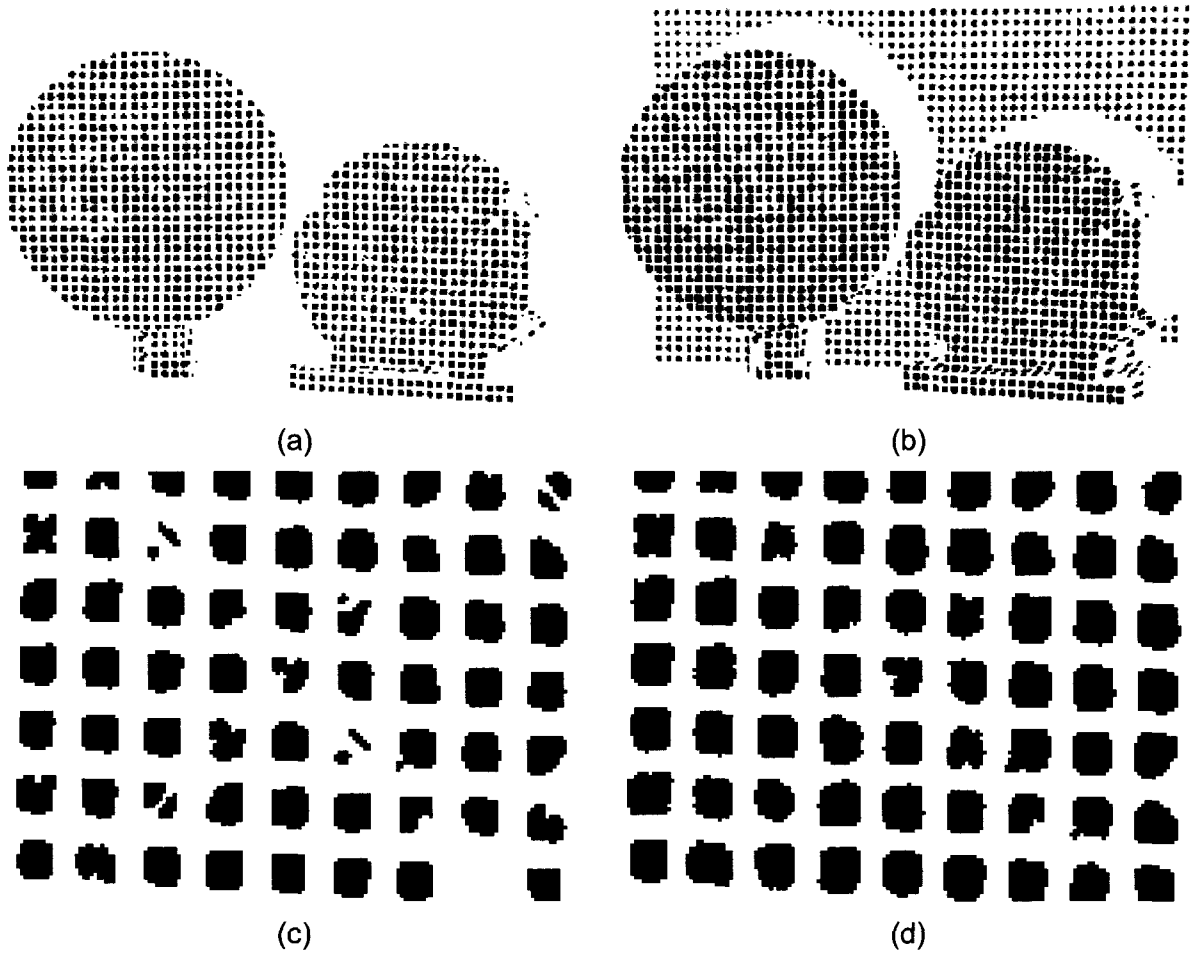
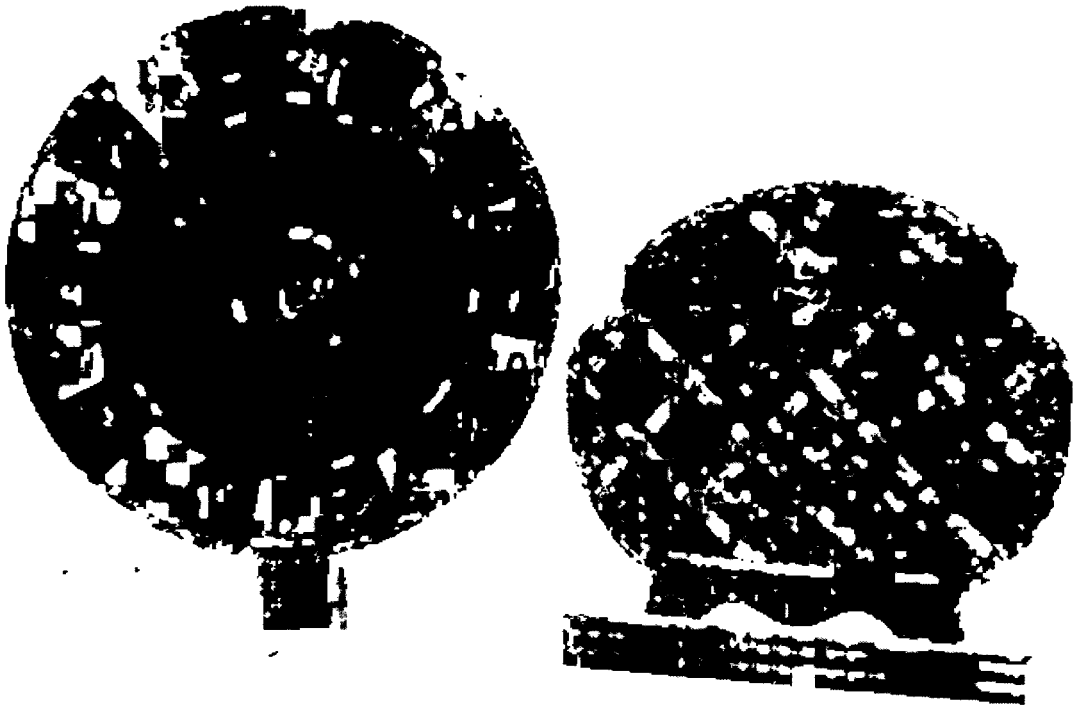


Figure 4.6. Red, green and blue masks using the time-multiplexed acquisition mode (a) without and (b) with adaptive thresholding. Zoomed in view of the masks for the basket (c) without and (d) with adaptive thresholding.

To easily visualize the advantages of the time-multiplexed acquisition mode, a high spatial density scan of the scene is performed using both methods. In this test, the pattern is marched six times in the horizontal and vertical directions to achieve a spatial density of approximately 2.5mm. The resulting meshes for both methods are shown in Figure 4.7.



(a)

(b)

Figure 4.7. Views of the dartboard and basket scene mesh using (a) the colour and (b) the time-multiplexed acquisition modes.

The surface mesh obtained using the colour acquisition mode, shown in Figure 4.7.a, has many holes due to undetected squares in key areas. As previously discussed, the red targets on the dartboard, the reflective strands on the basket and the entire black background are not detected. Figure 4.7.b shows that all of the problems are addressed with the new acquisition mode as it can handle surfaces with significant colour variability, as well as high and low reflective characteristics. Another benefit is that the new mode produces surface meshes that are less susceptible to noise since the projected squares are segmented more uniformly and with better defined borders.

As a final note, it is important to stress that the improved results, shown above, arise from the combination of both the new time-multiplexed acquisition mode and the adaptive thresholding segmentation. Although the two techniques overlap and address similar problems, their combined use adds robustness to the acquisition stage as opposed to using one of the techniques by itself. Moreover, the use of the exposure fusion algorithm during image capture, as explained in the next section, is another solution that addresses much of the same problems discussed in this section, and works in parallel with the new acquisition mode and the adaptive thresholding.

4.3 Exposure Fusion

Since the PR pattern is projected using a maximum intensity, objects and areas with high reflectance properties lead to saturated regions in the acquired images. The previous implementation does not compensate for objects with multiple reflectance characteristics and naively uses a global exposure. Also, this parameter has to be adjusted manually for each and every object and scene. To perform automated acquisitions, the exposure time must be automatically selected since it is the most important parameter to set when capturing digital images.

However, with most computer vision algorithms, a single global exposure is not sufficient to properly image the entire scene. In order to capture the pattern on objects of multiple colours and reflection properties, the proposed solution is to achieve an optimized local exposure when acquiring images using an exposure fusion (EF) technique. Several images of the same scene are acquired while varying the exposure time and then fused to produce a single image that has a dynamic range greater than what is possible to obtain from a single exposure. In other words, this algorithm allows for a normalization of colour and reflectance properties regardless of the characteristics of the scene. Moreover, the selection

of an exposure time parameter disappears since it is inherently selected via the EF algorithm.

4.3.1 Algorithm Application

The principal motivation of using an EF algorithm is to increase the dynamic range of the cameras through software. Essentially, an optimal local exposure is computed for each pixel when acquiring an image from the camera. To achieve this using conventional camera hardware, the EF algorithm is applied every time the structured light sensor performs an image acquisition.

In the context of the range sensor, there are three different types of image captures. The first capture is to secure a reference colour image that is used to map colour measures to the triangulated 3D points. Due to the nature of the EF algorithm, the colours in the acquired images tend to fade and get distorted when the algorithm is applied. Therefore, the EF algorithm is not invoked when acquiring a reference colour image in order to ensure high accuracy and exactness of the colours mapped to the range dataset.

The second and third captures are to detect the pattern projected onto the scene and in this case, the EF algorithm is invoked when acquiring images. The second capture consists of acquiring images of varying exposure, without the projected pattern, and fusing them to produce a base image, which is used for the background subtraction. The third and final capture consists of acquiring another set of images, using the same exposure times, with the pattern projected onto the scene, and fusing them to produce a pattern image.

In both the second and third captures, the EF algorithm is applied to the respective sets of images and the weights for the algorithm are recomputed each time, since the variations in scene reflectance from the changing pattern must be accounted for. Finally, the base image is subtracted from the pattern image, essentially resulting in a mask of the pattern. This described procedure is applied for the detection of all patterns, including the coloured PR pattern as well as the three pseudo-colour channels of the pattern.

Although there are many techniques and algorithms to perform exposure fusion, the literature does not generally deal with the problem of selecting the number of images required and their respective exposure time. The proposed solution is to develop an algorithm that has the ability of selecting relevant exposures automatically. The first step is to run the on-board automatic exposure calibration routine of the cameras, which is usually

available on current camera hardware. This estimates the best global exposure time for the camera given its current configuration and a target brightness level. In this case, a low brightness level target of 30% is specified since the sensor will be projecting light onto the scene latter on. Secondly, an exposure time range is defined using Equations 4.2 through 4.4. The global exposure time parameters of each camera E_{global}^{left}, E_{global}^{right} are queried using the camera API and the average global exposure E_{global} is computed. The minimum exposure E_{min} is set to 0 and the maximum exposure E_{max} is set to 1.5 times E_{global}.

$$E_{global} = \frac{E_{global}^{left} + E_{global}^{right}}{2} \tag{4.2}$$

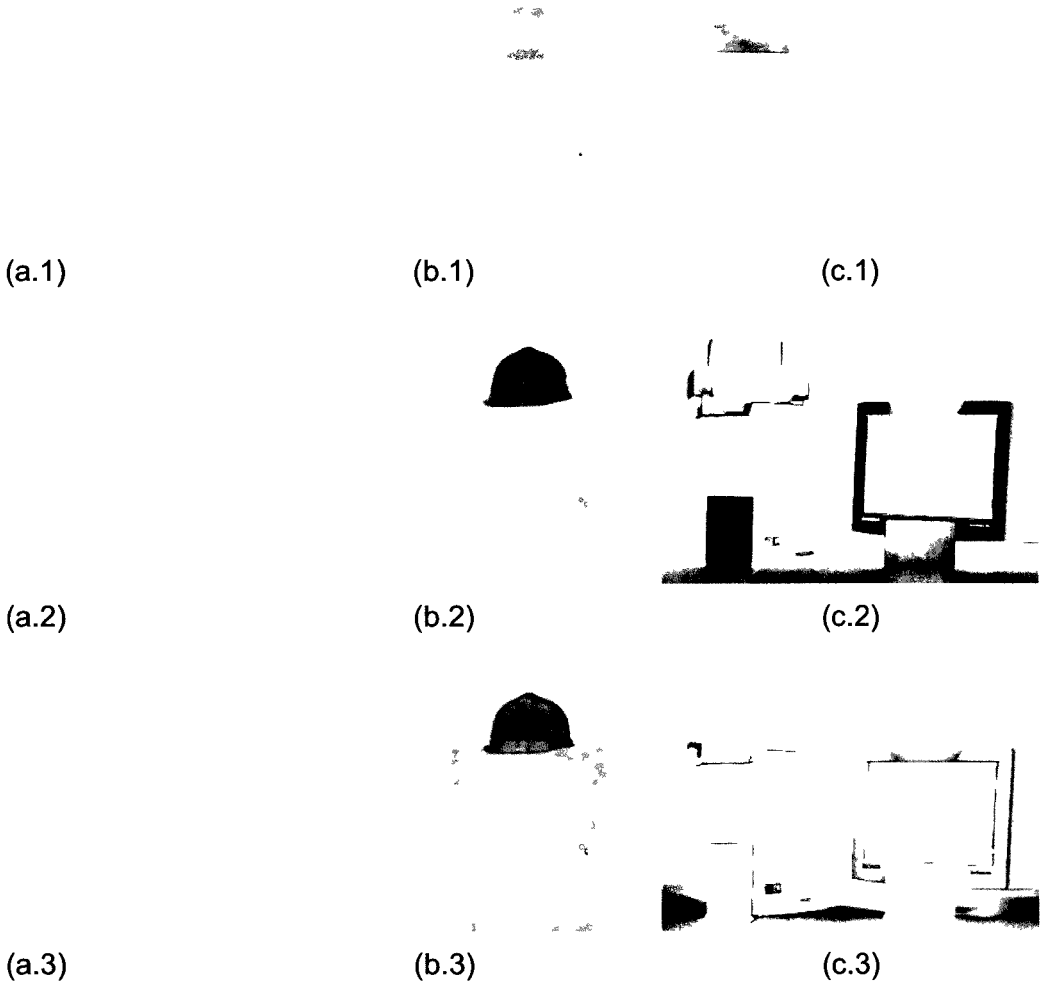
$$E_{min} = 0 \tag{4.3}$$

$$E_{max} = 1.5 E_{global} \tag{4.4}$$

Finally, the exposure range is divided by a parameter N, specifying the number of different exposure times to acquire. This results in N+1 distinct exposure times, where the first time corresponding to E_{min} is dropped. The parameter N is specified before acquisition; a default of 5 gives good results with a wide variety of different scenes. This establishes a trade-off within the acquisition stage; as E_{max} and N are increased, the sensor's effectiveness in handling colours and reflectance characteristics also increases at the cost of a longer acquisition time.

The EF algorithm of Mertens *et al.* [14], introduced in Section 2.2.2, makes use of quality measures (QM) to determine which areas of the input images are optimally exposed. In [14], three quality measures including contrast, saturation and well-exposedness are computed for each pixel of each image in the input set. Given the application of the EF algorithm to the range sensor system, the relevance of the quality measures is determined by performing several tests. An example scene with several objects of different reflectance characteristics, shown in Figure 4.8, is considered. Figure 4.9 shows the three QM maps for three different exposure times respectively. It is easy to see that the contrast QM does not contribute much to the overall weight map. Zoomed in views of the contrast QM maps, shown in Figure 4.10, demonstrate that only a very faint component is present around edges. For this reason, the contrast QM maps are not used in this implementation.

Figure 4.8. Original image of the computer scene from the right-hand camera.



(a.1) (b.1) (c.1)

(a.2) (b.2) (c.2)

(a.3) (b.3) (c.3)

Figure 4.9. (a) Contrast, (b) saturation and (c) well-exposedness QM maps of a scene acquired using an exposure time of (1) 9ms, (2) 24ms and (3) 39ms, where darker shades of gray represent higher weight values.

(a) (b)

Figure 4.10. Zoomed in views of contrast QM maps for exposure times of (a) 24ms and (d) 39ms, where darker shades of gray represent higher weight values.

The same cannot be easily determined for the saturation and well-exposedness QM maps. To compare the latter, the weight maps are computed for different combinations of quality measures. As detailed in [14], the final weight maps are defined by computing a point-based multiplication of the QM maps and then normalizing the result. Figure 4.11 shows the computed weight maps for three combinations of QM maps including all QM maps, the saturation and well-exposedness QM maps, and only the well-exposedness QM map. As shown in Figure 4.11.a, the contrast QM adds a lot of speckle noise in the weight map. Also, the saturation QM, shown in Figure 4.11.b, does not differ very much from the well-exposedness QM. Given these observations, it is clear that all three combinations lead to very similar weight maps, and that the contrast and saturation QM do not contribute much more information when multiplied to the well-exposedness QM. This is not surprising since the premise of the EF algorithm is to detect properly exposed pixels, which is what the well-exposedness QM estimates. Due to these observations, only the well-exposedness QM is computed when performing the EF algorithm in this implementation. The main advantage is that the execution time of the algorithm is reduced significantly with no detectable loss of quality with respect to the exposure fusion of images.

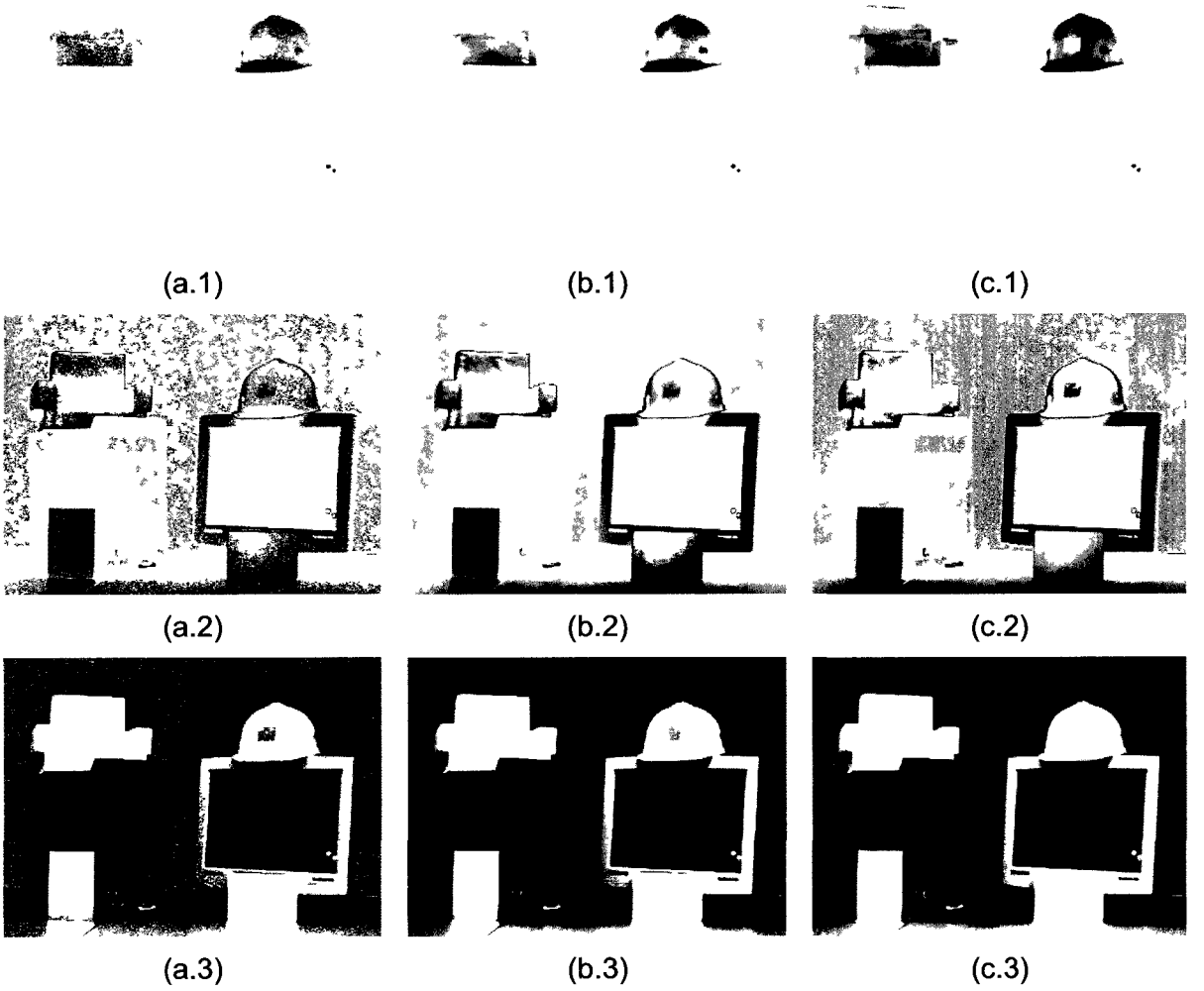


Figure 4.11. Weight maps using (a) all QM maps, (b) saturation and well-exposedness QM maps, and (c) only the well-exposedness QM map for exposure times of (1) 9ms, (2) 24ms and (3) 39ms, where darker shades of gray represent higher weight values.

In order to perform the blending of the input images, Mertens *et al.*'s EF algorithm [14] uses a multiresolution technique based on pyramid decomposition. The original input images and the weight maps are decomposed into Laplacian and Gaussian pyramids respectively, multiplied together and the synthesized image is generated by applying the inverse Laplacian pyramid transformation. This results in the need to define a parameter controlling the number of levels of decomposition. Although Mertens *et al.* do not discuss the selection of this parameter, in most situations, a value of 5 levels proves sufficient to achieve good results. More levels do not show significant increases in quality, while fewer levels greatly reduce the effectiveness of the multiresolution blending.

Another detail with respect to the pyramid decomposition that Mertens *et al.* do not discuss

is the size of the original input images. Since pyramid decomposition involves down-sampling the images by a factor of 2, in both dimensions at each level, the width and height of the input images must be selected such that the width and height of the last level are even numbers. To achieve an automated acquisition, the dimensions of the input images are dynamically adjusted to ensure that the input images are divisible by 2 up to N levels. In the context of the range sensor, only a region of interest (ROI) of the images obtained from the cameras is processed and sent to the EF algorithm. This ROI only contains the area of the field of view that is projectable by the projector and its estimation is presented in Section 5.1.1. However, this estimated ROI is refined using Equation 4.5, where N represents the number of decomposition levels, W represents the width and H represents the height, in order to ensure that the input images can be properly decomposed up to N levels.

$$\begin{aligned} F &= 2^{N-1} \\ W' &= W + \left[\left(\lceil W/F \rceil \cdot F \right) - W \right] \\ H' &= H + \left[\left(\lceil H/F \rceil \cdot F \right) - H \right] \end{aligned} \tag{4.5}$$

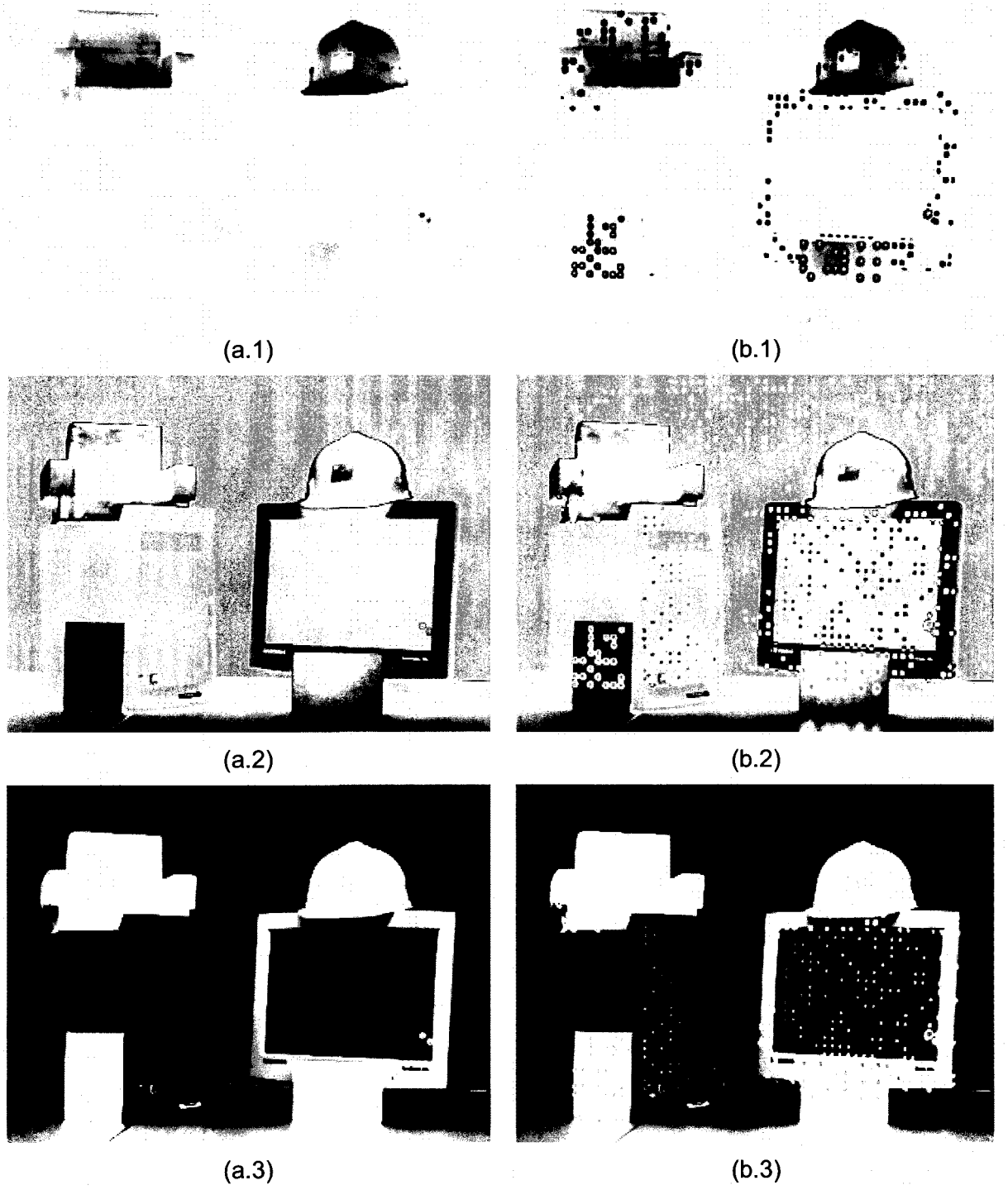
An important aspect to consider when using an EF algorithm is its effectiveness on the areas of the input images that contain the PR pattern. The square shaped symbols do not pose a problem to the algorithm proposed by Mertens *et al.* since it is pixel-based. On the contrary, block-based algorithms, as detailed in Section 2.2.1, tend to distort the symbols when they are located along the block boundaries, even when the techniques perform blending between the blocks. Another advantage to Mertens *et al.*'s algorithm is that it executes efficiently and is very easy to parallelize because of its pixel oriented approach. Finally, the algorithm is flexible since it is possible to define new quality measures depending on the application. This is possible since the weight map computation is separated from the pyramid decomposition, unlike other techniques.

4.3.2 Results

The enhancements provided by the application of the EF algorithm are evaluated by imaging the computer scene of Figure 4.8 and comparing the results with an acquisition of the scene without the use of the algorithm. This particular scene is relevant since it has surfaces made of different materials, including plastic, glass, paper, metal and fabric. Each surface type has different reflectance characteristics that affect how the pattern is reflected and how the cameras acquire images.

Since the projected PR pattern affects the effectiveness of the exposure fusion algorithm, the weight maps are first analyzed. The weight maps generated by Mertens *et al.*'s algorithm for three exposure times are shown in Figures 4.12.a and 4.12.b for the base image and for one of the pattern images respectively. A stronger intensity of black indicates more weight attributed to the pixels of the corresponding exposure time. The base image weight maps show that the highly reflective surfaces such as the chair, the hat, the monitor border and the metal planes are captured at lower exposures and the less reflective surfaces such as the computer, the monitor screen and the background fabric are captured at higher exposures.

The pattern image weight maps show that the exposure fusion algorithm is capable of dealing with the projected squares, which introduce sharp fluctuations in reflectance characteristics. Since Mertens *et al.*'s algorithm is pixel-based, the areas with projected light use different exposure times than the adjacent areas without projected light. This is not possible with block-based exposure fusion techniques since the latter attempt to find the best exposure for the entire block, containing areas with and without projected light.



(a.1) (b.1)

(a.2) (b.2)

(a.3) (b.3)

Figure 4.12. Weight maps of (a) base images and (b) pattern images for exposure times of (1) 9ms, (2) 24ms and (3) 39ms.

The above weight maps are used to generate composite base images and pattern images, which have a proper exposure over all regions. These images are subtracted to eliminate

the background and isolate the projected pattern as shown in Figure 4.13. Although the EF algorithm makes use of different exposures for various image regions, background subtraction is still possible since the same exposure is used for the background region of the base and pattern images, as no additional light is projected in that region. It is evident that the exposure fusion algorithm generates much more uniform pattern images as seen in Figures 4.13.b and 4.13.d when compared against images obtained without exposure fusion shown in Figures 4.13.a and 4.13.c. The squares have a much more consistent size and shape when exposure fusion is used, which allows for a more reliable and precise segmentation. As well, the projected light on the black background is visible and therefore detectable, which is not the case when using a single exposure.

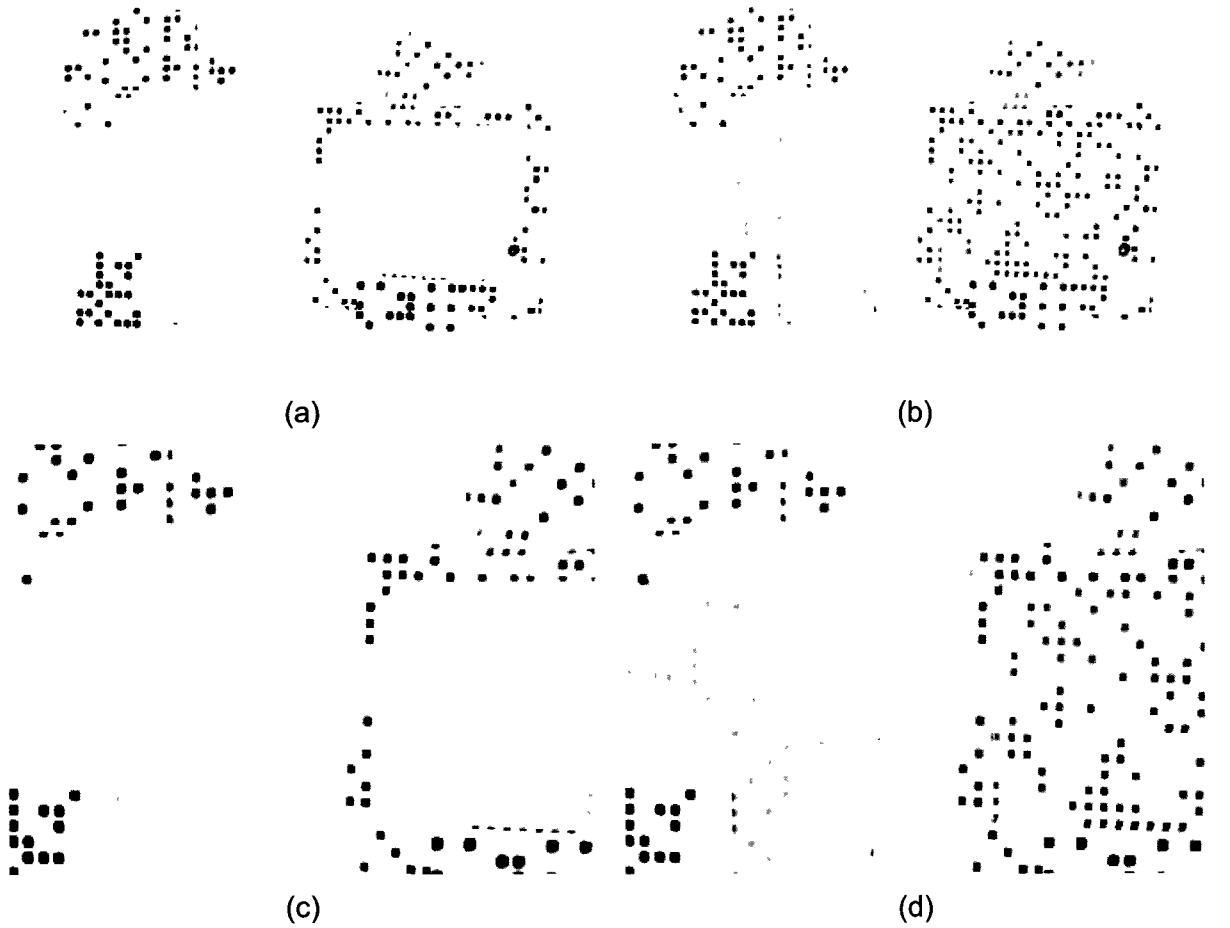


Figure 4.13. Background subtraction of pattern (a) without and (b) with exposure fusion. Zoomed in view of the pattern (c) without and (d) with exposure fusion.

The more uniform background subtraction images lead to higher quality segmentation. The results of the projected squares segmentation algorithm using only one exposure is shown

in Figure 4.14.a, whereas the segmentation results using the exposure fusion algorithm with five exposure times is shown in Figure 4.14.b. These images represent the squares that are actually detected by the acquisition stage and used for further processing. Not only is the entire background detected, apart from occluded areas, with the exposure fusion algorithm, but the squares are much more uniform over the entire image. This is clearly visible on the computer, as the projected squares are properly detected over both faces. It is even possible to detect the squares projected into the recess of the disk drive.

The zoomed in views of Figures 4.14.c and 4.14.d demonstrate the change in shape of the blobs. When only using one exposure, the blobs are segmented into more circular shapes as opposed to more square shapes when using exposure fusion. The benefit is that the centroid of the blob is better estimated with square blobs, which leads to more accurate triangulation results and overall better precision.

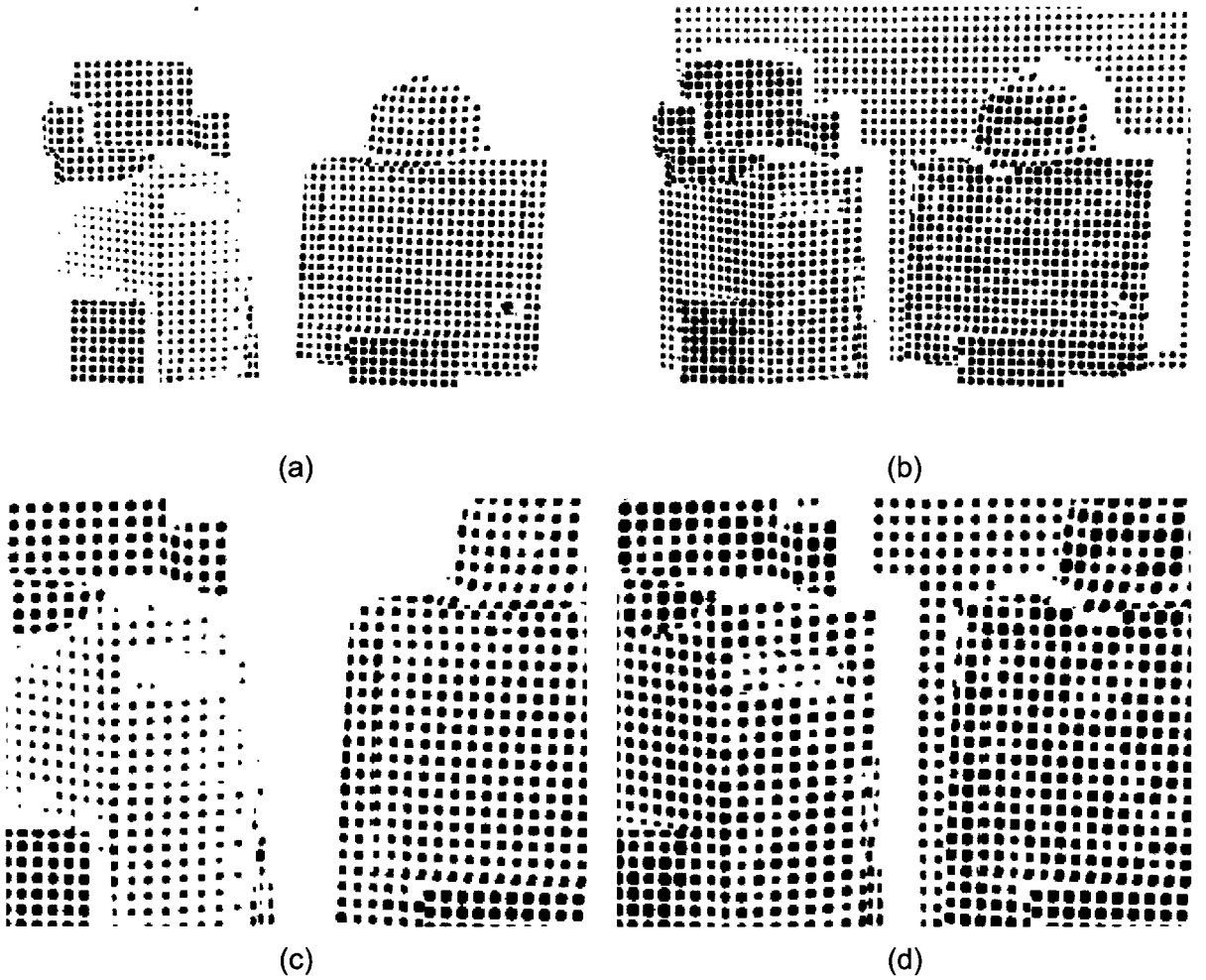
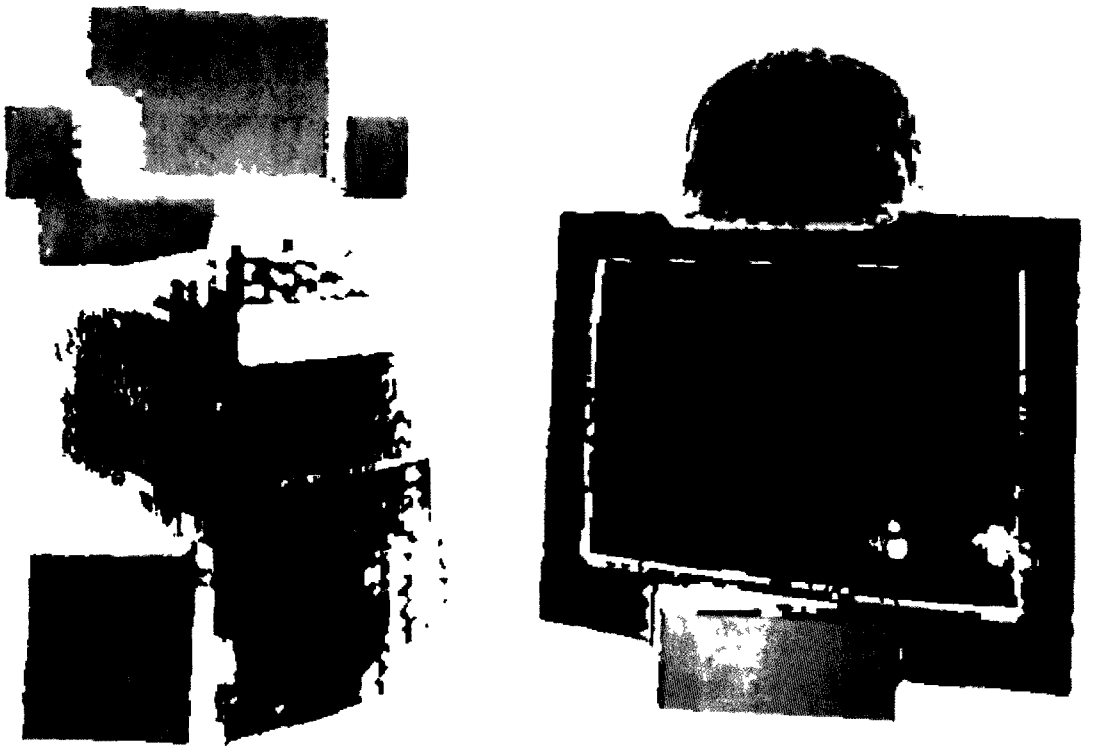


Figure 4.14. Red, green and blue masks (a) without and (b) with exposure fusion. Zoomed in view of the masks (c) without and (d) with exposure fusion.

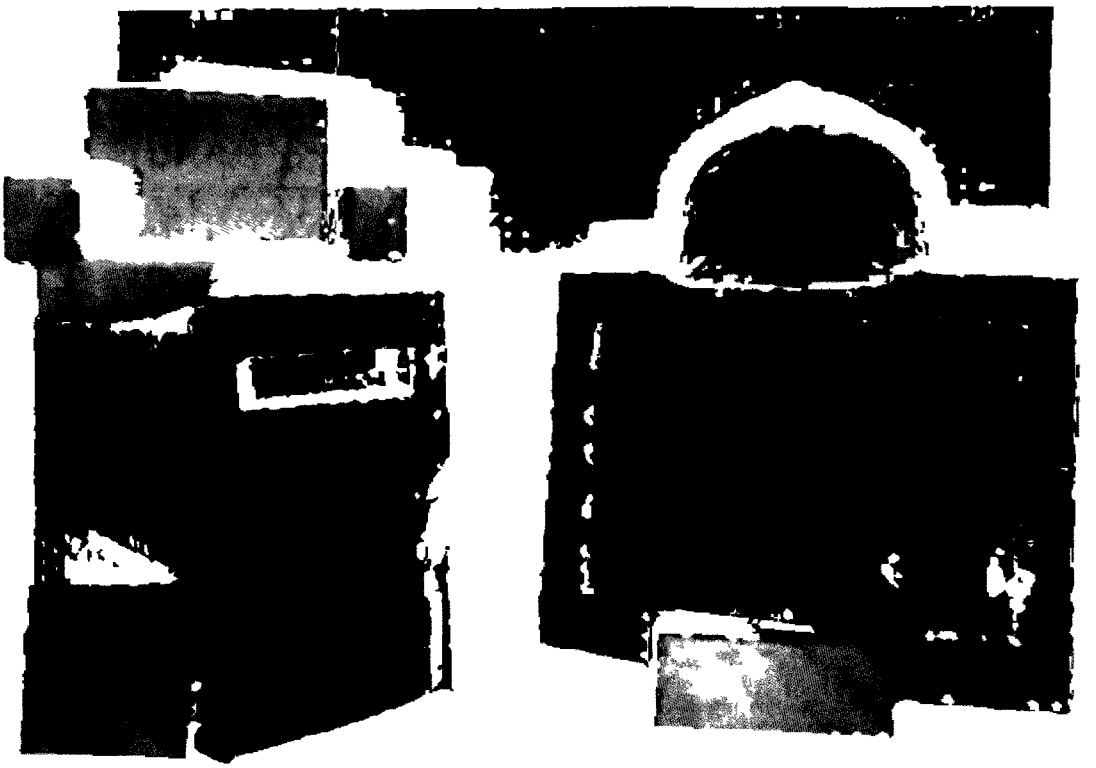
The benefit of using the exposure fusion algorithm during camera capture is best shown by comparing the surface meshes generated without and with the algorithm. Again, a high spatial density scan of the computer scene is performed by marching the pattern six times in the horizontal and vertical directions, which results in a spatial density of approximately 2.9mm. The surface mesh obtained using a single exposure capture is shown in Figure 4.15.a and is missing some information when compared with the mesh obtained using the exposure fusion algorithm, as shown in Figure 4.15.b. Once more, the new method allows for the proper detection of the entire computer, the recessed disk drive, the seam between the monitor screen and plastic border as well as the black background.

Although the exposure fusion algorithm compensates for multiple reflectance characteristics, it has some limitations. One limitation that is shown in the surface mesh of Figure 4.15.b, is the inability of dealing with intense specularities. For example, the bottom right hand corner of the monitor screen is characterized by two areas that have not been properly detected. This is due to the fact that the glass surface produces a specular reflection of the projector's light at those locations. This can be seen for the right-hand camera in Figure 4.8. Since the left-hand camera is not at the same location, the specularity appears in a different location and a total of two holes are present in the final mesh.

On the whole, the exposure fusion algorithm does not radically improve the resulting surface meshes but definitively ensures a better coverage of complex scenes. It plays an important role, along with the time-multiplexed acquisition mode and the adaptive thresholding segmentation, to intelligently handle different reflectance characteristics of objects in the scene. The three algorithms work in parallel to provide a robust detection of projected squares over many different combinations of surface colours, materials and angles. Also, the sensor is able to adapt to shadows in the scene, as well as different lighting conditions.



(a)



(b)

Figure 4.15. Views of the computer scene mesh (a) without and (b) with exposure fusion.

4.4 Focus Fusion

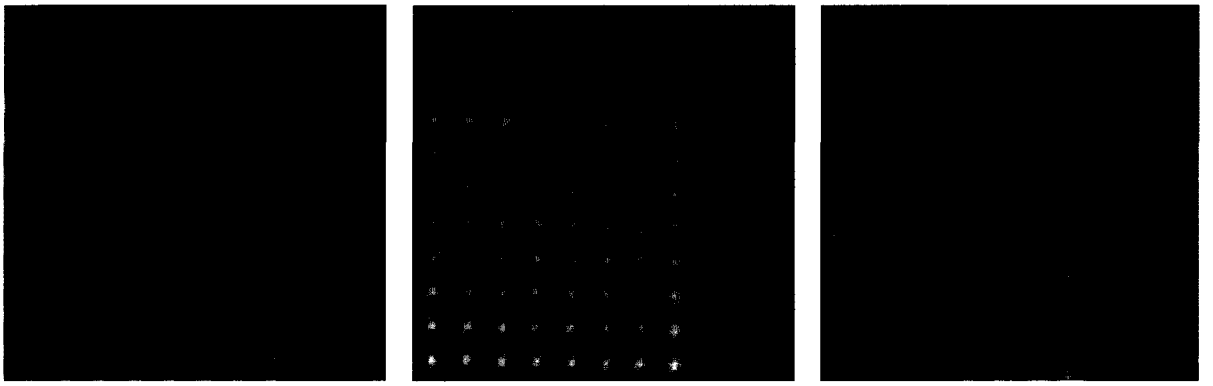
With the implementation of functionalities detailed in the previous sections, the sensor can now acquire data from surfaces with multiple colours and reflectance properties as long as they all lie within the same focus plane and the projector is properly focused to that plane before the acquisition. A focus plane is defined as a planar region, perpendicular to the sensor's principle axis, where the projected pattern has similar focus. The assumption that objects will lie in the same focus plane targeted by the projector's current focus setting cannot be made when designing a sensor capable of operating automatically in unconstrained environments.

The workspace of a structured light range sensor is usually constrained by the focus and the intensity of the projector. The minimum distance of the workspace is bound by the focal capabilities of the projector while the maximum distance is bound by the intensity of the projector. In most realistic applications, the cameras and lenses of the structured light setup can be configured such that the entire workspace is in focus. Since the projector is already using a maximum intensity to project the pattern, the only parameter that can be adapted is the focus of the projector as it varies considerably within the workspace. The proposed solution is to vary the focus of the projector from the closest to the farthest focus planes, capture images for each corresponding depth, perform the image processing and triangulation at each plane, and then merge the results. Although the process makes use of a focus fusion (FF) algorithm, the latter is only used to determine which regions of the images are in focus at the different focus planes.

4.4.1 Algorithm Application

Unlike the exposure fusion process, Hariharan *et al.*'s FF algorithm [16], introduced in Section 2.3.2, is not used to fuse multiple images of different projector focus into a synthesized image. As the projector lens is focused, by adjusting the focal length, the changing field of view angle of the lens makes the projected pattern shift slightly. The shift radiates out from the principle axis of the projector and becomes more prominent as the distance from the principle axis increases. For example, Figure 4.16 shows the top-right corner of the projected pattern under three different settings of projector focus, all imaged from the same viewpoint with the same camera and lens settings. As the focus is adjusted for nearer distances, the pattern tends to shift outwards from the centre of the projection as shown in Figure 4.16.a. The opposite occurs as the focus is adjusted for farther distances

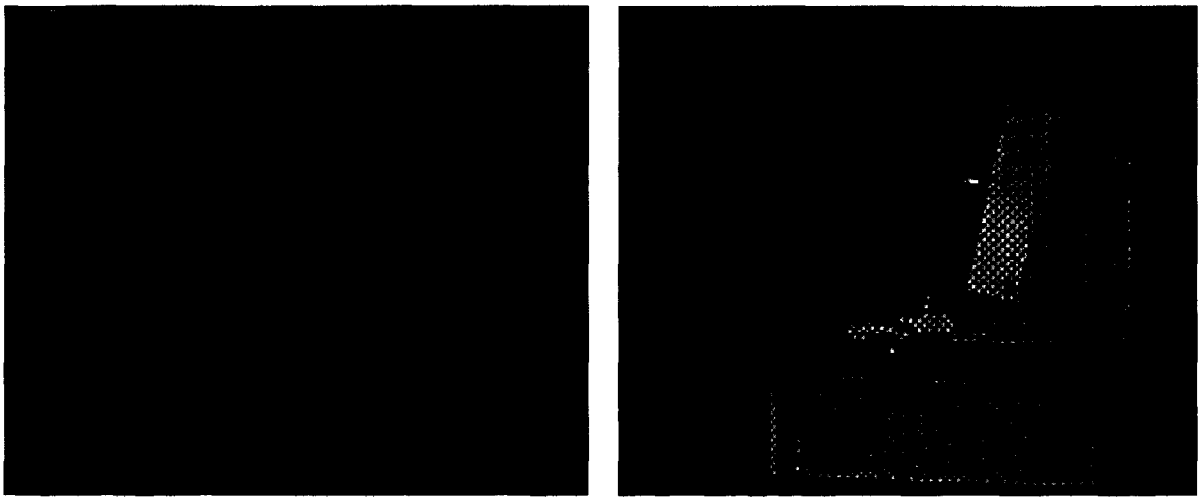
as shown in Figure 4.16.c. Because of this shifting effect, the FF algorithm is not used in its entirety since this would produce a composite image that contains a pattern that does not line up across focus regions. Instead, the algorithm's fusion step is dropped and only the analysis stage is used to identify focally connected regions. In other words, the FF algorithm is used to determine which regions of the field of view are in optimal focus at each focus plane.



(a) (b) (c)

Figure 4.16. Top-right corner of projected pattern using (a) near, (b) correct and (c) far projector focus settings.

Since it is not possible to apply the FF algorithm directly, a novel two-pass procedure is developed in order to capture range information from all focus planes. During the first pass, a special chessboard pattern, as show in Figure 4.17.b, is projected in order to measure the projector focus on different parts of the scene. The width and height of the pattern are as large as the resolution of the projector, ensuring that the entire projectable area is considered. To begin, the projector focus is set such that the nearest focus plane is in focus. The projector's lens is automatically cycled in one direction through each focus plane and images of the chessboard pattern are acquired. The analysis stage of Hariharan *et al.*'s FF algorithm [16] is used on the difference images, from background subtraction, to compute sharpness and partition masks of each plane. It should be noted that the sharpness masks estimate the focal characteristics of the image, and are used to generate the partition masks. The partition masks ultimately determine which areas of the chessboard pattern are in focus for the different focus planes. The masks are analyzed to determine the coverage of their focally connected regions and the focus planes that are in focus for less than 5% of the scene are dropped. By eliminating the focus planes that do not contribute any meaningful data, only a subset of focus planes is processed, which speeds up the execution of the second pass.



(a) (b)

Figure 4.17. (a) Original image of the laboratory scene and (b) chessboard pattern projected onto the scene to measure focus.

For the second pass, the projector focus is cycled in the opposite direction from the farthest to the nearest focus settings. At each focus plane of the subset, a full structured light acquisition is launched. The image processing is only performed on a region of interest of the image as defined by the respective partition mask of the focus plane. This results in the measurement of the scene in properly focused regions only and the generation of independent 3D point clouds for each focus plane. A surface is interpolated independently for each point cloud and the result is concatenated into a single mesh that contains the measured surfaces over the entire depth of field.

The control of the projector focus is done by communicating with the projector via the serial port. Many LCD projectors allow the programmatic control of most functions via similar interfaces. Commands are sent using an asynchronous communication channel to adjust many settings available on the projector. More details are presented in Appendix 1. In this case, only the control of the lens focus is necessary. The problem with the projector used in this experimentation and most others, is that the serial port commands only emulate the buttons available on the projector itself or its infra-red remote control. All of the setting changes are therefore relative to their current position, meaning that there is no way to command a setting to an absolute value. For example, the focus setting of the lens can only be increased or decreased. Also, since the serial port feedback only verifies whether the command was accepted or not, it is impossible to determine by how much the focus was modified. It is assumed that the focus increments are somewhat constant throughout the

entire range of focus.

With these specifications, a control algorithm is developed. A manual calibration is first performed in order to determine the number of increments N available within the focus range. The projector focus is set to the beginning of the focus range, which corresponds with the nearest focus plane, and the number of increments needed to displace the lens to the end of the range is counted. This defines N+1 focus planes that can be accessed by displacing the lens to the beginning of the range and sending commands to increase the focus a certain number of times. The optimal solution for the two-pass procedure is to perform the first pass in one direction while increasing the focus and the second pass in the opposite direction while decreasing the focus. However, in practice, the projector gives an extra focus plane over the entire range when decreasing the focus setting as opposed to increasing it. To ensure that the same focus planes are used for both passes, the latter are both performed from the beginning to the end of the focus range using the same serial port command to increase the focus. Therefore, between the two passes, there is a re-initialization stage that brings the projector focus back to the beginning of the range.

The focus setting increments do not map to equal distances between focus planes. Table 4.1 and Figure 4.18 show the non-linearity between the focus increment and the distance from the projector at which the projected pattern is in focus. It is clear that as the focus is set for farther distances, the distance between focus planes increases. Also, the projector focus setting is not as sensitive at farther distances. This is evident by the increasing tolerance margin in Table 4.1, as the focus increment approaches the infinity focus setting.

Focus Increment	Distance (cm)	Focus Increment	Distance (cm)
1	80 ±5	10	168 ±20
2	85 ±7	11	190 ±25
3	90 ±10	12	210 ±30
4	98 ±10	13	247 ±40
5	106 ±10	14	285 ±50
6	115 ±10	15	340 ±80
7	125 ±15	16	420 ±110
8	137 ±15	17	540 ±200
9	150 ±20	18	infinity

Table 4.1. Non-linearity between focus increments and distance from projector where pattern is in focus.

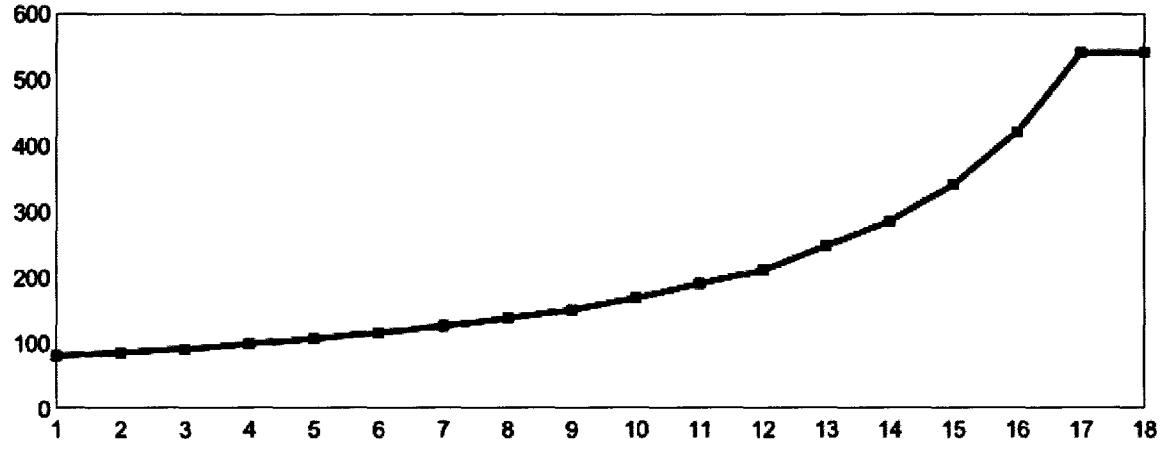


Figure 4.18. Non-linearity between focus increments and distance from projector where pattern is in focus.

Using all of the focus planes defined in Table 4.1 does not give good results for the acquisition of data at different planes, since there is a lot of overlap in the adjacent focus planes. For this reason, only every third focus plane including 1, 4, 7, 10, 13 and 16 is used during the acquisition stage. This ensures that there is little overlap between focus planes and improves the quality of the sharpness and partition masks generated by the FF algorithm. It is necessary to have significant differences between focus planes such that clear differences in the sharpness masks are produced.

The analysis stage of the FF algorithm specifies that the input images are first analyzed for sharpness using horizontal and vertical Sobel masks. Next, sharpness masks are generated by combining the two gradients and then low-pass filtering the result to reduce noise and increase neighbourhood relevance. It is mentioned in [16] that the size of the kernel must be selected relative to the size of objects in the scene. Since the focus of the projector is targeted for adjustment and not the focus of the cameras, the kernel size is selected such that it is slightly larger than the resolution of the chessboard pattern used in the first pass. This ensures that the focus of the projector is measured as opposed to the focus of the cameras as they are in focus over the entire workspace of the projector. In this application, a square 31x31 pixel averaging kernel is used to perform the filtering.

A couple of modifications are made to Hariharan *et al.*'s [16] FF algorithm in order to improve the quality of the partition masks. In this application, the FF algorithm is used to find focally connected regions of the projected chessboard pattern. Since the pattern has a lot of inherent variability, with the sharp transitions between filled and empty squares, extra smoothing is required. The same averaging kernel is therefore applied four times, in series,

to the sharpness masks in order to further smooth out the data. Figure 4.19 shows the improving quality of the sharpness masks as they are filtered in series. In Figure 4.19.a, the chessboard pattern is still visible and can lead to noisy partition masks. However, after the fourth iteration in Figure 4.19.d, the sharpness mask is more uniform and the darker regions give a better estimation of image regions that are in focus.

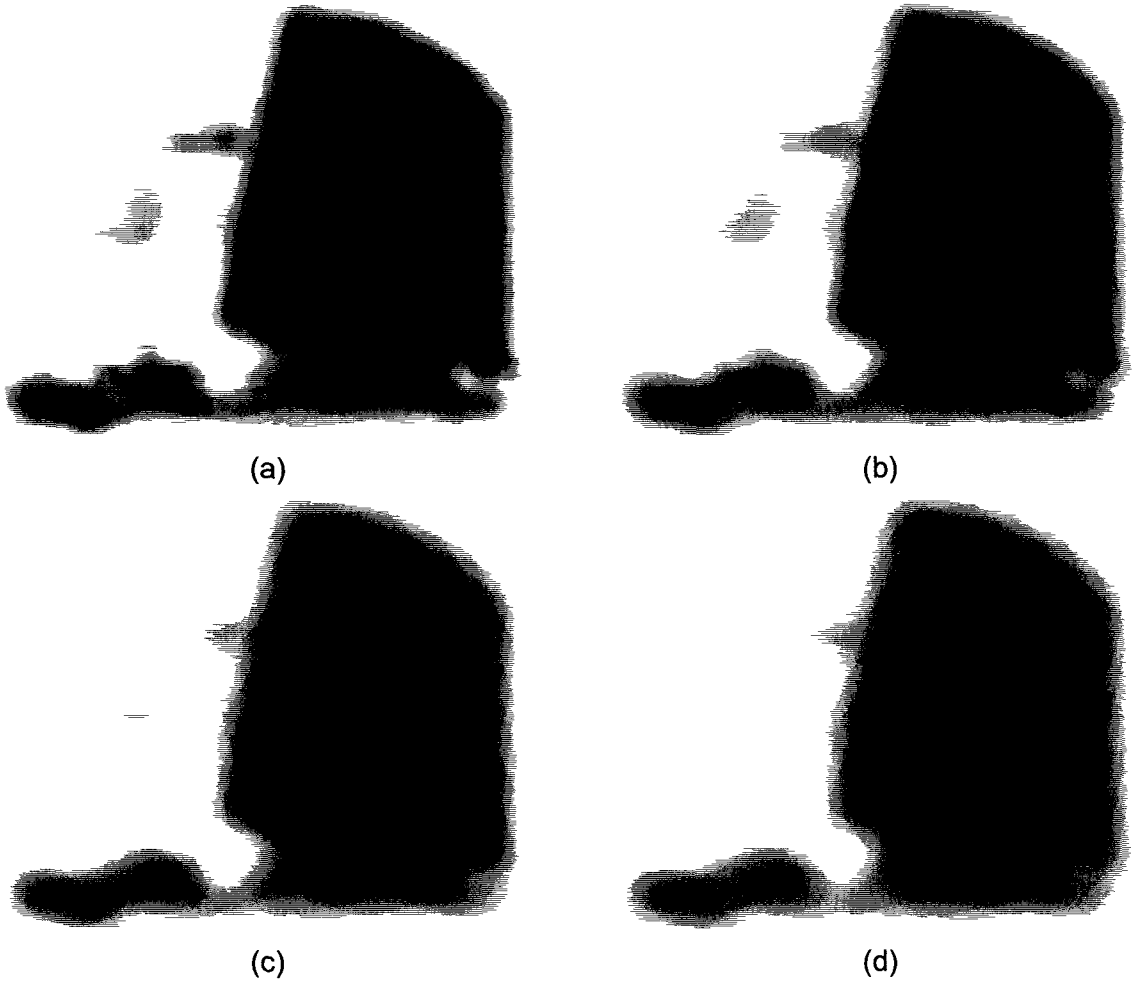


Figure 4.19. Filtered sharpness masks after (a) one, (b) two, (c) three and (d) four smoothing operations, where darker shades of gray represent higher values.

The partition masks are generated by performing a pixel-wise comparison of the sharpness masks of each focus plane. For each pixel of the image plane, the sharpness mask with the highest value is mapped to its respective partition masks. This leads to a partition mask for each focus plane that defines a region of interest where the plane is in proper focus. The second modification consists of adding a minimum threshold, when performing the mapping, in order to eliminate background noise present in low value areas of the sharpness masks. Equation 4.6 defines how the partition masks P_k are computed from the sharpness masks S_k

while respecting a minimum threshold T. The indices k and l represent the focus planes. The noise reduction is shown in Figure 4.20 using a minimum threshold of 5. Essentially, the threshold ensures that the partition masks are not defined in regions of the scene where there is no projected chessboard pattern. This results in the partition mask of Figure 4.20.b, which defines a region of interest limited to the object in focus at the centre of the image.

$$P_k(i,j)=S_k(i,j),\ \textit{if}\ S_k(i,j)>S_l(i,j)\ \text{and}\ S_k(i,j)>T,\ \textit{where}\ k\neq l$$
$$=0,\ \textit{otherwise} \tag{4.6}$$

(a) (b)

Figure 4.20. Partition masks computed (a) without and (b) with minimum threshold, where black represents the region of interest.

4.4.2 Results

The benefits of adapting to the focus planes of a scene are demonstrated by imaging the laboratory scene of Figure 4.17.a and presenting the generated 3D surface mesh. It is difficult to compare the results to the old implementation of the sensor, since the functionality of adapting to focus planes is new. The laboratory scene is particularly interesting since it contains a large depth of field; the closest surface is roughly 1.5m from the structured light sensor while the farthest surface is over 7m away. Also, many distinct objects are present, which provides higher scene complexity to ensure a proper evaluation of the focus fusion algorithm.

The first pass is performed in order to determine the focally connected regions of the different focus planes. The first step of the algorithm is to compute sharpness masks of the chessboard pattern projected onto the scene. Again, the algorithm is applied to the

background subtraction images since it is the focus of the pattern that is of interest and not the focus of the cameras. The sharpness masks for each of the six focus planes are shown in Figure 4.21. Since the masks are computed using horizontal and vertical Sobel filters, a higher intensity of black indicates a higher level of sharpness, which translates to an area of accurate focus. By comparing the different image regions between focus planes it can be seen that the side of the desk has sharp focus in Figure 4.21.c, the closest computer monitor in Figure 4.21.d, the chair in Figure 4.21.e and the rest of the scene in Figure 4.21.f.

Due to the high frequency and square-like nature of the chessboard pattern, the sharpness masks cannot be directly compared to isolate focally connected regions. This would lead to many small connected regions as there is significant noise in the sharpness masks. To address this, the masks are low pass filtered over four iterations and shown in Figure 4.22. This greatly increases the neighbourhood relevance of the sharpness masks and produces better defined focally connected regions. These new masks are much more useful for the partitioning of the image space in the next step.

By mapping the maximums in the filtered sharpness masks, the corresponding partition masks are computed and shown in Figure 4.23. These are straightforward binary masks that express where the projected pattern is in focus for each of the focus planes. For the most part, the focally connected regions are well defined, however some artifacts, in the shape of arbitrary lines, remain. The latter occur in regions where adjacent focally connected regions are separated by one or more focus planes. Although the regions are visually displeasing, they are not problematic since they do not lie in regions where the pattern is projected.

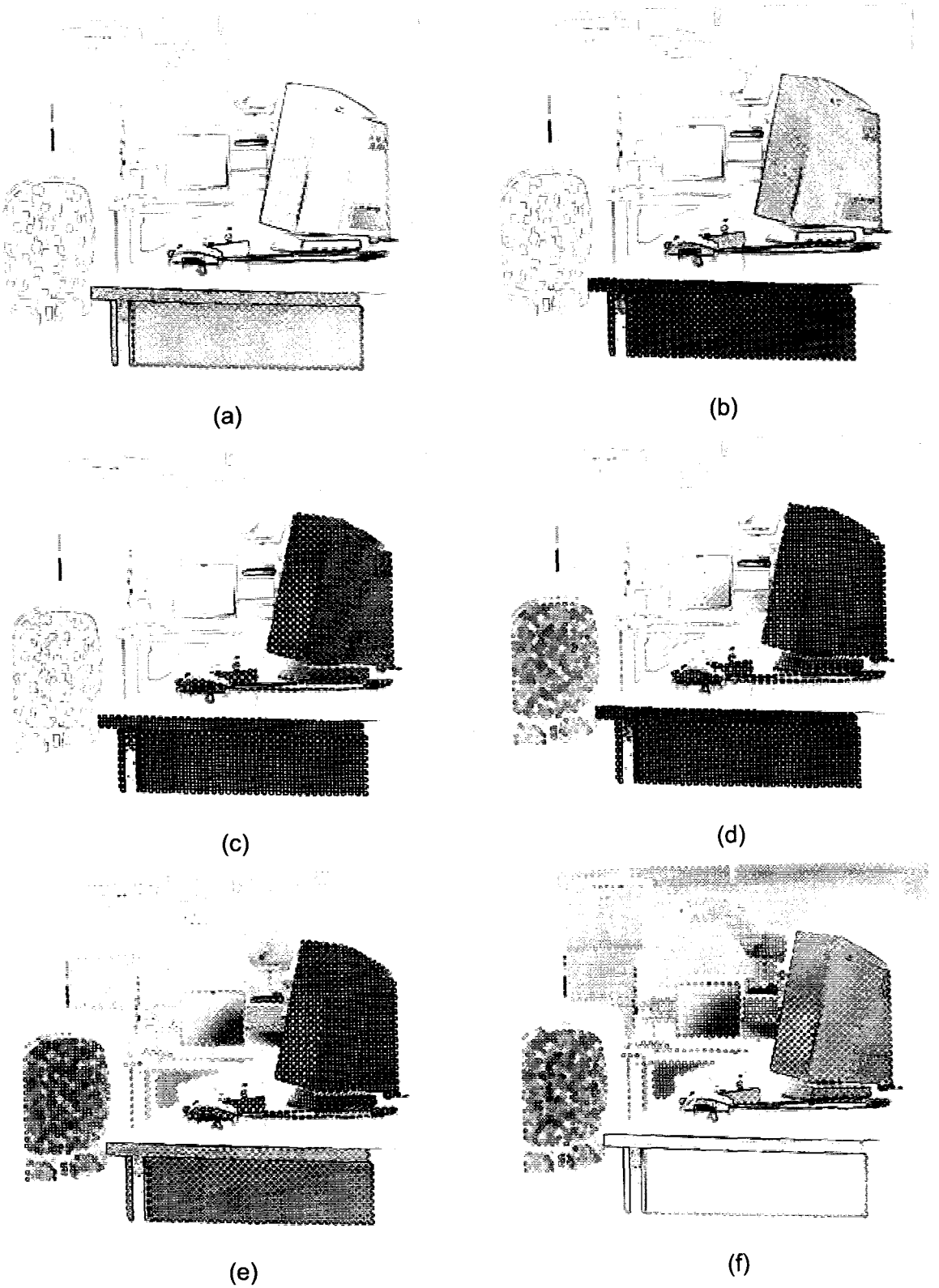


Figure 4.21. Sharpness masks for focus planes (a) 1, (b) 4, (c) 7, (d) 10, (e) 13 and (f) 16, where darker shades of gray represent higher values.

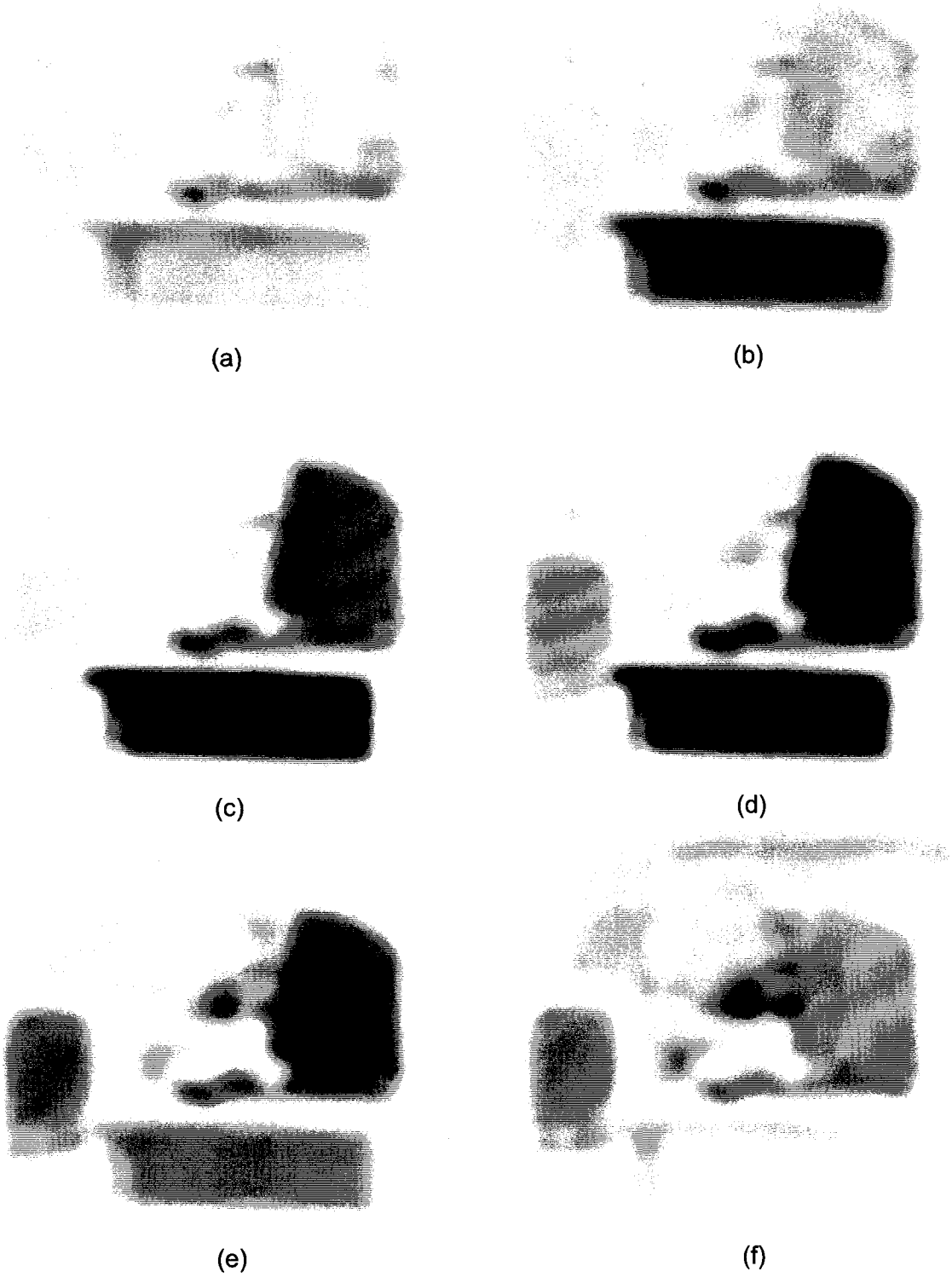


Figure 4.22. Filtered sharpness masks for focus planes (a) 1, (b) 4, (c) 7, (d) 10, (e) 13 and (f) 16, where darker shades of gray represent higher values.

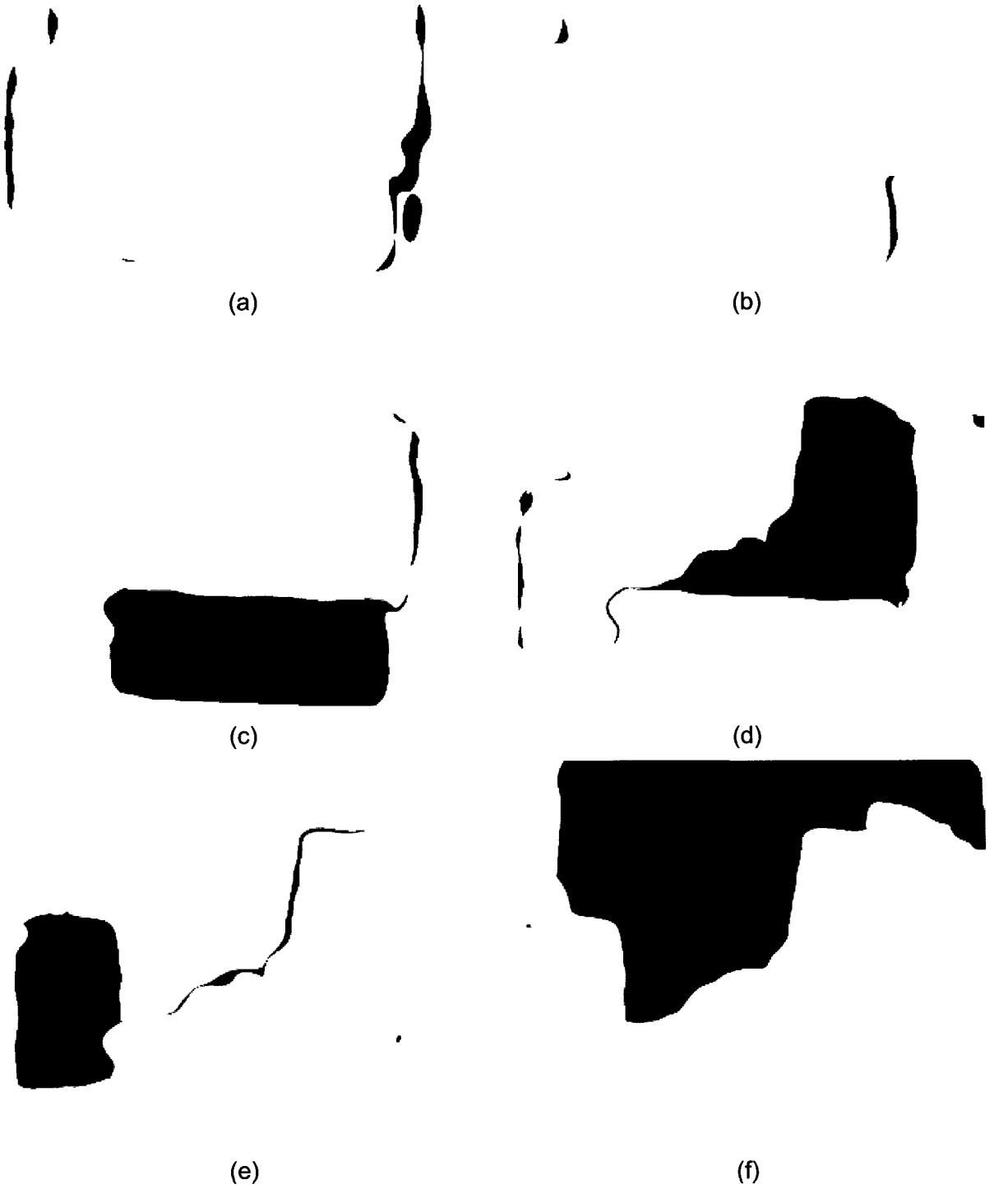


Figure 4.23. Partition masks for focus planes (a) 1, (b) 4, (c) 7, (d) 10, (e) 13 and (f) 16.

The first and fourth focus planes are very close to the structured light sensor in order to permit the acquisition of small objects in close proximity. Such objects are not present in the laboratory scene, which is shown by the empty partition masks of the first and fourth focus planes in Figures 4.23.a and 4.23.b respectively. Since both of these partition masks each cover less than 5% of the scene, they are dropped from further processing and a focus plane subset, containing planes 7, 10, 13 and 16, is considered.

The partitioning step, which includes the comparison of sharpness masks for maximum values, is performed a second time using only the focus planes of the subset. The updated partition masks are shown in Figure 4.24. This folds the regions from the discarded focus planes into those of the subset, ensuring that no region of the scene is neglected. An example of this in the laboratory scene is with the region of interest of the first focus plane, shown in Figure 4.23.a, which is folded into the seventh focus plane, shown in Figure 4.24.a. Again, the new partition masks also contain some artifacts, however this does not affect the focus fusion in any way since the artifacts normally occur in areas not covered by the projected pattern. Figure 4.24.e shows the superposition of the four partition masks in order to illustrate the coverage of the scene. All of the regions where the pattern is projected are considered when comparing with Figure 4.17.b.

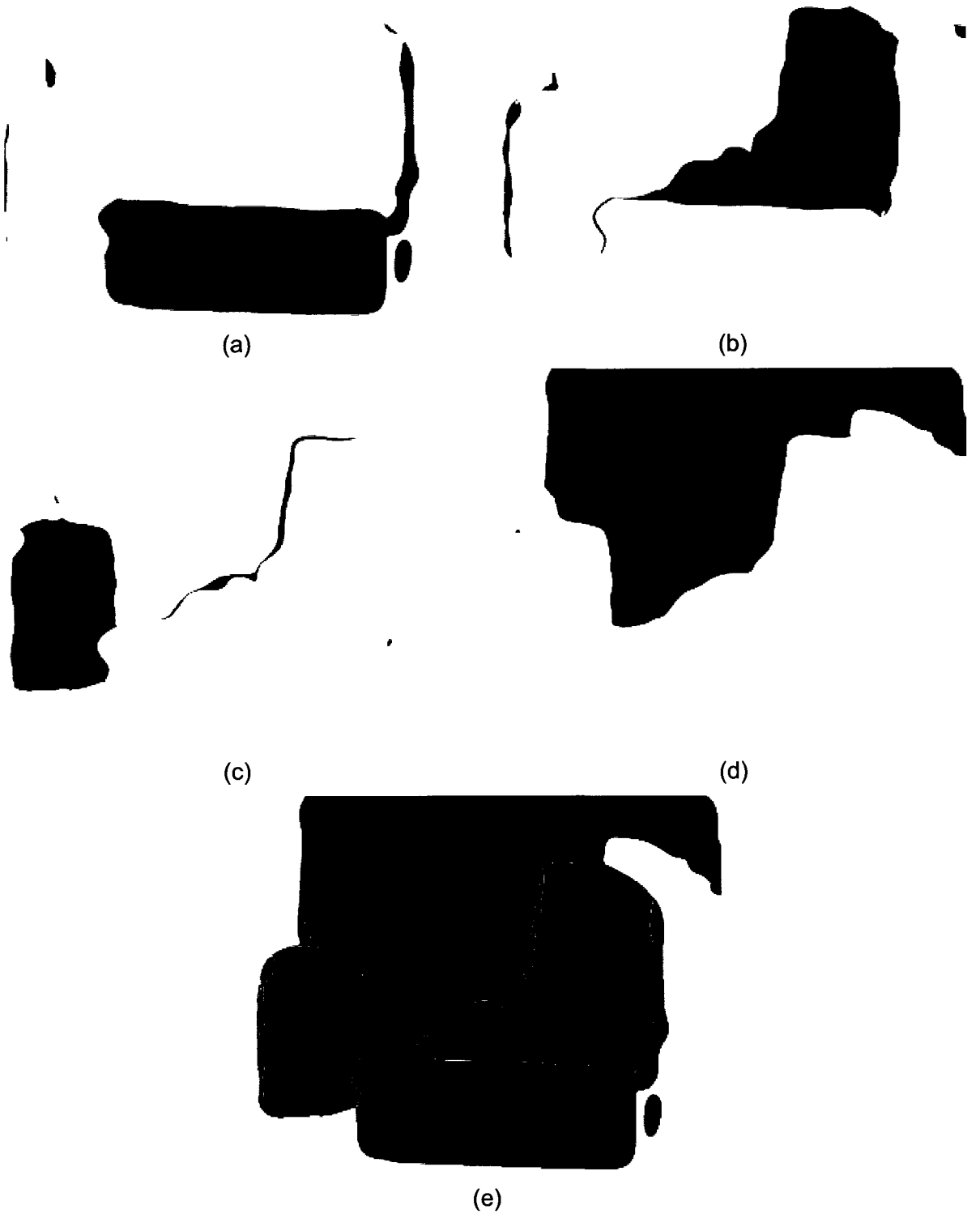


Figure 4.24. Re-computed partition masks for focus planes (a) 7, (b) 10, (c) 13 and (d) 16. (e) Superposition of the four partition masks to illustrate the coverage of the scene.

The second pass is now performed; where full structured light acquisitions are done for each of the focus planes in the regions specified by the respective partition masks. Each focus plane is treated independently with a separate pattern projection, segmentation, code detection, 3D triangulation and surface reconstruction within its region of interest as defined by the partition mask. The main motivation for the independent analysis is that the average distance between vertices is different for each focus plane. As the distance from the sensor increases, so does the average distance between vertices. By treating each focus plane separately, different parameters for the ball pivot algorithm are used to ensure a proper surface generation at the multiple depths. For example, with three pattern marches in the horizontal and vertical directions, the average distances between vertices is 4.6, 6.0, 8.8 and 18.9mm for the focus planes 7, 10, 13 and 16 respectively.

The surface meshes of each focus plane are concatenated into a global mesh of the laboratory scene, which is shown in Figure 4.25. All objects in the scene are clearly visible and the depth of field of the sensor is best demonstrated in the side view of Figure 4.25.b. It should be noted that closer objects are imaged with a higher spatial density than farther objects since the projected squares get larger as the distance from the sensor increases. Also, the accuracy of the 3D points decreases as the distance from the sensor increases. This is seen by the noisy surfaces of the boxes (5) and rear wall (7) as compared with the accurate and sharp mapping of the texture on the chair (3).

The main limitation with the focus fusion algorithm is that it does not properly handle objects that span multiple focus planes. In such cases, the surface patch that spans multiple planes will be divided into several sections by the common border between the partition masks involved.

The addition of the focus fusion algorithm to the acquisition stage considerably increases the range of the sensor. The structured light sensor is now adaptable to the depth of field of the scene, while still remaining fully autonomous during the acquisition stage. Also, due to the nature of the algorithm, the projector focus is inherently set, alleviating the operator from setting the object at a predetermined distance or adjusting the focus of the projector.

Figure 4.25. (a) Front view and (b) side view of the laboratory scene mesh showing (1) the desk, (2) the first monitor, (3) the chair, (4) the second monitor, (5) the boxes, (6) the robotic arm and (7) the rear wall.

4.5 Summary

This chapter presented three enhancements to the acquisition stage that significantly improve the sensor's range of operation. First, a new time-multiplexed acquisition mode was developed in order to project the most accurate and detectable PR pattern using white light. This was coupled with adaptive thresholding during the segmentation to achieve a robust detection of the pattern on objects of multiple colours. Second, an exposure fusion step was added, when capturing images from the cameras, such that a local exposure setting is simulated. This allows the sensor to adapt to many reflectance characteristics in the scene and removes the need to manually calibrate the exposure time. Third, a focal analysis was added to the acquisition stage in order to extend the depth of field of the structured light sensor by adapting the focus of the projector. It is now possible to image scenes with objects at multiple depths. This also eliminates the constraint of having the projector in focus prior to the acquisition. Throughout the chapter, detailed results were presented to validate the design decisions and illustrate the adaptability of the new structured light sensor.

Chapter 5 Sensor Overview

All of the improvements to the sensor, which have so far been presented and analyzed independently, are now integrated together. This chapter gives an overview of how the sensor operates as a complete system for 3D measurement and reconstruction. First, the entire acquisition and processing procedure is reviewed to summarize the high-level operation of the new sensor. Second, the operating characteristics of the structured light sensor are presented to understand how it compares to other sensors.

5.1 Acquisition and Processing Procedure

The structured light range sensor measures the scene in two stages. First, the acquisition stage programmatically controls the cameras and projector, performs the exposure and focus fusion and saves images to disk. Second, the processing stage analyzes the captured images, identifies the pseudo-random code correspondences and performs a triangulation to extract 3D points. The acquisition stage is completely overhauled from the implementation in [1] by incorporating the new time-multiplexed mode, the exposure fusion and the focus plane fusion, as described in Chapter 4. The processing stage is modified by adding the new colour segmentation, the improved code detection and the surface generation step, as described in Chapter 3. The processing stage is also adapted to work with the time-multiplexed patterns and the multiple focus planes.

5.1.1 Acquisition Algorithm Summary

Before presenting the algorithm, the region of interest of the cameras is discussed since it has not been presented so far. When developing a structured light sensor, the projected pattern is required to be fully visible within the field of view of the cameras. This is achieved by carefully positioning the LCD projector with respect to the cameras and ensuring that the latter have compatible lens settings. However, it is common that the projected pattern appears in a sub-region of the captured images. Although the optimal setup consists of the entire projected pattern appearing as large as possible within the image plane, regions with no projected pattern around the borders will always remain. To maximize the efficiency of the system, the regions of interest of each camera, which contain the projected pattern, are computed and only these regions are considered for the subsequent image processing steps.

To determine the regions of interest, a solid white mask is projected over the entire field of view of the projector. Background images, without the projected white mask, and pattern images, with the projected white mask, are acquired for both cameras. The pattern images are subtracted from the background images, a threshold is applied and the bounding boxes of the masks are computed giving the regions of interest.

The pseudo-code, presented in Table 5.1, illustrates the high-level details of the acquisition stage. First, an exposure set is computed in order to perform the exposure fusion algorithm. Second, a region of interest that defines where the projector can project is computed for each camera. Third, the first pass through the focus planes is performed to calibrate the system by determining regions that are in focus and dropping insignificant focus planes. Finally, the second pass through the subset of focus planes is performed to acquire data; the time-multiplexed pattern channels are projected separately and images are acquired using the exposure fusion algorithm.

- Perform global onboard auto exposure on both cameras and compute E_g.
- Compute E_{min}, E_{max} and determine the exposure set for the EF algorithm.
- Project a solid white mask onto the scene and acquire left and right images using the EF algorithm.
- Compute a region of interest for each camera that contains the white mask.
  - Only consider and process these regions of interest for all subsequent steps.
- Iterate through all focus planes for calibration.
  - Adjust projector focus to current focus plane.
  - Project chessboard pattern using maximum intensity and acquire images using the EF algorithm.
  - Compute sharpness and partition masks for current focus plane.
- Drop focus planes that contain a proportionally small partition mask.
- Re-compute partition masks for subset of focus planes.
- Iterate through the subset of focus planes for acquisition.
  - Adjust projector focus to current focus plane.
  - Perform structured light data acquisition in region specified by current partition mask.
    - Project time-multiplexed pattern channels at maximum intensity and acquire images using EF algorithm.
    - Save images to disk for further processing.

Table 5.1. Acquisition stage pseudo-code.

The above steps are repeated whenever an acquisition is triggered. Even if the cameras do not move with respect to the projector, the region of interest must be recalculated since the configuration of the new scene can change where the pattern is located on the image planes. Also, the weight maps for the exposure fusion algorithm as well as the masks for the focus fusion algorithm must be recomputed since they must adapt to the changing scene.

5.1.2 Processing Algorithm Summary

The high-level pseudo-code for the processing stage is presented in Table 5.2. First, each focus plane is processed independently to segment the pattern squares and recover the 3x3 codes, only in the region of interest defined by the partition masks. Second, the code correspondences are determined, the triangulation is performed and the point cloud is interpolated to generate a surface. Finally, the surfaces from all focus planes are concatenated into one final mesh.

- Iterate through the subset of focus planes.
  - Process the saved images to segment the pattern squares and perform statistical analysis.
  - Recover the 3x3 codes from the left and right images.
  - Perform code correspondence and triangulation to generate 3D range data.
  - Interpolate the point cloud to generate a surface mesh.
- Concatenate the surfaces from all focus planes into one mesh and save to disk.

Table 5.2. Processing stage pseudo-code.

5.1.3 Discussion

In the implementation of the acquisition and processing stages, which is further detailed in Appendix 2, all new algorithms presented in this work are fully configurable. The colour or time-multiplexed acquisition mode can be selected depending on the scene. The exposure fusion algorithm can be manually configured, automatically configured or turned off completely. Similarly, the focus fusion algorithm can be run automatically or turned off in order to process only one focus plane.

The acquisition procedure is not only designed to be adaptable to colour, reflectance characteristics and depth of field, but also flexible to the type of scene imaged and the desired results. For example, to perform object modelling of a single object placed close to the sensor in a controlled environment, the sensor is configured to acquire data at one focus plane and use a high spatial density via marching patterns. On the other hand, to explore an unknown environment, the sensor is configured to use all focus planes and a lower spatial density to get the general layout of the environment. It is then possible to concentrate on a focus plane of interest and scan it again using a higher spatial density to obtain more detailed information. The same flexibility applies to the acquisition modes of the sensor. If the scene is known to be uniform in colour, the colour acquisition mode can be used to improve the acquisition time. Otherwise, the time-multiplexed acquisition mode is used in

order to ensure maximum performance.

Since the operation of the structured light range sensor is divided into stages and implemented using a set of independent modules, it is easily possible to control which modules are activated for different types of acquisitions. For example, consider the application of selective scanning where a scene is first scanned using a coarse spatial density, regions of interest based on discontinuities or features are found and then a second scan is performed in the detected regions with a finer spatial density to extract more details. In this application, the scene is static and there is no need to re-compute the regions of interest of the cameras or to re-calibrate which focus planes are of interest after the first scan, provided that the sensor did not move. Also, the first scan would be performed using a coarse spatial density by only marching the patterns a small number of times. For the second and subsequent scans, the marching would be respectively increased to obtain a finer spatial density and only certain regions of the acquired images would need to be processed as defined by the areas with discontinuities or features. The current sensor framework is adaptable to this particular use-case and many others, as defined by different requirements from various applications.

5.2 Operating Characteristics

Although the new implementation significantly enhances the robustness and adaptability of the sensor, the latter still has a defined operating range. The main operating characteristics, including field of view, spatial density and execution time, are presented in this section. It should be noted that the operating ranges of the sensor can vary considerably depending on the configuration of the acquisition stage as well as the scene to be imaged.

The field of view of the range sensor is mostly constrained by the capabilities of the LCD projector and is therefore determined by the projectable area of the projector. These characteristics, which are relative to the projector's projection axis, are presented in Table 5.3. The horizontal field of view is equally divided by the projection axis, and the minimum and maximum depths of field are determined by the focal characteristics of the projector's lens. By considering the aspect ratio of the projector, the surface area covered at the minimum depth of field is 43x32cm and 289x217cm at the maximum depth of field.

Field of View (°)	Minimum Depth of Field (cm)	Maximum Depth of Field (cm)
30	80	540+

Table 5.3. Field of view and depth of field characteristics.

The spatial density of the resulting point cloud is directly controlled by adjusting the number of horizontal and vertical shifts of the pseudo-random pattern during the acquisition. The minimum spatial density is obtained using only one projection and not marching the pattern, while the maximum spatial density is the result of shifting the pattern 18 times horizontally and vertically. This is the maximum as the pattern squares have a dimension of 9x9 pixels with 9 pixels of empty space between them. The spatial densities are presented in Table 5.4 for several different depths of field, since the latter affects the size of the reflected squares and in turn the spatial density. It should be noted that at maximum spatial density, the noise present in the 3D point triangulation is roughly on the same order as the actual spatial density, which determines the accuracy of the depth estimation.

Depth (cm)	Minimum Spatial Density (mm)	Maximum Spatial Density (mm)
100	10.17	0.56
200	19.53	1.08
300	29.35	1.63
400	39.58	2.20
500	49.60	2.76

Table 5.4. Point cloud spatial density at specific depths of field.

The execution time required for the sensor to acquire and process data depends heavily on the configuration of the acquisition stage. Tables 5.5, 5.6 and 5.7 contain the execution time for different configurations to show how the new improvements affect the sensor and what the trade-offs are between the quality of the results versus the speed of the acquisition. The metrics for the acquisition stage include the time to calibrate the exposures and region of interest as well as the time to project and acquire the pattern. The metrics for the processing stage include the image processing time to detect codes, the time to perform 3D triangulation and the time to generate a mesh via the ball pivot algorithm. It should be noted that the new processing stage is used for all configurations, and that it has a similar execution time when compared with the old implementation.

Table 5.5 presents execution times for the previous version of the sensor that uses the colour acquisition mode and no exposure fusion. As expected, the time to project and

acquire patterns, and the image processing to detect codes are proportional to the number of shifts selected. Also, the bottleneck of the processing stage is the code detection step.

Shifts	Acquisition Stage (sec)			Processing Stage (sec)			
	Calibrate	Pattern	Total	Codes	Triangle	Mesh	Total
1x1	4.928	1.251	6.471	15.740	1.438	0.563	18.101
2x2	4.912	3.251	8.440	61.680	2.125	3.971	69.260
3x3	4.866	6.597	11.567	138.974	3.907	16.663	162.763

Table 5.5. Execution time using the old colour acquisition mode and no exposure fusion algorithm.

Table 5.6 presents execution times for the new sensor configured to use the time-multiplexed acquisition mode and no exposure fusion. In this case, the pattern acquisition time is essentially doubled as 2 extra images are acquired for each shift. The processing time remains the same since the same amount of data is ultimately acquired.

Shifts	Acquisition Stage (sec)			Processing Stage (sec)			
	Calibrate	Pattern	Total	Codes	Triangle	Mesh	Total
1x1	4.883	2.250	7.394	16.537	1.454	0.453	18.820
2x2	4.836	7.300	12.192	66.697	2.032	4.158	74.230
3x3	4.992	15.693	20.804	147.461	3.673	15.991	170.156

Table 5.6. Execution time using the new time-multiplexed acquisition mode and no exposure fusion algorithm.

Table 5.7 presents execution times for the new sensor configured to use all new enhancements, including the time-multiplexed acquisition mode and the exposure fusion. The main difference is that the pattern acquisition time is multiplied by a factor of 10 to account for the 5 extra images that must be acquired and processed for each shift. Again, the processing stage has a similar execution time.

Shifts	Acquisition Stage (sec)			Processing Stage (sec)			
	Calibrate	Pattern	Total	Codes	Triangle	Mesh	Total
1x1	19.358	24.087	43.594	16.475	1.391	0.594	18.897
2x2	19.336	77.997	97.474	65.962	2.001	3.689	73.090
3x3	19.477	167.718	187.273	144.522	3.377	15.927	167.062

Table 5.7. Execution time using the new time-multiplexed acquisition mode and the exposure fusion algorithm.

It should be noted that with the addition of the focus fusion algorithm, the acquisition and processing stage execution times are essentially multiplied by the number of valid focus planes processed, plus some overhead for the focal analysis. Again, there is a trade-off between the speed and the robustness of the acquisition that must be considered. Also, much of the execution times presented in this section are highly dependant on the objects in the scene. They provide a general estimate for the worst case scenario, since a flat surface where the entire pattern is visible was imaged. With a general scene, the execution time is normally slightly reduced since some areas of the pattern are not visible, due to occlusions or the absence of surfaces, which reduces the amount of subsequent processing.

Nevertheless, the execution times for the improved sensor can be quite long, especially when acquiring data at a high spatial density and adapting to the focus planes. To address this drawback, the notion of selective scanning is introduced and further detailed in Chapter 6. With the integration of this sensor into an autonomous mobile exploration platform, the idea is not to model the entire environment at high spatial density, but rather to progressively refine a coarse scan of the environment. For example, a quick low density scan of the environment could be analyzed to identify interesting or important areas, which would then be scanned using a higher spatial density or an exposure fusion with finer steps. This highlights the importance of having a flexible and adaptable structured light sensor that can operate in many different configurations.

Finally, to compare the sensor with other similar acquisition systems, a configuration that is commonly used consists of placing the objects of interest at 1.5m from the sensor, using 3x3 horizontal and vertical shifts, and all of the improvements presented in this thesis. This results in the acquisition of a surface area of 80x60cm, at a spatial density of 5mm for a total acquisition and processing time of roughly 6 minutes.

5.3 Summary

This chapter gave a high-level overview of the new and improved 3D acquisition sensor and presented how all of the improved components are integrated to produce a complete system. First, a detailed overview of the revised acquisition and processing stages was given to demonstrate the flexibility with which the sensor can adapt to the scene. Second, the relevant characteristics of the sensor were presented detailing its operating range.

Chapter 6 Sensor Mobility and Data Fusion

One of the primary motivations for the research presented in this thesis is its use in mobile robotic exploration applications for the autonomous mapping of environments and the generation of accurate 3D texture representations. This particular application is explained in detail and experimental results are presented and discussed. The idea is to mount the range sensor, including the projector, the cameras, a laptop computer and power supply, onto a mobile platform. This would allow the platform to map an unknown environment by moving about the environment, capturing 3D point clouds with enough overlap and then fusing the data to generate a 3D representation of its surroundings. What follows is an initial proof of concept where a mobile platform is manually operated to move the sensor about a scene and range data acquisitions are also manually triggered from several viewpoints. However, it should be noted that the range sensor's operation is fully automated due to the enhancements presented in Chapter 4.

6.1 System Requirements

A scene reconstruction system usually consists of three main components, the acquisition, the registration and the data fusion parts. First, the acquisition stage, powered by the structured light range sensor, takes care of acquiring multiple 3D maps of an arbitrary scene. Secondly, the registration component estimates the transformation between the multiple views. Finally, the data fusion procedure merges the respective data sets and estimates a surface mesh over the entire set of objects in the scene.

Much literature has dealt with the simpler object reconstruction problem using many structured light active vision technologies. However, many constraints on the objects being imaged and on the surrounding environment still remain. Most current structured light systems require that the registration component be highly coupled to the acquisition component. For example, the classical turntable approach [40], [41], [42] requires calibration and synchronization between the acquisition and rotation devices. In order to create a flexible reconstruction system that is operational with minimal constraints on its motion, a reduction of the coupling between the acquisition and the registration is necessary. As a result, the position of the acquisition component with respect to the scene being imaged can remain flexible and multiple views can be collected by moving the 3D range sensor without impairing the registration system.

In order to build a scene reconstruction system with low coupling between components, the technologies of the sub-systems must be appropriately selected. The structured light range sensor, described in the previous sections, is used as the acquisition system since it is capable of adapting to different scenes and measure 3D surfaces with and without features from multiple points of view. The registration algorithm developed by Curtis *et al.* [21] is used to compute the translation and rotation transformations between the multiple viewpoints without any initial estimation. Finally, a point set surface (PSS) algorithm [28] is used to merge and simplify the overlapping surfaces meshes, producing a final 3D surface model.

6.2 Data Acquisition, Registration and Fusion

The proposed framework for the scene reconstruction system, combining the acquisition, registration and fusion modules, is illustrated in Figure 6.1. The interconnections between the modules as well as the flow of data are shown.

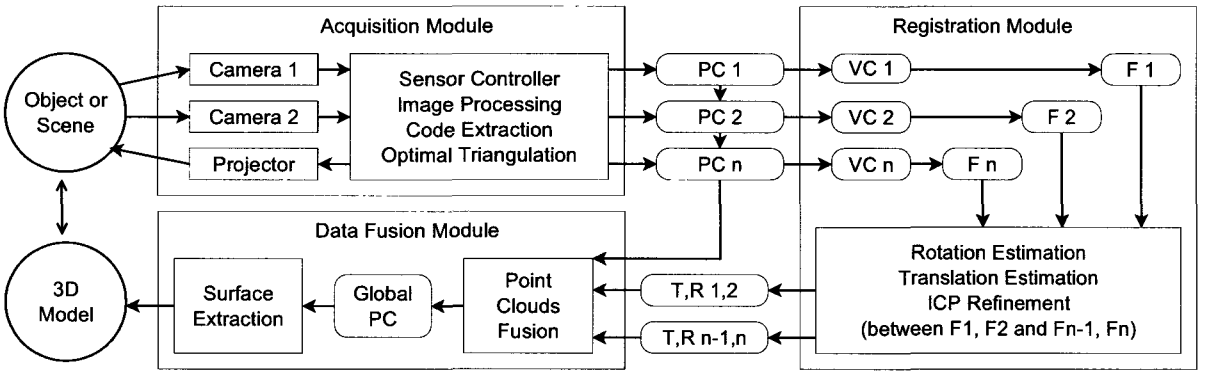


Figure 6.1. Interconnection of acquisition, registration and data fusion modules.

The acquisition module is used to capture many 3D point clouds of the scene. For conventional object reconstruction, it is possible to move and rotate the object under analysis, which makes the process less cumbersome. However, for general scene reconstruction, such as in the case of robotic exploration, the entire range sensor is translated and rotated to capture multiple viewpoints. The only constraint is that 3D point clouds contain significant overlap between the successive views in order to assure proper registration. The acquisition and processing stages of the structured light range sensor are carried out exactly as explained in the previous sections and are therefore not detailed here.

According to the technique of Curtis *et al.* [21], the point clouds PC_i are transferred to the

registration module where they are first mapped to voxel clouds VC_i and then converted to the frequency domain F_i via the Fourier transform. First, the amplitude component of the point clouds is compared in order to extract the rotation. Second, the phase component is compared to estimate the translation between the data sets. Since the point clouds are captured in succession, the frequency domain registration is only applied to successive point clouds. This establishes a chain of transformations, including translation T_i and rotation R_i, that link the successive views and describe the path of the mobile robotic platform as it moved through the environment. To achieve a proper registration, an overlap of roughly 75% between data sets is required, thereby ensuring that dominant components of the data are present in both data sets. The final step is to refine the transformations using the iterative closest point (ICP) algorithm [19].

The data fusion module takes the outputs of both the acquisition and registration modules and merges the data sets obtained from respective viewpoints into a 3D model of the scene. Since the registration component determines the transformations between successive views, each point cloud is first expressed with respect to a common reference frame, located at the first viewpoint. All of the respective transformations between the latter and all other views are computed and applied to their respective point clouds. Finally, all of the point clouds are merged by concatenating the list of points to produce a single large point cloud modelling the scene with respect to the first location and orientation of the 3D range sensor.

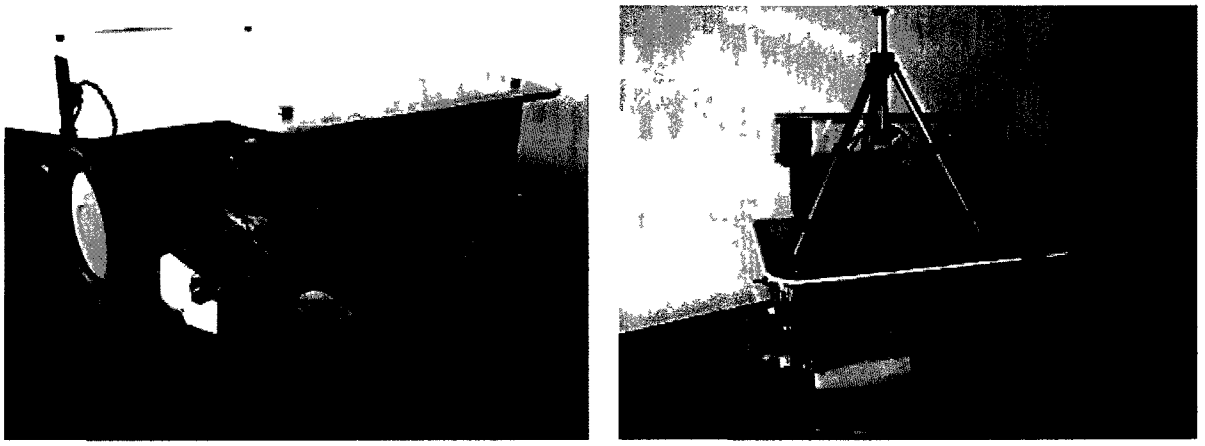
The current model is the result of many disjoint data sets, each with significant overlap, which are all merged into a single mesh. This leads to a significant oversampling and many areas with unnecessary high densities of 3D points. To simplify the model, locally redundant points are removed and a surface mesh that approximates the noisy data is generated. First, the global mesh is decimated using a voxelization over the bounding box of the point cloud and keeping only one point in each voxel as a representative entity for that voxel. The voxel size is selected as the average distance between points such that only redundant points are dropped. Not only does this remove redundant points, it also greatly reduces the size of the point cloud, which in turn speeds up the processing of all subsequent steps. Secondly, a surface is extracted from the unorganized point cloud using PSS techniques [28]. Points are projected onto a moving least squares (MLS) surface [29] that approximates the point cloud locally. This allows the extraction of a surface in the presence of noise or error due to the registration procedure. A coarse mesh is extracted using the marching

cubes algorithm [24] and the latter is further refined using another more accurate projection back onto the MLS surface. The result is a global surface mesh that approximates and interpolates all of the individual point clouds from the different points of view. All mesh operations, including the ICP and PSS algorithms, are performed using the implementations within Meshlab [43].

6.3 Mobile Platform Prototype

Since the long term goal of this research is to develop a mobile acquisition device, capable of imaging an entire 3D environment, a prototype mobile platform was designed and built. The mobile platform, shown in Figure 6.2.a, is built off an existing robotic base that houses three wheels, a drive train and two DC motors. The onboard electronics are all custom designed and include an operation mode and a charging mode. The platform can be operated via a radio control (RC) controller for manual movement or via the onboard microcontroller for automated navigation. It should be noted that presently, the navigation algorithms are not yet implemented on the microcontroller and the only practical way of using the platform is with the RC controller. Finally, the platform is designed with a large payload bay, consisting of a sheet of aluminum, which can be used to carry different sensor equipment for various experiments and research projects that require a medium size mobile platform. There is also room for a large battery and power inverter to supply power to the components on the payload bay, although this functionality is not currently implemented.

In the context of the present research, the mobile platform is equipped with the structured light range sensor, consisting of an LCD projector, two cameras and a laptop, as shown in Figure 6.2.b. The process of acquiring 3D scans from an environment is mostly manual and requires a human operator. The platform is positioned accordingly and a 3D acquisition is triggered. Once complete, the platform is moved to the next point of view using the RC controller and another acquisition is triggered. The process is repeated in order to capture the necessary views of the environment. It should be noted that the mobile platform does not have to be moved precisely between viewpoints; the only requirement is that adequate overlap is provided between views. Once the data has been collected, the process of registering the successive point clouds via the registration algorithm and generating the point set surface model is performed with the assistance of the human operator.



(a) (b)

Figure 6.2. Mobile platform with (a) empty payload bay and (b) fitted with the structured light acquisition system.

6.4 Results

The scene reconstruction system is tested by imaging two environments, in order to determine the feasibility and some of the problem areas of such a system. The first environment is a robotic workcell, shown in Figure 6.3.a, of approximately $1.5m^3$ that is contained within a laboratory with fluorescent lighting. The scene is imaged from 12 separate viewpoints, starting perpendicular to the computer monitor and moving left along an arc by about 25 to 50cm between each view. The platform is rotated to keep the scene within the field of view of the 3D sensor. In this test, the structured light pattern is shifted 3 times in the horizontal and vertical directions, resulting in point clouds of approximately 4000 points with a spatial density of 5mm for each view. Figures 6.3.b to 6.3.d show three independent surface meshes for the first, sixth and twelfth viewpoints respectively, to give an idea of the change in perspective from the views.

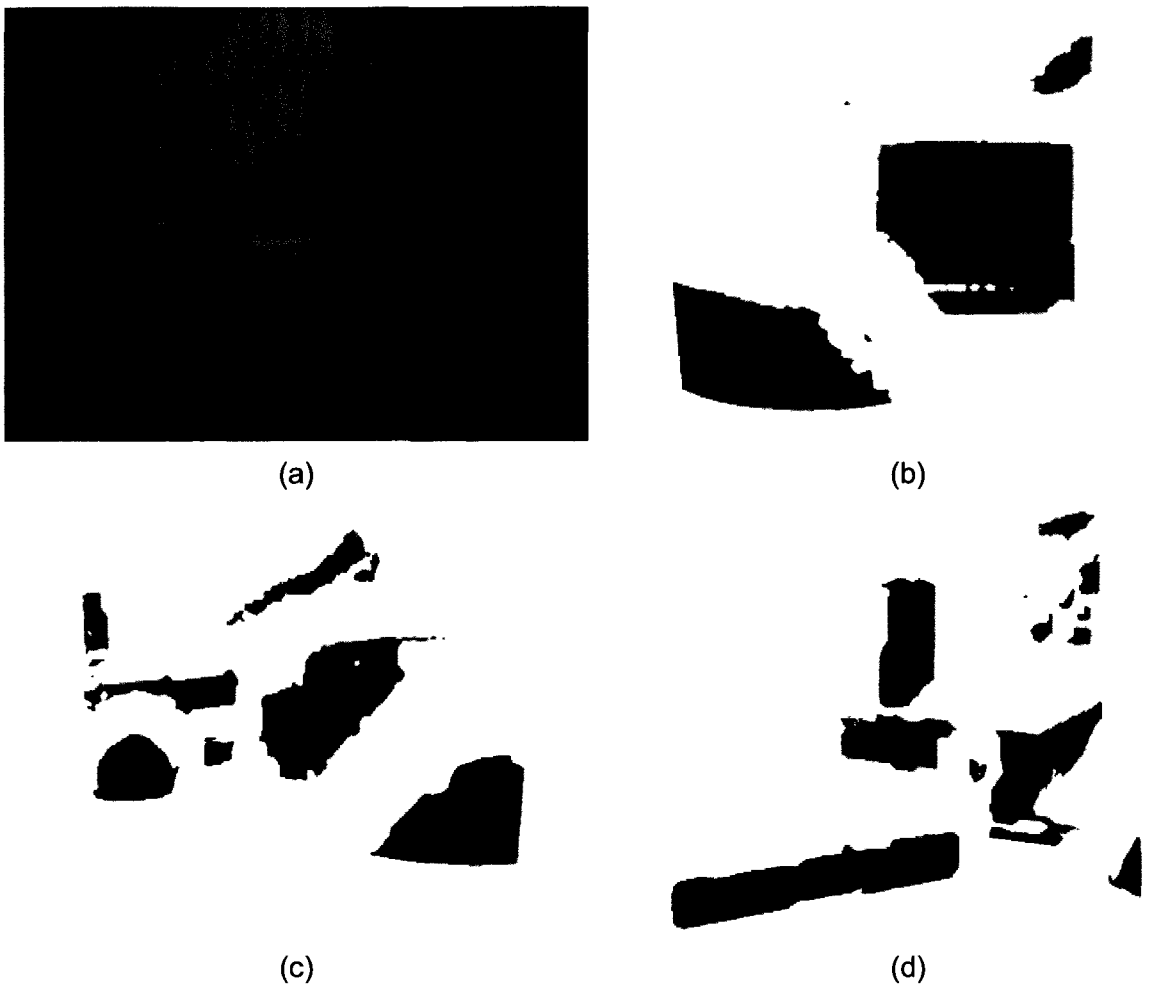
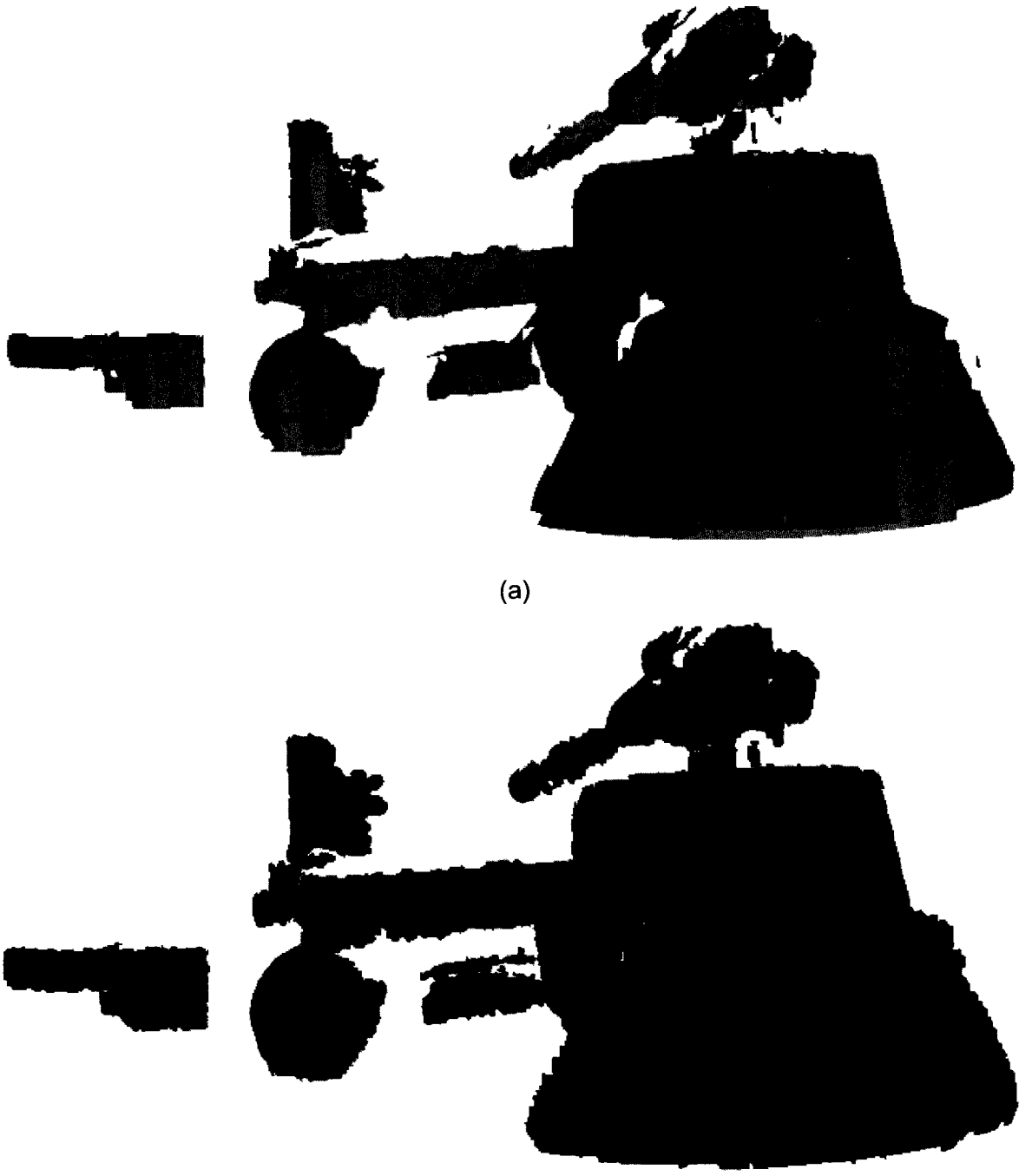


Figure 6.3. (a) Original image of robotic workcell scene and individual surface meshes from (b) first, (c) sixth and (d) twelfth points of view.

A pairwise registration between the 12 point clouds is performed, using the registration algorithm explained in Section 6.2, with a 256x256x256 voxelization of the datasets. A refinement of the translation and rotation between successive viewpoints is performed using the ICP algorithm. Next, the surface meshes are all expressed with respect to a common reference frame, the first view, and concatenated together. Figure 6.4.a shows the result of the registration step and displays each mesh with a unique colour to better see the global alignment. Interleaved meshes, such as on the glass screen of the monitor, are great results whereas the registration of the seat back is less optimal since the meshes are not interleaved. However, the resulting registration is still relatively accurate for the datasets.



(a)

(b)

Figure 6.4. (a) Concatenation of all individual meshes represented by unique colours. (b) Point set surface model of the scene with mapped colour information.

To fuse the datasets and remove the redundant vertices from the multiple overlapping meshes, a global surface mesh is extracted from the concatenated meshes, as defined in Section 6.2. First, the mesh is decimated using a uniform voxel dimension of 5mm to reduce

vertex redundancy. Second, the PSS algorithm is applied using an MLS filter scale of 7mm to obtain a new global mesh interpolating all of the concatenated meshes. Finally, colour is mapped from the individual meshes to the global mesh by averaging the colour component of the 5 closest neighbouring points in order to minimize the colour variation from the different points of view. The final surface mesh is rendered from the approximate point of view of Figure 6.3.a and shown in Figure 6.4.b.

The model of the robotic workcell scene demonstrates that the system is capable of imaging a real environment with multiple objects of varying colour and unconstrained placement. The scene is particularly interesting since the track of the manipulator robot is occluded in the first views, but correctly appears behind the monitor, as the sensor is moved along the arc. This is illustrated by a second point of view of the scene along with another rendering of the model shown in Figure 6.5. Also, the resulting model consists of a set of surface patches, which can be seen from the sensor's point of view, and merged to create a partial map of the environment. The horizontal surfaces of the table and bench do not appear in the reconstructed model because the mobile sensor only images horizontally, as is typical with a mobile platform exploring an unknown world. It should be noted that the vertical wall is not detected since it was occluded in many of the viewpoints as the sensor was positioned close to the robotic workcell objects.

(a) (b)

Figure 6.5. (a) A second point of view of the robotic workcell scene along with (b) the same surface model rendered from the corresponding point of view.

The second environment is a section of an atrium with an approximate width of 9m and depth of 3m. This environment, shown in Figure 6.6.a, is interesting since it is outside of the laboratory and contains a combination of artificial lighting and natural light coming in from

the left hand side. The scene contains a mixture of featureless areas, such as the cement walls, and highly textured regions, such as the poster. Also, some significant depth variation is provided by the columns extending away from the rear wall. The scene is imaged from 11 separate viewpoints, starting from the recycling centre on the left, pivoting toward the bench and then moving parallel to the rear wall toward the right until the poster. The exact locations are shown in Figure 6.6.d and labelled from 1 to 11. In this test, the pseudo-random pattern is shifted 2 times in the horizontal and vertical directions to generate point clouds of approximately 7000-8000 points with a spatial density of 20-25mm for each view.

Again, a pairwise registration between the 11 point clouds is performed, using the frequency domain registration algorithm summarized in Section 6.2, with a 128x128x128 voxelization. Since the algorithm is based in the frequency domain, it is sensitive to the dominant peaks of the datasets. For example, if two successive datasets have similar dominant peaks in their frequency signature, a proper registration will be achieved. This constraint is normally satisfied when performing inward imaging of a single object where the sensor is rotated around the subject. However, the constraint is almost never satisfied when performing outward imaging of a general environment, like the atrium scene, where the sensor is pivoted about itself or moved along a straight path. To address this issue, a manual segmentation is performed prior to the registration. Each pair of successive point clouds is segmented such that only the common overlapping points are considered for registration. This ensures that the dominant peaks of the datasets are similar and a proper registration is achieved. Once more, the ICP algorithm is executed to refine the transformation estimates, obtained with the frequency-domain registration, between the successive viewpoints. Finally, a global registration of all the point clouds is run, again using the ICP algorithm, to refine the entire dataset as a whole.

The registered point clouds are fused in the same way as in the robotic workcell scene example. The global mesh is decimated using a voxel dimension of 15mm, the PSS algorithm is applied using an MLS filter scale of 7mm and the colours are mapped by averaging the closest 5 neighbouring points. The final surface mesh is rendered from different viewpoints and shown in Figures 6.6.b to 6.6.d. It should be noted that the initial range data is acquired at a low spatial density and interpolated using the PSS algorithm at a much higher spatial density. This explains why the 3D data of the final mesh is of high spatial density but the colour data is not.

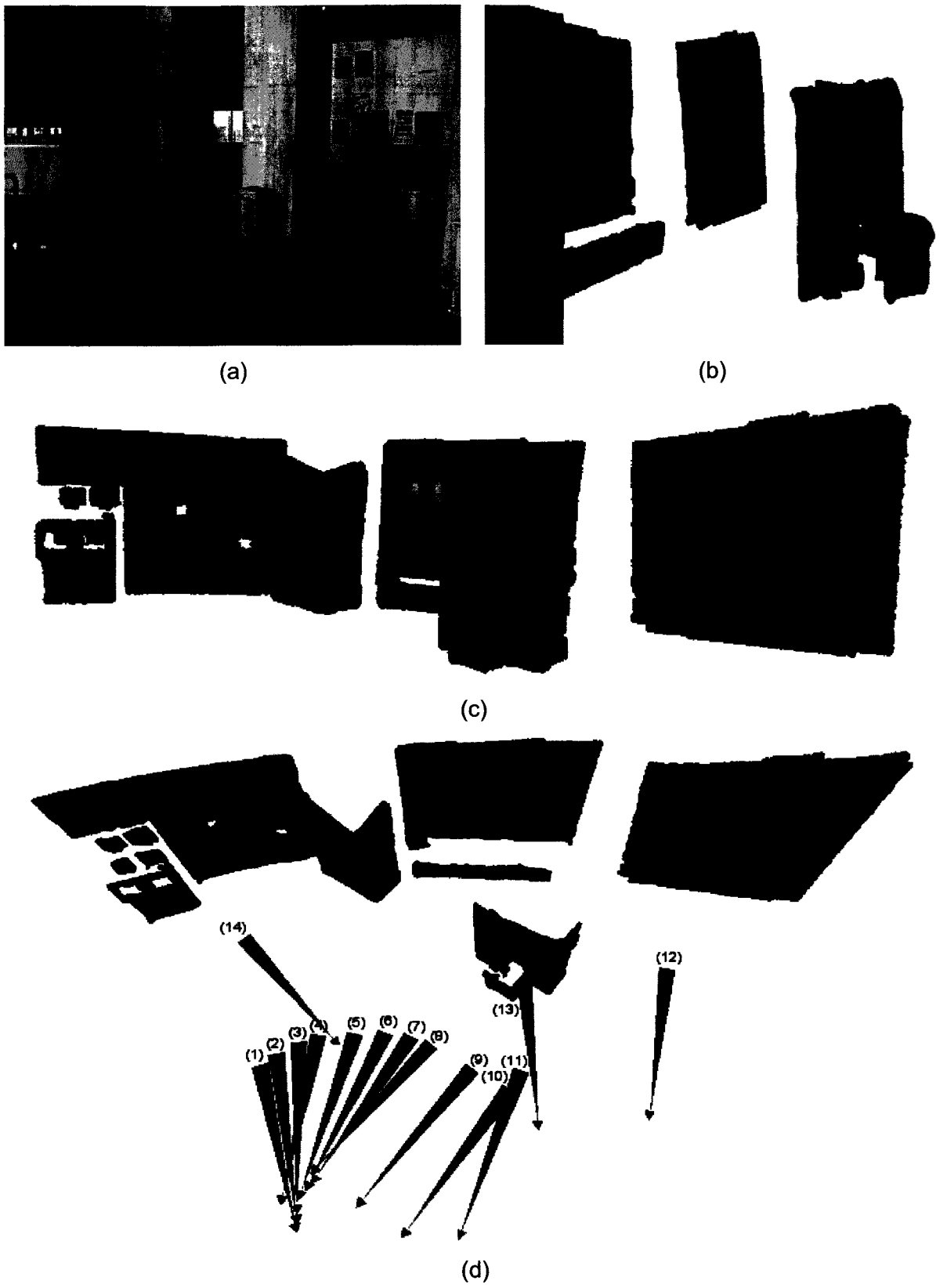


Figure 6.6. (a) Original image of the atrium scene and corresponding surface model from (b) a zoomed in view, (c) a lateral view and (d) a top view with the locations of the sensor.

To demonstrate how the structured light sensor can adapt to the required spatial density, a selective scanning example is presented. Three regions of interest in the atrium scene are manually selected and scanned at a higher spatial density. The locations are shown in Figure 6.6.d and labelled 12 to 14. This is accomplished by shifting the structured light pattern 6 times in the horizontal and vertical directions, and obtaining point clouds of roughly 65000 points per view at a spatial density of 3-6mm. The poster, shown in Figure 6.7.a, has fine detail and high contrast in areas that contain text. There is also a combination of highly reflective aluminum on which less reflective paper is affixed. Figure 6.7.b shows the scan of the garbage can where the black plastic garbage bag is particular difficult to image since it is dark in colour, yet highly reflective when creased. Since this scene is much closer to the sensor, the projected codes are smaller and the resulting mesh is a lot smoother than other datasets. Finally, the recycling centre is shown in Figure 6.7.c and illustrates the difficulty of imaging small dark objects such as the phone hanging on the wall. In this example, the phone is simply too textured and dark for the sensor to detect. Positioning the sensor much closer to the phone would allow for the size of the pattern squares to match the resolution of the phone and produce better results. Also, the upper part of the recycling centre is not captured since the angle of the surface is too large with respect to the sensor's principle axis and a large amount of natural light is also present in that direction. A more perpendicular placement of the sensor would capture that region at the cost of loosing the right-hand side of the recycling centre.

Since the high spatial density scans are acquired in much the same way as the low spatial density ones, they can be registered to the existing surface model following the same procedure. Once registered, the colour information of the scans is mapped to the global mesh, which is shown in Figure 6.7.d. This demonstrates that the structured light sensor is capable of using a low spatial density acquisition to get the general model of the scene and then perform additional high spatial density acquisitions of areas that are deemed interesting.

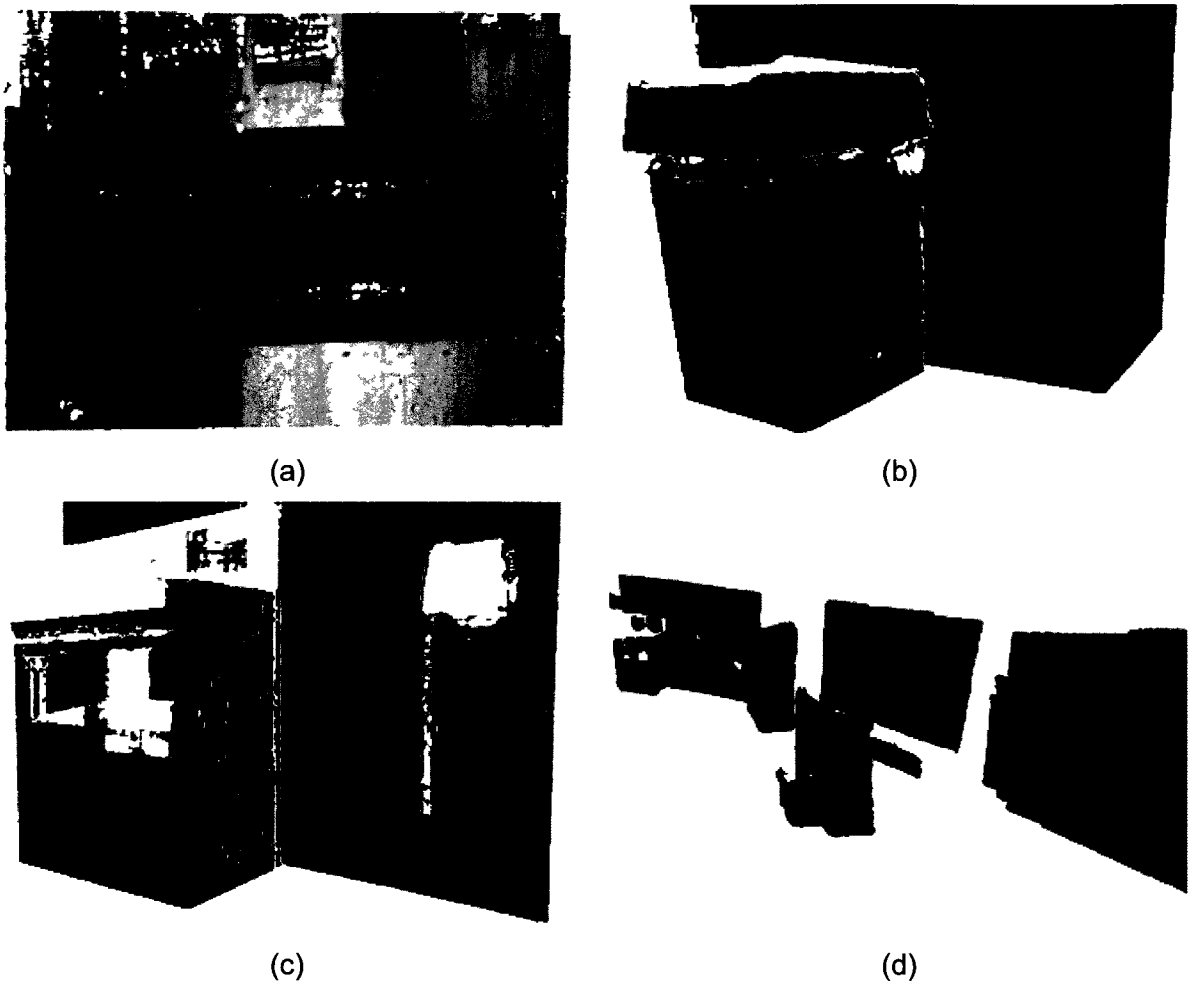


Figure 6.7. High spatial density surface mesh of (a) a poster, (b) a garbage can and (c) a recycling centre. (d) Same surface model of the atrium scene with high spatial density scans mapped to the lower density surface model.

Most of the results in this section are obtained by performing the operations manually. This includes segmenting the meshes prior to registration, applying the frequency domain and ICP registration algorithms to successive points of view, running the PSS algorithms on the registered mesh and determining where to move the robot to ensure proper overlap. However, the operation of the structured light sensor is completely automatic since it is capable of adapting to colour, reflectance characteristics, depth of field and spatial density. Since this present research is focused on the sensor itself, the results of this section validate the improvements and enhancements made to the structured light sensor as it can now be used as a reliable acquisition device for a selective scanning mobile platform system.

6.5 Summary

This chapter demonstrated the capabilities of the improved structured light sensor when integrated into a mobile robotic exploration system for autonomous mapping. First, the system requirements for a typical mapping system were presented, with emphasis on the need for low coupling between components. Second, the data acquisition, registration and fusion components were briefly explained to give an overview of how the system generates accurate 3D models of an environment. Third, a mobile platform prototype was presented and the procedure to acquire and generate models of an environment was detailed, explaining which steps are fully automated and which ones currently need to be performed by a human operator. Finally, surface models for two real world scenes were presented along with a discussion of some remaining limitations in the system.

Chapter 7 Conclusion

This research sought to advance structured light technology using modern image processing and computer vision techniques. This last chapter presents some concluding thoughts and insight on the work conducted in this thesis. First, a summary of the many enhancements and algorithms developed for the acquisition and processing stages is presented. Second, key contributions are highlighted along with their implications to the field of computer vision. Finally, possible directions for future work are discussed.

7.1 Summary

The research presented in this thesis consisted of many separate enhancements and improvement applied to a previously developed structured light range sensor. The key algorithms of the processing stage were improved and the entire acquisition stage was redesigned such that it is more flexible and adaptable.

The first thing addressed in the processing stage was the extrinsic calibration of the cameras. The translation and rotation between the cameras is computed using epipolar geometry via a decomposition of the fundamental matrix. This technique yields more precise calibration results and a more solid mathematical foundation to integrate the multiple views of the chessboard target. Also, the scale factor of the calibration is reliably and automatically computed, using a least squares approach on the calibration data, in order to obtain a metric reconstruction. Second, improvements to the PR pattern segmentation algorithm were facilitated by only processing the most important pixels. This was done by thresholding the value channel of the HSV image to obtain a mask that eliminates the background without using hard-coded colour thresholds. Also, a new histogram analysis technique was employed to detect the dominant colours of the image without the need of post-processing the returned peaks. Third, a new code detection algorithm was developed to better identify clusters of 3x3 squares that are skewed as a result of angled surfaces. Vector-based computations are preformed to identify opposing neighbours of a central square of interest as well as selecting the optimal pairs of neighbours via an empty distance measurement. Finally, a surface reconstruction step was added to interpolate the resulting 3D point cloud into a triangular mesh in order to better visualize the quality of the acquired data. The ball pivot algorithm was selected since it inherently considers the proximity of vertices when constructing triangular faces. This is useful when evaluating the sensor's output quality as

the generated surfaces accurately represent the underlying spatial density and organization of data points.

To achieve the goal of making the structured light range sensor adaptable to many different scenes, the acquisition stage was significantly extended by integrating three key algorithms. First, a new acquisition mode was introduced, which eliminates the use of colour to encode the PR pattern. Instead, the pattern is projected using a time-multiplexing of pseudo-colour channels that are constructed using the colour white at full intensity. The result is the ability to recover codes from surfaces with multiple colours as well as surfaces with more complex textures. This acquisition mode is also complemented with a more robust adaptive thresholding since the projected pattern is now monochrome. Second, an exposure fusion algorithm was added and is invoked each time images are captured from the cameras. The process involves acquiring multiple images while varying the exposure time and fusing the images together such that all pixels are properly exposed. By using this algorithm, the sensor is able to detect the projected pattern from surfaces with varying reflective characteristics as well as under many different lighting conditions. Finally, a focus analysis technique is integrated into the acquisition procedure to adjust the projector's focus since the latter is the main limitation of the sensor's depth of field. A first pass is performed, while varying the projector focus, to determine relevant focus planes of the workspace. A second pass is then executed to perform a full structured light acquisition on the relevant focus planes and fuse the resulting 3D point clouds.

Although beneficial on their own, the above improvements truly enhance the performance and robustness of the sensor when used together. Their integration produces a sensor that is much more automated and fully capable of adapting to numerous scenes without constraint.

7.2 Contributions

Most of the literature on structured light range sensors always focuses on solutions developed for constrained environments and for specific object types and properties. The recurring applications are the measurement of small objects and the modelling of the human face or museum artifacts, all of which are performed in controlled environments with proper lighting and specific object placement. The overall contribution of this research is to move away from these constraints and develop a structured light sensor capable of imaging unconstrained scenes. The contributions in this thesis all work toward increasing the

robustness and adaptability of the sensor to its environment, while reducing the amount of parameters to configure prior to an acquisition. This is useful when using a range sensor as an autonomous input device for a robotic mobile platform as opposed to a 3D measurement tool used by a human operator.

The main contributions are in the enhancements developed for the acquisition stage of the structured light sensor. Much of the difficulty in implementing a structured light sensor lies in the acquisition of quality images that contain a clearly defined pattern. The processing to detect the pattern and compute a 3D point cloud is mostly handled by widely known image processing and triangulation algorithms respectively. For these reasons, the acquisition stage was significantly extended and the four main contributions resulting from this work are as follows.

- A time-multiplexed acquisition mode was proposed to improve the capture of colour pseudo-random patterns projected onto multicoloured objects and scenes with significant colour variability.
- The application of an exposure fusion algorithm to the image capture component of a structured light sensor was proposed to enhance the robustness of detecting the projected pattern reflected from many objects with different reflectance characteristics.
- The novel adaptation of a focus fusion algorithm to a new focal analysis process was proposed to automatically detect the focal planes of a scene and significantly increase a structured light sensor's depth of field.
- The above contributions made it possible to adapt classical structured light technology so that it can be used on an automated robotic platform as an input device to model and map the surrounding environment.

These algorithms all work in parallel and their novel combination allows the refined range sensor to measure scenes regardless of object colour, lighting conditions, reflectance characteristics and depth of field. Although all of the enhancements presented in this thesis were demonstrated on an existing sensor, they can be applied to structured light sensors in general.

Secondary contributions are in the incremental improvements to the processing stage of the structured light sensor. Since most of the processing stage algorithms are well known, much of the work consisted in refining the existing implementation which resulted in the following

three secondary contributions.

- The extrinsic calibration was re-implemented in a more mathematically accurate way to increase precision and a reliable method to compute the scale factor was presented in order to achieve automatic metric reconstruction.
- An improved colour pattern segmentation and pseudo-random code detection were proposed to improve the detection of a structured light pattern projected using different combinations of colours onto a scene with many angled surfaces.
- A surface interpolation procedure, which uses an improved meshing algorithm and automatic parameter computation, was designed specifically for range data in order to visualize and analyze the results of a structured light sensor.

These improvements contributed to increased precision and adaptability when processing the structured light patterns projected onto a scene. In addition, more edge cases are handled, which further increases the robustness of the detection algorithms and allows for a better automated operation.

7.3 Future Work

Chapter 6 presented a proof of concept for a robotic exploration platform capable of mapping an unknown environment using the improved structured light range sensor. This initial work has already been published in [44]. The continuation of this research is to fully implement the autonomous mobile platform and further develop its algorithms such that automatic modelling of unknown environments becomes possible.

In order to accomplish this, the physical setup of the mobile platform needs to be refined and a high-level controller needs to be developed. The latter would manage the acquisition and registration modules as well as decide where the next point of view should be. Finally, the output 3D range data would need to be analyzed in order for the autonomous system to map the environment and be capable of locating itself within it.

Regarding the structured light sensor, future work includes making it even more autonomous. This involves detecting which acquisition mode to use depending on the scene. For example, if the colour of the target scene is uniform, the coloured structured light pattern can be used. On the other hand, the more robust time-multiplexed acquisition can be used as a fall-back for more complex scenes. Another optimization would be to define

some metrics that determine how many exposures and focus planes to use, for the respective exposure fusion and focus fusion algorithms, depending on the scene. This would help reduce the execution time of the acquisition stage when possible. Finally, an algorithm to determine the desired spatial density of scans would be beneficial. As the 3D structure of the scene becomes more complex, the number of marching patterns can be increased to achieve a greater spatial density of 3D points. This could also be used in conjunction with a selective scanning framework which would first scan the environment at a lower spatial density and then augment interesting regions of the model with higher density data.

References

[1] D. Desjardins, "Structured Lighting Stereoscopy with Marching Pseudo-Random Patterns," M.A.Sc. thesis, University of Ottawa, Ottawa, ON, Canada, 2008.

[2] D. Desjardins and P. Payeur, "Dense Stereo Range Sensing with Marching Pseudo-Random Patterns", in Proceedings of the *4th Canadian Conference on Computer and Robot Vision*, Montreal, QC, 2007, pp. 216-226.

[3] P. Payeur and D. Desjardins, "Structured Light Stereoscopic Imaging with Dynamic Pseudo-random Patterns," in Proceedings of the *6th International Conference on Image Analysis and Recognition*, Halifax, NS, 2009, pp. 687-696.

[4] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, Nov. 2000.

[5] R. A. Morano, C. Ozturk, R. Conn, S. Dubin, S. Zietz and J. Nissanov, "Structured Light Using Pseudorandom Codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 322-327, Mar. 1998.

[6] J. Salvi, J. Batlle and E. Mouaddib, "A Robust-Coded Pattern Projection for Dynamic 3D Scene Measurement," *Pattern Recognition Letters*, vol. 19, no. 11, pp. 1055-1065, Sep. 1998.

[7] R. Jain, R. Kasturi and B. G. Schunck, *Machine vision*, New York, NY: McGraw-Hill, 1995.

[8] R. Hartley and A. Zisserman, *Multiple View Geometry*, Cambridge, UK: Cambridge University Press, 2003.

[9] A. A. Goshtasby, "Fusion of Multi-Exposure Images," *Image and Video Computing*, vol. 23, no. 6, pp. 611-618, Jun. 2005.

[10] Z. Xiang, "Color Image Quantization by Minimizing the Maximum Intercluster Distance," *ACM Transactions on Graphics*, vol. 16, no. 3, pp. 260-276, Jul. 1997.

[11] I. Zafar, E. A. Edirisinghe and H. E. Bez, "Multi-Exposure & Multi-Focus Image Fusion in Transform Domain," in Proceedings of the *IET International Conference on Visual Information Engineering*, Bangalore, 2006, pp. 606-611.

[12] P. J. Burt and R. J. Kolczynski, "Enhanced Image Capture Through Fusion," in Proceedings of the *4th International Conference on Computer Vision*, Berlin, 1993, pp. 173-182.

[13] P. E. Debevec and J. Malik, "Recovering High Dynamic Range Radiance Maps from Photographs," in SIGGRAPH 1997, Proceedings of the *24th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, 1997, pp. 369-378.

[14] T. Mertens, J. Kautz and F. Van Reeth, "Exposure Fusion," in Proceedings of the *15th Pacific Conference on Computer Graphics and Applications*, Maui, HI, 2007, pp. 382-390.

[15] I. De and B. Chanda, "A Simple and Efficient Algorithm for Multifocus Image Fusion using Morphological Wavelets," *Signal Processing*, vol. 86, no. 5, pp. 924-936, May 2006.

[16] H. Hariharan, A. Koschan and M. Abidi, "An Adaptive Focal Connectivity Algorithm for Multifocus Fusion," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, Knoxville, TN, 2007, pp. 1-6.

[17] H. Hariharan, A. Koschan and M. Abidi, "Multifocus Image Fusion by Establishing Focal Connectivity," in Proceedings of the *IEEE International Conference on Image Processing*, San Antonio, TX, 2007, pp. III-321-III-324.

[18] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Upper Saddle River, NJ: Prentice Hall, 1998.

[19] P. J. Besl and H. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, Feb. 1992.

[20] P. Curtis and P. Payeur, "A Frequency-Domain Approach to Registration Estimation in 3-D Space," in Proceedings of the *IEEE Instrumentation and Measurement Technology Conference*, Sorrento, Italy, 2006, pp. 293-298.

[21] P. Curtis and P. Payeur, "A Frequency Domain Approach to Registration Estimation in Three-Dimensional Space," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 1, pp. 110-120, Jan. 2008.

[22] R. Fabio, "From Point Cloud to Surface: The Modeling and Visualization Problem," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Science*, vol. XXXIV-5/W10, Feb. 2003.

[23] J. -D. Boissonnat, O. Devillers, S. Pion, M. Teillaud and M. Yvinec, "Triangulation in CGAL," *Computational Geometry*, vol. 22, no. 1-3, pp. 5-19, May 2002.

[24] W. E. Lorensen, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics*, vol. 21, no. 4, pp. 163-169, Jul. 1987.

[25] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349–359, Oct.-Dec. 1999.

[26] P. J. Frey and P. L. George, *Mesh Generation: Application to Finite Elements*, Oxford, UK: Hermes Science, 2000.

[27] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle, "Surface Reconstruction from Unorganized Points," *Computer Graphics*, vol. 26, no. 2, pp. 71-78, Jul. 1992.

[28] M. Alexa, J. Behr, D. Coher-Or, S. Fleishman, D. Levin and C. T. Silva, "Computing and Rendering Point Set Surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 1, pp. 3-15, Jan.-Mar. 2003.

[29] D. Levin, "Mesh-Independent Surface Interpolation," in *Geometric Modeling for Scientific Visualization*, G. Brunett, B. Hamann, K. Mueller and L. Linsen, Eds. New York, NY: Springer, 2003, pp. 37-50.

[30] G. Guennebaud and M. Gross, "Algebraic Point Set Surfaces," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 23, Jul. 2007.

[31] D. Tsai and Y. Chen, "A fast histogram-clustering approach for multi-level thresholding," *Pattern Recognition Letters*, vol. 13, no. 4, pp. 245-252, Apr. 1992.

[32] S. Zhang and S. Yau, "Three-dimensional shape measurement using a structured light system with dual cameras," *Optical Engineering*, vol. 47, no. 1, pp. 013604, Jan. 2008.

[33] D. G. Aliaga and Y. Xu, "Photogeometric Structured Light: A Self-Calibrating and Multi-Viewpoint Framework for Accurate 3D Modeling," In Proceedings from the *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008, pp.1-8.

[34] C. Je, S. W. Lee and R. Park, "High-Contrast Color-Stripe Pattern for Rapid Structured-Light Range Imaging," in Proceedings of the *8th European Conference on Computer Vision*, Prague, Czech Republic, 2004, pp. 95–107.

[35] H. Kawasaki, R. Furukawa, R. Sagawa and Y. Yagi, "Dynamic scene shape reconstruction using a single structured light pattern," in Proceedings from the *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008, pp.1-8.

[36] P. Fong, "Sensing, Acquisition, and Interactive Playback of Data-Based Models for Elastic Deformable Objects," *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 630-655, May 2009.

[37] D. Caspi, N. Kiryati and J. Shamir, "Range Imaging with Adaptive Color Structured Light," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 470-480, May 1998.

[38] T. P. Koninckx, P. Peers, P. Dutré and L. Van Gool, "Scene-Adapted Structured Light," in Proceedings of the *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, 2005, pp. 611-618.

[39] D. Skocaj and A. Leonardis, "Range Image Acquisition of Objects with Non-Uniform Albedo using Structured Light Range Sensor," in Proceedings of *15th International Conference on Pattern Recognition*, Barcelona, Spain, 2000, pp. 778-781.

[40] W. -K. Leow, Z. Huang, Y. Zhang and R. Setiono, "Rapid 3D Model Acquisition from Images of Small Objects," in Proceedings of *Geometric Modeling and Processing*, Hong Kong, China, 2000, pp. 33-41.

[41] H. -Y. Lin and M. Subbarao, "Vision System for Fast 3-D Model Reconstruction," *Optical Engineering*, vol. 43, no. 7, pp. 1651-1664, Jul. 2004.

[42] S. -Y. Park and Murali Subbarao, "A Multiview 3D Modeling System Based on Stereo Vision Techniques," *Machine Vision and Applications*, vol. 16, no. 3, pp. 148-156, May 2005.

[43] Visual Computing Lab – ISTI – CNR, *Meshlab*. [Online]. Available: http://meshlab.sourceforge.net

[44] A. Boyer, P. Curtis and P. Payeur, "3D Modeling from Multiple Views with Integrated Registration and Data Fusion," in Proceedings of the *2009 Canadian Conference on Computer and Robot Vision*, Kelowna, BC, 2009, pp. 252-259.

[45] SP Controls, *Electrohome EPS1024 Driver*. [Online]. Available: http://www.spcontrols.com/drivers.php

[46] P. Bourke, *PLY – Polygon File Format*. [Online]. Available: http://local.wasp.uwa.edu.au/~pbourke/dataformats/ply

Appendix 1 – Implementation Details

The hardware, software libraries and programming tools that were used to implement the applications, developed for this thesis, are presented in this appendix. The stereo vision setup is composed of two Lumenera LU135C-IO digital cameras capable of acquiring 1392x1040 pixel colour images. The cameras are connected to the computer via USB and have a fully configurable and programmable interface. The projector consists of an Electrohome EPS1024 with a native resolution of 1024x768 pixels. It is connected to the computer via VGA and serial port, and has a limited programmable interface.

The LuCam software library, distributed with the cameras, is used to interface with the cameras and control them programmatically. This allows for the setting and querying of all camera settings as well as the remote acquisition of images. Since the projector does not have an official software library to programmatically control it, an asynchronous serial port communication library was developed. The latter makes use of the codes [45] for the Electrohome projector and allows a basic control via the emulation of button presses on the projector. The *OpenCV software library is widely used in computer vision research as it* implements many commonly used algorithms. It is used for all of the image processing, matrix manipulation and data storage requirements of the applications. Finally, the VCG software library provides triangular mesh data structures as well as many mesh related algorithm implementations. It is used to handle all meshes, perform the ball pivot algorithm, read and write meshes to the disk using the PLY format as well as provide an OpenGL wrapper to facilitate the implementation of a custom mesh viewer.

All of the applications developed for this thesis are implemented using the C++ programming language along with the Qt application and user interface framework. The Qt framework provides many useful programming components and all the tools necessary to create user interfaces. Since Qt includes cross-platform support, the applications developed can be deployed on Windows, Linux and Mac platforms. It should be noted however, that the camera acquisition only works on Windows since the LuCam software library is only supported on that platform.

Appendix 2 – Applications Developed

Over the course of this research, several applications were developed in order to validate the methods proposed in this thesis. Two separate applications are used to perform the intrinsic and extrinsic calibration of the cameras and a third application is used run the structured light acquisition and process the collected data to generate 3D models. This appendix gives an overview of the applications as well as a brief explanation of how to use them.

The *Calibrator* application, shown in Figure A2.1, is used to independently calibrate the intrinsic parameters of both cameras. First, a black and white chessboard is printed and then affixed to a flat surface, a piece of corrugated cardboard in this case. The *chessboard settings* are adjusted to match the characteristics of the printed chessboard, including the dimensions of the boxes and their horizontal and vertical arrangement. Second, the *number images* parameter is set to specify how many images to acquire for the calibration. In most cases, 10 images are more than sufficient.

Third, the camera to calibrate is selected from the *camera selection* drop down menu and the *start preview* button is clicked to start streaming video from the camera. This aids the calibration procedure as feedback is given to the operator in order to keep the target within the field of view. Fourth, the calibration procedure is started by clicking the *calibrate* button. While the application acquires images and detects the chessboard pattern, the operator must wave the calibration target in front of the camera in order to obtain multiple viewpoints. To effectively model the radial distortion of the lens, it is important to cover as much of the image plane as possible with the chessboard pattern, therefore the chessboard should be as close as possible to the edges of the image. Also, audio cues are played back during the calibration procedure to instruct the operator when images are being taken and if the chessboard pattern was properly detected.

Finally, once all of the images are acquired, the application performs the intrinsic calibration and stores the result in *output/intrinsic.cal*. This file contains the intrinsic and distortion matrices for the camera. Once the first camera is calibrated, the above procedure is repeated for the second camera of the stereo pair. The calibration files are then renamed to *left.cal* and *right.cal*.

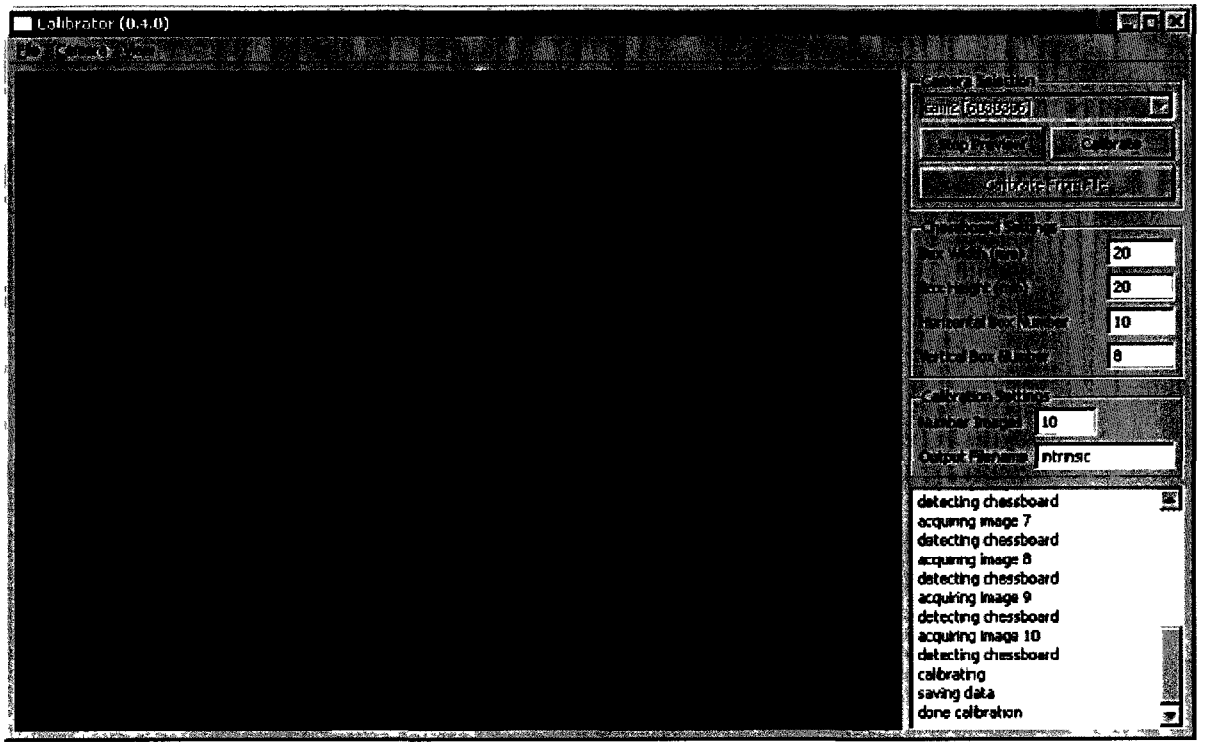


Figure A2.1. Calibrator application used to perform intrinsic calibration.

The *Calibrator2* application, shown in Figure A2.2, is used to calibrate the extrinsic parameters of the camera pair. First, the *left.cal* and *right.cal* calibration files, obtained in the previous step, are copied to the *input/* directory of the *Calibrator2* application. Second, the *chessboard settings* are set accordingly as well as the *number images* parameter. In most cases, 10 images are sufficient, but 30 images are recommended in order to obtain the best precision of the extrinsic parameters.

Third, the left and right cameras are selected from the *camera selection* drop down menus and the *start preview* button is clicked to start streaming video from the cameras. Fourth, the calibration procedure is started by clicking the *calibrate* button. While the application acquires images, detects the chessboard pattern and matches the chessboard between both images, the operator must again wave the calibration target in front of the cameras. To effectively estimate the translation and rotation between the cameras, it is important that the chessboard pattern be fully visible in both images and cover the entire volume of the workspace as it is waved. Again, audio cues are played back during the calibration procedure to guide the operator.

Finally, once all images are acquired, the application performs the extrinsic calibration, estimates the pixel error and computes the scale factor. The results are stored in

output/extrinsic.cal, which contains the fundamental matrix, the essential matrix, the extrinsic matrix, the average backprojection error, the maximum backprojection error and the scale factor.

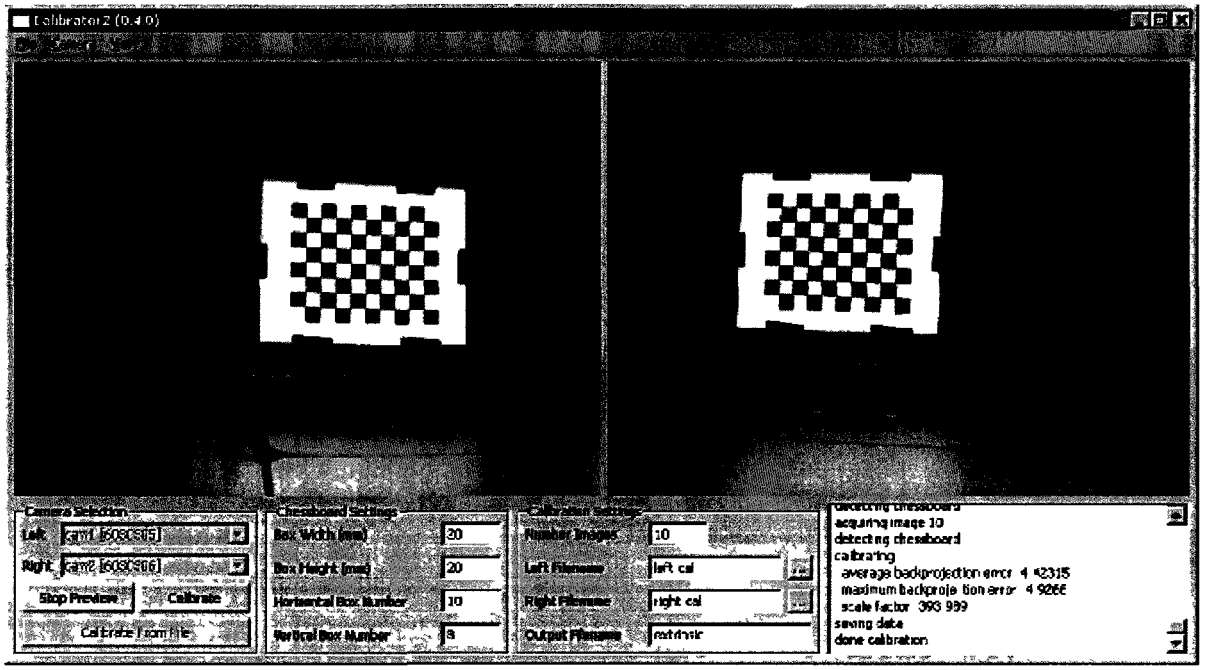


Figure A2.2. Calibrator2 application used to perform extrinsic calibration.

The *Sensor* application, shown in Figure A2.3, is used to perform the structured light acquisition and process the captured data to generate 3D models. Its operation is fully automatic, however, some parameters can be adjusted for a finer control of how the sensor should adapt to the environment. First, the *intrinsic_left.cal, intrinsic_right.cal* and *extrinsic.cal* calibration files are copied to the *input/* directory of the *Sensor* application. Alternatively, the *process settings* section of the *settings* dialog (*File* > *Settings*) can be used to specify the calibration file names.

Second, the structured light pattern window is configured such that it is shown on the proper display adaptor. If the computer is configured with a single display adaptor, which sends the same output to the monitor and the projector, this step can be skipped. On the other hand, if the computer is configured with a dual display adaptor, which sends different parts of the desktop to the monitor and the projector, the following configuration must be performed. The structured light pattern window is shown (*View* > *Preview Pattern*), moved to the appropriate display adaptor, maximized and hidden (*View* > *Hide Pattern*). This assures that when the *Sensor* application needs to display the structured light pattern, it appears on the proper

display adaptor and gets projected onto the scene.

Third, the left and right cameras are selected from the *camera selection* drop down menus and the *start preview* button is clicked to start streaming video from the cameras. If the fusion of multiple focus planes is desired, a connection to the projector must be established. The *serial port* is selected from the drop down menu and the *connect projector* button is clicked to open the connection. For proper operation, the focus of the projector must be set to the nearest setting prior to the opening of the connection. If the focus fusion is not desired, the focus of the projector must be set manually, such that the projected pattern is in focus on the objects of interest.

Fourth, the *acquire settings* are configured for the desired acquisition. The acquisition mode is selected by specifying the *mode* parameter; *time* indicates a time-multiplexed acquisition with three pseudo-colour patterns and *color* indicates an acquisition with a pattern encoded using three colours. If the colour mode is selected, the *pattern colors* to use can be specified. More colour combinations can be added to the *input/colors.col* file. The most important parameter is *pattern offset*. This controls how many times the structured light pattern is shifted or marched horizontally and vertically, which allows for the increase or decrease in spatial density of the acquired point cloud. Optimal offsets with an even distribution of shifts are denoted with a star. The *exposure* parameters respectively specify the number of images to acquire for one exposure fusion operation and the maximum exposure time that determines the upper limit of the exposure range. The exposure fusion algorithm is disabled by setting the number of images to 1 and the upper limit of the exposure range is computed automatically when the maximum exposure is set to 0. The *focus planes* parameter specifies how many focal planes to consider when performing the focus fusion operation. The focus fusion algorithm is disabled by setting the number of planes to 1. The acquisition stage is started by clicking the *acquire* button. This calibrates the exposure time of the cameras, determines the projectable area of the scene, projects the pseudo-random structured light pattern and acquires image data via the exposure and focus fusion algorithms.

Finally, the *process settings* can be configured for finer control, however, they rarely need to be modified from the default values. The *ball pivot dimension* parameter specifies the size of the voxel space used to divide the point cloud working volume in order to compute the minimum distance between points. The *ball pivot radius ratio* parameter specifies the maximum size of the ball radius with respect to the minimum distance between points. The

processing stage is started by clicking the *process* button. This analyzes the acquired images for unique codes, performs the triangulation to produce a 3D point cloud and generates a surface via the ball pivot algorithm. The resulting surface mesh is displayed in the main 3D scene graph and stored in *output/sensor.ply* using the PLY Polygon File Format [46]. The scene graph allows the surface mesh to be viewed from any viewpoint by panning, rotating and zooming as well as rendered (*View* > *Draw Mode*) using only the acquired 3D points, the wireframe generated by triangulation or the interpolated smooth surface.

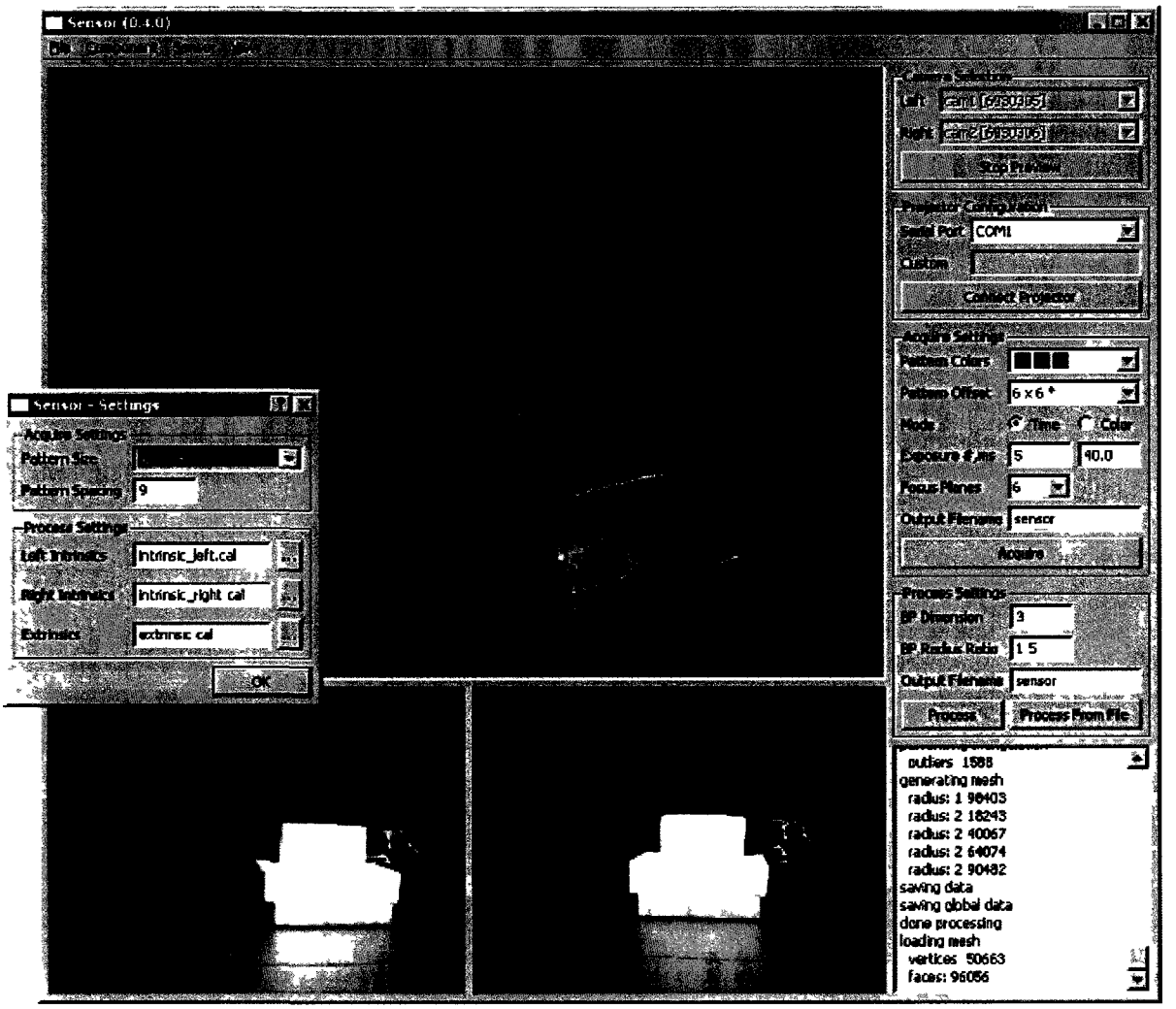


Figure A2.3. Sensor application used to perform structured light acquisition and generate 3D models.