NOTE TO USERS

This reproduction is the best copy available.

UMI®

Université d'Ottawa · University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DE ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Philippe LAVOIE

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

Ph.D. (Electrical Engineering)

GRADE - DEGREE

School of Information, Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

A NURBS Based 3D Reconstruction Using Colour-coded Strutured Light

D. Ionescu

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

CO-DIRECTEUR DE LA THÈSE - THESIS CO-SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

D. Capson

É. Dubois

R. Goubran

E. Petriu

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

A NURBS Based 3D Object Reconstruction Using Colour-Coded Structured Light

Philippe Lavoie

A thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfilment of the requirements
for the degree of Ph.D. in Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering (OCIECE)
School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

©Philippe Lavoie, Ottawa, Canada, 2004

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-494-01728-7
Our file Notre référence
ISBN: 0-494-01728-7

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

ABSTRACT

This thesis introduces a new robust and high precision methodology and corresponding techniques for reconstructing three-dimensional (3-D) views of real objects from one single two-dimensional (2-D) image. The reconstruction process relies on obtaining the two dimensional image of a real object from a camera, on which a colour coded structured light is projected. The method does not require any *a-priori* knowledge of the absolute positioning or orientation of the camera and the projector which illuminates the scene with the colour coded structured light. Prior to the 3-D reconstruction steps, a calibration process is used to provide a high precision calculation of both the camera's and the projector's intrinsic and extrinsic parameters. These parameters are essential to the 3-D object reconstruction technique as introduced in this thesis. The structured light is used to determine unique patterns on the object's surface. The lines provide a series of control points which once extracted from the object's 2-D view, are used in the creation of Non-Uniform Rational B-Spline (NURBS) curves in 2-D. Those NURBS curves are then projected into space to eventually re-create a 3-D surface. The technique's precision depends on the structured light sampling rate adaptation on the object surface. In a recent test, a 7 tau (thousands of an inch) precision was achieved with a relatively smooth object.

The thesis will focus on the 3-D reconstruction of an object based on its single 2-D view. The approach proposed is end-to-end since it handles all the steps necessary to go from a 2-D view of an object to its modelization using NURBS curves and eventually to a surface in 3-D which can be manipulated with 3-D

editing software on a computer monitor. The thesis's results are applicable to many domains. One of those domains is the CAD/SLA systems for manufacturing applications. Furthermore, NURBS surfaces can compress the 3-D image of a real object in a very efficient manner.

Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

Various abbreviations used frequently in this thesis are summarized below. All notations are defined in the text when they first appear.

2-D	Two Dimensions
3-D	Three Dimensions
DLP	Digital Light Processing
NURBS	Non Uniform Rational B-Spline
POV	Persistence of Vision
PRBA	Pseudo Random Binary Array
PRBS	Pseudo Random Binary Sequence

PRMVS	Pseudo Random Multi-Variate Sequence
SLA	Stereolithography
SLR	Single Lens Reflex
Std Dev	Standard Deviation

LIST OF SYMBOLS

Various symbols are often used in this text. A list of these is given below.

$B_{i,n}$	The classical nth-degree Bernstein polynomial
C	A NURBS curve
C^w	A NURBS curve in homogeneous space
F	The fundamental matrix
N	The B-spline basis function
P	A control point
P^w	A control point in homogeneous space
S	A NURBS surface
S^w	A NURBS surface in homogeneous space
U	The knot vector
V	The know vector in the column direction
w	The weight of a control point

Chapter 1

Introduction

Creating the three dimensional (3-D) representation of an object or a set of objects (a scene) inside a computer can be done using different image processing techniques. Each varies in terms of precision, costs, computational requirements and advantages in regards to what can and can't be reconstructed. Low cost techniques, with the advent of low cost and very high resolution cameras, are extremely appealing from a research point of view and normally rely on the processing of one or multiple bi-dimensional (2-D) images. The problem researchers try to solve is to re-create the scene in 3-D using the information content of 2-D images. This thesis introduces a new methodology to solve the problem of 3-D object reconstruction based on the projection of a colour encoded grid and the reconstruction of the object in 3-D using a network of NURBS curves, which are used in conjunction with the Gordon surface Algorithm to effectively capture the real 3-D topology of the object.

1.1 The Goal

This thesis addresses the following problem: generate a 3-D object inside a computer based on 2-D images. There are several ways one can choose to accomplish this. Therefore, it is important to know the basic assumptions made in order to narrow down the problem space.

The first assumption is that the technique must rely on low-cost equipment. With sufficient funding, it is possible to obtain a laser system to recreate a 3-D object with very high precision. However, it is important to explore low-cost alternatives. One of the most inexpensive solutions is to capture information about an object using a camera. With the advances made in the design of digital cameras, it is now possible to get a 6.3 mega pixel digital SLR camera for $1,450. The availability of such high precision, low cost equipment can be leveraged to solve the problem of recreating an object in the 3-D space.

The second assumption is that stereo fusion is used. Stereo fusion recreates the 3-D content of an image through triangulation. The estimation process requires the knowledge of the global position and orientation of each camera, as well as the model of the camera and the correspondence between all the feature points of both images. Stereo fusion can be active or passive. For this thesis, an active stereo fusion process is used.

In an active setup, one of the cameras is replaced by a projector which illuminates the scene with a structured light pattern image. The structured pattern can be made out of dots, lines, grids, or any other pattern. The choice of the pattern is often dictated by the application domain. For example, in a cookie factory where cookies are baked on a conveyor belt which runs at high speed, a line pattern can be used to illuminate the cookies. A camera will be used to rapidly estimate the

height and shape of each cookie.

There are a lot of similarities between active and passive stereo fusion methods. In the active stereo fusion model, the projector acts as a second camera with a pre-fixed image. The correspondence between the camera and the projector is obtained by matching the pre-fixed image with the one viewed by the camera.

The third assumption is that a software calibration technique is used to further limit the cost and also simplify the set up required. The less the amount of hardware required, the easier it is to setup the required environment. Furthermore, the calibration of the system is a requirement of the proposed methodology to result in a high precision of the 3-D reconstruction of an object.

A fourth assumption is that a high precision has to be attained by the system. The use of an active stereo system allows the technique to be of high precision without the need of an expensive setup. An active stereo system is necessary to obtain sub-pixel accuracy in establishing a correspondence between the camera and the projector's view. The system uses a digital camera to capture the object.

A fifth assumption is that the camera used has a much higher resolution than the projected image. This allows the camera to be effective at capturing the projected line information.

The last assumption is that one can control the lighting and other environmental conditions while the 3-D capture occurs. While some methodologies would trade precision in order to be able to capture the 3-D information of an object in any environment, this is not the case here. The algorithm is also limited to situations for which a light can be projected and accurately perceived by a camera. There are situations where either the light will not reflect well on the object, or the ambient light causes the projected pattern to be non-retrievable by the camera

or by the computational algorithm chosen.

Figure 1.1 depicts the general setup proposed in this thesis. It contains a projector, a camera, a rotating table, an object and a computer. The projector sends a structured light on the object by illuminating a transparent film which has the structured information printed on it. This scene is picked up by the camera and the information is processed by the computer to generate a 3-D view. The rotating table can be used to get different views of the object. The combination of all the views would recreate the full object in 3-D space.

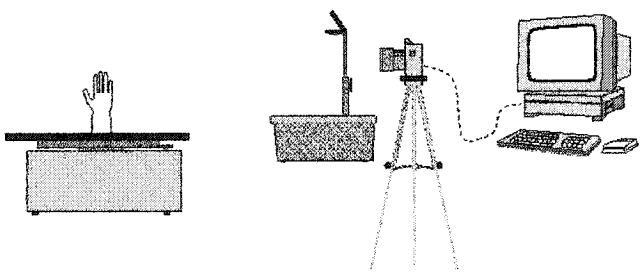


Figure 1.1: The general setup for 3-D capture is composed of a rotating table, a projector, a camera and a computer.

1.2 The 3-D Object Reconstruction Approach

This work focuses on using the 3-D information contained inside a single view of an object on which the pattern is being projected. Multiple views of the object could be merged to generate a more complete understanding of the object in 3-D. However, this last step to be taken in the 3-D object reconstruction algorithm is outside the scope of this thesis.

The methodology and the techniques developed and proposed in this thesis are based on the standard "active stereo fusion" approach. It differs from most of the other "active stereo fusion" methods, which rely on point mapping between the camera view and the image projected. The approach proposed herein relies on modelling the objects using Non Uniform Rational B-Splines (NURBS) and using their projective invariant properties to recreate the object in 3-D. This is a new methodology in 3-D object reconstruction. It was introduced and used by the author of this thesis in [40].

With an active stereo fusion approach, ambiguities can exist when establishing the correspondence between the structured light projected by the projector and the image perceived by the camera. The order in which the dots, strips or grid are projected may not be perceived correctly by the camera. One can have problems locating a particular dot, strip or grid from the captured image. The ambiguities in finding a correspondence between the image and the structured light can be removed by encoding the projected pattern [41]. The methodology proposed in this thesis relies on Pseudo Random Colour Encoding of a grid pattern to eliminate ambiguities in the recovery of the 3-D estimates. This is also a new technique introduced by the author in [38].

The encoding is done in 2-D and therefore allows the camera to know the exact relationship between the grid it perceives and the grid projected by the projector. At the intersections of the grid lines, a point mapping can be obtained. However, the system relies more on line mapping to generate the 3-D content of the object.

The computing complexity required by the preprocessing step is greatly reduced by illuminating the object. Traditionally, this is done by either one of the following methods:

1. Calibrating the position of the cameras with a known object: the object must contain easily recognizable features, or

2. Complex search in 2-D space from the image in order to find correspondences. This method usually involves correlations of areas around the points of interest, for example edge or corner points.

Eventually the method 1 has to be used if a complete model of the 3-D world has to be generated (global orientation). The method 2 is sufficient if a scaled version is acceptable (relative orientation). The structured light has the advantage of simplifying both methods 1 and 2. It provides easily recognizable feature points. The method which determines those feature points is the same for both the calibration and reconstruction processes.

The method proposed here for 3-D reconstruction involves the following steps:

Step 1 Calibrate the camera by using a known object which has a Pseudo Random Colour Array printed on it. That encoded grid is the same as the one used by the projector.

Step 2 Calibrate the projector by illuminating the object of step 1.

Step 3 Place an object properly inside the volume of interest and illuminate it with the structured light without modifying the camera's and the projector's location.

Step 4 Match the projected lines obtained by the camera with the ones from the projector's image.

Step 5 Fit the NURBS curve through each grid line obtained in Step 4.

Step 6 Project the NURBS curves in 3-D using the calibration parameters.

Step 7 Reconstruct the NURBS surface using the network of 3-D NURBS curves obtained in step 6.

The accuracy of the approach will be measured by the following steps. To begin with, the accuracy of the calibration will be demonstrated. The precision of the calibration will be determined by analyzing the precision of the location of a cloud of points with known 3-D world coordinates.

Then the accuracy of the 3-D reconstruction will be assessed. Simple objects with precise 3-D coordinates are captured by the system for the precision of the reconstruction to be assessed. The objects are used to verify how the system handles flat surfaces, curved surfaces and edges.

1.3 Contributions

The major contributions of this work are:

- The first contribution is with the calibration process. The calibration process in most techniques normally uses a different methodology than the methodology used for the 3-D reconstruction process to acquire feature points. The approach proposed here has high precision in terms of feature extraction. This high precision benefits both the calibration and the 3-D reconstruction processes [39].

- The pseudo-random encoding technique proposed here used to map a structured light pattern is unique. It is a grid composed of lines which are colour coded using a pseudo random multivariate sequence (PRMVS) [40]. The

PRMVS removes ambiguities when registering information contained in the images viewed from the camera and the projector. This form of encoding allows the methodology to match the whole line as opposed to only matching interest points such as intersections or other meaningful feature points located on an image.

- The 3-D reconstruction process doesn't generate a cloud of points but instead generates a 3-D NURBS surface. This kind of surface is more easily manipulated by a computer than a cloud of points and it conveys more meaning information. If multiple views are to be used, then a network of 3-D NURBS curves is also a lot more meaningful in terms of 3-D information than a cloud of points. Finally, a network of 3-D NURBS curves can be viewed as a way to compress the 3-D information obtained from the image.

- Feature extraction with the proposed algorithm does not rely on single points but instead relies on the information contained on a grid line. Furthermore, the feature extraction uses a parametric representation which adapts to any shape, from linear lines to free hand drawing. Most methods rely on the matching of points between the left and right images to create a depth value at that point. This limits their ability to work at sub-pixel accuracy. To work at sub-pixel accuracy, those methods must rely on multiple views registration. Multiple views registration is not a substitute for the real sub-pixel feature extraction which is proposed in this thesis.

- The approach is global. It accounts for all the necessary steps to go from an image of an object to its model obtained with NURBS curves inside the 3-D world coordinate system. Several approaches do not use 3-D world coordi-

nates and instead rely on relative coordinates. By doing so, they minimize the issues related to going from the relative to the absolute world.

Multiple minor contributions are also generated by the proposed algorithm.

- The proposed structured light also enables the calibration process. Most structured lights cannot be used when performing calibration since they rarely encode in 2-D. This characteristic is shared with some previous work that was done with pseudo-random encoded binary arrays. However, the accuracy of the current method is higher.

- Sub-pixel accuracy is possible with the proposed approach since it doesn't rely on the grid lines themselves but on the edges which constitute these lines.

- This thesis does not rely on the line definition itself, but on the accurate location of the edges on both sides of the line. This not only improves the accuracy of the edge detection in 2-D but also generates a more accurate 3-D image of the object. The 3-D reconstruction is improved because a line's true center is different in the 2-D view perceived by the camera than it is on the 3-D object illuminated by the projector.

1.4 Thesis Outline

The thesis is structured as follows. In Chapter 2, a review of the techniques for stereo fusion and structured lighting is presented.

The following chapter gives a primer on the mathematics and algorithms associated with NURBS curves and surfaces. NURBS forms the basis on which the

3-D reconstruction process of an object relies.

Chapter 4 introduces the algorithm for creating a 3-D object at a high level. It lists the steps needed along with the algorithms and techniques required for the 3-D reconstruction.

The structured light used in this thesis is encoded with a PRMVS. The supporting theory for this type of encoding along with a description on how it is used to recreate the object in 3-D is given in Chapter 5.

Chapter 6 deals with line and edge detection to sub-pixel accuracy, while Chapter 7 discusses the 3-D reconstruction algorithm used. Chapter 8 explains the calibration process proposed by this thesis. Chapter 9 shows the results obtained for the calibration and with the reconstruction of 3-D objects with both synthetic and real objects.

Chapter 2

Methods for 3D reconstruction

Several methods are used to solve the problem of recreating a 3-D view of an object from the information contained within two 2-D images. Barnard and Fischler [5] along with Dhond and Aggarwal [17] analyzed some of the methods used for stereo fusion. Jarvis [35] wrote a perspective on range finding techniques such as structured lights, stereo fusion, focusing the cameras, camera motion, etc. More recently, Salvi [67] wrote a report on the different codification strategies available in structured light systems. In this chapter, the structured light and stereo fusion methods will be reviewed and a description of the newer techniques will be presented.

2.1 Methods for Stereo Fusion

Establishing a correspondence between the matching points of a set of stereo images is the most important and difficult step toward the reconstruction of the observed 3-D scene. Matching strategies can be differentiated by the matching

primitive and geometry used. The matching primitive selected differentiates the area-based from the feature-based matching methods. Imaging geometry creates a distinction between parallel axis and non-parallel axis stereo systems.

Dhond and Aggarwal [17] have described the difference between area-based and feature-based matching as follows:

> "'Area-based stereo techniques use correlation among brightness (intensity) patterns in the local neighborhood of a pixel in one image with brightness patterns in a corresponding neighborhood of a pixel in the other image. First, a point of interest is chosen in one image. A cross-correlation measure is then used to search for a point with a matching neighborhood in the other image. The area-based techniques have a disadvantage in that they use intensity values at each pixel directly, and are hence sensitive to distortions as a result of changes in viewing position (perspective) as well as changes in absolute intensity, contrast, and illumination. Also, the presence of occluding boundaries in the correlation window tends to confuse the correlation-based matcher, often giving an erroneous depth estimate.
>
> Feature-based stereo techniques use symbolic features derived from intensity images rather than image intensities themselves. Hence, these systems are more stable concerning changes in contrast and ambient lightning. The features used most commonly are either edge points or edge segments (derived from connected edge points) that may be located with subpixel precision. Also feature-based methods allow for simple comparisons between attributes of the features being matched, and are hence faster than correlation-based area matching

methods."'

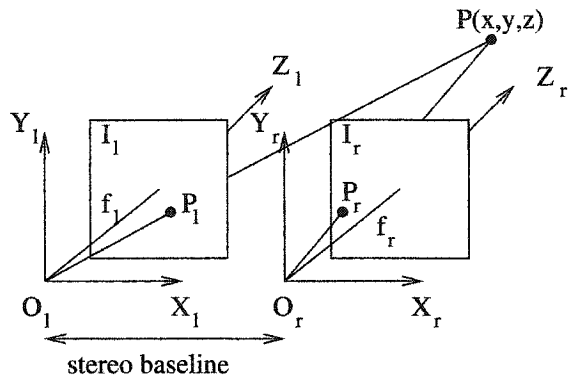


Figure 2.1: Conventional stereo system.

The image geometry being used affects the stereo matching paradigm. The conventional stereo system involves a pair of cameras with their optical axes parallel and horizontally separated. The stereo baseline denotes the horizontal distance between these. The cameras have their optical axis perpendicular to the stereo baseline and their image scanline is parallel to the stereo baseline. Such a system is shown in Figure 2.1.

In the conventional system shown in Figure 2.1, O_l and O_r represent the origin of the left and the right cameras respectively, f_l and f_r represent their focal length, and P_l and P_r show where the point P is located in both images. The point P is the 3-D location of a point of interest.

The epipolar constraint is valid for each scanline in a conventional system. This means that a point in a scanline from the left image is present in the corresponding scanline of the right image.

The epipolar constraint simplifies the matching problem to one dimension.

This explains why the majority of the stereo matching algorithms assume the validity of the epipolar constraint. However, this requires a stereo rig of very high precision and, in practical situations, the epipolar constraint for each scanline is generally not valid. A method to transform stereo images from a non-parallel system to an epipolar constrained one is described by Lavoie [37].

Every method has to deal with half-occlusion points: points visible in one of the image but not in the other. A recent overview [19] of different half-occlusion approaches compared and described the following five methods: bi-modality, match goodness jumps, left-right checking, ordering and occlusion constraint. The results indicate that, in general, the occlusion constraint method seems to yield better results but that in textured images left-right checking yields better results.

2.1.1 Area Matching

The area matching techniques attempt to match points from the left and the right image by analyzing an area around each point and then by trying to establish a correspondence between each area. Figure 2.2 shows the general approach used by area matching techniques. In that figure, the grayed region in the left and right image are found to have a high level of correlation and therefore are marked as matching regions.

Most of the earlier work used a simple correlation measure around points of interest to match neighborhoods of points. To improve the results, Moravec [49] used a coarse-to-fine technique. Initially, an interest operator is used to identify feature points. For each feature point from the left image, a search is made in the right image at various resolutions $(\times 16, \times 8, \ldots, \times 1)$ starting from the coars-

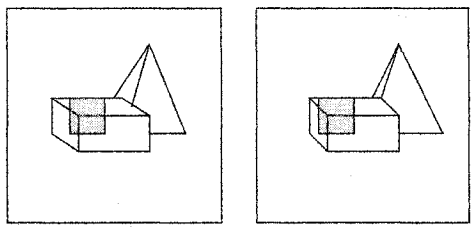


Figure 2.2: Area matching

est. At each resolution, the position yielding the highest correlation coefficient is enlarged to the next resolution. The process continues until the resolution $\times 1$ is reached. A disparity was found for each of the feature points with such a technique.

Further improvements were made by Gennery [22] who developed a high-resolution correlator which gave better results and also was able to estimate the accuracy of the match in the form of a variance and a covariance of the point (x, y) of the match in the second image. Other improvements include the ones made by Hannah [28] who was able to obtain subpixel matching by parabolic interpolation of correlation values.

2.1.2 Relaxation Process

Rosenfeld and al. [62] proposed a fairly general model for scene labeling called relaxation labeling. To match stereo pairs using relaxation labeling, a set of feature points (nodes) are identified in each images. The problem involves assigning unique labels (or matches) to each node out of a list of possible matches. For each candidate pair of matches, a matching probability is updated iteratively depending upon the matching probabilities of neighboring nodes. Strong neighboring

matches improve the chances of weaker matches in a globally consistent manner.

Marr and Poggio [47] [46] used the neighborhood information of that match primitives in a simple iterative scheme. A two dimensional network of nodes is set up for each scanline pair in the stereo images. The horizontal and vertical connections are described as inhibitory. The diagonal connections are termed excitatory. All the horizontal and vertical nodes inhibit each other until finally only one match remains on each horizontal and vertical line.

Barnard and Thompson [4] suggest to extract feature points (nodes) using the Moravec[49] interest operator from each image. Each node in the left image is assigned a set of labels that represent the possible candidate matches from the right image within a disparity range. A relaxation process is then applied to impose global consistency. The consistency of a node is given by a probability factor. The iterative procedure used is continued until the probabilities reach a steady state or a maximal number of iterations is reached.

Kim and Aggarwal [36] propose a relaxation scheme that combines three disambiguity constraints: continuity of disparity, figural disparity and smoothness of probability (certainty) of matching. The matching probability corresponds to a correlation factor.

2.1.3 Using Edge Segments

It has been shown by Ayache and Faverjon [1] and by Medioni and Nevatia [48] that piecewise-linear approximation to connected edge points as a matching primitive is a viable alternative to matching individual edge points (Figure 2.3). Linear edge segments have some advantages over single edge points in the matching process:

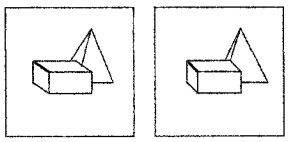


Figure 2.3: Matching using edge segments

- A positional error at an isolated point has little effect on the position and orientation of the edge segment;
- Most of the edge points lie very close to the best fit of the edge segment;
- The edge connectivity constraint is always met.

On the other hand, the edge segments might be fragmented in the preprocessing stage. Provision must be taken to allow a segment to match one, two or more segments in the other image and vice-versa.

Two main approaches are used for edge segment matching: the minimal differential disparity criterion [48] and the Ayache-Faverjon algorithm [1]. The main differences between the two is that the Ayache-Faverjon method utilizes a generalized non-parallel axis imaging geometry and uses disparity between midpoints of matching line segments rather than average disparity between corresponding points that lie on matching line segments.

2.1.4 Integrated Methods

Integrated methods are methods where the disparity map is obtained by joining the information received from two or more different approaches.

Cochran and Medioni [10] developed a method where an area-based approach

is fused with a feature-based approach. They use a coarse-to-fine approach. Initially, the feature and area-based methods are computed separately. Their results are combined to form a dense disparity map. For the finer levels, the dense disparity map of the previous level is used to improve the matching.

Maître and Luo [45] propose to combine the stereoscopic and monocular information of a scene. Their stereoscopic information (the disparity image) is obtained by any conventional stereo algorithm. The monocular information is used to generate homogeneous regions in the left and right images. A planar or quadratic model of each region is made using the disparity map. The model is then used to refine the disparity map.

Guerrero [25] combine the geometric description of the scene with information about the image brightness. Doing so, they can compute the motion direction of a camera from the brightness information on a straight edge. A heuristic filtering is proposed to solve inaccuracies when they compute the local gradient of the brightness. This allows Guerrero to consider topological relations between pixels of lines and brightness information in motion computation.

Pajares [53] approaches the problem of local stereo-vision matching by using edge segments as features with four attributes. A matching probability is assigned between pairs of feature of the stereo images. He introduces a nonparametric strategy based on Parzen's window to estimate a probability density function which is used to obtain the matching probability.

2.1.5 Dynamic Programming

Baker and Binford[2] use the Viterbi algorithm, a dynamic programming technique, to partition the stereo matching problem recursively based upon the con-

straint that a left-to-right ordering of edges is preserved along a scanline in a pair of stereo images. A first pass is done to match half edges in the left image to those in the right and vice-versa. A cooperative procedure then identifies surface contours that are not continuous in disparity. Finally, an intensity-based Viterbi correlation, performed between intensity pixels from scanline intervals lying between a match pair of edges, yields a denser depth map.

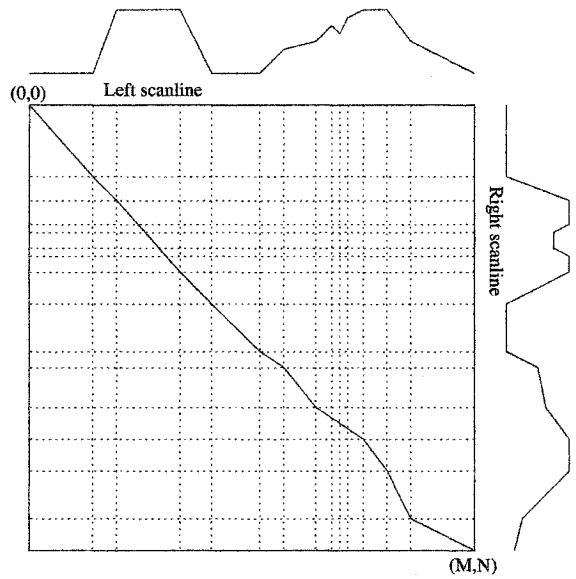


Figure 2.4: 2-D search plane for intra-scanline search

Ohta and Kanade [51] use pixel intensities of scanline intervals (delimited by edge points) to guide the intra-scanline matching search by dynamic-programming. It is formulated as a path finding problem in a 2-D search space in which the horizontal and vertical axes are respectively the right and left scanlines. The 2-D search plane is shown in Figure 2.4. A cost function is associated with each partial path. The edges are numbered from left to right on each scanline. The end

points of each scanline are also numbered as nodes. The solution can be represented as a sequence of straight lines from node $(0, 0)$ to node (M, N) where M is the number of nodes in the left image and N the number of nodes in the right image. Next inter-scanline matching is done to impose consistency among matches obtained at each scanline using edge connectivity.

Roy and Meunier [63] proposed a Dynamic programming method where the cost function depends on the luminosity of each pixels. Their method generates a dense disparity map from a sequence of two or more stereoscopic images. They allow the camera displacement between each image to be any combination of rotation and translation.

2.2 Methods Using Structured Light

In a laboratory or an industrial location where a secluded space can be created and isolated from light variation, special lighting effects can be used to reduce the computational complexity of passive stereo matching methods and to improve the reliability of 3-D object analysis. This class of methods involves illuminating the scene with a controlled light and interpreting the pattern of the projection in terms of the surface geometry of the objects.

One of the controlled light schemes is to use a single ray of light scanned over the scene. When the light source is displaced from a viewing TV camera, the camera's view of the stripe shows displacements along a stripe which are proportional to the depth. A kink in the displacement indicates a change of plane and a discontinuity. A discontinuity corresponds to a physical gap between surfaces. The proportionality constant between the beam displacement and depth is dependent

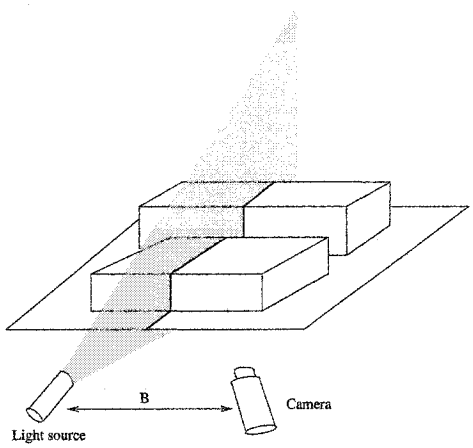


Figure 2.5: Range measurement by a structured light approach.

upon the displacement between the light source and the camera. A more accurate depth measurement can be made with larger displacements. One of the drawbacks of this method is the need to analyze a huge sequence of images in order to obtain a 3-D representation of the scene. More recently, Park [54] created a dual beam version of the standard structured light algorithm.

The number of images to analyze is reduced by projecting multiple lines onto the scene. However, line identification for tracing purposes becomes difficult. The strips patterns are encoded to simplify the identification process. The encoding techniques used are colour coding[8], space encoding[59] [42], thickness labelling [42] and identification by PRBS [77]. In the case of grid patterns, the encoding used can be colour coding, feature coding and the identification by PRBA coding [74][71].

Will and Pennington[82] describe a method by which the locations and orientations of planar areas of polyhedral solids are extracted through linear frequency

domain filtering applied to images of scenes illuminated by a high contrast rectangular grid of lines. The edges are defined to be the intersections of the extracted planes.

Chen and al. [9] proposed a method to reconstruct and model a polyhedral surface of the scene using grid encoding. They model the polyhedral plane surface based on a model first described by Tseng and Chen [76]. They then use sensitivity analysis and a method to fuse line segments from different planes to refine the modelling. The sensitivity analysis is a function based on the Normal Vector estimation and the depth estimation.

A PRBA identification is used by Petriu and al. [71] to reconstruct a 3-D object. This method finds the 3-D locations of all the intersection points of the grid. It doesn't model the surfaces on which the grid is projected. Petriu introduced the idea to use De Bruijn sequences or Galois fields to code the location of points in an area. That idea is applied to a colour grid by Lavoie [40] and also later on by others [52].

Forster [21] projects a coloured structured light made of coloured lines and uses the edges to compute a 3-D correspondence. He uses non-trivial Hamming distance between adjoining patterns and binary values for each of the colour components (0 and full) to obtain robust results.

Huynh and al [31] combines a novel calibration approach for stripe patterns to simplify the shape reconstruction stage. The light stripes are encoded in binary and are projected in succession on the scene: first all stripes are on, then only the odd ones, then 4 consecutive, then 8, 16, 32 and finally 64 consecutive strip rows are projected. This approach yielded for a fan object a mean deviation of 1.778 mm and a standard deviation of 1.220 mm.

A method similar to Huynh was developed by Holt and Rusinkiewicz [26]. The method allows the capture of moving objects.. The precision of their real-time system is under 0.1mm not taking into account outliers and distortion. The precision for highly textured objects decreases to 0.2 mm. Other works with time-multiplexing systems using structured light can be found in [23, 27].

Zhang and al [84] use colour coded strips of lights and a multi-pass dynamic programming algorithm which allows his technique to eliminate the global smoothness assumptions and the strict ordering constraints often present. Their approach can generate high-speed scans of moving objects from a single stripe pattern. It can also generate high-resolution scans of static scenes using a sequence of time-shifted stripe patterns.

A review of the progress on the use of structured light to solve the correspondence problem in stereo vision is given by Mouaddib and al [50]. This review covers the use of structured light from 1982 to 1993. More recently, Salvi [67] wrote a review of the different codification strategies used in structured light systems.

2.3 Hybrid Methods

In [37] a hybrid method is defined which use a structured light and two camera views to reconstruct an object in 3-D. The structured light was encoded using a PRBA.

The encoding of the structured light removed most of the ambiguities in matching the left and the right camera views. It relied on a robust method developed by Zhang and al. [86] to estimate the fundamental matrix F. The F matrix gives the

relation between the location of two different views of the same scene.

Definition 2.1. *The epipolarization process consists of aligning the images such that a match between the left and the right image along each scan-line can occur,* i.e. *the epipolar constraint becomes true.*

The estimation of the fundamental matrix F allows the epipolarization of the input images. Once the epipolarization is performed, a dynamic programming technique is used to match pixels between the left and right camera view. Dynamic programming is a fast and optimal method to determine an optimal path. In the case of that thesis, the optimal path was the disparity between points from the left and right images.

Zhang and al. [85] propose a new hybrid method based on his earlier work with colour stripe structured light. Their work extends the traditional binocular stereo problem into the spacetime domain, in which a pair of video streams is matched simultaneously instead of matching pairs of images frame by frame. Their approach can be used to augment the quality of a reconstruction of any stereo matching system by replacing the image matching algorithm with a space-time algorithm.

2.4 Multi-View Registration

The methods presented above create in most cases a single view of the object. The object being scanned or the camera must be moved if a different view of the object is to be obtained. This creates multiple views of the object and the process to integrate all those views to form a single surface representation of the object is multi-view registration.

There are currently off-the-shelf software which can be bought that will automatically register cloud of points and create an integrated view based on the multi-view information. Two such products are named Cyclone [13] and RapidForm[33]. These products also have the ability to triangulate the resulting clouds for further processing by Computer Aided Design (CAD) systems.

There is a well known methodology [7] to perform multi-view registration called Iterative Closest point (ITC). This method consists of iteratively minimize the mean square error between points in one view and their closest point in an another view. An improvement over the ITC approach is proposed by Silva and al [69]. In their approach, they use Genetic Algorithms to optimize the search in a space of transformations. Their approach improves on the ITC methodology and also guarantees that the approach will converge to a solution. Their approach requires a set of corresponding points between two views.

Recently, Sharp and al [68] create a new methodology to register range data from unknown view points. They separate the local problem of pair-wise registration and the global problem of minimizing the distribution of accumulated errors inside the integrated view. The strength of their approach is that it doesn't rely on the knowledge of corresponding control points.

2.5 Conclusion

3-D object reconstruction involves trade-off between speed, quality and precision. The literature offers a wide range of methods to recreate an object or a scene in 3-D from a set of images, and this with or without the use of a structured light. Those methods all have shortcomings when it comes to the problem of recreating

an object with a smooth texture.

One of the features (or problem) of the techniques used in the literature is that they work mostly pixel by pixel except for feature based matching techniques. This means that the pixel is the input from which a 3-D location is determined. This precludes subpixel accuracy methods, and the ability to infer knowledge from the structure of the image itself to recreate a 3-D object. Feature based matching techniques do not rely on pixel matching, but are limited to edges seen from the camera view and have problems inferring the 3-D information when the object is smooth and has relatively no features.

In an industrial setting, objects being measured are often smooth and texture-less. Passive stereo fusion methods do not provide enough information to fully re-create these kinds of objects. Dynamic stereo fusion methods must be used instead. Current techniques in dynamic stereo fusion often require the *a-priori* knowledge of the extrinsic and intrinsic parameters of the camera and the projector.

The work presented in this thesis solves the problem of only using the pixel as the unit for 3-D object reconstruction. A colour encoded grid is projected onto a surface which adds features on smooth objects and also enables the calibration process to work to a great precision. The grid information viewed by the camera can be computed to subpixel accuracy. Because it relies on the information contained on a line instead of a single pixel, the 3-D reconstruction process is more precise than a per pixel methodology.

Chapter 3

Non Uniform Rational B-Splines

One of the key concepts introduced by this thesis to solve the 3D object reconstruction problem is the use of NURBS curves and surfaces that allows one to model a line or a surface using a parametric representation. NURBS have useful properties which make them suitable for the 3D object reconstruction problem. For example, commercial entities such as RSI [64] or Raindrop Geomagic [32] use NURBS to reverse engineer a 3-D model by reconstructing a NURBS surface from a cloud of points. The NURBS surface model is then used inside CAD systems for manufacturing or entertainment purposes.

The derivation of various NURBS related formulas and the algorithms associated with them are the essence of the work being done in this thesis. This chapter provides an introduction to NURBS. It is followed by algorithms used to interpolate or approximate a set of points with a NURBS curve. These algorithms are the theoretical foundation on which the 3D object reconstruction algorithm presented here holds. Finally, a method to generate a NURBS surface based on a set of points and on a set of curves is presented.

3.1 NURBS Introduction

A three dimensional (3-D) object is composed of curves and surfaces. A method is needed to represent the curves and surfaces in order to model accurately an object. The two most common methods used are the implicit and the parametric method [58].

The implicit method uses a function which depends on the axis variables and is usually equal to 0. It describes a relationship between the axis variables. Given a two dimensional Euclidian space $E(x, y)$ where $x \in \mathbb{R}$ represents the horizontal axis and $y \in \mathbb{R}$ represents the vertical axis. Then the function $f(x, y) = x^2 + y^2 - 1 = 0$ represents a circle of radius 1.

In the parametric method each of the axis variables is a function of an independent parameter. In this form, a curve would be defined with the independent variable u as

$$C(u) = [x(u), y(u)] \qquad a \leq u \leq b$$

To represent the first quadrant of a circle in a parametric form, the equation becomes

$$C(u) = [cos(u), sin(u)] \qquad 0 \leq u \leq \frac{\pi}{2}$$

or

$$C(u) = \left[\frac{1-u^2}{1+u^2}, \frac{2u}{1+u^2}\right] \qquad 0 \leq u \leq 1$$

This shows that the parametric representation of a curve is not unique.

To visualize how a parametric curve is drawn, imagine that as time increases a new point on the curve is plotted. In the function above, the time is represented with the variable t, it goes from 0 to 1 and it generates a curve like the one in Figure 3.1.

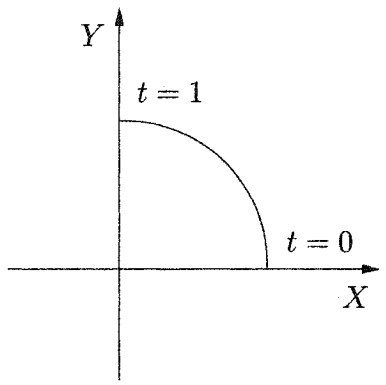


Figure 3.1: A quadrant of a circle generated with $C(u) = \left[\frac{1-u^2}{1+u^2}, \frac{2u}{1+u^2}\right]$ for $0 \leq t \leq 1$

By allowing the coordinate functions $x(u), y(u)$ to be arbitrary, it is possible to obtain a large variety of curves with the parametric form. A NURBS curve and a NURBS surface are classes of parametric curves and parametric surfaces.

NURBS are being used for computational work because they are easily processed by a computer, they are stable to floating points errors and having little memory requirements and because of their ability to represent any kind of curves or surface. They are the generalization of non-rational B-splines which are based on rational Bézier curves. Finally, the rational Bézier curve is a generalization of the Bézier curve, which is explained in subsection 3.1.1.

The material introduced in this chapter is mainly based on the theory and re-

sults as given in [6, 15, 58, 43, 56, 57, 73, 72, 11, 14, 24, 20]

3.1.1 Bézier Curves

Bézier curves are parametric curves which are based on Bernstein polynomials and are defined as follows.

Definition 3.1. *The Bézier curve of degree n is defined as*

$$C(u) = \sum_{i=0}^{n} B_{i,n}(u) P_i \qquad 0 \leq u \leq 1 \tag{3.1}$$

The geometric coefficients $P_i \in \mathbb{R}^3$ are called control points. *The basis functions $B_{i,n}$ are the classical nth-degree Bernstein polynomials defined as*

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \tag{3.2}$$

For interactive shape design, the control points of the Bézier curve convey a lot of geometric information as seen in Figure 3.2.

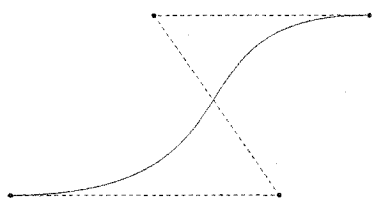


Figure 3.2: A Bézier curve of degree 3.

The Bézier curves are incapable of representing conic curves. A conic curve (such as a circle) can be represented using a *rational function*, defined as the ratio of two polynomials such as

$$x(u) = \frac{X(u)}{W(u)} \qquad y(u) = \frac{Y(u)}{W(u)} \qquad z(u) = \frac{Z(u)}{W(u)} \tag{3.3}$$

Definition 3.2. *The rational Bézier curve is defined as*

$$C(u) = \frac{\sum_{i=0}^{n} B_{i,n}(u) w_i P_i}{\sum_{i=0}^{n} B_{i,n}(u) w_i} \quad 0 \leq u \leq 1 \tag{3.4}$$

where the P_i and $B_{i,n}$ are defined in Definition 3.1 and the $w_i \in \mathbb{R}$ are scalars called the weights.

When the weight associated to a control point is varied, the resulting curve will seem to be "attracted" or "repulsed" by that control point. This is best explained with an example. Four Bézier curves are drawn in Figure 3.3. The only difference between them, is the weight of the control point P_2. The weight of 0.5 makes the curve go outside the boundary drawn by its control points. With the weight equal to 1, the curve is equivalent to the one depicted in Figure 3.2. The weight of 2 tends to push the curve away from the second point and the weight of 10 pushes it even farther.

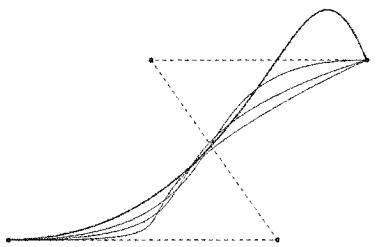


Figure 3.3: The effects of changing the weight of the second control point with $w = 0.5, 1, 2$ and 10.

A curve consisting of only one rational Bézier curve segment is often inadequate. The problems with a single segment range from the need of a high degree curve to accurately fit a complex shape, which is inefficient to process and is numerically unstable, to the need of interactive design for which a single segment has

limitations concerning local control of the shape. To overcome these problems, a piecewise rational curve is used.

3.1.2 B-spline Curves

A piecewise Bézier curve or a B-spline curve is constructed from several Bézier curves joined together at *breakpoints* with some level of continuity between them.

Theorem 3.1. *A piecewise Bézier curve is C^k continuous at a breakpoint u_i if $C_i^{(j)}(u_i) = C_{i+1}^{(j)}(u_i)$ for all $0 \leq j \leq k$ where $C_i^{(j)}$ represents the jth derivative of C_i. [58]*

Such a curve is depicted in Figure 3.4.

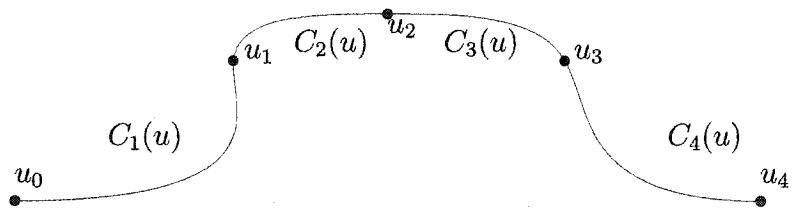


Figure 3.4: A piecewise rational Bézier curve.

The curve $C(u)$ is defined on $u \in [0, 1]$ and it is composed of $m = 4$ segments $C_i(u)$ where $1 \leq i \leq m$. The segments are joined together at the breakpoints $u_0 = 0 < u_1 < u_2 < u_3 < u_4 = 1$ with level 1 continuity.

The different control points composing the B-spline curve are shown in Figure 3.5. The circled control points are used in more than one Bézier segment. It should be clear that storing these points more than once is not memory efficient. If the curve is said to be C^1 continuous, then some of the points inside a Bézier segment are dependent on the position of the previous points to satisfy the continuity constraint. Therefore storing these points in memory is not necessary.

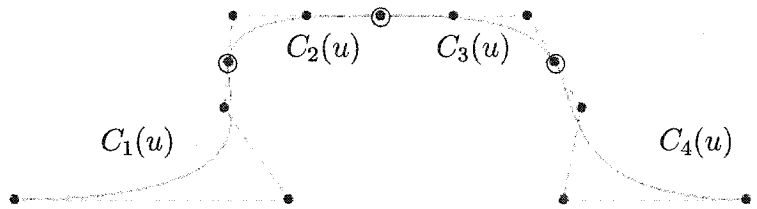


Figure 3.5: The Bézier segments composing the curve.

The equation of the B-spline should therefore be memory efficient and should also allow for the local control of the curve, *i.e.* the basis functions should not be defined over $[u_0, u_m]$. Instead, they should be constrained to a limited number of subintervals.

Definition 3.3. *A B-Spline curve is defined as*

$$C(u) = \sum_{i=0}^{n} N_{i,p} P_i \qquad a \leq u \leq b \tag{3.5}$$

where P_i are the control points and $N_{i,p}$ are the pth degree B-spline basis functions.

There are different methods to define the B-spline basis functions. They consist of: divided differences of truncated power functions [12], blossoming [61] and recurrence formula [11, 14, 15]. The recurrence definition is used since it is well suited to a computer implementation.

Definition 3.4. *Let the knots represent the breakpoints of a B-spline curve. Then a sequence of those knots is the* knot vector *and it is defined as $U = u_0, \ldots, u_m$ which is a nondecreasing sequence of real numbers,* i.e., *$u_i \leq u_{i+1}$ for $i = 0, \ldots, m$ of the form*

$$U = \{\underbrace{a, \ldots, a}_{p+1}, u_{p+1}, \ldots, u_{m-p+1}, \underbrace{b, \ldots, b}_{p+1}\} \tag{3.6}$$

Definition 3.5. *The B-spline basis function of p-degree is defined using the recurrence formula as*

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \qquad (3.7)$$

The above equation can result in a $\frac{0}{0}$ quotient; that quotient is defined to be zero.

The B-spline curve of Figure 3.4 is presented with its *control polygon* in Figure 3.6. The control polygon is the polygon formed by joining the control points P_i.

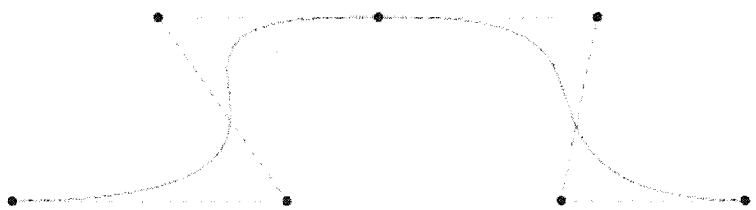


Figure 3.6: A B-spline curve with its control polygon.

3.1.3 NURBS Curve

As mentioned above, only rational functions can represent conic curves. Therefore one could generalize the B-spline curve to obtain a rational representation.

Definition 3.6. *A NURBS curve is the generalization of the B-spline curve and it is defined as*

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \qquad a \leq u \leq b \tag{3.8}$$

where P_i are the control points, w_i are the weights and $N_{i,p}$ are the B-spline basis functions defined on the non-periodic and non-uniform knot vector defined in equation 3.6.

Rational curves with coordinate functions in the form expressed in equation 3.3 have efficient processing and have an elegant geometric interpretation.

Theorem 3.2. *Homogenous coordinates can represent a rational curve in n dimensions as a polynomial curve of $n+1$ dimensions.*

The homogeneous control points are written as $P_i^w = w_i x_i, w_i y_i, w_i z_i, w_i$ in a four dimensional space where $w \neq 0$. To obtain P_i, we divide all the coordinates by the fourth coordinate w_i. This operation corresponds to a perspective map with the center at the origin.

Definition 3.7. *A NURBS curve is defined using homogenous coordinates as*

$$C^w(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i^w \tag{3.9}$$

where P_i^2 are the control points in homogenous space and $N_{i,p}$ are the B-spline basis functions defined on the non-periodic and non-uniform knot vector defined in equation 3.6.

A circle can be represented with a NURBS curve. Its representation will contain weight values different than 1 to be able to represent this shape. For instance, the first quadrant arc shown in Figure 3.1 can be obtained using $\{P_i\} = \{(1,0),(1,1),(0,1)\}$ and $\{w_i\} = \{1, \sqrt{2}/2, 1\}$.

3.1.4 NURBS Surface

NURBS are also used to represent surfaces.

Definition 3.8. *A NURBS surface in homogeneous coordinates is defined as*

$$S^w(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)P^w_{i,j} \tag{3.10}$$

where $P^w_{i,j}$ forms a bidirectional control net, and $N_{i,p}(u)$ and $N_{j,q}(v)$ are the non-rational B-spline basis functions defined on the knot vectors

$$\begin{aligned} U &= \{\underbrace{a,\ldots,a}_{p+1}, u_{p+1},\ldots,u_{r-p+1}, \underbrace{b,\ldots,b}_{p+1}\} \\ V &= \{\underbrace{c,\ldots,c}_{q+1}, u_{q+1},\ldots,u_{s-q+1}, \underbrace{d,\ldots,d}_{q+1}\} \end{aligned}$$

where $r = n+p+1$, $s = m+q+1$ and the limits $[a,b]$ and $[c,d]$ are often set to $[0,1]$.

3.2 NURBS Properties

NURBS curves and surfaces have useful properties when used to represent 3D objects. NURBS curves and NURBS surfaces are both defined as functions of the B-Spline basis function. It is thus important to understand the properties of the B-Spline basis functions before the properties of a NURBS curve and a NURBS surface are addressed.

3.2.1 Properties of B-Spline Basis Functions

The equation 3.7 is re-written here to facilitate the discussion:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

where $U = u_0, \ldots, u_m$ is a nondecreasing sequence of real numbers, *i.e.*, $u_i \leq u_{i+1}$ for $i = 0, \ldots, m$. U has the form defined in 3.6.

Here are some comments about equation 3.7 as noted by Piegl and Tiller [58]:

- $N_{i,0}(u)$ is a step function which is non-zero on the half-open interval $u \in [u_i, u_{i+1})$.
- For $p > 0, N_{i,p}(u)$ is a linear combination of two $(p-1)$-degree basis functions.
- The equation can yield $0/0$ and is defined in this case to be zero.
- The $N_{i,p}(u)$ are piecewise polynomials, defined on the entire real line; only the interval $[u_0, u_m]$ is of interest.
- The half-open interval $[u_i, u_{i+1})$ is called the *ith knot span*. It can have zero length, since knots need not be distinct (just nondecreasing).

The important properties of the B-Spline basis functions are listed below. These properties determine the many desirable geometric characteristics of B-Spline curves and surfaces.

Property 1.1 $N_{i,p}(u) = 0$ if u is not inside $[u_i, u_{i+p+1})$. This is the local support property.

Property 1.2 In any given knot span, $[u_i, u_{i+1})$, at most $p+1$ of the $N_{i,p}(u)$ are nonzero.

Property 1.3 $N_{i,p}(u) \geq 0$ for all i,p and u. This is the nonnegativity property.

Property 1.4 For any knot span $[u_i, u_{i+1})$ the following equations holds true $\sum_{j=i-p}^{i} N_{j,p}(u) = 1$ for all $u \in [u_i, u_{i+1})$. This is the partition of unity property.

Property 1.5 Inside a knot span, $[u_i, u_{i+1})$, all the derivatives of $N_{i,p}(u)$ exist in the interior of that span. At a knot, where k is the multiplicity of that knot, $N_{i,p}(u)$ is $p-k$ times continuously differentiable.

Property 1.6 Except for the case $p = 0$, $N_{i,p}(u)$ attains exactly one maximum value.

3.2.2 Properties of NURBS Curves

The properties of B-Spline basis functions also apply to the rational B-Spline basis functions. From Property 1 to Property 6 enumerated above, one can list the following important geometric characteristics of NURBS curves assuming that the curve is of degree p and that the knot vector goes from 0 to 1:

Property 2.1 $C(0) = P_0$ and $C(1) = P_n$.

Property 2.2 Affine invariance: an affine transformation is applied to the curve by applying it to the control points. NURBS curves are also invariant under perspective projections ([57, 43]).

Property 2.3 Strong convex hull property: if $u \in [u_i, u_{i+1})$, then $C(u)$ lies within the convex hull of the control points $P_{i-p}, \ldots, P_i$.

Property 2.4 $C(u)$ is infinitely differentiable on the interior of a knot span and is $p - k$ times differentiable at a knot of multiplicity k.

Property 2.5 Variation diminishing property: no plane has more intersections with the curve than with the control polygon.

Property 2.6 A NURBS curve with no interior knots is a rational Bézier curve. A NURBS curve can contain nonrational B-spline, and rational and nonrational Bézier curves as special cases.

Property 2.7 Local approximation: if the control point P_i is moved, or the weight w_i is changed, it affects only that portion of the curve on the interval $u \in [u_i, u_{i+p+1})$.

3.2.3 Properties of NURBS Surfaces

A NURBS surface will have very similar geometrical properties than that of a NURBS curve. Assuming that the NURBS surface has degree p in the u direction and degree k in the v direction and that its control points form a $n \times m$ bidirectional net, their geometric properties are:

Property 3.1 Corner point interpolation: $S(0,0) = P_{0,0}$, $S(1,0) = P_{n,0}$, $S(0,1) = P_{0,m}$ and $S(1,1) = P_{n,m}$.

Property 3.2 Affine invariance: an affine transformation is applied to the surface by applying it to the control points. They are also invariant under perspective projections.

Property 3.3 Strong convex hull property: assume $w_{i,j} \geq 0$ for all i, j. If $(u, v) \in [u_{i_0}, u_{i_0+1}) \times [v_{j_0}, v_{j_0+1})$, then $S(u, v)$ is in the convex hull of the control points $P_{i,j}$, $i_0 - p \leq i \leq i_0$ and $j_0 - q \leq j \leq j_0$.

Property 3.4 Local modifications: if $P_{i,j}$ is moved or $w_{i,j}$ is changed, it affects only the surface shape in the rectangle $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$.

Property 3.5 Nonrational B-Spline and Bézier and rational Bézier surfaces are special cases of NURBS surfaces.

Property 3.6 Differentiability: $S(u, v)$ is $p - k$ times differentiable with respect to u at a u knot with multiplicity k and $S(u, v)$ is $q - k$ times differentiable with respect to v at a v knot with multiplicity k.

3.3 NURBS Curve Interpolation

The geometric properties mentioned in the previous section are very useful for interactive shape design and they are also useful when fitting a NURBS curve to a set of points. Local support and interpolation properties are especially important for the problem of fitting points.

Fitting any curve through a set of points does not always appear correct visually but is correct mathematically since it minimizes a given formula, *i.e.* minimize the distance between the curve and the set of points. This is true for any curves regardless if they are NURBS curves or not, but NURBS curve are usually better suited for fitting then most other types of curves because of their geometric properties.

This section looks at interpolation as a method of fitting a set of points with

a NURBS curve. There are two general methods for the interpolation of a set of points: global interpolation and local interpolation. Global interpolation solves a set of functions by looking at all the points at once. Local interpolation looks at a subset of points and creates a NURBS curve by adjoining pieces together using some continuity constraint. In this thesis, only global interpolation is discussed.

3.3.1 Global Interpolation

The global interpolation method fits points globally.

Theorem 3.3. *Given a set of points* $\{Q_k\}$, $k = 0, \ldots, n$, *the global interpolation method determines the NURBS curve of degree* p *which interpolates these points.*

An appropriate parameter value $\overline{u}_k$ for each Q_k needs to be chosen and an appropriate knot vector $U = u_0, \ldots, u_m$ selected such that a system of $(n+1) \times (n+1)$ linear equations can be set up as shown in the following equation:

$$Q_k = C(\overline{u}_k) = \sum_{i=0}^{n} N_{i,p}(\overline{u}_k) P_i \tag{3.11}$$

The control points P_i are the $n+1$ unknowns. Let r be the number of coordinates in the set of data points Q_k (typically 2,3 or 4). Equation 3.11 has one coefficient matrix with r right hand sides and, correspondingly, r solution sets for the r coordinates of the P_i.

The choice of $\overline{u}_k$ and U affects the shape and parameterization of the curve, assuming, that u is bounded in $[0, 1]$. There are three common methods to choose the $\overline{u}_k$:

- Equally spaced:

$$\overline{u}_0 = 0 \qquad \overline{u}_n = 1$$
$$\overline{u}_k = \frac{k}{n} \quad k = 1, \ldots, n-1$$

This method can produce erratic shapes (such as loops) when the data is unevenly spaced.

- Chord length: Let d be the total chord length

$$d = \sum_{k=1}^{n} |Q_k - Q_{k-1}| \tag{3.12}$$

then

$$\begin{aligned} \overline{u}_0 &= 0 \\ \overline{u}_n &= 1 \\ \overline{u}_k &= \overline{u}_{k-1} + \frac{|Q_k - Q_{k-1}|}{d} k = 1, \ldots, n-1 \end{aligned} \tag{3.13}$$

This is the most widely used method. It approximates a uniform parameterization.

- Centripetal method: Let

$$d = \sum_{k=1}^{n} \sqrt{|Q_k - Q_{k-1}|}$$

then

$$\begin{aligned} \overline{u}_0 &= 0 \\ \overline{u}_n &= 1 \\ \overline{u}_k &= \overline{u}_{k-1} + \frac{\sqrt{|Q_k - Q_{k-1}|}}{d} k = 1, \ldots, n-1 \end{aligned} \tag{3.14}$$

This method gives better results than the chord length method when the data takes very sharp turns.

Knots can be equally spaced, that is,

$$\begin{aligned} u_0 = \cdots = u_p = 0 \quad & u_{m-p} = \cdots = u_m = 1 \\ u_{j+p} = \tfrac{j}{n-p+1} \quad & j = 1, \ldots, n-p \end{aligned} \tag{3.15}$$

With equation 3.15, the knots reflect the distribution of the $\overline{u}_k$. Combining equation 3.13 or equation 3.14 with equation 3.15 leads to a system of linear equations (equation 3.11) where the coefficients can be represented with a totally positive and banded matrix having a semi-bandwidth less than p (see [15]). This means that $N_{i,p}(\overline{u}_k) = 0$ if $|i-k| \geq p$. This system of linear equation can thus be solved by Gaussian elimination without pivoting.

The $(n+1) \times (n+1)$ coefficient matrix of the system 3.11 is set up by evaluating the nonzero basis functions at each $\overline{u}_k, k = 0, \ldots, n$.

3.4 Least Square Fitting

Least Square fitting can also be used to create a NURBS curve which fits the set of data points $Q_0, \ldots, Q_m$ in the least square sense. Let p be the degree of the desired NURBS curve. It has the following properties $p \geq 1$ and $n \geq p$. The variable n is the number of control points desired. Then the least square fitting method is to generate a curve $C(u)$ defined as

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i \qquad u \in [0,1] \tag{3.16}$$

and satisfying the following

- $Q_0 = C(0)$ and $Q_m = C(1)$.
- The other Q_k are approximated in the least square sense to minimize the equation

$$\sum_{k=1}^{m-1} |Q_k - C(\bar{u}_k)|^2 \tag{3.17}$$

with respect to the $n+1$ variables P_i. The $\{\bar{u}_k\}$ are precomputed parameter values.

Algorithm 3.1. *Let*

$$R_k = Q_k - N_{0,p}(\bar{u}_k)Q_0 - N_{n,p}(\bar{u}_k)Q_m \qquad k = 1, \ldots, m-1 \tag{3.18}$$

Then a system of system of $n-1$ *equations in* $n-1$ *unknowns can be obtained and is defined as*

$$(N^T N)P = R \tag{3.19}$$

where

$$N = \begin{bmatrix} N_{1,p}(\bar{u}_1) & \cdots & N_{n-1,p}(\bar{u}_1)l \\ \vdots & \ddots & \vdots \\ N_{1,p}(\bar{u}_{m-1}) & \cdots & N_{n-1,p}(\bar{u}_{m-1}) \end{bmatrix} \tag{3.20}$$

and

$$R = \begin{bmatrix} N_{1,p}(\bar{u}_1)R_1 + \cdots + N_{1,p}(\bar{u}_{m-1})R_{m-1} \\ \vdots \\ N_{n-11,p}(\bar{u}_1)R_1 + \cdots + N_{n-11,p}(\bar{u}_{m-1})R_{m-1} \end{bmatrix} \tag{3.21}$$

and

$$P = \begin{bmatrix} P_1 \\ \vdots \\ P_{n-1} \end{bmatrix} \tag{3.22}$$

Chord length parametrization can be used to obtain the set $\{\bar{u}_k\}$. *The knots should respect this distribution. Let*

$$d = \frac{m+1}{n-p+1} \tag{3.23}$$

then the internal knots can be defined in the range $j = 1, \ldots, n-p$ *by*

$$i = floor(jd) \tag{3.24}$$

$$\alpha = jd - i \tag{3.25}$$

$$u_{p+j} = (1-\alpha)\bar{u}_{i-1} + \alpha\bar{u}_i \tag{3.26}$$

where floor returns the highest integer which is smaller or equal to the variable.

3.5 NURBS Curve Approximation

There is another methodology to fit a curve to a set of points. NURBS curve fitting goes through the set of points and contains at least the same number of control points as the number of points to be fitted. When using approximation, the number of control points that will be used to fit through the set $\{U_k\}$ of points is not known in advance. The curve will be fitted to satisfy a minimizing function, i.e. reduce the distance between the curve and the points to be fitted. This type of fitting is more difficult than least squares fitting. However, it often yields more visually correct results within an error tolerance ϵ.

The approximation methods are iterative in nature and can be global or local. The global approach uses two methods. The first (Type 1) starts with a minimal set of control points, tries to fit a curve with that number of points, then check if the deviation is smaller than E everywhere. If it is smaller, it stops the iteration

process. Otherwise, it tries to fit a new curve by increasing the number of control points used for the fitting process and doing the process again.

The second method (Type 2), starts with a large number of control points and makes sure that the error deviation is still lower then the accepted error deviation E. If it is, it reduces the number of control points and fits a curve through this reduced number of control points. It then does another check for error deviation. If it is small enough, it continues the process of elimination.

The central process in the global methods mentioned above is that one must fit a curve through a known number of control points. In order to do this, a non-linear optimization problem can be set up with the control points, parameters (u_k), knots and even the weights as unknowns. The objective minimizing function must measure the error in some way.

A type 2 method is explained in more details below. This method relies on degree elevation and knot reduction to perform its fitting instead of least square fit or non-linear approximation.

Algorithm 3.2. *The general knot reduction algorithm defines u_r to be an interior knot of a pth degree non-rational curve $C(u)$ where $u_r \neq u_{r+1}$, and the multiplicity of u_r is s. Let $\hat{C}(u)$ denote the curve obtained by removing one occurrence of u_r.*

The new control points P_i^1 and P_j^1 after the removal of the knot u_r are computed from the left and the right by using the following equation.

$$P_i^1 = \frac{P_i^0 - (1-\alpha_i)P_{i-1}^1}{\alpha_i} \quad r-p \leq i \leq \tfrac{1}{2}(2r-p-s-1)$$
$$P_j^1 = \frac{P_j^0 - \alpha_j P_{j+1}^1}{(1-\alpha_j)} \quad \tfrac{1}{2}(2r-p-s+2) \leq j \leq r-s \qquad (3.27)$$

with

$$\alpha_k = \frac{u - u_k}{u_{k+p+1} - u_k} \qquad k = i, j$$

After removal, $\hat{C}(u)$ differs from $C(u)$. Tiller [73] derives error bounds for two cases:

- assume $(p+s) \mod 2 = 0$, set $k = (p+s)/2$, and

$$B_r = |P_{r-k} - \alpha_{r-k} P^1_{r-k+1} - (1 - \alpha_{r-k}) P^1_{r-k-1}| \tag{3.28}$$

where

$$\alpha_{r-k} = \frac{u_r - u_{r-k}}{u_{r-k+p+1} - u_{r-k}}$$

then

$$|C(u) - \hat{C}(u)| \leq N_{r-k,p}(u) B_r \qquad \text{for} \quad u \in [0,1] \tag{3.29}$$

- assume $(p+s) \mod 2 = 1$, set $k = (p+s+1)/2$, and

$$B_r = |P^1_{r-k} - P^1_{r-k+1}| \tag{3.30}$$

then

$$|C(u) - \hat{C}(u)| \leq (1 - \alpha_{r-k+1}) N_{r-k+1,p}(u) B_r \qquad \text{for} \quad u \in [0,1] \tag{3.31}$$

with

$$\alpha_{r-k+1} = \frac{u_r - u_{r-k+1}}{u_{r-k+p+2} - u_{r-k+1}}$$

The above provides the deviation error computed by knot removal. One can then create an algorithm which removes as many control points as possible while keeping the $e_k \leq \epsilon$ for all k representing the set of points $\{U_k\}$ and its associated set of parameters $\{\bar{u}_k\}$. With the information above, a type 2 algorithm can be defined.

Algorithm 3.3. *The algorithm to approximate data points within a certain error bound ϵ proposed by Piegl [58].*

Step 1 *Start with interpolating the points with a degree 0 NURBS rational curve.*

Step 2 *Use Algorithm 3.2 to remove as many control points as possible. The algorithm accumulates the error deviations $\{e_k\}$.*

Step 3 *Do a least square fitting through the data point using the knot vector obtained in Step 2.*

Step 4 *Increase the degree of the curve obtained from the previous step.*

Step 5 *Go back to 2 until the degree has the degree p required.*

The degree elevation approach has the following benefits:

- Geometric characteristics such as cusps and discontinuities in curvature inherent to the data, tend to be captured at the appropriate stage.
- The evolving curve tends to "settle" into a natural parameterization.
- The above algorithm reduces the wiggle in the final curve. A general rule when globally fitting large numbers of points is that the higher the degree and the more knots (interpolation being the limit), the worst the wiggle.

The author of this thesis proposes two other algorithms to approximate data.

Algorithm 3.4. *The first proposed algorithm to approximate a curve within a certain error bound ϵ.*

Step 1 *Start with interpolating the points with a degree 0 NURBS rational curve.*

Step 2 *Use Algorithm 3.2 to remove as many control points as possible. The algorithm accumulates the error deviations* $\{e_k\}$.

Step 3 *Increase the degree of the curve obtained from the previous step.*

Step 4 *Go back to Step 2 until the degree has the degree p required.*

Algorithm 3.5. *The second proposed algorithm to approximate a curve within a certain error bound* ϵ.

Step 1 *Start with interpolating the points with a degree 0 NURBS rational curve.*

Step 2 *Increase the degree of the curve to the desired degree.*

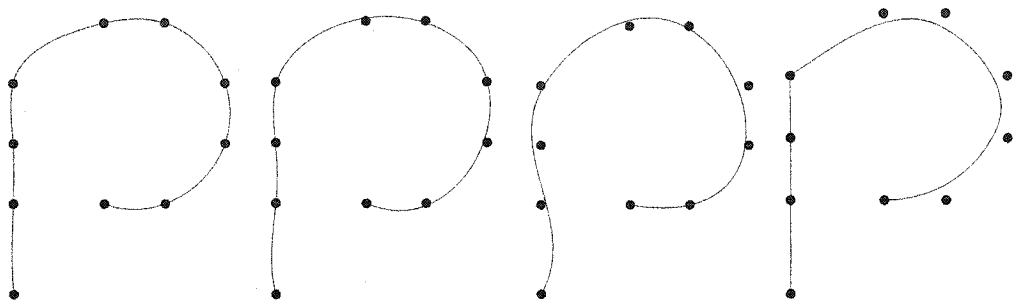
Step 3 *Use Algorithm 3.2 to remove as many control points as possible within the error bound* ϵ.

3.5.1 Comparison with Interpolation

Fitting a set of data points with interpolation or approximation yields different results. This section shows how those two techniques behave when fitting a small set and a large set of data points. For completeness, results of fitting the data set with a least square fitting method are also shown.

The points in Figure 3.7 are hard to fit since they are part of a straight line followed by a curved segment. Only the approximation method can capture the cusp between the straight segment and the curve segment.

The points in Figure 3.8 are close to each other. This is where the approximation method works best. It captures all the points and doesn't over-smooth



(a) Global Interpolation.

(b) Least Square fitting with 8 control points.

(c) Least Square fitting with 5 control points.

(d) Global Approximation with an accepted error of 10.

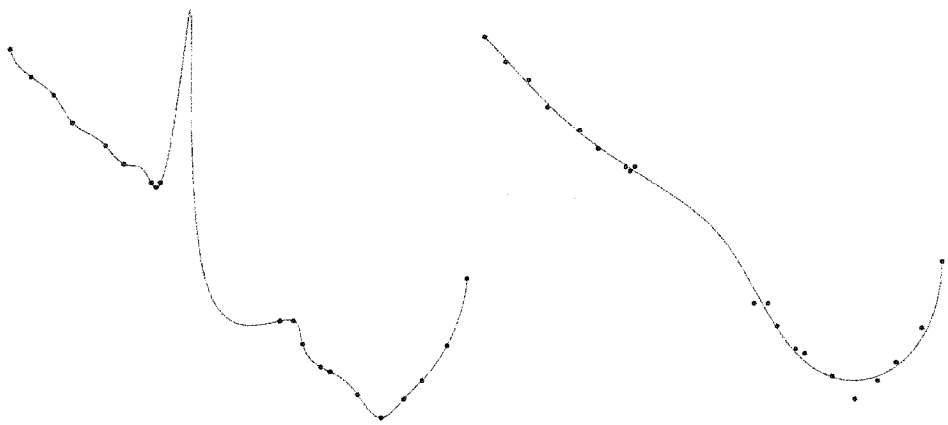
Figure 3.7: Point fitting functions.

the curve like the least square method. The interpolation method introduces instabilities and is impractical in those cases. The images that will be processed in this thesis all fall in this category due to the projection of a colour grid onto the observed scene. This is why the approximation method will be preferred as the mean to fit the data points to a NURBS curve.

3.6 NURBS Surface Generation

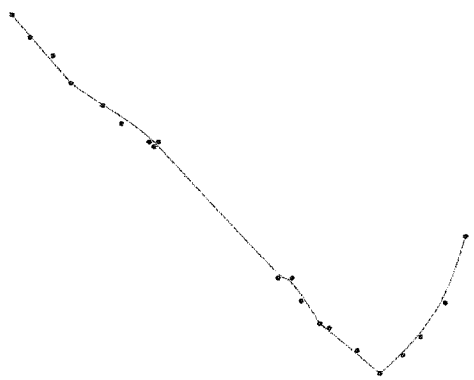
The techniques of interpolation and approximation for NURBS curves can be extended to the case of NURBS surfaces. NURBS surface generation offers more possibilities than the algorithms used for curve interpolation and approximation.

The interpolation of a bidirectional curve network is important for creating a 3D object surface. Its algorithm relies on two other algorithms: the NURBS global surface interpolation and the NURBS surface skinning. Both will be described



(a) Global interpolation method (20 control points).

(b) Least square fitting method with 8 control points.



(c) Approximation within error bound method using 0.1 as the error bound (21 control points).

Figure 3.8: Fitting points with a NURBS curve with (a) global interpolation, (b) least square fitting and (c) approximation within error bound.

prior to explaining the bidirectional curve network interpolation algorithm.

3.6.1 NURBS Surface Interpolation

It is possible to generalize the global curve interpolation technique to a global surface interpolation technique. Let $\{Q_{k,l}\}, k = 0, \ldots, n$ and $l = 0, \ldots, m$ be a set of $(n+1) \times (m+1)$ data points. A (p,q)th-degree B-spline surface which interpolates through these points is constructed by solving the following equation.

$$Q_{k,l} = S(\overline{u}_k, \overline{v}_l) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(\overline{u}_k) N_{j,q}(\overline{v}_l) P_{i,j} \tag{3.32}$$

Reasonable values for $(\overline{u}_k, \overline{v}_l)$ and the knot vectors U and V need to be computed. The $\overline{u}_k$ and $\overline{v}_l$ are computed using the same algorithm. Therefore only the methodology for computing the $\overline{u}_k$ is given. There are different possible methods to compute $\overline{u}_k$ and one of the most widely used is as follows: equation 3.13 is used to compute parameters $\overline{u}_0^l, \ldots, \overline{u}_n^l$ for each l and then, each $\overline{u}_k$ is obtained by the average across all $\overline{u}_k, l = 0, \ldots, m$. This corresponds to

$$\overline{u}_k = \frac{1}{m+1} \sum_{l=0}^{m} \overline{u}_k^l \qquad k = 0, \ldots, n$$

With the $(\overline{u}_k, \overline{v}_l)$ computed, the knot vector U and V can be obtained by using equation 3.15. The control points of $S(u,v)$ can be obtained as $(n+1) \times (m+1)$ linear equations in the unknown $P_{i,j}$. However, since $S(u,v)$ is a tensor product, the $P_{i,j}$ can be obtained more simply and efficiently as a sequence of curve interpolations. Where l is fixed, equation 3.32 becomes:

$$Q_{k,l} = \sum_{i=0}^{n} N_{i,p}(\overline{u}_k) \left(\sum_{j=0}^{m} N_{j,q}(\overline{v}_l) P_{i,j} \right) = \sum_{i=0}^{n} N_{i,p}(\overline{u}_k) R_{i,l} \tag{3.33}$$

where

$$R_{i,l} = \sum_{j=0}^{m} N_{j,q}(\overline{v}_l) P_{i,j} \tag{3.34}$$

Equation 3.33 is just a curve interpolation through the points $Q_{k,l}$, $k = 0, \ldots, n$. The $R_{i,l}$ are the control points of the isoparametric curve on $S(u, v)$ at fixed $v = \overline{v}_l$. When i is fixed and l is variable, equation 3.34 is a curve interpolation through the points $R_{i,0}, \ldots, R_{i,m}$, with $P_{i,0}, \ldots, P_{i,m}$ as the computed control points. This algorithm is symmetric and the same surface can be obtained instead by fixing k first.

3.6.2 Skinning

Skinning was previously referred to as *lofting*. Skinning is a process of blending the section curves together to form a surface. The section curves are given by a set of curves $\{C_k(u)\}$ defined for $k = 0, \ldots, K$. The blend direction is the v direction and the skinning method interpolates through the $C_k(u)$ curves. The $C_k(u)$ curves end up as isoparametric curves on the resulting skinned surface.

Based on B-splines, Tiller [72] defines skinning as follows. Let

$$C_k^w(u) = \sum_{i=0}^{n} N_{i,p}(u) P_{i,k}^w \qquad k = 0, \ldots, K \tag{3.35}$$

be the rational or non-rational section curves, and assume that all the curves are defined on the same knot vector U with common degree p. For the v direction a degree q is chosen as well as parameters $\{\overline{v}_k\}, k = 0, \ldots, K$.

A knot vector V is then computed. This information is used to do $n + 1$ curve interpolations across the control points of the section curves, yielding the control points $Q_{i,j}^w$ of the skinned surface.

The degree q is arbitrary as long as the relationship $q \leq K$ holds. The parameters $\{\overline{v}_k\}$ are computed by averaging the chord length for all the curves. This means

$$\begin{aligned} d_i &= \sum_{k=1}^{K} |P_{i,k} - P_{i,k-1}| \\ \overline{v}_0 &= 0 \qquad \overline{v}_K = 1 \\ \overline{v}_k &= \overline{v}_{k-1} + \frac{1}{n+1} \sum_{i=0}^{n} \frac{|P_{i,k} - P_{i,k-1}|}{d_i} \qquad k = 1, \ldots, K-1 \end{aligned} \tag{3.36}$$

and the knot vector V is computed using equation 3.15.

It is possible that the curves do not have the same knot vector nor the same degree. For those cases, performing degree elevation, knot insertion and knot refinement becomes necessary.

3.6.3 Bidirectional Curve Network Interpolation

The work of this thesis relies on the projection of a pseudo random encoded colour grid on an object to perform the 3D recreation of the object. This grid creates a network of NURBS curves. Gordon [24] created an algorithm that takes a network of NURBS curves and creates a NURBS surface from it.

Definition 3.9. *Let*

$$C_k(u) = \sum_{i=0}^{n} N_{i,p}(u) P_{k,i} \qquad k = 0, \ldots, r \quad u \in [0,1]$$
$$C_l(v) = \sum_{j=0}^{m} N_{j,q}(u) P_{l,j} \qquad l = 0, \ldots, s \quad v \in [0,1] \tag{3.37}$$

be two sets of non-rational B-spline curves. The two sets of curves are defined to be compatible if they satisfy the following two compatibility conditions:

- *Each set is compatible in the B-spline sense: all the $C_k(u)$ are defined on a common knot vector U^C and all the $C_l(v)$ are defined on a common knot vector V^C.*
- *There exists parameters $0 = u_0 < u_1 < \cdots < u_{s-1} < u_s = 1$ and $0 = v_0 < v_1 < \cdots < v_{r-1} < v_r = 1$ such that*

$$Q_{l,k} = C_k(u_l) = C_l(v_k) \qquad k = 0, \ldots, r \qquad l = 0, \ldots, s \tag{3.38}$$

The second condition implies that each curve of each set is intersected by all the other curves from the other set.

The Gordon algorithm consists of the interpolation of a bidirectional curve network. A NURBS surface is created such that it interpolates the two sets of compatible non-rational curves, that is

$$S(u_l, v) = C_l(v) \qquad l = 0, \ldots, s$$
$$S(u, v_k) = C_k(u) \qquad k = 0, \ldots, r \tag{3.39}$$

This interpolation is limited to the case of non-rational curves. In theory, rational curves could be used, but it was found [58] to be not practical as the 3-D

results obtained could be unpredictable. The technique that satisfies 3.39 was developed by Gordon when working at General Motors Corporation in the late 1960s and bears his name: *Gordon Surfaces* [24]. Gordon showed that $\{\phi_l(u)\}_{l=0}^{s}$ and $\{\psi_k(v)\}_{k=0}^{r}$ are two sets of blending functions satisfying

$$\phi_l(u_i) = \begin{cases} 0 & \text{if } l \neq i \\ 1 & \text{if } l = i \end{cases}$$
$$\psi_k(v_i) = \begin{cases} 0 & \text{if } k \neq i \\ 1 & \text{if } k = i \end{cases} \tag{3.40}$$

and that the surface satisfying 3.39 is then given by

$$\begin{aligned} S(u,v) &= \sum_{l=0}^{s} C_l(v)\phi_l(u) + \sum_{k=0}^{r} C_k(u)\psi_k(v) - \sum_{l=0}^{s}\sum_{k=0}^{r} Q_{l,k}\phi_l(u)\psi_k(v) \\ &= L_1(u,v) + L_2(u,v) - T(u,v) \end{aligned} \tag{3.41}$$

The resulting surface is composed of three simpler surfaces: two lofted surfaces $L_1(u,v)$ and $L_2(u,v)$ and the tensor product $T(u,v)$. Creating a NURBS representation of $S(u,v)$ is simplified as $L_1(u,v)$, $L_2(u,v)$ and $T(u,v)$ can easily be made compatible in the B-Spline sense. Because of this compatibility, they can be added and subtracted by applying the corresponding operations to their control points.

This means that $P_{i,j}$ of $S(u,v)$ are computed by

$$P_{i,j} = P_{i,j}^{L_1} + P_{i,j}^{L_2} - P_{i,j}^{T} \tag{3.42}$$

where $P_{i,j}^{L_1}$, $P_{i,j}^{L_2}$ and $P_{i,j}^{T}$ are the control points of $L_1(u,v)$, $L_2(u,v)$ and $T(u,v)$.

The first two surfaces are skinned surfaces and the latter is obtained by interpolating the points $Q_{l,k}$. The choice of parameters and knots for the skinning and point interpolation processes is determined by the given parameters $\{u_l\}$, $\{v_k\}$ and the knots U^C and V^C.

3.7 Conclusion

NURBS have useful properties which make them very suitable for the 3D object reconstruction problem. Different methods can be used with NURBS to represent a line based on a set of points: global or local interpolation, global approximation, etc. An efficient means of representing a line is what drives the precision of the algorithm for 3D object reconstruction presented in this thesis. Once 3D lines are obtained, a bidirectional curve network interpolation is performed to generate a NURBS surface.

The mathematics behind NURBS curves and surfaces is non-trivial. A NURBS library was created by the author to implement all the algorithms described in this chapter. This library is created in C++ and has around 18 000 lines of code. These numbers also include a matrix library to represent the control points and a numerical computation library to solve the linear equations systems required to perform interpolation.

Chapter 4

High Level Approach to the 3D Object Reconstruction Problem

This thesis proposes a new way to solve the 3-D object reconstruction problem. An overview of this approach is provided in this chapter. The approach builds on the experience acquired by previous works [37] with structured light. The first section states the motivation for this work, which helps understand the rationale behind this novel approach. The second section provides an introduction to the algorithm proposed in this thesis.

4.1 Motivation

Previous work [37] used a mixed approach, where both passive and stereo fusion methods were used at the same time. The dynamic fusion methods consisted of illuminating the object with a PRBA encoded grid. That illuminated scene was captured by two cameras and a dynamic programming matching technique was

used to perform the 3-D reconstruction.

In that work, the structured light information was used to help the passive stereo fusion method establish a correspondence between the left and right images. A description of the structured light encoding used for that work is given in Section 5.1. The structured light was also used in the calibration step, and to create a coarse disparity map which helped the registration algorithm. The correspondence between the left and right image information was still done pixel by pixel. This approach limited the potential precision of the technique. To improve the quality of the 3-D reconstruction it was necessary to improve the quality of the information which could be generated by the structured light.

Improving the quality of the information provided by the structured light resulted in the current research. The first problem to solve was the encoding itself. It relied on the presence or the absence of a disk at the intersection points inside the grid. Computing to a great precision the center point of a disk or of an intersection could not be done reliably due to the deformations of the projected structured light when it intercepted a 3-D object. The second problem to solve was that only the intersection points on the grid were used for the 3-D reconstruction process. The remainder of the grid information was not used. Finally the usefulness of using a passive stereo fusion method was questionable as the projection of the grid created a high contrast image which removed most of the texture which could have been seen by the camera if the grid hadn't been projected on the object.

The current research has two key features which sets it apart from previous work and at the same time solves the two problems highlighted above. The first is the use of colour lines to encode the structured light, as opposed to the spatial cues: intersection or circle. The second is the use of NURBS curves to model the

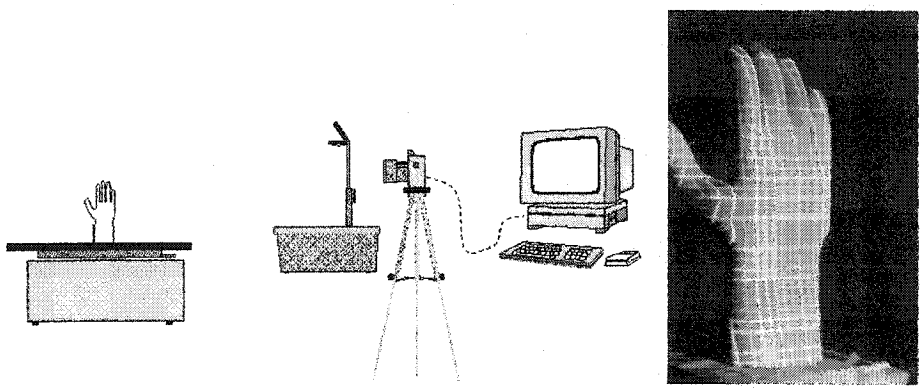
grid view from the camera.

The first has great benefits in terms of the usefulness of the structured grid as a means to capture the spatial dimensions of the object. The matching between the structured line and the camera view of the illuminated scene is not done area by area like previously.

The use of coloured lines to encode the grid has many benefits. The first is that the difficulty of finding the precise center of a circle or an intersection point is removed. Only the information defining the line itself is used. The line information near an intersection is not taken into account. The second is that there is no area matching required to label points from the camera. Instead, a whole line is labelled based on its colour and the colour of its left and right neighbours.

The use of NURBS curves to model the grid lines perceived by the camera greatly improves the precision of the 3D reconstruction. A line conveys more information than a single point. A line can be modelled by a series of dots connected with each other. It can also be modelled with a NURBS curve.

As described in Section 3.2, NURBS curves have interesting properties which make them ideal for modelling an object. One of those properties is their invariance under affine and projective transformation. This property enables the projection of the control points which make up the NURBS curve onto the plane of light projected by each grid line of the structured light. The intersection of the projection of the control points and the plane of light created by the projector yields a new control point. The set of those new control points creates a 3D NURBS curve which corresponds to the projection of the NURBS curve from the camera view on the 3D world coordinate system.



(a) (b)

Figure 4.1: a) The setup for the 3-D reconstruction of an object. b) A captured image from that setup.

4.2 The High Level Algorithm

The problem of reconstructing an object in 3-D is done with multiple steps. First, a PRMVS is generated and a distinct colour is assigned to each value. In the case of this thesis, five colours are used: magenta, red, green, yellow and cyan. The PRMVS is constructed such that three successive colours are always unique through the sequence. This sequence is applied to the X and Y axis of a grid to generate the pseudo random colour encoded grid that the projector uses to illuminate a scene.

An object of interest is put on a rotating table. The projector illuminates the object and the camera takes a snapshot of the scene. The computer then processes the information. This setup is depicted in Figure 4.1

The processing starts with finding where the coloured lines are. Once identified, each line is assigned to its colour, based on the hue of the pixels defining that

line. All lines are put into two groups: the horizontal and the vertical lines.

Within a group, the lines are properly ordered and assigned a temporary location. Once the left and right (or up and down) neighbours of each line are found, their true location inside the PRMVS is determined. This process assigns a true label to each line.

Each line has two edges and the edge points can be determined to subpixel accuracy. From the two sets of edge points, two NURBS curves are created for each grid line. Either interpolation or approximation is used (tests are needed to determine which solution is preferred) to create a NURBS curve based on the set of edge points.

The equation for the plane of light, which corresponds to the projection of the edge of the grid line on the 3-D world coordinate, is computed. Each control point from the NURBS curve is projected on that plane. This projection creates a new set of control points from which a NURBS curve is created in the 3-D world coordinate system.

All NURBS curves are projected on the 3-D world coordinate system. This creates two sets of parallel NURBS curves: horizontal and vertical curves. A NURBS surface is created by a bidirectional network interpolation of the NURBS curves.

The bidirectional network interpolation requires that the NURBS curves satisfy the compatibility condition. In practise, this implies that the network of curves must be split into compatible zones before a surface can be generated. Knot removal techniques are used to isolate zones of interest. A NURBS surface can then be generated for each zone of interest.

The projection on the 3-D world coordinate requires the calibration of the

camera and the projector. The calibration process used is the Tsai calibration method. This process requires the following steps. The first step is to put a test cylinder object which has a pseudo-random colour encoded grid painted on it and to take a picture of that cylinder with the camera. From the camera view, 2D NURBS curves are found along with their proper sequences in the PRMVS.

In Tsai's method a set of co-linear or non co-linear points to calibrate the camera properly is used. For the camera, the intersection points between each NURBS curve is found and mapped to their corresponding 3-D point on the cylinder. This gives a set of points which is fed to Tsai's non-linear optimization algorithm. After this step, the camera is calibrated.

The projector is calibrated by illuminating another known object and taking a picture of that scene with the camera. The object is a white plane. The set of 2D NURBS curves and their proper sequence in the PRMVS are determined based on the camera's view of that plane. The intersections of the NURBS curves yield a set of co-linear points which are fed to Tsai's algorithm. Once this is done, the projector is also calibrated properly.

4.3 Conclusion

The approach proposed in this thesis improves the quality of the reconstruction of an object in 3-D. It could be categorized as a dynamic feature based matching method. Unlike previous work, the feature is not a collection of edge points linked together but instead is the parametric representation of the underlying curve as seen by the camera. This approach can use subpixel accuracy and uses a robust calibration technique to preserve the precision of the algorithm when generating

points in the 3-D world coordinate system.

Chapter 5

Multidimensional Pseudo Random Encoding

This thesis work is based on using structured light to recreate three-dimensional objects. A technique must be used to create a relationship between the camera view of the scene and the structured light used to illuminate the scene. This relationship, in this thesis, is obtained by encoding the structured light with a technique known as PRMVS.

This technique grew from the body of work previously done with structured light [37]. This previous work encoded a grid using a technique called pseudo-random binary array (PRBA) encoding.

5.1 Pseudo-random Binary Encoding

For the 3-D reconstruction process one needs the information contained in the image by using a relationship between the structure being projected and its view

captured by a camera. The camera captures the projection of a structured light on an object. This view is limited to a small portion of the information being projected as most of the structured light is lost to the background.

The structured light can be projected without any form of encoding. In this situation, trying to establish a correspondence between the projector's information and the camera's capture of it is very difficult and can not be done "absolutely". "Absolutely" in this context means that a given point $P_{i,j}$ from the projector can be matched to point $C_{k,l}$ inside the image from the camera.

The structured light information can be encoded. The encoding will enable the creation of a correspondence between the camera's and the projector's views.

Definition 5.1. *A pseudo-random binary array (PRBA) is defined by a $n_1 \in \mathbb{N}$ by $n_2 \in \mathbb{N}$ binary array encoded using a pseudo-random sequence such that a k_1 by k_2 window sliding over the array is unique and fully identifies the window's absolute coordinate (i, j) within the array.*

The following relations hold for the PRBA [44].

$$\begin{aligned} 2^n - 1 &= 2^{k_1 k_2} - 1 \\ n_1 &= 2^{k_1} - 1 \\ n_2 &= \frac{2^n - 1}{n_1} \end{aligned} \tag{5.1}$$

where n_1 and n_2 must be relatively prime.

Definition 5.2. *Two integers are defined by Weisstein [80] as relatively prime if they share no common positive factors (divisors) except 1.*

A $2^n - 1$ PRBS is generated by the "primitive polynomials modulo 2" method [78]. Table 5.1 is based on results obtained by Watson [78] and gives the non-zero coefficients of some of the primitive polynomials used by the method up to order 30.

1,0	11,2,0	21, 2, 0
2,1,0	12,6,4,1,0	22,1,0
3,1,0	13,4,3,1,0	23,5,0
4,1,0	14,5,3,1,0	24,4,3,1,0
5,2,0	15,1,0	25,3,0
6,1,0	16,5,3,2,0	26,6,2,1,0
7,1,0	17,3,0	27,5,2,1,0
8,4,3,2,0	18,5,2,1,0	28,3,0
9,4,0	19,5,2,1,0	29,2,0
10,3,0	20,3,0	30,6,4,1,0

Table 5.1: Non-zero coefficients of some primitive polynomials

A primitive polynomial defines a recurrence relation to obtain a new random bit from the n preceding ones. This relation is guaranteed to produce a sequence of maximal length, *i.e.* to cycle through all possible sequences of n bits (except all zeros) before it repeats and gets $2^n - 1$ random bits before the entire sequence is repeated.

As previous work showed [37], the PRBA method for encoding a grid does not allow sub-pixel accuracy. The previous work projected the PRBA on a surface and then used stereoscopy methods to reconstruct the object in 3-D. However, there are problems in terms of accuracy when retrieving the intersection points and the disks centres. This problem was partially solved in previous work by using two cameras to observe the scene and only use the PRBA as a coarse disparity map between the two scene. The coarse disparity map was useful in the early stages of stereo reconstruction. The PRBA used in previous work allowed to combine the

strengths of structured light and stereoscopy.

5.2 Pseudo Random Multi-Variate Sequence

One method to solve the problems associated with a PRBA is to use multi-variate sequences for the X and Y axis to form the grid that will then be projected on an object. Both the PRBA the PRMVS are an extension of the PRBS method. Beside from allowing for a greater number of variables, the PRMVS extension can also be applied to more than one dimension. PRMVS are generated by using Galois field arithmetic.

Definition 5.3. *A Galois field is the set of number* $(0, 1, 2, \ldots, N-1)$ *where* N *is prime and on which operations such as addition or multiplication are done using modulo* N *arithmetic.*

Theorem 5.1 (Fermat). *Any number* n *drawn from the Galois field set* $(1, 2, \ldots, N-1)$ *has the following relationship* $n^{N-1} = 1$.

From the Fermat theorem, it is apparent that a number to an integral power is periodic with periodicity of $N-1$:

$$n^{m+(N-1)} = n^m \times n^{(N-1)} = n^m$$

The property which enable a Galois field to create a PRMVS is that any Galois field always contains at least one number (a) called a generator (or primitive root). This generator has the property that the set of numbers $(a^i | i = 0, \ldots, N-2)$ is a permutation of the set $(1, 2, \ldots, N-1)$. The sequence generated by the generator is periodic (from Fermat's theorem). So, for any integer (m), the set of numbers $(a^i | i = m, \ldots, m+N-2)$ is a permutation of the set $(1, 2, \ldots, N-1)$.

Theorem 5.2 (Petriu [55]). *A PRMVS can be used to map a two dimensional area by assigning each value of the Galois field to a unique colour*

One approach tested was to use the basic red, green and blue colours along with the combination of those basic colours: cyan, yellow, magenta and white. This yields seven colours which could be used to generate a PRMVS with a base of 7.

There is a practical limit to the number of colours a camera can perceive under varying lighting situations. The blue colour in particular was found to be very hard to register properly by the camera when doing practical tests. Furthermore, white does not have a hue associated with it and creating a reliable colour extractor is easier and more robust if the hue is used when performing the computations. Therefore instead of using the initial 7 colour scheme presented above, only the following colours were actually used: magenta, red, green, yellow and cyan. This combination allows to use a base 5 encoding for the PRMVS. An example with a sequence of 2 is shown in Figure 5.1.

The creation of a structured light pattern using a PRMVS is simple. The PRMVS is used to encode the X and Y axis. To retrieve a point on the structured light structure, one simply moves along the X axis until n data points are found, where n is the size of the viewing window. This will yield the X position. The same process is applied for the Y coordinate: one moves n data points along the Y axis to obtain the necessary information. With the information obtained, one then only needs to transfer the data points into values and look at a table lookup to find the coordinate.

To illustrate the overall process of retrieving a coordinate from the view of a projected grid, a PRMVS encoded colour grid was generated and projected on the

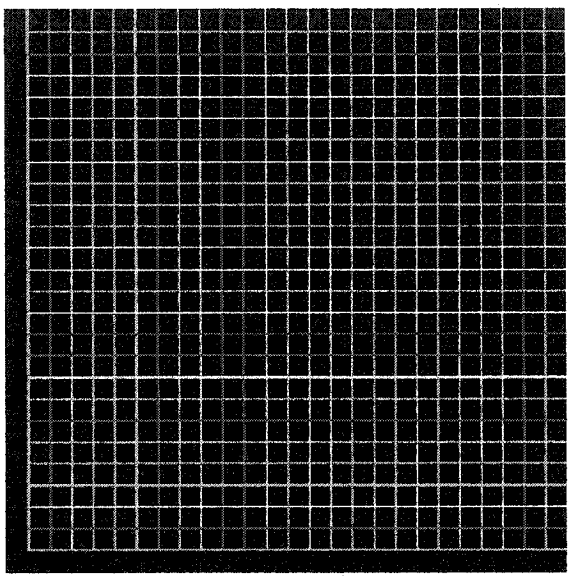
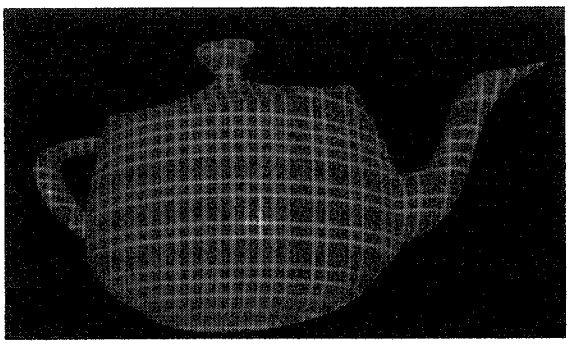


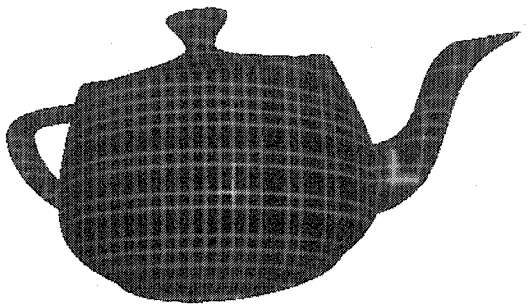
Figure 5.1: A grid encoded with a PRMVS of base 5 and window length of 2.

famous teapot object. The grid was generated by using a base of 5. Each colour was given a value: magenta (0), red (1), green (2), yellow (3) and cyan (4). The window size was set to 3 and the PRMVS was generated using the polynomial 3102 (expressed in base 5 and not in decimal). This gives a sequence of 125 numbers over which a window of size 3 can slide over without finding any repeating patterns. The sequence obtained is: 3031224113342112124302404101114010422344003244242301231341444314132203311023323201420403433304340221004412002021432131030001.

In Figure 5.2 the result of projecting the grid described previously is shown. Looking at the region near the beak and the body of the Utah teapot (highlighted in Figure 5.2b) one can see that to find the coordinate of that point one must find the location of the sequence yellow-red-cyan on the X axis and yellow-cyan-red



(a) Generated image with a structured light projected on the Utah teapot



(b) Background is removed and the area of interest near the beak is highlighted.

Figure 5.2: Utah teapot with a grid projected onto it.

on the Y axis. This corresponds to finding the sequence 314 and 341 on the PRMVS. Those sequences occur at position 61 and 55 respectively. That point is then labelled as being at $(61, 55)$ inside the structured light grid.

It should be noted that points will not be labelled directly by the process presented in this body of work. Instead, lines will be labelled. A line belongs to the vertical or horizontal set of lines that constitute a $X - Y$ coordinate grid. Because NURBS curves represent lines and not points, there is no need to limit ourselves to the location of intersections.

Figure 5.2 also shows that encoding can't solve all problems but can help spot problem areas. For instance, on the handle on the teapot, the vertical lines do not create a smooth sequence with the vertical lines from the body. This indicates the presence of a discontinuity. A discontinuity is also observed for the teapot's lid. There are discontinuities in the horizontal lines between the body and the lid of the teapot. The 3-D rendering process can use this information to adjust its

algorithm appropriately.

5.3 Conclusion

Previous work focused on the fusion of stereo and projected structured light methods. For this, the PRBA method was used with great efficiency. The PRBA creates a uniquely identified area which can easily be mapped to the structured light coordinates. In the stereo process, this allowed the left and right views to create a set of known points from which the matching process could begin, starting with the epipolarization process.

This new work focuses only on an active method (projecting a structured light grid). To this end, the PRMVS technique is used. By allowing the use of any base (in our case 5) to create a PRMVS, the size of the viewing window can be significantly reduced. In previous work, the viewing area was 5×2. Any problems in recognizing a disk or an intersection in that area would create a wrong label, *i.e.* the misplacement of a point. Now, a 3×1 area has to be looked at, once in the X direction and once in the Y direction, to obtain the desired information. The problem of mislabelling a point is greatly minimized because the process will label whole lines instead of labelling points directly.

Analyzing the line as a whole also solves some of the problems which arise when images are highly saturated. Projecting a grid on an object always creates a very strong probability that some of the areas of the image will be saturated.

Chapter 6

Line Detection

Line detection is a classical problem in computer vision [29, 34]. As such, this chapter will focus on the method which is used in this thesis to obtain the lines that form the basis of the 3-D recreation process. The proposed method uses the line detection process in a somewhat unusual way: the two edges of the line are used instead of the line itself. The rationale for this is further explained below.

6.1 Filtering

In a normal setting, filtering is used to limit the noise obtained when capturing the image. However, the edge operator selected for this thesis is very robust to noise. Further analysis is required to determine whether or not filtering is actually required to obtain accurate results. Preliminary work indicates that it may not be necessary to filter the input image to obtain accurate results in edge detection.

Filtering modifies the input image and may lead the edge detector astray. The high precision of the algorithm depends on the precision of the edge detection

of each grid line of the structured light. Tests will be performed to measure if median filtering or other standard noise cleaning methods can be used to improve the accuracy of the 3-D reconstruction process.

6.2 Line Detection

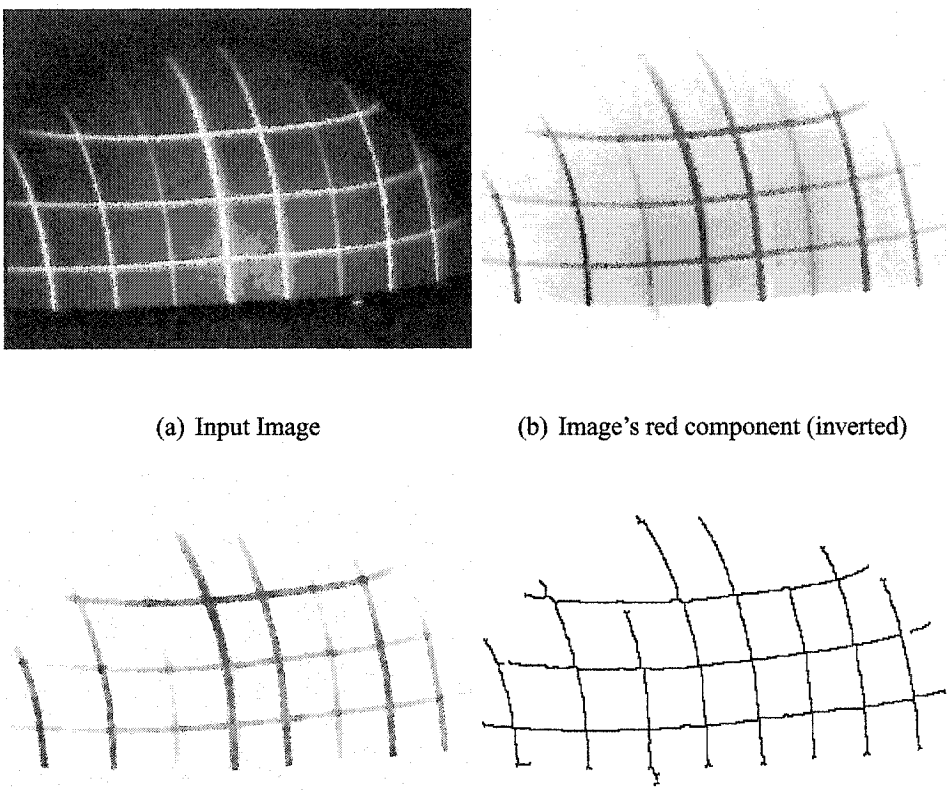
The methodology for 3-D object reconstruction presented in this thesis requires the generation of NURBS curves from the line information contained within the grid information projected by the projector. Although the line detection algorithm is ultimately not used in this thesis, it is important to understand why it was not utilized. Earlier work [38] did use this line detection algorithm but a better alternative was found later on. The earlier methodology for NURBS curves generation is described in this section.

The process of generating NURBS curves representing the lines projected on the surface requires three steps: extracting a coarse grid from the red, green and blue components of the image, determining a precise NURBS curve for each line and matching the line to its proper colour.

A coarse grid is extracted from its different colour components using mathematical morphology operators. The grid generated is coarse due to the nature of mathematical morphology. Since each colour component has a different saturation and reflectivity value, it is individually processed.

The first step in extracting the coarse grid is to obtain an image without highlights from the projector. The second step is to determine where the object is located in the image. This ensures that a grid will not be found outside of the illuminated object. From this information, a grid is extracted using dynamic thresh-

olding and morphological operators. These steps are depicted in Figure 6.1.



(a) Input Image (b) Image's red component (inverted)

(c) Projector's highlights are removed (d) Coarse grid over the object is determined.

Figure 6.1: Extracting a coarse grid from the image's red component.

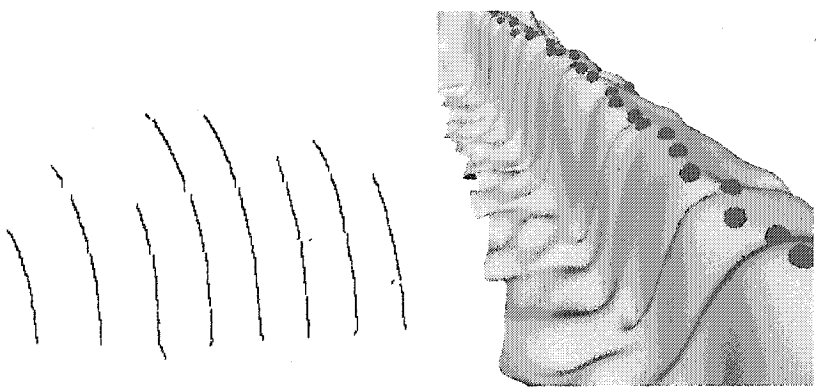
One colour component can not fully represent the grid seen in the colour image. However, merging the grids obtained from all the colour components yields a grid which is close to the real grid projected on it. A merged grid is shown in Figure 6.2(a) The intensity level of the image near this coarse grid is viewed as a 3-D landscape. Maximum points on this landscape are then used to find the pre-

cise grid. The landscape is processed line by line, starting with the vertical ones and continuing with the horizontal ones.

A NURBS surface interpolates the landscape for each line and the maximum points are found along that surface (Figure 6.2(b)). The maximum points found with this scheme are not necessarily at a pixel location: they can also be at an inter-pixel location, thus giving the line a sub pixel accuracy. Due to the interpolation, some of the maximum points found will not correspond to the optimal location of a point in a grid line.

Definition 6.1. *Outliers are points which are not valid and should not be used to compute results.*

To remove outliers, a Monte Carlo technique is used along with a bucketing technique [40, 86]. At this step, mathematical morphology can not be used to eliminate outliers since it can not operate at sub-pixel accuracy.



(a) (b)

Figure 6.2: (a) The grid merged from all the colour components and (b) the landscape for one of the lines.

6.2.1 Finding Outliers

The least median of squares (LMedS) method gives the necessary information to determine which points are outliers.

Definition 6.2. *Let f_i be the result of applying the function f to the data point i. Then the least median of squares (LMedS) method is defined as*

$$\min med_i\, f_i^2$$

This must be solved by a search in the space of possible estimates generated from the data. For practical reasons, only a randomly chosen subset of data can be analyzed. The subset of data is obtained from the maximum points on the landscape for each line. A Monte Carlo type technique is used to draw m random subsamples of p different points for each line.

For each subsample, a NURBS curve is interpolated and the median of the distance between each point on the line and that curve is computed. This yields a median value M_J

$$M_J = \text{med}_{i=1\ldots n} |\min(C_J(u) - p_i)|^2$$

The M_J that is minimal for all m M_J's indicates which NURBS curve is a valid interpolation of the line.

A subsample is "good" if it consists of p good correspondences. Assuming that the whole set of correspondences may contain up to a fraction ϵ of outliers, the probability that at least one of the m subsamples is good is given by

$$P = 1 - [1 - (1 - \epsilon)^p]^m \tag{6.1}$$

By requiring that P must be close to 1, we can determine the value of m given values for p and ϵ.

The LMedS method is poor in presence of Gaussian noise. To compensate for this deficiency, a weighted least-square procedure is used. The *robust standard deviation* estimate is given by

$$\hat{\sigma} = 1.4826[1 + \frac{5}{n-p}]\sqrt{M_J} \tag{6.2}$$

where M_J is the minimal median. Using σ, we can assign a weight for each correspondence:

$$w_i = \begin{cases} 1 & \text{if } d_i^2 \leq (2.5\hat{\sigma})^2 \\ 0 & \text{otherwise} \end{cases} \tag{6.3}$$

If the weight is set to 0, then the point is an outliers, otherwise it is not. A new curve is interpolated by using only the valid points.

As mentioned above, a sample is "good" if it consists of p good correspondences. Choosing random data points may yield a sample with points being too close together and thus represent an invalid curve. A bucketing technique [40, 86] will spread the points such that it minimizes the occurrence of such "bad" samples.

The bucketing technique creates b buckets to which a set of consecutive data points of equal length is attached. To generate a set of p points, p buckets are randomly chosen, as well as a point inside each bucket. Since the data point must cover the length of the curve being analyzed, the first and last buckets are always chosen.

6.2.2 Line Detection Issues

Figure 6.3 shows the theoretical line intensity when it leaves the projector. It also shows a cross-section of an image captured by camera showing two grid lines. The cross-section shows that a grid line as perceived by the camera does not

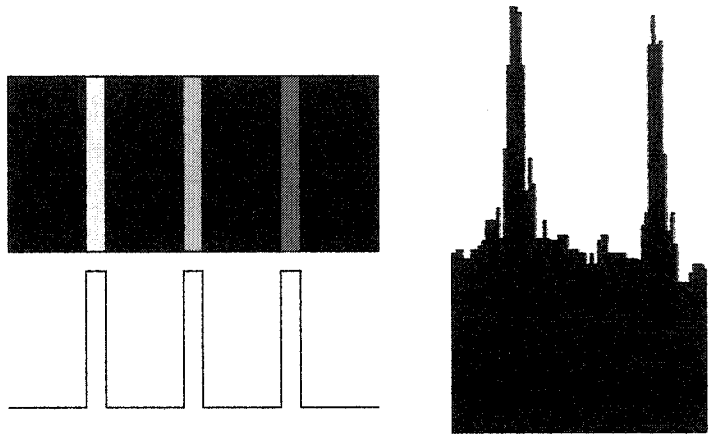


Figure 6.3: *a)* the theoretical coloured lines intensity from the projector *b)* a cross-section of the input image as captured by the camera

correspond to a parabola. A parabola is required by the line detection algorithm above to obtain sub pixel accuracy. The combination of camera and projector lens distortions caused the image to be perceived not as a double step edge, but instead as the image depicted in the figure.

This explains why the technique described above to perform line detection includes a method to remove outliers. The method is statistically correct, but it is preferable to use a method which would not rely on the elimination of outliers to properly and accurately detect a line.

6.3 Double Edge Detection

This work tries to rely on methods which are easy to implement and also have great precision. Edge detection has great precision; line detection does not. Since

a line is made of two edges, we can define it by the location of its edges.

The location of the edges is easier to compute and leads to better results when trying to recreate a three dimensional object. One of the premises of line detection is that the center of the line is at the same place as the center being viewed. This premise does not hold when viewing a 3-D object illuminated by a grid.

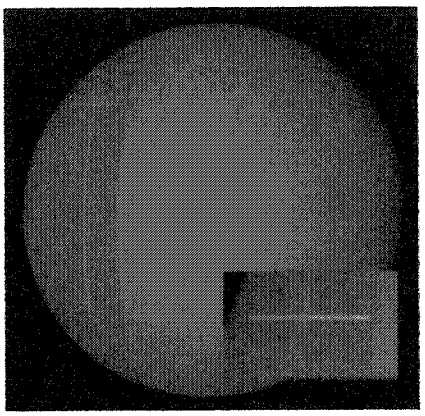


Figure 6.4: Blue line projected on a sphere. The white line shows the center.

A very simple example with a line illuminating a basic sphere object will illustrate the problem of detecting the center of a line with precision. In this example, depicted in Figure 6.4, a wide blue line with a thin white line in its center was projected on a sphere. A small green line was drawn over to illustrate a problem area for finding the true center of the line. There are 19 pixels to the left of the white center line and 23 to the right, in the small test region.

This example amplifies the problem so that it can be viewed with the naked eye. This problem is always amplified in practise because subpixel accuracy can be achieved with edge detection but not with line center detection.

In some settings one cannot use the two edges to describe a line. However, in the setup proposed in this thesis there is no such limitation.

6.3.1 Subpixel Edge Detection

The precision of the 3-D reconstruction relies on edge detection to sub pixel accuracy. A type of edge detector which can work to sub pixel accuracy are the zero-crossing edge detectors. These edge detectors do not place an edge at locations of high gradient, but at locations of spatial gradient maxima. A pixel is marked as an edge pixel, if in its immediate area, there is a zero crossing having negative slope of the second directional derivative taken in the direction of the gradient ([30]).

The integrated directional derivative gradient operator [87, 88] enables a more accurate calculation of step edge direction than do techniques that use values of directional derivatives estimated for a point. The directional derivative can be computed from the knowledge of the directional derivative D_1 and D_2. The gradient magnitude is computed as $\sqrt{D_1^2 + D_2^2}$ and its direction as $\arctan(D_2/D_1)$.

The Sobel [70] edge detector computes the directional derivative at a point. It estimates the edge direction for step edges using a function of the true edge direction and the displacement from the pixel's center. This is a problem as the step edge does not match the polynomial model, which in turn creates a bias in the computation of the directional derivative. The integrated directional derivative gradient operator computes the first directional derivative taken over a square area.

The noise sensitivity of that operator is comparable to the Sobel operator and, unlike the Sobel operator, increasing the neighbourhood size decreases both estimate biases and noise sensitivity. For the proposed 3-D reconstruction technique,

-3	-348	-555	-624	-555	-348	-3
-142	-372	-510	- 556	-510	-372	-142
-113	-228	-297	-320	-297	-228	-113
0	0	0	0	0	0	0
113	228	297	320	297	228	113
142	372	510	556	510	372	142
3	348	555	624	555	348	3

Table 6.1: The row derivative mask for integrated directional derivative gradient operator on a 7×7 neighbourhood size.

the 7×7 neighbourhood size for the integrated directional derivative gradient operator is used. The row derivative mask values are shown in Table 6.1, and the column derivative mask is obtained by rotating the row mask by $90\,°$.

With the integrated directional derivative gradient edge detector operator, it is possible to compute the edge location with subpixel accuracy. To this end, linear, quadratic or cubic fitting can be used to compute the zero-crossing value of the gradient within subpixel accuracy; this yields the location of the edge of the line. It has been shown [16] that for a step edge, the quadratic fitting yields the proper result.

Figure 6.5 shows an example of subpixel edge detection. This example was created using the procedure described below:

Step 1 Apply the integrated directional derivative gradient operator with a kernel of size 7×7 to the x and y directions.

Step 2 Use quadratic fitting to obtain subpixel accuracy for the location of the edge

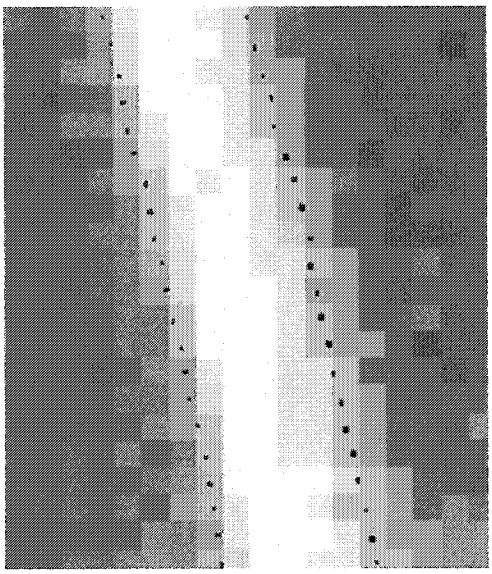


Figure 6.5: Subpixel edge detection near a line

pixels.

The subpixel location is computed for both the column and the row direction. Inside that figure, the edge point is shown as a black dot inside a pixel.

6.4 Line Detection Algorithm

Once edge points are found, the system still needs to interpret the edge data and assign them to their lines. The following algorithm is used to do this. This algorithm is first applied to the horizontal edge points to create a list of horizontal lines. Then it is applied to the vertical edge points to generate a list of vertical lines.

Algorithm 6.1. *The line detection algorithm consists of the following steps:*

Step 1 *Label the connected segments using an iterative labelling algorithm [29].*

Step 2 *Each segment is marked as an up edge or a down edge. The up edge is associated with a positive gradient value and the down edge with a negative gradient value.*

Step 3 *Merge segments that belong with each other by the following rules:*

1. *A segment's end must be close to the other segment's start. Close in this context is a function of the distance between grid lines.*
2. *The segments must both be of the same type. Both have an up edge or a down edge.*

Step 4 *Remove segments if they don't have a supporting segment. A supporting segment in the case of an up edge segment is a down edge segment that is close to the segment. An up edge supporting segment is associated with a down edge segment.*

Step 5 *Estimate the colour of the edge lines by looking at a minimum of 10 points along each grid line. The colour of a grid line is estimated by checking the hue value of some of the pixels inside that grid line.*

6.5 Conclusion

This chapter described the problems associated with center line detection. These problems prompted the author of this thesis to use an alternative method which would be less computationally intensive and more precise. The use of edge lines instead of center lines is thus proposed to solve most of the problems of center

line detection. It gives us the ability to compute edge point locations to subpixel accuracy and thus improve the 3-D recreation process.

The use of the edge line instead of the center line improves the accuracy of the 3-D reconstruction process. One of the premises for illuminating an object with a line is that the illumination creates a plane of light in 3-D. The object intersects this plane. That premise is changed by the use of the two edge lines instead of a single line. It implies that two planes instead of one plane are projected for each grid line.

Chapter 7

A NURBS Based Methodology for 3-D Reconstruction

There is still no true answer when trying to infer the 3-D location of a point based on the information contained within an image. The precision of the reconstruction depends on different factors. One of these factors is the camera model. Historically, complex camera models were computationally intensive and therefore not used. Today's state of computational power allows one to use more accurate representation of the camera model in order to reconstruct in 3-D a scene as viewed by a camera.

7.1 The Camera Model

The camera model chosen was influenced by the calibration method selected. Tsai [75] has created an efficient method to calibrate a camera and because it is the approach used for the calibration process, it is also the camera model chosen.

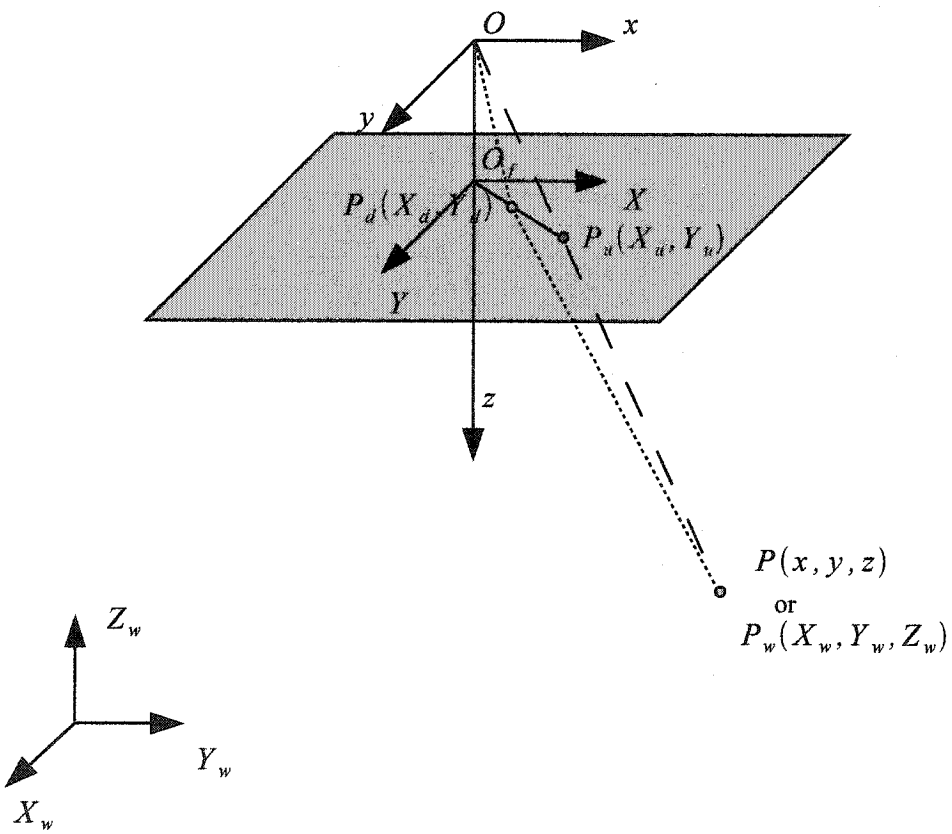


Figure 7.1: Tsai's camera model.

The model is the famous pinhole model with the radial lens distortion taken into account [75]. The model is depicted in Figure 7.1. The following notation is used inside that figure:

$P_w(X_w, Y_w, Z_w) \in \mathbb{R}^3$ is the 3-D world coordinate of the object point P.

$P(x, y, z) \in \mathbb{R}^3$ is the point P in the 3-D camera coordinate system.

$O \in \mathbb{R}^3$ is the origin of the 3-D camera coordinate system.

z axis is the z axis of the camera coordinate system, which is the same as the optical axis.

$(X, Y) \in \mathbb{R}^2$ is the image coordinate system centred at O. It is the intersection of the optical axis z and the front image plane. It is parallel to the x and y axes.

$O_f \in \mathbb{R}$ is the focal distance between the front image plane and the optical center O.

$(X_u, Y_u) \in \mathbb{R}^2$ is the image coordinate of the point $P(x, y, z)$ if a perfect pinhole camera is used.

$(X_d, Y_d) \in \mathbb{R}^2$ is the actual image coordinate of the point $P(x, y, z)$. It differs from (X_u, Y_u) due to lens distortion.

$(X_f, Y_f) \in \mathbb{R}^2$ is the pixel location of the point (X_d, Y_d). This point is not shown in the graphic. The origin for the pixels is not centred on the image, but is usually at the lower left corner of the image.

The camera model is described by the set of equations used to transform a point residing in the 3-D world coordinates (x_w, y_w, z_w) to a point (X_f, Y_f) that is the pixel location of that point on the image obtained by the camera. This transformation requires four steps:

1. Rigid body transformation from the object world coordinate system (x_w, y_w, z_w) to the camera 3-D coordinate system (x, y, z)

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \tag{7.1}$$

where T is the translation vector

$$T \equiv \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{7.2}$$

and R is the 3×3 rotation matrix

$$R \equiv \begin{bmatrix} \cos\phi\cos\kappa & \sin\omega\sin\phi\cos\kappa + \cos\omega\sin\kappa & -\cos\omega\sin\phi\cos\kappa \\ -\cos\phi\sin\kappa & -\sin\omega\sin\phi\sin\kappa + \cos\omega\cos\kappa & \cos\omega\sin\phi\sin\kappa + \sin\omega\cos\kappa \\ \sin\phi & -\sin\omega\cos\phi & \cos\omega\cos\phi \end{bmatrix} \tag{7.3}$$

The equation above uses the pan, swing and tilt notation often used in the photogrammetric literature. This notation corresponds to the following sequence of rotations:

(a) A clockwise rotation of κ around the z-axis (swing).

(b) A clockwise rotation of ϕ around the y-axis (pan).

(c) A clockwise rotation of ω around the x-axis (tilt).

To avoid long trigonometric expressions, the following convention is used to describe the elements of R

$$R \equiv \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

where

$$r_{11} = \cos\phi\cos\kappa \tag{7.4}$$

$$r_{12} = \sin\omega\sin\phi\cos\kappa + \cos\omega\sin\kappa \tag{7.5}$$

$$r_{13} = -\cos\omega\sin\phi\cos\kappa \tag{7.6}$$

$$r_{21} = -\cos\phi\sin\kappa \tag{7.7}$$

$$r_{22} = -\sin\omega\sin\phi\sin\kappa + \cos\omega\cos\kappa \tag{7.8}$$

$$r_{23} = \cos\omega\sin\phi\sin\kappa + \sin\omega\cos\kappa \tag{7.9}$$

$$r_{31} = \sin\phi \tag{7.10}$$

$$r_{32} = -\sin\omega\cos\phi \tag{7.11}$$

$$r_{33} = \cos\omega\cos\phi \tag{7.12}$$

Note that the rigid body transformation from one Cartesian coordinate system, *i.e.* (x_w, y_w, z_w), to another, *i.e.* (x, y, z), is unique, if the transformation is defined as a 3-D rotation around the origin followed by the 3-D translation. Some techniques define the transformation as a translation, followed by a rotation.

2. Transformation from 3-D camera coordinate (x, y, z) to ideal (undistorted)

image coordinate (X_u, Y_u) using perspective projection with pinhole camera geometry:

$$X_u = f\frac{x}{z} \tag{7.13a}$$

$$Y_u = f\frac{y}{z} \tag{7.13b}$$

The pinhole model is the most often used camera model in the literature. It lacks the radial lens distortion described next.

3. Radial lens distortion is defined by:

$$X_u = X_d + D_x \tag{7.14a}$$

$$Y_u = Y_d + D_y \tag{7.14b}$$

where (X_d, Y_d) is the distorted or true image coordinate on the image plane and

$$D_x = X_d(\kappa_1 r^2 + \kappa_2 r^4 + \cdots) \tag{7.15a}$$

$$D_y = Y_d(\kappa_1 r^2 + \kappa_2 r^4 + \cdots) \tag{7.15b}$$

$$r = \sqrt{X_d^2 + Y_d^2} \tag{7.15c}$$

and κ_i are the distortion coefficients.

The are two kinds of distortion: radial and tangential. For each kind of distortion an infinite series is required. However, Tsai's experience shows that for industrial machine vision application, only radial distortion needs to be considered, and only one term is needed from the series. Any more elaborate modelling not only would not help but also would cause numerical instability in the calibration process. The effect of radial distortion on projected lines is shown in Figure 7.2.

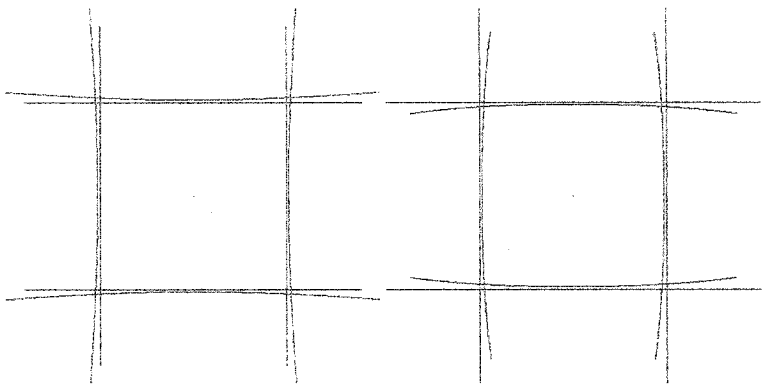


Figure 7.2: Effect of radial distortion on projected lines. The variable κ_1 is positive in the left image and negative in the right one.

4. The transformation from the real image coordinate (X_d, Y_d) to the computer image coordinate (X_f, Y_f) transformation is defined as

$$X_f = s_x d'^{-1}_x X_d + C_x \tag{7.16a}$$

$$Y_f = d_y^{-1} Y_d + C_y \tag{7.16b}$$

$$d'_x = d_x \frac{N_{cx}}{N_{fx}} \tag{7.16c}$$

where

(X_f, Y_f) is the row and column numbers of the image pixel in computer frame memory

(C_x, C_y) is the row and column numbers of the center of the computer frame memory

d_x is the center to center distance between adjacent CCD sensor elements in X (scan line) direction.

d_y is the center to center distance between adjacent CCD sensor elements in the Y direction.

N_{cx} is the number of sensor elements in the X direction

N_{fx} is the number of pixels in a line as sampled by the computer

s_x is the uncertainty image scale factor

These four steps will assign a world coordinate point (x, y, z) to the pixel of the image captured by the camera.

The calibration process described by Tsai uses non linear search techniques to find values for the parameters. Some values, like the number of pixels, have to be provided.

In thick lens cameras, there are two focal lengths: the front focal length and the back focal length. With a thin lens, but focal lengths are the same but this is not the case with a thick lens camera. Tsai's model fails to consider this situation. However, in a controlled environment, it is possible to minimize the effect of a thick lens by positioning the object of interest at a proper distance from the camera and by capturing an image with good photographic lenses.

For a real camera, another type of distortion which can be accounted for is the depth of field distortion. The depth of field defines the area in front and behind a focused subject in which the image taken with the camera appears sharp. Outside the depth of field, the image will appear blurry. The depth of field varies according to the lens' focal length, aperture value and shooting distance. It is therefore possible to capture an object for 3-D reconstruction in such a way that it will reside within the depth of field. This will remove the distortions caused by analyzing picture pixels residing outside the depth of field.

7.2 The Projector Model

The projector model is essentially the same as the camera model. The grid acts as a CCD array and the projector, like the camera, adds radial distortion to the simple pinhole model. The difference lies in the expression of the pixel for the projector.

There are actually two kinds of projectors that can be used: the standard passive projector and the newer active projectors. The standard passive projector works by having a bright light source illuminating a sheet of plastic with the structured light information on it, then a 45 degree mirror and a lens refocuses that light to the desired location.

For that kind of projector, since it is working with a printed grid, a meaningful value for a pixel might be based on the dots per inch (dpi) of the printed material. Usually, 300 dpi is the maximum effective resolution of a low cost ink-jet printer. There are higher precision printers than this, however in the case of an ink jet, an ink droplet can only be so small. Depending on the quality of the printer, one may be able to get true 300 dpi resolution. If this is the case then setting the grid image at 300 dpi resolution serves a double role. The first is to get a grid for the projector, the other is that the image used to generate the grid can also be viewed as the image of the camera model.

The other kind of projector is dynamic. This projector is basically a computer monitor that can project its image. The resolution is usually set to 800×600, but some models are more precise. For those projectors, the pixel location is simply the location of a pixel on the 800×600 display. Digital Light Processing (DLP) projector also offer great advantages and other research avenues: dynamically moving the grid to take a series of snapshot for example.

The mathematical model for the projector is the same as the one used for the

camera. It is described in the previous Section.

7.3 Generation of a Colour Encoded Grid

The projector and the calibration object require a pseudo random multi-variate sequence encoded grid. The grid in this thesis is generated by following the equations introduced in Section 5.2 and with the following parameters:

- The window size is set to 3.
- The Galois field uses a base 5 and the numbers 0 to 4 were assigned the colour magenta, red, green , yellow and cyan respectively.
- The generating number used is 3102.
- The sequence S is computed using a register R initialized with $R = \{1, 0, 0, 0\}$ by repeatedly applying the following rules:

$$\begin{aligned} x &= 3R_0 + 1R_1 + 0R_2 + 2R_3 \\ S_i &= x \\ R_{n+1} &= R_n \\ R_0 &= x \\ i &= i + 1 \text{ for } i < 5^3 \end{aligned}$$

- The horizontal and vertical lines are drawn at every 10 pixels.
- The image is printed on the desired medium.

It is possible to use any base when constructing the coloured grid, however for this thesis it was found that the proper balance between the window size, the total grid size and the accuracy at which grid colours are extracted from the observed image is optimized with a base 5. This value is therefore dependent on the quality of the camera, the projector and the printed grid itself.

The proposed method relies on the information contained in the edges. The distance between the grid lines affects the object being captured in multiple ways. The farther apart the lines are, the greater the contrast between an edge and the black area between grid lines. The closer the lines are to each others, the greater the amount of information is available inside a single image. The intersections tend to reduce the precision of the edge detector since the gradient operator tends to smooth the surface around a point of interest and artificially rounds the corners. There is therefore a balance between the amount of information available and the precision at which an edge can be extracted. A ratio of 10 was used in the current implementation as it allowed for precise location of the edges as well as good coverage of the objects.

7.4 Back Projection of an Image Point

Given a point $P_f = (X_f, Y_f)$ in the image frame buffer, it is possible to back project this point into a 3-D world coordinate system. This creates a line in 3-D. Fixing the value for Z yields a point in the 3-D world coordinate system.

Algorithm 7.1. *The back projection algorithm projects an image point into the 3-D world coordinate system. The inputs of the algorithm are the coordinates of the image point inside the frame buffer* (X_f, Y_f) *and the desired* P_z *value in the*

world coordinate system. The steps are:

1. *Compute the real image coordinate* (X_d, Y_d) *from the pixel in frame memory* (X_f, Y_f).

$$X_d = \frac{X_f - C_x}{s_x d'^{-1}_x} \tag{7.17a}$$

$$Y_d = \frac{Y_f - C_y}{d_y^{-1}} \tag{7.17b}$$

2. *Compute the radial distortion* (D_x, D_y)

$$D_x = X_d(\kappa_1 r^2) \tag{7.18a}$$

$$D_y = Y_d(\kappa_1 r^2) \tag{7.18b}$$

$$r = \sqrt{X_d^2 + Y_d^2} \tag{7.18c}$$

3. *Compute the undistorted point* (X_u, Y_u)

$$X_u = X_d + D_x \tag{7.19a}$$

$$Y_u = Y_d + D_y \tag{7.19b}$$

4. *The inverse of the projection is used to compute the values for* P_x *and* P_y.

$$denom = Y_u(r_{11}r_{32} - r_{12}r_{31}) + X_u(r_{22}r_{31} - r_{21}r_{32}) \tag{7.20}$$

$$+f(r_{12}r_{21} - r_{11}r_{22}) \tag{7.21}$$

$$P_x = \frac{\begin{array}{c} P_z\left(Y_u(r_{12}r_{33} - r_{13}r_{32}) + X_u(r_{23}r_{32} - r_{22}r_{33}) + f(r_{13}r_{22} - r_{12}r_{23})\right) \\ +(r_{12}T_z - r_{32}T_x)Y_u + (r_{32}T_y - r_{22}T_z)X_u + f(r_{22}T_x - r_{12}T_y) \end{array}}{denom} \tag{7.22}$$

$$P_y = \frac{\begin{array}{r} P_z\left(Y_u(r_{11}r_{33} - r_{13}r_{31}) + X_u(r_{23}r_{31} - r_{21}r_{33}) - f(r_{13}r_{21} - r_{11}r_{23})\right) \\ +Y_u(r_{11}T_z - r_{31}T_x) + X_u(r_{31}T_y - r_{21}T_z) + f(r_{21}T_x - r_{11}T_y) \end{array}}{-denom} \tag{7.23}$$

where the variables are defined as per the camera's model of the Section 7.1.

7.5 3-D Reconstruction

The camera model makes it clear that if the only information known about a point is its location (X_f, Y_f) in pixels, then (after calibration) the only information which can be obtained about its location in the 3-D world coordinates is the 3-D "line" in which it lies. Actually, it is not even a line as a pixel is not an infinitely small point but a square of a finite size. This situation is minimized for the camera model and is helpful for the projector model.

Corollary 7.1. *The effect of only being able to follow a point by following a 3-D line is minimized since the proposed methodology does not attempt to recreate the 3-D location of a point, but rather the 3-D location of a NURBS curve.*

A NURBS curve is used to describe each grid line illuminated by the projector and perceived by the camera. Each NURBS curve is constructed by using the edge points obtained with sub-pixel accuracy and then using the global approximation method (see Algorithm 3.3) with an error bound of 0.1 pixel. The NURBS curves form a network which is used to create a 3-D representation of an object.

It is possible to use a network of NURBS curves to create a 3-D representation of an object because of one of their properties. Property 2.2 states that a NURBS curve is invariant under affine as well as perspective transformations. This implies that an affine or a perspective transformation is applied to a curve by applying

it to the control points. Therefore, the control points of the NURBS curve are projected on a plane inside the 3-D world coordinate system and the new NURBS curve will lie on that plane. By applying this perspective transformation to every curves inside the network of NURBS curves, a 3-D representation of the object is obtained.

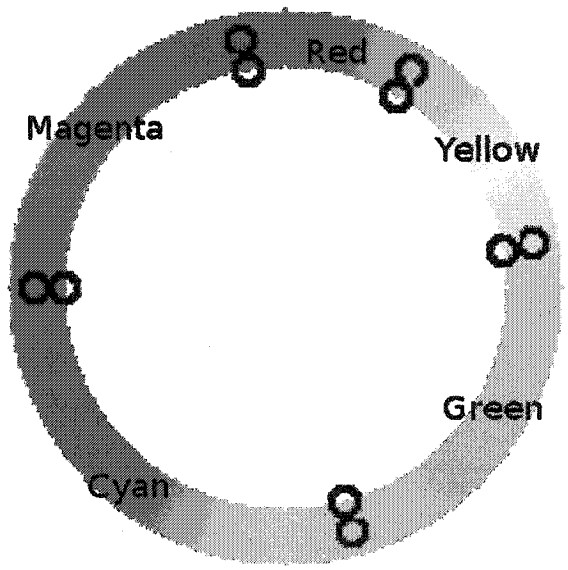


Figure 7.3: Hue boundaries used to associate a line with a colour.

Algorithm 7.2. *The 3-D reconstruction process is as follows*

Step 1 *Extract the lines from the image using Algorithm 6.4.*

Step 2 *Determine the colour of each line by analyzing the hue of the middle points. Figure 7.3 shows the hue boundaries for each colour that is part of the PRMVS being projected. The boundaries are located at* $30°, 80°, 165°, 270°$ *and* $350°$. *Those values were chosen to account for the type of bulb used*

inside the projector and also to account for the digital camera sensor. The colour yellow goes from 30° *to* 80°. *The colour green goes from* 80° *to* 165°. *The colour cyan goes from* 165° *to* 270°. *The colour magenta goes from* 270° *to* 350°. *The colour red goes from* 350° *to* 30°. *The colour of multiple points along the line are examined and when more than half of these points have the same colour c then the line is also set to the colour c.*

Step 3 *Using a sequence of 3 lines, assign a grid index to each line.*

Step 4 *Using NURBS curve approximation (Algorithm 3.3), generate a NURBS curve* C_i.

Step 5 *For an horizontal NURBS curve* C_i, *the corresponding horizontal grid line* j *from the projector is used to create a 3-D world coordinate surface.*

The calibration process takes into account the radial distortion of the projector. The radial distortion implies that, in a perfect world, a plane of light would be created by the projection of a line through space. However, with radial distortion, the situation looks more like Figure 7.2

A NURBS curve can be used to represent a conic or in the case of radial distortion, a parabolic curve. Let P_0 *and* P_2 *be the end points of the conic arc, let* T_0 *and* T_2 *be the derivative of the arc at those end points and let* P *be an arbitrary point on the parabolic curve. Then, it is possible to create a NURBS curve which corresponds to this parabolic curve. The control points* P_0 *and* P_2 *are already defined and the point* P_1 *is obtained by intersecting the line* $[P_0, T_0]$ *and* $[P_2, T_2]$. *The weights* w_0 *and* w_2 *are both set to 1. The only unknown is* w_1 *which can be solved for using geometric arguments (see [56]). This method yields the following equation for* w_1:

$$w_1 = \frac{(1-u)^2(P-P_0)\cdot(P_1-P)+u^2(P-P_2)\cdot(P_1-P)}{2u(1-u)|P_1-P|^2} \quad (7.24)$$

where

$$u = \frac{a}{1+a} \qquad a = \sqrt{\frac{|P_0Q|}{|QP_2|}} \quad (7.25)$$

and Q is the projection of P onto the line $[P_0, P_2]$ using P_1 as a center of projection.

To create the surface of projection of this line onto space the ruled surface algorithm is used. The ruled surface algorithm requires two curves and it is the same algorithm as skinning (see Section 3.6.2) applied to only two curves. The second curve is obtained by projecting the curve obtained above on a plane behind the object. The projection equations are the same as the ones in step 2 of the reconstruction algorithm. The ruled surface is then simply defined as $U = \{0,0,0,1,1,1\}, V = \{0,0,1,1\}, W_j^i = w_i^1, R_i^1 = P_i^1$ and $R_i^2 = P_i^2$.

Step 7 *For each control point P_j inside C_i, a projection line is computed inside the 3-D world coordinate. The projection follows the steps of Algorithm 7.1. That line is created by finding a point at $Z = 0$ and $Z = 50$.*

Step 8 *For each intersection between the line and the projection surface, a new control point P_j' is inferred.*

Intersecting a line with a NURBS surface is done using a Newton iterative method to compute the intersection point. The method uses the geometrical properties of a NURBS surface and its control net to limit the search space.

Step 9 *The 3-D world coordinate NURBS curve is created by using the same knot vector as C_i and the control point set defined by P'_j. Figure 7.4 depicts the steps 7, 8 and 9.*

Step 10 *Repeat the process for every line.*

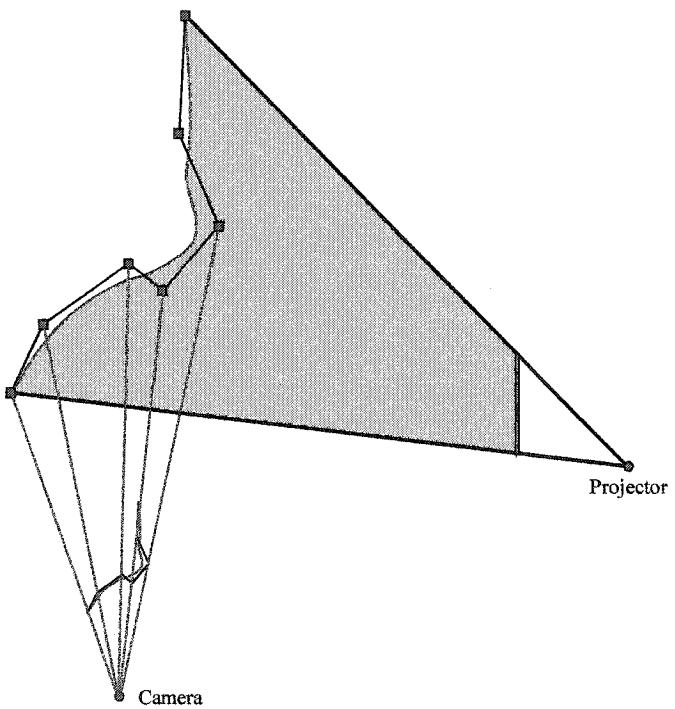


Figure 7.4: Projection of a 2-D NURBS curve in 3-D world coordinate space.

The process is the same for the vertical lines, and is depicted in Figure 7.4. It illustrates how a NURBS curve's control point polygon is projected on the light plane of a grid line originating from the projector. The new NURBS curve is now in 3-D world coordinates.

As mentioned in section 6.3, the reconstruction is not made in terms of the grid line illuminating the object but in terms of the edges for that line. The 3-D reconstruction process is the same, except that two NURBS curves are generated for each grid line. A close-up of this situation is shown in Figure 7.5.

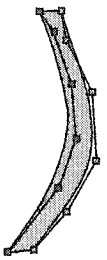


Figure 7.5: 3-D reconstruction is performed with two edges.

The equation of each projection surface is different as each edge line has an offset of $\pm d_x$ for the vertical lines and an offset of $\pm d_y$ for the horizontal lines.

Creating a surface that represents the object is done by doing an interpolation of a bidirectional curve network (see Section 3.6.3). A consistent network of curves is required to generate a Gordon surface. Let d be the degree of the surface desired, then the NURBS curve network must form at least a $(d+1) \times (d+1)$ matrix of intersections for the Gordon surface methodology to be usable.

There are multiple ways to handle a grid network. The approach chosen in this thesis ensures full coverage of the grid network but creates multiple NURBS surfaces and it also creates overlapping surfaces.

Algorithm 7.3. *Let $\{H_i\}$ be the set of horizontal curves in 3-D and $\{H_i'\}$ be the set of corresponding curves in 2-D. Also, let $\{V_i\}$ be the set of vertical curves in 3-D and $\{V_i\}$ be the set of corresponding curves in 2-D, then, the algorithm to*

generate a set of NURBS surfaces based on the network of 3-D NURBS curves obtained with Algorithm 7.2 is as follows:

Step 1 *Find an intersection node $n_{i,j}$ for which a contiguous region of $(d+1) \times (d+1)$ intersections exists.*

Step 2 *Create an initial intersection matrix of size $(d+1) \times (d+1)$.*

Step 3 *For all the intersection points, find the parametric value $u_{k,l}$ and $v_{k,l}$ where $k = 0, \ldots, d+1$ and $v = 0, \ldots, d+1$ that satisfy the relationship $H'_{i+k}(u_{k,l}) = V'_{j+l}(v_{k,l})$.*

Step 4 *Populate all the intersection points with the values of $(H_{i+k}(u_{k,l}) + V_{j+l}(v_{k,l}))/2$ for $k = 0, \ldots, d+1$ and $v = 0, \ldots, d+1$*

Step 5 *Create a set of horizontal NURBS curves $\{h_k\}$ by doing the following. For each k, compute the compatible NURBS curve by splitting the NURBS curve H_k between $u' = u_{k,0}$ and $u'' = u_{k,d}$.*

Step 6 *Create a set of vertical NURBS curves $\{v_l\}$ by doing the following. For each l, compute the compatible NURBS curve by splitting the NURBS curve V_l between $v' = v_{0,l}$ and $v'' = v_{d,l}$.*

Step 7 *Compute a NURBS surface with Gordon's algorithm (see Section 3.6.3) using the network of curves $\{h_k\}$ and $\{v_l\}$ and also using the intersection points computed in Step 4.*

The presence of protrusions or holes might hinder the surface interpolation method. For those cases, the surface then becomes a set of NURBS patches, which taken together, constitute the overall surface.

In practise, it was found that two edges could introduce instabilities into the generated NURBS surfaces. Therefore only the upper edge is used when generating a NURBS surface.

Definition 7.1. *The upper edge is the edge which corresponds to a positive gradient value on an edge. This corresponds to the top edge of a horizontal line and the left edge of a vertical line.*

7.6 Conclusion

The methodology to generate a 3-D object model using NURBS starts by defining the camera model. Tsai's calibration method defines a camera model which can relate 3-D world coordinate points to the pixels captured by camera. This model is used to project the control points of a NURBS curve, defined in 2-D and corresponding to the edge lines of the coloured grid as seen by the camera, in the 3-D world coordinate system. This creates a set of curves in the horizontal and vertical direction. From the two set of curves, it is possible to create a NURBS surface.

This thesis takes into account the radial distortion when recreating the surface unlike most other techniques which rely solely on the pinhole model. It doesn't project a single point in space but it projects a line that keeps the structure of the data and allows to create a surface with great precision.

Chapter 8

Calibration

The calibration uses Tsai's calibration method at its core (see Section 7.1). This method is well known in the literature. It compares well with other methods [66] and a reference implementation is readily available on the Internet [83]. Tsai's method can calibrate both co-planar and non co-planar points and it is available in an optimized as well as a non-optimized version.

The method proposed by Tsai does not address the problem of generating a high precision calibration object nor does it address the problem of precisely extracting feature points inside an image. This thesis' calibration process address these issues with different objects, on which a PRMVS grid could be printed on, were tried to determine how to obtain the most accurate results possible.

8.1 Calibration Process

The calibration process is characterized by two important factors: accuracy and reliability. To be accurate, the calibration points must be non co-planar. To be

reliable, the calibration object should stay in the same position when calibrating the camera and the projector and only one calibration object should be used.

Most calibration techniques use two objects: one to calibrate the camera and the other to calibrate the projector. That kind of method relies on the following steps:

1. Put the camera calibration object in place.
2. Calibrate the camera
3. Remove the calibration object used to calibrate the camera
4. Put the calibration object for the projector at the exact same location
5. Calibrate the projector

That kind of methodology can be error prone since it is very difficult to place the two calibration objects at the same location within an error smaller than 0.1 mm. With this in mind, it was decided to use a single object to calibrate both the camera and the projector therefore removing the need to move two objects in order to calibrate the setup.

In order to satisfy the two calibration criteria with a single object, a cylinder was first tried. The surface of a cylinder is non co-planar and when an axis goes through the centre it can simply be turned to show either a face with a structured pattern on it or a blank face.

The cylinder object was also chosen because it could be manufactured with high precision. A prototype of such an object is shown in Figure 8.1. The prototype is composed of the cylinder and a support structure. The support allows the cylinder to be turned on its axis.

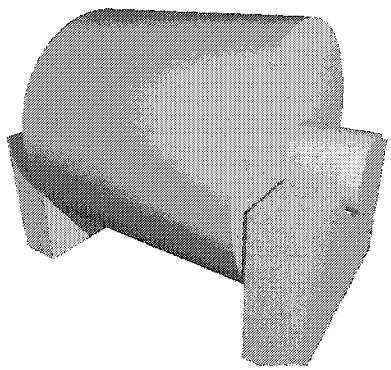


Figure 8.1: Prototype of the cylinder object.

The principle of using a cylinder is sound and it seems to solve the problem of having two calibration objects. When analyzing the calibration points, the cylinder gives very precise results. However, when analyzing the precision of the calibration obtained across the volume of interest, the precision drops rapidly as the data points are further away from the surface of the cylinder.

Definition 8.1. *The volume of interest is defined as the volume for which scanning data can be obtained. The volume of interest is also referred to as the cube of interest to reflect that a cubic area is covered.*

The cylinder was therefore found inadequate as a calibration object. An alternative solution is now proposed. This alternative solution is better than the original solution. It is based on the observations made by Tsai regarding calibration:

- The calibration points must cover a wide area of the frame buffer to calibrate for radial distortion.
- The calibration points should span the volume of interest

The modification of the structured pattern used for the calibration allows us to solve the problem of following Tsai's advice regarding where the points are to be located. In addition, this enables the system to be calibrated without moving the calibration object.

A structured grid divided into four quadrants with two of the four quadrants being totally white was devised and used. The calibration points for the camera are in quadrant 2 and 4, and for the projector in quadrant 1 and 3 as shown in Figure 8.2.

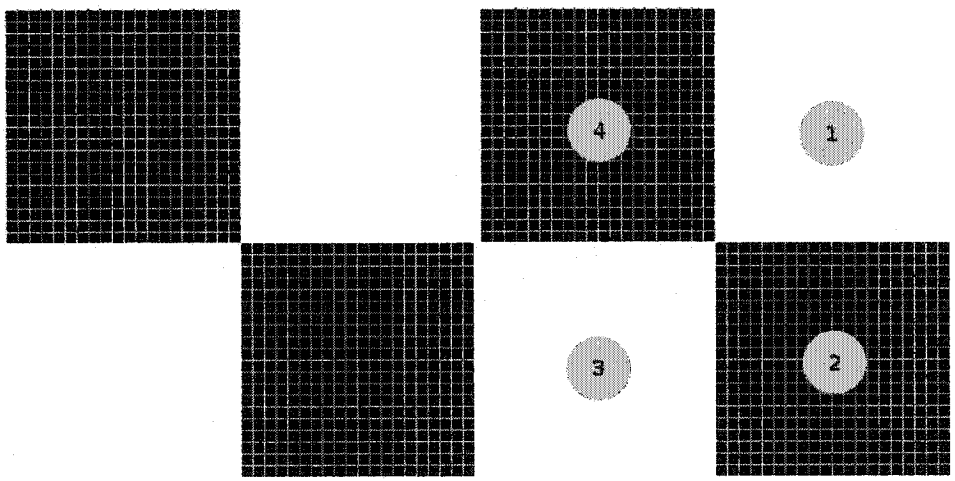


Figure 8.2: (a) Image to be printed on a calibration object. (b) Numbering of the quadrants.

The algorithm to calibrate an object is as follows:

Algorithm 8.1. *The calibration algorithm.*

Step 1 *Position the calibration object containing the proposed grid pattern.*

Step 2 *Take a picture of the calibration object.*

Step 3 *Extract the intersection points and compute the 3-D location of each of these points.*

Step 4 *Use Tsai's method to calibrate the camera in a non-optimized mode since it creates good initial values for the next step.*

Step 5 *Use Tsai's method to calibrate the camera with an optimized method.*

Step 6 *Turn on the projector*

Step 7 *Take a picture of the object while illuminated by the projector.*

Step 8 *Extract the intersection point location and compute its ideal 3-D location.*

Step 9 *Use Tsai's method to calibrate the projector in a non-optimized mode since it creates good initial values for the next step.*

Step 10 *Use Tsai's method to calibrate the projector with an optimized method.*

8.2 Determining the 3-D Location for Calibration

Several types of calibration objects were used to test the calibration process. The chosen objects are a cylinder, a cylinder with a truncated portion, two planes at a 135 degree angle and a single plane. The Figure 8.3 depicts all these objects together.

The ideal 3-D location is a required input for the calibration process. This ideal point corresponds to the expected location of the point in 3-D. Each calibration

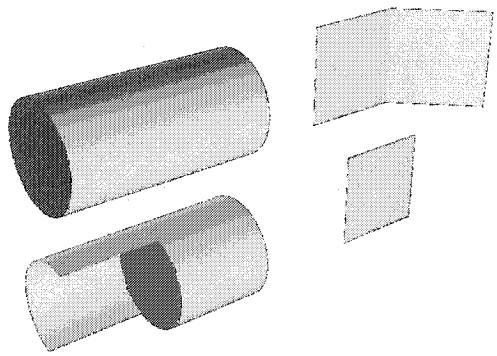


Figure 8.3: The calibration objects are a cylinder, a hollow cylinder, two planes and a single plane.

object requires a different equation to compute the true 3-D locations needed for the camera and projector calibration.

8.2.1 The 3-D Location for the Cylinder

The 3-D Location of Points for the Camera

The equation that maps the grid points printed on the cylinder object to a 3-D world coordinate point is as follows:

$$\alpha = a_s Y + a_o \tag{8.1a}$$

$$P_x = s_x X + O_x \tag{8.1b}$$

$$P_y = r\cos(\alpha) + O_y \tag{8.1c}$$

$$P_z = r\sin(\alpha) \tag{8.1d}$$

where $P = (P_x, P_y, P_z)$ is the location of the ideal 3-D grid intersection point, (X, Y) the grid intersection index, r the radius, a_s the angle distance between two grid lines, a_o the angle offset, s_x the size along the X axis between grid lines and $O = (O_x, O_y, O_z)$ the offset value.

The 3-D Location of Points for the Projector

The location of a projector's grid point in 3-D is achieved by back projecting the camera view of that point onto the cylinder. The back projection of a point (see Algorithm 7.1) allows to define a line in the 3-D world coordinate system that corresponds to the pixel of the camera.

Algorithm 8.2. *The line-circle intersection algorithm shown here is the same as the one given by Weisstein [79]. It defines a circle of radius r which is centred around the origin $(0, 0)$. The line is defined from two points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$.*

A line can intersect a circle at 2 points, 1 point or 0 points. It intersects 2 points when it crosses the circle. It intersects at one point when it is a tangent to the circle. It can also not intersect the circle.

Given

$$d_x = x_2 - x_1 \tag{8.2}$$

$$d_y = y_2 - y_1 \tag{8.3}$$

$$d_r = \sqrt{d_x^2 + d_y^2} \tag{8.4}$$

$$D = x_1 y_2 - x_2 y_1 \tag{8.5}$$

then the intersection points are at

$$x = \begin{cases} \frac{Dd_y \pm d_x\sqrt{r^2d_r^2 - D^2}}{d_r^2} & d_y \geq 0 \\ \frac{Dd_y \mp d_x\sqrt{r^2d_r^2 - D^2}}{d_r^2} & d_y < 0 \end{cases} \tag{8.6}$$

$$y = \frac{-Dd_x \pm |d_y|\sqrt{r^2d_r^2 - D^2}}{d_r^2} \tag{8.7}$$

The discriminant is defined as

$$\Delta = r^2d_r^2 - D^2 \tag{8.8}$$

When the discriminant is smaller than 0, there are no intersection points. When it is 0, the line is tangent to the circle. When it is greater than 0, there are two intersection points.

Algorithm 8.3. *Given a plane defined with the point-normal form as* $(X-P)\cdot v = 0$ *and a line given in parametric form as* $Q(t) = Q + tw$ *then the intersection* i *between the line and the plane is found by replacing* X *by* $Q(t)$ *which results in*

$$i = Q + \frac{(P-Q)\cdot v}{w \cdot v}w \tag{8.9}$$

Algorithm 8.4. *The intersection between a line and the calibration cylinder relies on a single observation: the cylinder's axis is parallel to the* X *axis. This fact allows to compute the intersection between the calibration cylinder and a line by using the following steps:*

Step 1 *Compute the intersection of the line and the circle which lies in the* $Z - Y$ *plane. Before using Algorithm 8.2, an offset must be applied such that the centre of the cylinder lies at* $(0, 0)$ *inside that plane. This yields two intersection points. The point closest to the camera is kept, the other is discarded.*

Step 2 *Construct a plane which is parallel to the X axis and passes through the closest intersection point.*

Step 3 *The intersection between the plane and the line yields the location of the intersection between the line and the cylinder. The intersection is found by using Algorithm 8.3.*

The projector's ideal 3-D points are computed by using Algorithm 8.5. The locations of the intersection points as perceived by the camera are projected on the cylinder and these 3-D points are used to calibrate the projector. The pixel location of a point $p = (p_x, p_y)$ on the projector's grid is simply given by

$$p_x = s_x X \tag{8.10}$$

$$p_y = s_y Y \tag{8.11}$$

where s_x and s_y are the size in pixels between the grid lines. X and Y are the grid line indexes along the column and row axes.

Algorithm 8.5. *The method to intersect the calibration cylinder and an image pixel is as follow:*

Step 1 *Back project the view of a grid intersection as perceived by the camera to create a line in the 3-D world coordinate by using Algorithm 7.1.*

Step 2 *Find the intersection between that line and the calibration cylinder by using Algorithm 8.4.*

8.2.2 The 3-D Location for the Hollow Cylinder

The hollow cylinder is very similar to the cylinder in terms of equations, however one of its sections is concave while the other is convex.

The 3-D Location of Points for the Camera

The 3-D locations of the ideal grid points are a variation of the equations and algorithm used for the cylinder calibration object. The 3-D location of the ideal intersection point when calibrating the camera is given by the following equation:

$$\alpha = \begin{cases} a_s Y + a_o & \text{if } X \geq 10 \\ a_s - Y & \text{if } X < 10 \end{cases} \tag{8.12a}$$

$$P_x = s_x X + O_x \tag{8.12b}$$

$$P_y = r\cos(\alpha) + O_y \tag{8.12c}$$

$$P_z = r\sin(\alpha) \tag{8.12d}$$

where $P = (P_x, P_y, P_z)$ is the location of the ideal 3-D grid intersection point, (X, Y) the grid intersection index, r the radius, a_s the angle distance between two grid lines, a_o the angle offset, s_x the size along the X axis between grid lines. and $O = (O_x, O_y, O_z)$ the offset value.

The 3-D Location of Points for the Projector

The algorithm to find the 3-D points necessary to calibrate the projector is Algorithm 8.6.

Algorithm 8.6. *This algorithm computes the location of an image point, back projected, on the hollow cylinder object. The value C_x is the location along the X axis of the center of the hollow cylinder.*

Step 1 *Use Algorithm 8.5 to compute an intersection point.*

Step 2 *Modify Algorithm 8.5 by discarding the intersection closes to the camera and not the one farthest from the camera. This yields another intersection point.*

Step 3 *If the x value of the intersection point obtained in Step 1 is greater than C_x then the intersection point is this one.*

Step 4 *If the x value of the intersection point obtained in Step 1 is smaller than C_x then the intersection point is the one obtained in Step 2.*

8.2.3 The 3-D Location for the Two Planes

The use of two planes as a calibration object has been around for a long time [75].

The 3-D Location of Points for the Camera

The 3-D points used to calibrate the camera are defined by the following equation:

$$P_z = \begin{cases} s_x(X-20)\cos 45° & \text{if } X >= 20 \\ 0 & \text{otherwise} \end{cases} \tag{8.13}$$

$$P_y = O_y - s_y Y \tag{8.14}$$

$$P_x = \begin{cases} O_x + 20s_x + z & \text{if } X >= 20 \\ O_x + s_x X & \text{otherwise} \end{cases} \tag{8.15}$$

$$\tag{8.16}$$

where $P = (P_x, P_y, P_z)$ is the location of the ideal 3-D grid intersection point, (X, Y) the grid intersection index, s_x the size along the X axis between grid lines and $O = (O_x, O_y, O_z)$ the offset value.

The 3-D Location of Points for the Projector

The location of a 3-D intersection point to calibrate the projector is calculated by computing the intersection between the line obtained by back projecting the pixel viewed by the camera (see Algorithm 7.1) and the plane at $Z = 0$ and by computing the intersection with the plane at an angle. The intersection point with the highest value of Z is the one that is kept to calibrate the projector.

8.2.4 The 3-D Location for the Single Plane

The single plane calibration is the classical method used to calibrate a camera [75].

The 3-D Location of Points for the Camera

The 3-D points used to calibrate the camera are defined by the following equation:

$$P_x = O_x + s_x X \tag{8.17}$$

$$P_y = O_y - s_y Y \tag{8.18}$$

$$P_z = 0 \tag{8.19}$$

$$\tag{8.20}$$

Where $P = (P_x, P_y, P_z)$ is the location of the ideal 3-D grid intersection point, (X, Y) the grid intersection index, s_x the size along the X axis between grid lines and $O = (O_x, O_y, O_z)$ the offset value.

The 3-D Location of Points for the Projector

The location of a 3-D intersection point to calibrate the projector is calculated by computing the intersection between the line obtained by back projecting the

pixel viewed by the camera (see Algorithm 7.1) and the plane at $Z = 0$ and by computing the intersection with the plane at an angle. The intersection point with the highest value of Z is the one that is kept to calibrate the projector.

8.3 Conclusion

This section presents the calibration technique proposed in this thesis. The technique is unique as it relies on a new coloured 2-D encoding of the structured light, it allows the calibration object to remain static throughout the calibration process and it can be applied to multiple type of calibration objects. The information contained within an image for calibration or for scanning is extracted using the same methodology. Most calibration systems have dedicated algorithms for the calibration. This implies that improvements in the accuracy of their methodology doesn't necessary imply a better calibration mechanism.

The calibration pattern proposed in this thesis allows to use different shapes to perform a calibration as can be seen in Chapter 9. Furthermore, the use of this calibration pattern enables the object to stay fixed when performing the calibration which greatly increases the precision of the calibration algorithm. Other calibration algorithms [65, 18] than Tsai's calibration algorithm can be used as long as the camera model also accounts for radial distortions.

Chapter 9

Experimental Results on NURBS based 3D Object Reconstruction using Colour-Coded Structured Light

In this chapter, experimental results are given on images obtained of different objects by using the proposed methodology for 3-D object reconstruction. The results are divided in two sections: the calibration results and the object reconstruction results.

The results are analyzed through statistical methods in order to characterize the two main aspects of the methodology: the precision of the calibration and the precision of the 3-D reconstruction process. To enable this analysis, results are given using synthetic images and real images. The synthetic images are obtained by using POV-Ray version 3.5 [60]. The advantage is that the geometry of the

scene can be known mathematically and therefore errors obtained in the reconstruction process can be evaluated with precision. The real images are obtained with a Canon Digital SLR camera model 300D which can capture 6.3 mega pixel images.

Synthetic images are generated with a ray tracer. These images are computer generated data which are further used to calculate the precision at which the methodology handles geometric changes to a surface. The use of synthetic images also allows to concentrate on the methodology itself in order to validate it prior to doing complex and time consuming real image tests.

9.1 Measuring the Calibration Accuracy

The calibration accuracy is measured using four different statistics. Each gives an error on different aspects of the camera model (see Section 7.1). The precision of all the calibration objects used are compared using the following type of statistics: distorted image plane error, undistorted plane error, object space error and normalized calibration error. The are defined below.

1. Distorted image plane error corresponds to the pixel error on the input image. The steps to compute this error are given inside Algorithm 9.1.

   Algorithm 9.1. *Estimating the distorted image plane error.*

   Step 1 *Convert the world coordinate points into camera coordinate points using Equation 7.1.*

   Step 2 *Project the points into the undistorted sensor plane using Equation 7.13.*

   Step 3 *Project those points into the distorted sensor plane (Equation 7.14)*

Step 4 *Transform the points into image coordinate points (Equation 7.16).*

Step 5 *The error is the difference between the ideal point's location on the image and the measured image pixel's location.*

2. Undistorted image plane error corresponds to the error between the ideal point's location on the undistorted sensor plane and the location of the measure image point on the undistorted sensor plane. The steps to compute this statistic are given in Algorithm 9.2.

   Algorithm 9.2. *Estimating the undistorted image plane error.*

   Step 1 *Convert the ideal point's world coordinate into camera coordinate (Equation 7.1),*

   Step 2 *Project that point on the undistorted sensor plane (Equation 7.13)*

   Step 3 *Convert the measured image point to the distorted sensor plane (Equation 7.17)*

   Step 4 *Transform it on the undistorted sensor plane (Equation 7.19).*

   Step 5 *The error is the distance between ideal point's location on the image undistorted plane and the measured image pixel's location inside the undistorted plane.*

3. Object space error corresponds to the distance between the ideal point in 3D and the line defined by back projecting the measured point out through the camera model. The equations required to back project are described in Section 7.5.

4. Normalized calibration error was proposed by Weng [81].

Definition 9.1. *The normalized calibration error corresponds to the error between the ideal point transformed in the camera coordinate system and the measure point when it is back projected to the camera coordinate system with the same Z value as the ideal point. That error is scaled with the z value and the focal distance in the camera coordinate system.*

The normalized calibration error normalizes the error across the volume of the object and gives a normalized error in finding the X and Y values. The equation to back project is the same as for the previous item except that the back projection occurs in the camera coordinate system and not in the world coordinate system. The normalization equation is defined as

$$error = \frac{1}{n}\sum_{1}^{n}\sqrt{\frac{(p_x - p'_x)^2 + (p_y - p'_y)^2}{p_z(\frac{1}{f_u^2} + \frac{1}{f_v^2})/12}} \tag{9.1}$$

where p is the measured point's location when back projected and p' is the ideal point's location on the camera coordinate system. The focal factors f_u and f_v are defined as

$$f_u = \frac{s_x f}{d_x} \tag{9.2}$$

$$f_v = \frac{f}{d_y} \tag{9.3}$$

where s_x, f and d are defined inside Section 7.1.

9.2 Measuring the System Accuracy

The statistics in the previous section yield accuracy results by analyzing only the calibration points. It doesn't yield global results regarding the system's accuracy

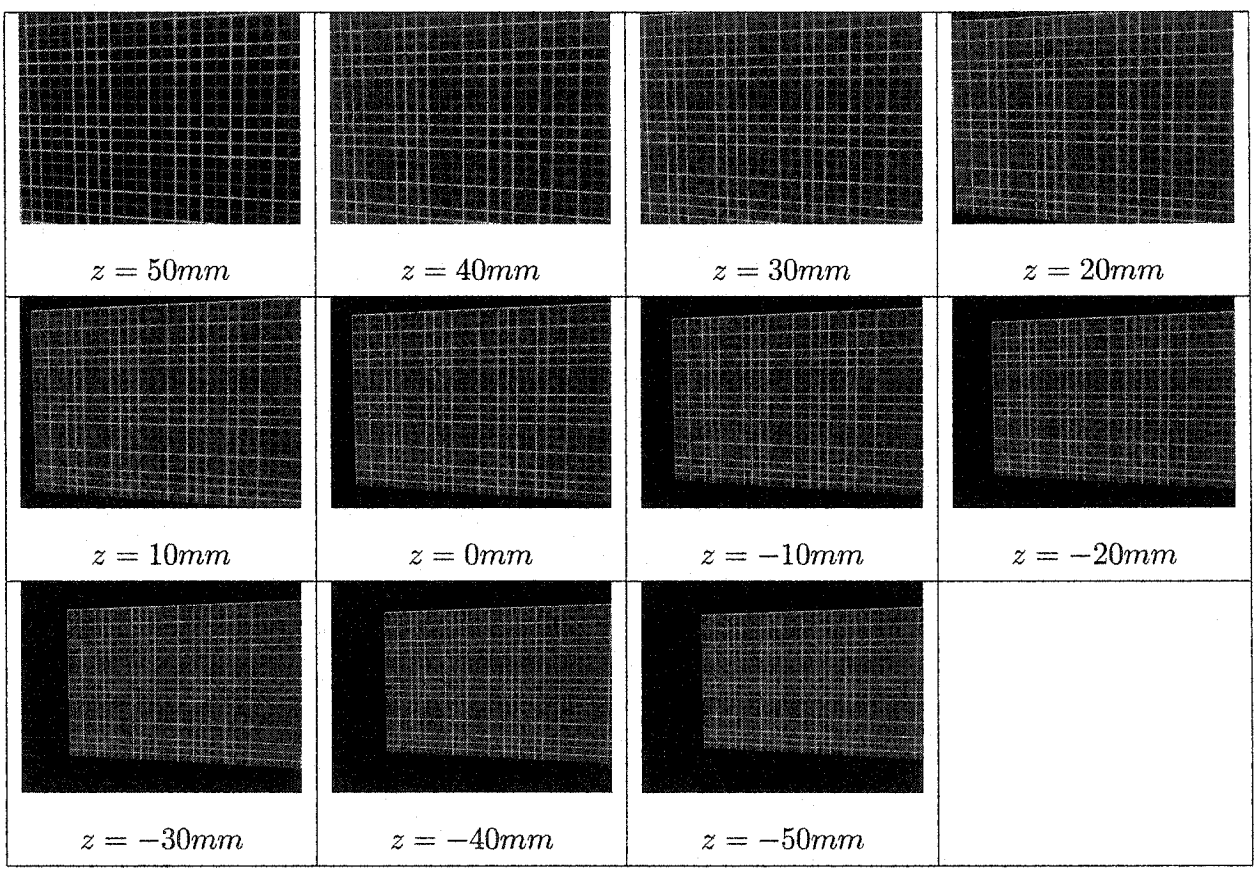


Figure 9.1: Testing the volume accuracy of the camera at different z values.

for the 3-D reconstruction of an object across the volume of interest. The volume of interest in the current experiments is $200 \times 100 \times 100mm^3$.

To assess the accuracy across that volume, the following two tests are performed. The first estimates the volume accuracy of the camera and the second estimates the global volume accuracy. The global volume test yields two results, one is based on the accuracy obtained by only using the horizontal grid lines and the other when only using the vertical grid lines.

The volume accuracy of the camera corresponds to the precision obtained by the camera calibration algorithm that is analyzed across the volume of interest. Figure 9.1 shows all the input images used to compute the volume accuracy of the camera inside a volume of interest varying from -50 mm to 50 mm. The accuracy is estimated by following these steps:

1. Position a plane with a structured grid printed on it perpendicular to the z-axis at different positions along the z-axis: -50 mm to 50 mm in 10 mm increments.

2. Take a snapshot of the plane at each of those locations as depicted in Figure 9.1.

3. Extract the intersection points

4. Use the 3-D reconstruction technique detailed in Section 7.5 to determine the location of each intersection point in 3-D.

5. The error is the difference between the ideal location and the computed location.

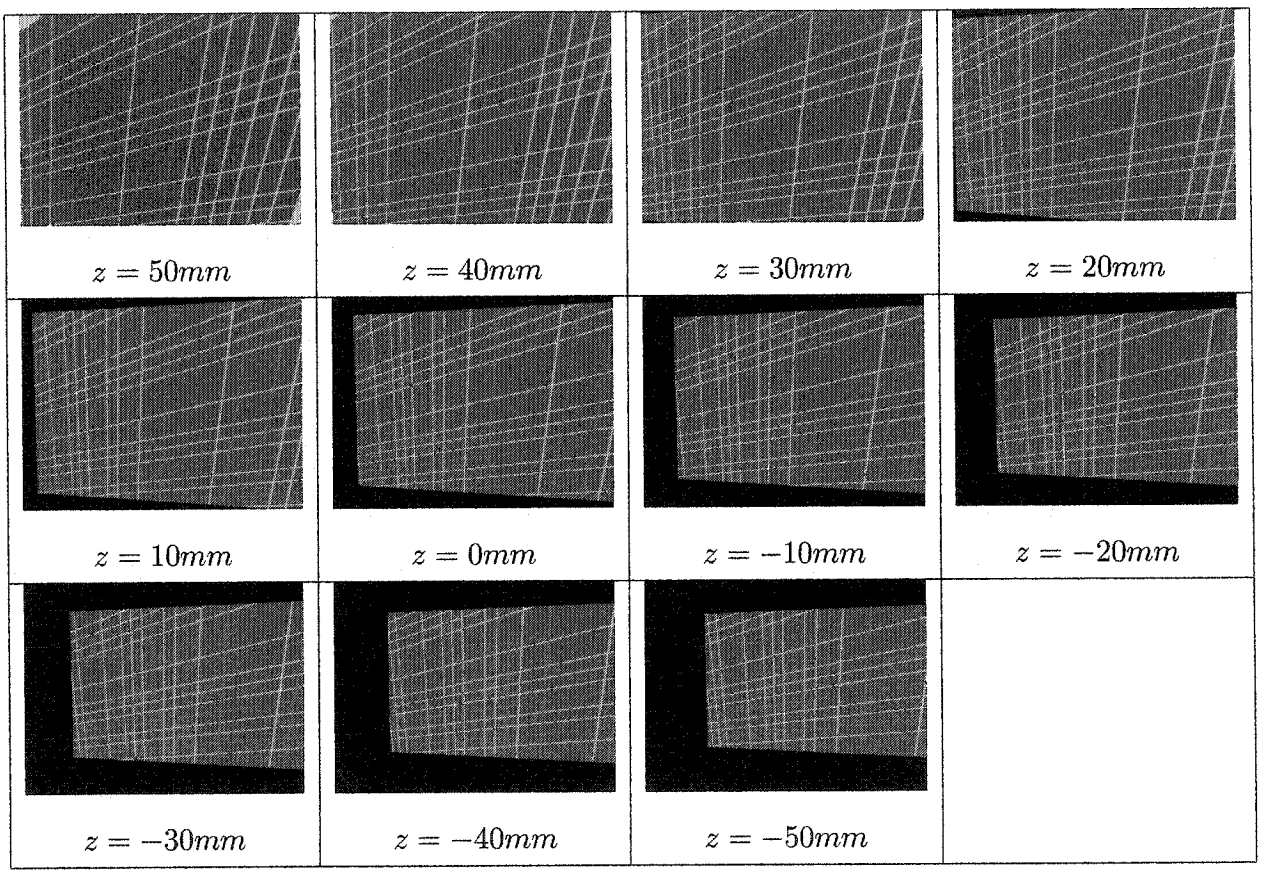


Figure 9.2: Testing the global volume accuracy at different z values.

The global volume accuracy test verifies the accuracy of the 3-D reconstruction technique proposed in Section 7.5 across the volume of interest. It verifies the accuracy at which an object illuminated with the structured grid can be reconstructed. Figure 9.2 shows all the input images used to compute the global volume accuracy. It is very similar to the global volume accuracy for the camera test and includes the following steps:

1. Activate the projector
2. Position a white plane perpendicular to the z-axis at different positions along the z-axis: -50 mm to 50 mm in 10 mm increments.
3. Take a snapshot of the plane at each of those locations. Those snapshots are depicted in Figure 9.2.
4. Extract the intersection points.
5. Use the back projection algorithm (see Algorithm 7.1) with the Z value set to the Z value of the plane to determine the location of each intersection point in 3-D.
6. The error is the difference between the Z-value of the plane and the Z value of the computed intersection.

The ideal location of the point in (x, y, z) could be estimated by interpolating the camera images taken above while also considering the perspective transformation. However, from a practical point of view, the Z-value error is proportional to the (x, y, z) error and it thus sufficient to only analyze this error metric.

9.3 Calibration Results

The calibration objects chosen are a cylinder, a cylinder with a truncated portion, two planes at a 135 degree angle and a single plane. Figure 8.3 of the previous section depicts all these objects together.

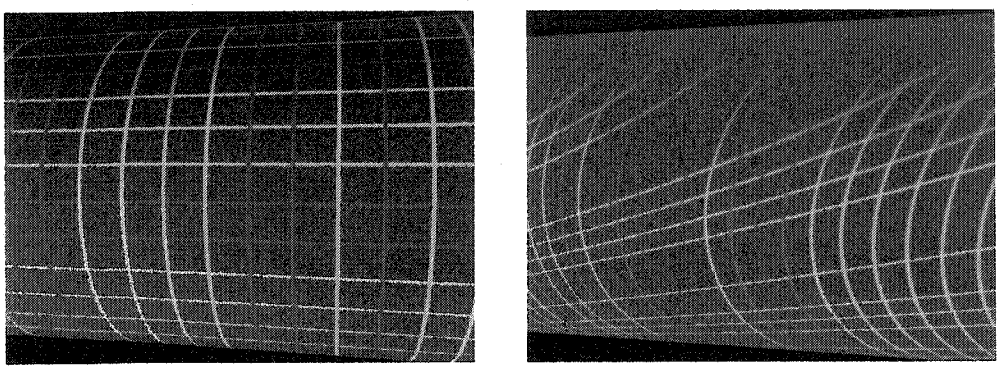
The calibration method proposed by Tsai requires a minimum of 11 intersection points. All the tests performed contained much more points than this minimum. Ultimately, it is not the number of points but the precise location of those points which influence more the ability of the Tsai calibration method to calibrate at high accuracy the camera and the projector. The tables inside this section show the number of points used to perform the calibration but the size of the samples is from five to ten times higher than the minimum required by Tsai's method and therefore the impact of the sample size is not a factor in the accuracy of the system.

9.3.1 The Cylinder

As mentioned in Chapter 8, the cylinder was the original answer to the problem of using a single object to calibrate. Figure 9.3 shows the picture used to calibrate the camera and the projector using the cylinder. The images are 800 by 600 pixels. The cylinder has a radius of 50 mm and a length of 200 mm. Its axis lies parallel to the z-axis.

In order to calibrate the camera, the ideal 3-D locations of the grid's intersection points is a required input for the calibration system.

The implementation of Tsai's algorithm offers two versions: one without optimization and one with optimization. Therefore the calibration is performed by first using the non-optimized version to get a good first estimate and then these



(a) Image for camera calibration.

(b) Image for projector calibration.

Figure 9.3: Calibration images with the cylinder calibration object.

values are used by the optimized calibration method to improve the calibration results. The results obtained for the calibration are shown in Table 9.1. This Table shows the precision of the algorithm when analyzing the input points used for the calibration. The different error statistics are explained in Section 9.1.

The image plane error of 0.12414 pixel of the camera calibration indicates that using an intersection point with sub-pixel accuracy greatly improves the calibration results. The overall object space distortion of 0.01068 mm as measured when calibrating the projector indicates a high precision.

Tsai notes that the calibration object should span the entire volume of interest to have valid calibration across that volume. The cylinder calibration object only covers its outer radius and only a small portion of it. This yields a calibration which is not stable across the desired volume of interest as it doesn't contain points which are spread throughout the volume of interest.

The Table 9.2 depicts the errors obtained by the calibration process when analyzing the estimated and real points at different plane positions. The values ob-

Camera (70 points)	Mean	Standard Deviation	Max
Distorted image plane error	0.12414 pixel	0.08608 pixel	0.44553 pixel
Undistorted image plane error	0.11995 pixel	0.08242 pixel	0.42167 pixel
Object space error	0.01651 mm	0.01236 mm	0.06614 mm
Normalized calibration error	0.29382 mm	0.20188 mm	1.03286 mm
Projector (165 points)	Mean	Standard Deviation	Max
Distorted image plane error	0.13335 pixel	0.08489 pixel	0.35496 pixel
Undistorted image plane error	0.01202 pixel	0.00762 pixel	0.04470 pixel
Object space error	0.01068 mm	0.00699 mm	0.04117 mm
Normalized calibration error	0.02944 mm	0.01868 mm	0.10968 mm

Table 9.1: Calibration statistics from the cylinder calibration object.

tained indicate that the camera's calibration accuracy is very stable across the volume of interest as the standard deviation is very small (from 0.01084 mm to 0.02014 mm). This seems to indicate that a translation error occurred at the calibration stage and that Tsai's method couldn't optimize it properly.

The global volume accuracy as listed inside Table 9.3 indicates that the z-axis error increases as the plane is farther from the outer radius of the cylinder which is close to the $z = 50mm$ plane. The standard deviation across the plane is very low (from 0.05060 mm to 0.12308 mm for the vertical lines and from 0.04384 mm to 0.06783 mm for the horizontal lines) which again indicates that the calibration is stable across the volume of interest.

Figure 9.4 depicts the accuracy of the camera and the global accuracy of the method proposed in this thesis. Only the mean error values are shown on the graphic. This figure clearly shows that the estimated point is different when using

Z (mm)	Mean (mm)	Min (mm)	Max (mm)	Std Dev (mm)	N
-50	0.39755	0.37505	0.42783	0.01084	549
-40	0.39484	0.36990	0.43331	0.01175	538
-30	0.39211	0.36680	0.43100	0.01262	540
-20	0.38932	0.36157	0.43010	0.01335	527
-10	0.38745	0.35791	0.43355	0.01446	524
0	0.38639	0.35441	0.43496	0.01560	511
10	0.38536	0.35213	0.44110	0.01741	512
20	0.38589	0.35006	0.45080	0.01915	485
30	0.38555	0.34750	0.45274	0.02014	436
40	0.38530	0.34787	0.44292	0.01918	359
50	0.38913	0.35286	0.44607	0.01907	314

Table 9.2: Volume accuracy of the camera at different Z values with the cylinder calibration object.

		Horizontal Lines				Vertical Lines			
Z (mm)	N	Mean (mm)	Min (mm)	Max (mm)	Std Dev (mm)	Mean (mm)	Min (mm)	Max (mm)	Std Dev (mm)
-50	238	1.50693	1.36237	1.64002	0.06783	0.11611	0.00074	0.34791	0.08785
-40	247	1.31089	1.18381	1.44564	0.05938	0.10494	0.00181	0.32532	0.07995
-30	260	1.12846	1.02153	1.24585	0.05203	0.09491	0.00014	0.26226	0.06828
-20	277	0.95432	0.85054	1.06937	0.04684	0.08784	0.00085	0.24064	0.06313
-10	291	0.79343	0.62031	0.89762	0.04384	0.08027	0.00002	0.21395	0.05541
0	307	0.64536	0.42101	0.77189	0.04648	0.07277	0.00003	0.25798	0.05060
10	320	0.50910	0.26804	0.64604	0.04917	0.06989	0.00020	0.37770	0.05565
20	331	0.38425	0.08376	0.53380	0.05706	0.06816	0.00004	0.38594	0.06243
30	309	0.28049	0.04128	0.41251	0.05364	0.07556	0.00017	0.45310	0.07833
40	273	0.19412	0.02119	0.32895	0.04422	0.09470	0.00014	0.46559	0.10022
50	238	0.11996	0.00188	0.25967	0.04178	0.12976	0.00069	0.54816	0.12308

Table 9.3: Global volume accuracy at different Z values with the cylinder calibration object.

the horizontal line than it is when using the vertical line to perform the 3-D reconstruction. The results obtained while applying the calibration method proposed in this thesis indicate that the precision in localizing the z-value is very accurate using the vertical lines and imply that the error mostly exist in estimating the x and y values.

The dispersion of the global volume accuracy errors is shown in Figure 9.5. The dispersion is shown at the plane at $+50mm$ and $-50mm$. The radius of the green circle corresponds to the error obtained using the vertical lines in localizing the z value while the blue circles are used for the errors obtained using the horizontal lines. The radius size corresponds to the error multiplied by 10. This scaling is done in order to enable the visualization of the error inside the graphics.

The dispersion depicts that the errors using the horizontal lines are very constant, whereas the errors with the vertical lines increase as the points of interest are further from the projector.

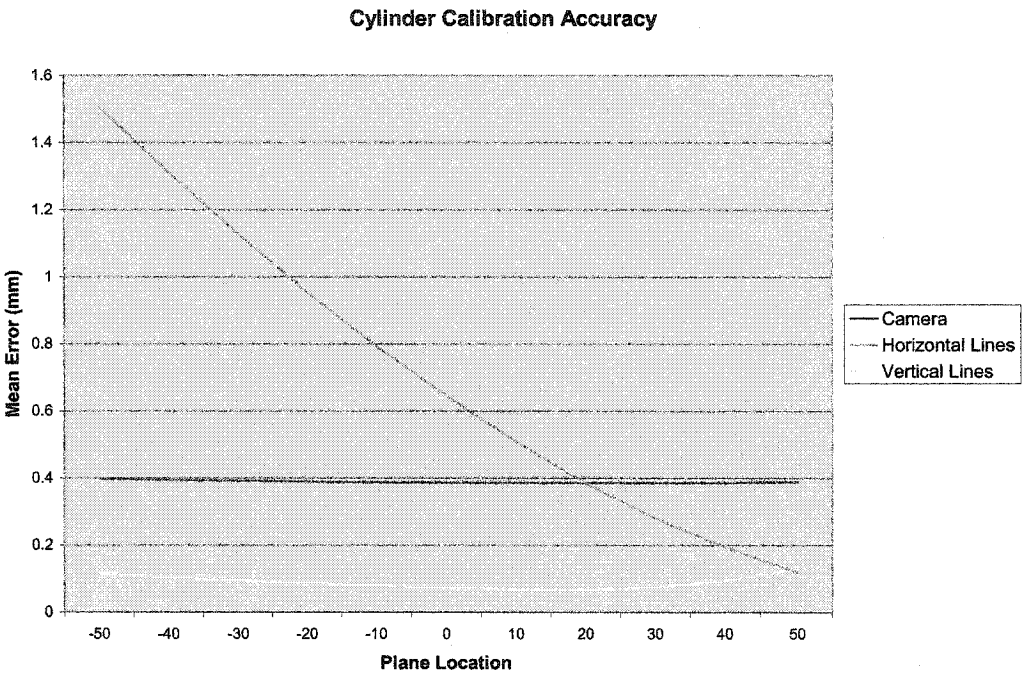


Figure 9.4: Volume accuracy at different Z value with the cylinder calibration object.

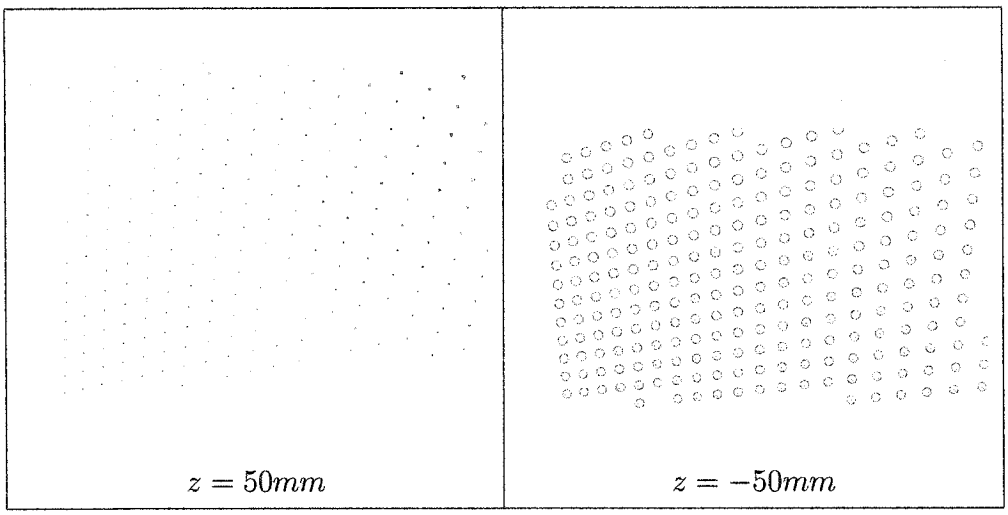
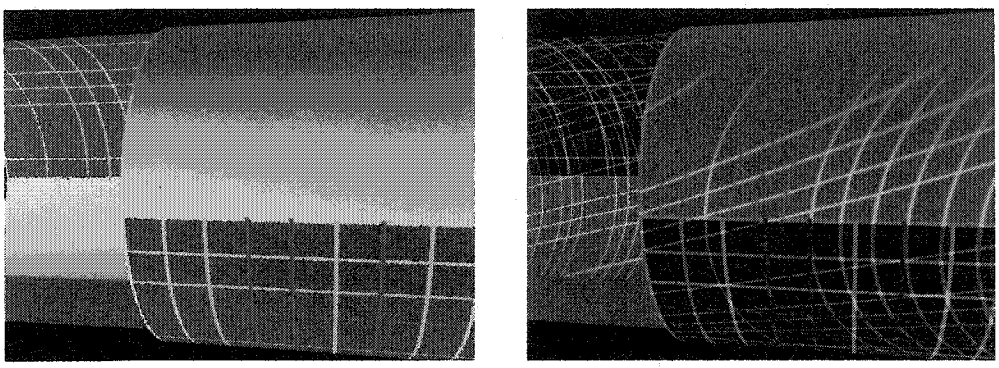


Figure 9.5: Global volume accuracy at 50 mm and -50 mm with the cylinder calibration object. The radius of a circle is the z-error $\times$ 10 with the horizontal line and vertical line errors using the blue and green colour respectively.

9.3.2 The Hollow Cylinder

There is a requirement to not move the calibration object to ensure a better precision of the 3-D reconstruction method. Tsai's methodology requires that the points be across a wide range of the image plane to ensure a good precision in the estimation of the calibration parameters. The cylinder calibration object fails to give points which cover a good percentage of the volume of interest.

The hollow cylinder object was introduced to obtain a broader depth of calibration points. With a section of the cylinder removed, it is possible to have points not only near the $+50mm$ region but also near the $-50mm$ region. Figure 9.6 depicts the camera and projector calibration images. Both images are of size 800 $\times$ 600 and the cylinder has a radius of 50 mm and a length of 200 mm. The hollow section has a length of 100 mm.



(a) Image for camera calibration.

(b) Image for projector calibration.

Figure 9.6: Calibration images with the hollow calibration cylinder.

This hollow cylinder calibration object can't be rotated since its geometry is not symmetrical. Two calibration objects could be needed to allow calibration to occur but the use of the calibration structured pattern proposed in this work and shown in Figure 8.2 solves the problem of requiring two calibration objects.

With the new structured grid pattern, the projector illuminates the white sections and the sections used for the camera calibration. To help the calibration process it is important to mask out from the projector image, the grid information which was used by the camera calibration. This is done by simply selecting the white pixels from the camera image and applying this mask to the projector image. The result is depicted in Figure 9.7.

The calibration results obtained with the hollow cylinder are given in Table 9.4. The results indicate that the camera calibration is less precise than the one obtained with the cylinder object (0.433 mm mean error instead of 0.29382 mm mean error). The difference can be explained by the limited number of intersection points which can be used by using the hollow cylinder calibration object

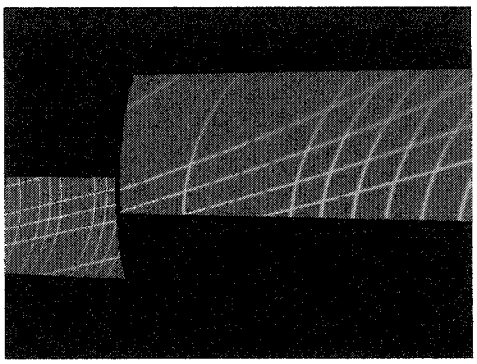


Figure 9.7: Masked image for projector calibration.

instead of the cylinder calibration object.

The volume accuracy of the camera with the hollow cylinder calibration object is listed in Table 9.5. The mean error is 0.06 mm higher than the accuracy observed in the cylinder object. The standard deviation is almost twice as much at -50 mm than at +50 mm and it is an order of magnitude higher than the standard deviation obtained using the cylinder object. However with a maximum standard deviation of 0.06973 mm, it is very stable.

The global volume accuracy of the hollow cylinder (see Table 9.6) is significantly more constant at all the different planes, than the global volume accuracy obtained with the cylinder object. It varies only by 0.0464 mm across all the planes with the hollow cylinder object instead of 1.3869 mm for the cylinder calibration object. The standard deviation in the case of the horizontal line accuracy test increases as the plane increases it's z value. The vertical line accuracy's behaviour is the opposite, it decreases as the z value increases. The behaviour of the global volume accuracy is depicted inside Figure 9.8 along with the volume accuracy of the camera.

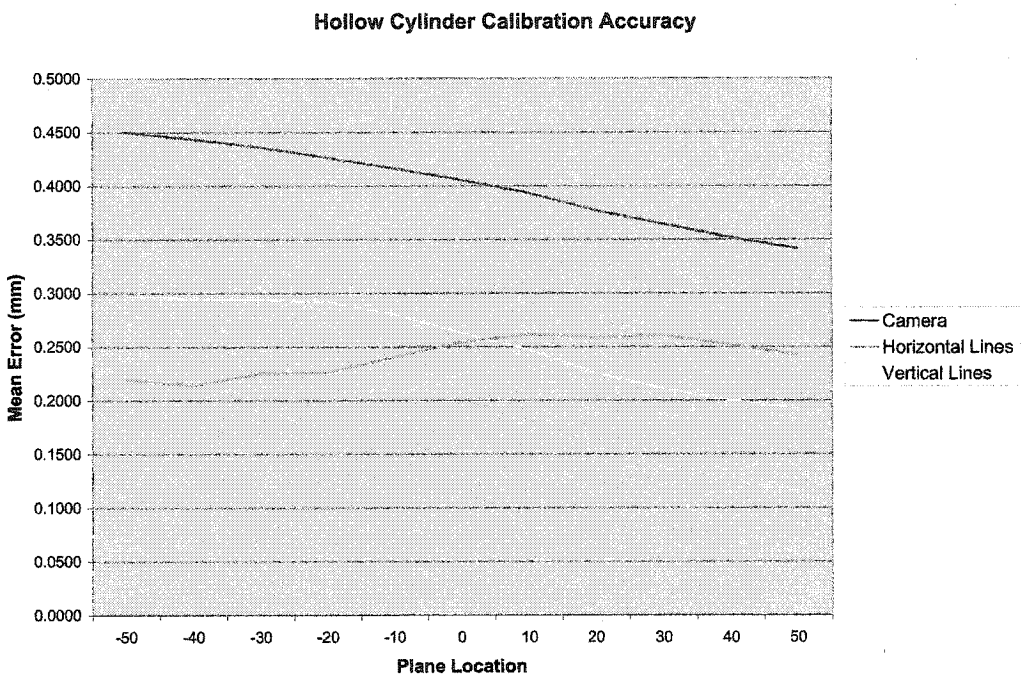


Figure 9.8: Volume accuracy at different Z value with the hollow cylinder calibration object.

Camera (56 points)	Mean	Standard Deviation	Max
Distorted image plane error	0.80827 pixel	0.51347 pixel	2.13556 pixel
Undistorted image plane error	0.17673 pixel	0.10732 pixel	0.52161 pixel
Object space error	0.03244 mm	0.01978 mm	0.08083 mm
Normalized calibration error	0.43300 mm	0.26293 mm	1.27798 mm
Projector (61 points)	Mean	Standard Deviation	Max
Distorted image plane error	0.24836 pixel	0.11027 pixel	0.51118 pixel
Undistorted image plane error	0.00926 pixel	0.00545 pixel	0.02515 pixel
Object space error	0.00923 mm	0.00539 mm	0.02460 mm
Normalized calibration error	0.02268 mm	0.01334 mm	0.06160 mm

Table 9.4: Calibration statistics from the hollow cylinder calibration object.

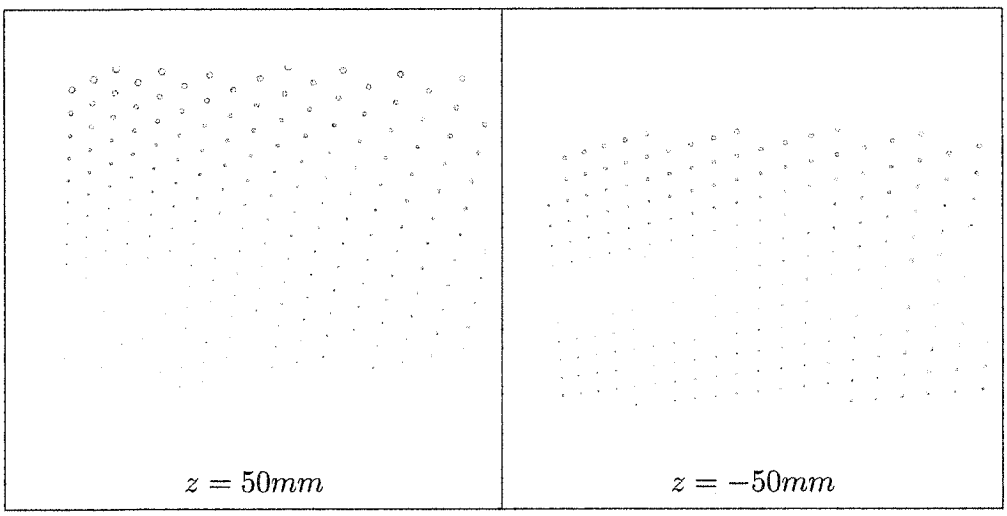


Figure 9.9: Global volume accuracy at 50 mm and -50 mm with the hollow cylinder calibration object. The radius of a circle is the z-error $\times$ 10 with the horizontal line and vertical line errors using the blue and green colour respectively.

Z (mm)	Mean (mm)	Min (mm)	Max (mm)	Std Dev (mm)	N
-50	0.45016	0.35336	0.64481	0.06973	549
-40	0.44346	0.35048	0.64293	0.06847	538
-30	0.43594	0.34998	0.61324	0.06560	540
-20	0.42617	0.32640	0.58791	0.06273	527
-10	0.41619	0.29481	0.56305	0.05931	524
0	0.40532	0.26675	0.54235	0.05740	511
10	0.39348	0.24326	0.51771	0.05567	512
20	0.37711	0.21861	0.49160	0.05570	485
30	0.36428	0.21761	0.48898	0.05157	436
40	0.35150	0.20480	0.48186	0.05009	359
50	0.34143	0.18815	0.48580	0.05311	314

Table 9.5: Volume accuracy of the camera at different Z values with the hollow cylinder calibration object.

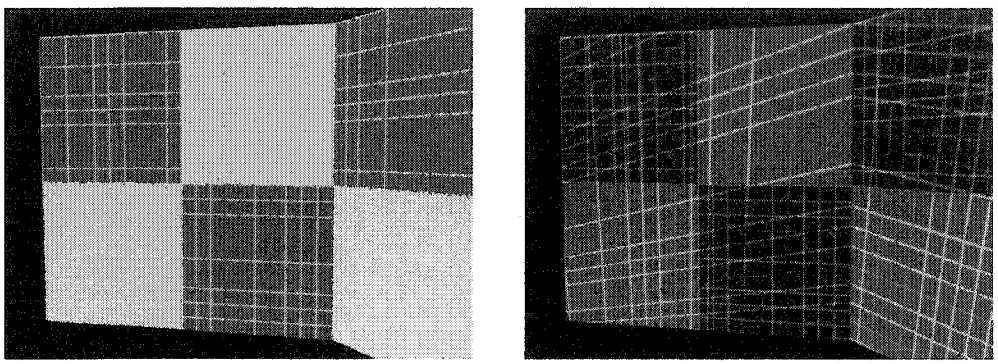
		Horizontal Lines				Vertical Lines			
Z (mm)	N	Mean (mm)	Min (mm)	Max (mm)	Std Dev (mm)	Mean (mm)	Min (mm)	Max (mm)	Std Dev (mm)
-50	238	0.21960	0.00032	0.76243	0.18730	0.29710	0.00226	1.14172	0.26431
-40	247	0.21408	0.00044	0.78560	0.20445	0.29525	0.00357	1.05159	0.24754
-30	260	0.22547	0.00057	0.83507	0.22488	0.29393	0.00061	1.08765	0.23976
-20	277	0.22599	0.00044	0.84373	0.22830	0.29226	0.00018	1.08045	0.22947
-10	291	0.24071	0.00039	0.87402	0.22935	0.28017	0.00039	0.99898	0.21564
0	307	0.25429	0.00023	0.90292	0.23828	0.26303	0.00097	0.90488	0.20180
10	320	0.26164	0.00065	0.94245	0.23951	0.24748	0.00081	0.96957	0.19464
20	331	0.25884	0.00066	0.98378	0.23934	0.22577	0.00084	0.88874	0.18241
30	309	0.26052	0.00056	1.08813	0.24341	0.21102	0.00006	0.87328	0.17549
40	273	0.25184	0.00189	1.11209	0.23672	0.20025	0.00204	0.79558	0.17539
50	238	0.24243	0.00025	0.95270	0.22618	0.19278	0.00005	0.80288	0.17382

Table 9.6: Global volume accuracy at different Z values with the hollow cylinder calibration object.

The Figure 9.9 shows that for both the images at $z = 50mm$ and $z = -50mm$, the error obtained with the horizontal lines is higher on the top-left corner and it decreases almost linearly toward the bottom-right corner. The exact opposite behaviour is seen on the error generated by the vertical lines as it increases when the points are further away from the top-left corner.

The hollow cylinder calibration object solves the problem of not having points that cover all the region of interest. It also introduces the concept of using a different pattern when calibrating. This new pattern removes the need to have two calibration objects when performing the calibration.

9.3.3 Two Planes



(a) Image for camera calibration.

(b) Image for projector calibration.

Figure 9.10: Calibration images with the two planes calibration object.

The calibration pattern introduced with the hollow cylinder can be applied to different calibration objects. One of the classic calibration objects is constructed by using two planes at a 135° angle. Each plane is of size $100mm \times 100mm$ and two different calibration patterns are printed on the planes. They are both similar to the ones used for the hollow cylinder however the pseudo random array covers a different region on each of the two planes. The first plane covers the region which the intersections varying from $(0, 0)$ to $(19, 19)$ and the second plane has the intersections ranging from $(20, 0)$ to $(39, 19)$.

Figure 9.10 depicts both calibration images. The images are of size 800×600. As is the case with the hollow cylinder, the projector's image has to be masked to remove the area which is used for the camera calibration.

The analysis of the camera calibration statistics listed inside Table 9.7 indicate that the distorted image plane error is higher than the one obtained with the

Camera (231 points)	Mean	Standard Deviation	Max
Distorted image plane error	0.17935 pixel	0.15777 pixel	0.71064 pixel
Undistorted image plane error	0.04829 pixel	0.03106 pixel	0.14456 pixel
Object space error	0.00870 mm	0.00563 mm	0.02487 mm
Normalized calibration error	0.11827 mm	0.07609 mm	0.35420 mm
Projector (151 points)	Mean	Standard Deviation	Max
Distorted image plane error	0.05606 pixel	0.04439 pixel	0.17812 pixel
Undistorted image plane error	0.02408 pixel	0.01471 pixel	0.06160 pixel
Object space error	0.02581 mm	0.01634 mm	0.07288 mm
Normalized calibration error	0.05894 mm	0.03600 mm	0.15075 mm

Table 9.7: Calibration statistics from the two planes calibration object.

cylinder calibration object (0.17935 pixel instead of 0.13335 pixel). The standard deviation is also higher for the distorted image plane (0.15777 pixel instead of 0.08489 pixel). However, the object space error (0.0087 mm instead of 0.01651 mm) and normalized calibration error (0.11827 mm instead of 0.29382 mm) is smaller by a factor of 1.898 and 2.48 respectively. A smaller error in the normalized calibration is more important than a higher error at the undistorted plane.

The smaller normalized calibration error translates itself in a volume accuracy with camera that is from 2.77 times (at $z = 50$) to 15.456 times (at $z = 0$) more accurate across the volume of interest. Table 9.8 lists the statistical results obtained for the volume accuracy with the camera with the two plane calibration object. The standard deviation is also very small since it has values varying from 0.01460 mm to 0.02216 mm.

The global volume accuracy is listed inside Table 9.9. This accuracy when

Z (mm)	Mean (mm)	Min (mm)	Max (mm)	Std Dev (mm)	N
-50	0.13426	0.09482	0.20482	0.02216	549
-40	0.11068	0.07371	0.19139	0.02165	538
-30	0.08800	0.05291	0.16135	0.02207	540
-20	0.06471	0.03263	0.14844	0.02092	527
-10	0.04272	0.01310	0.12509	0.02127	524
0	0.02498	0.00065	0.10900	0.01952	511
10	0.03095	0.00118	0.09929	0.01758	512
20	0.05166	0.00249	0.09231	0.01460	485
30	0.07801	0.01825	0.11007	0.01520	436
40	0.10731	0.05484	0.14053	0.01509	359
50	0.14043	0.08124	0.17479	0.01817	314

Table 9.8: Volume accuracy of the camera at different Z values with the two planes calibration object.

		Horizontal Lines				Vertical Lines			
Z (mm)	N	Mean (mm)	Min (mm)	Max (mm)	Std Dev (mm)	Mean (mm)	Min (mm)	Max (mm)	Std Dev (mm)
-50	238	0.25501	0.00214	0.78022	0.19827	0.06119	0.00061	0.26105	0.04265
-40	247	0.21971	0.00123	0.67990	0.18241	0.04979	0.00020	0.21549	0.03848
-30	260	0.18879	0.00039	0.63382	0.16253	0.04356	0.00002	0.24385	0.03802
-20	277	0.17000	0.00225	0.59662	0.14863	0.03664	0.00000	0.22813	0.03390
-10	291	0.15680	0.00320	0.56661	0.13342	0.03193	0.00003	0.22257	0.03056
0	307	0.14639	0.00001	0.52980	0.12027	0.02955	0.00023	0.24280	0.03068
10	320	0.14469	0.00082	0.48198	0.11285	0.02996	0.00030	0.19492	0.03072
20	331	0.14524	0.00011	0.49703	0.11317	0.03246	0.00038	0.17934	0.03072
30	309	0.14880	0.00042	0.58179	0.12505	0.03867	0.00013	0.16858	0.02759
40	273	0.15323	0.00015	0.61206	0.13752	0.04570	0.00006	0.20172	0.02815
50	238	0.16406	0.00016	0.65815	0.15008	0.05088	0.00078	0.16617	0.02781

Table 9.9: Global volume accuracy at different Z values with the two planes calibration object.

calibrating with two planes is also increased compared to both the cylinder and the hollow cylinder object. The mean error using the horizontal lines is higher than the cylinder object at $z = 50$, however this error increases only slightly when the z value decreases. The values vary from 0.1641 mm to 0.2550 mm for the two planes calibration object whereas these values vary from 0.1200 to 1.5069 for the cylinder. The accuracy of the hollow cylinder is more constant as it varies from 0.2424 mm to 0.2196 mm , however it also has a higher average value than the two planes calibration method with an average of 0.2479 mm instead of 0.1065 mm for the two plane calibration object.

The two planes calibration method is really precise when the intersection points are projected on the vertical lines. The mean error in that case has a maximum value of 0.0612 mm at $z = -50mm$. This error is 4.85 times smaller than the one obtained for the hollow cylinder at the same z value (0.2971 mm) and 1.90 times smaller than the value obtained with the cylinder object (0.1161 mm).

Figure 9.11 highlights the fact that the accuracy error increases as one moves away from the $z = 0$ value. As for all the other calibration objects, the error is higher when the horizontal lines are used to estimate the location of the intersection point in 3-D.

The dispersion of the errors is not uniform across the plane at $z = 50mm$ as can be seen in Figure 9.12.

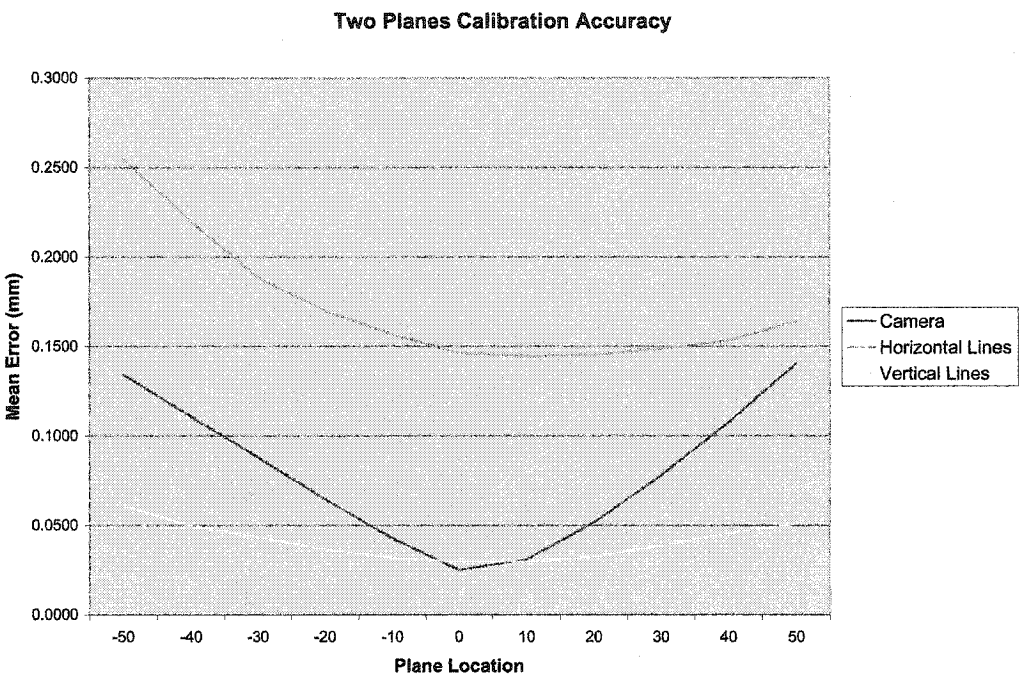


Figure 9.11: Volume accuracy at different Z value with the two planes calibration object.

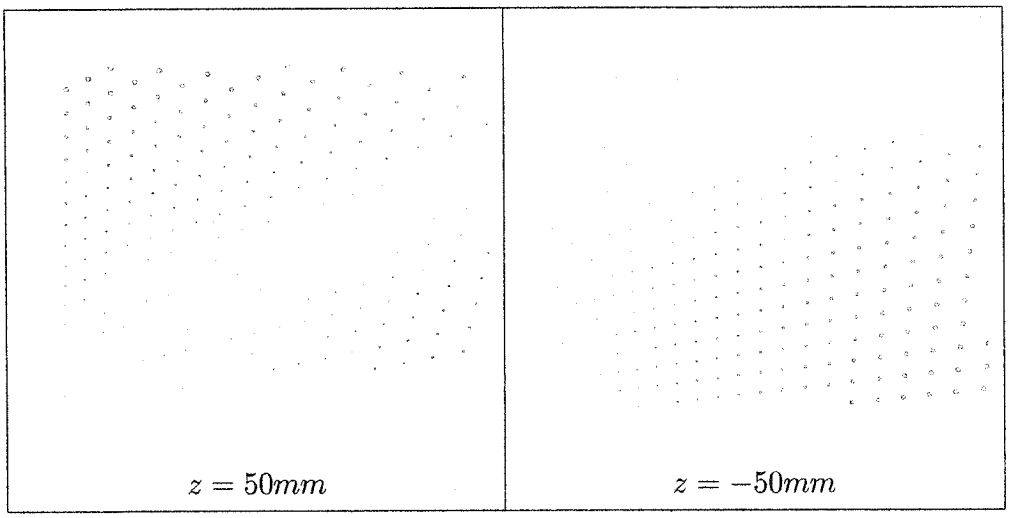
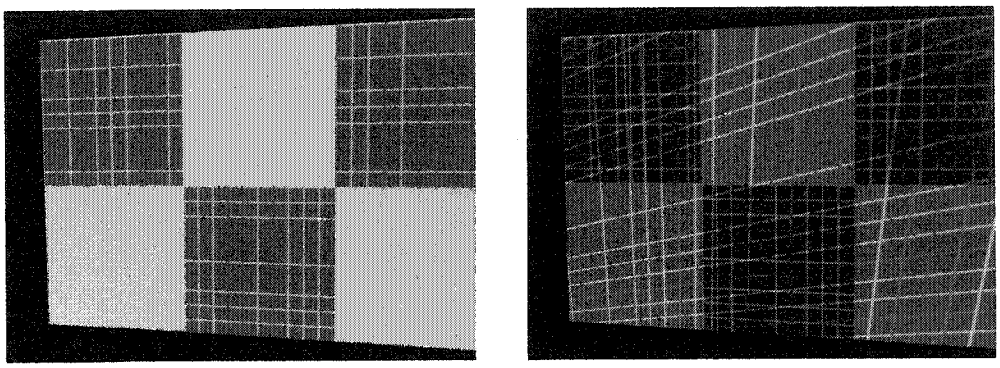


Figure 9.12: Global volume accuracy at 50 mm and -50 mm with the two planes calibration object. The radius of a circle is the z-error $\times$ 10 with the horizontal line and vertical line errors using the blue and green colour respectively.

9.3.4 Single Plane

The calibration with co-planar points using a single plane was originally performed to show how co-planar calibration is less accurate than non co-planar calibration. However, the results do not warrant this observation. The single plane is a plane of size $200mm \times 100mm$ and it uses an encoded structured pattern which covers the same region as for the two planes calibration object. Figure 9.13 depicts both calibration images. The images are of size 800×600. As is the case with the hollow cylinder, the projector's image has to be masked to remove the area which is used for the camera calibration.

The calibration using the single plane results in statistics which are smaller than all other calibration objects (see Table 9.10). The object space error is 0.00359 mm whereas it is 0.01651 mm, 0.03244 mm and 0.00870 mm respec-



(a) Image for camera calibration.

(b) Image for projector calibration.

Figure 9.13: Calibration images with the single plane calibration object.

Camera (224 points)	Mean	Standard Deviation	Max
Distorted image plane error	0.02103 pixel	0.01105 pixel	0.06440 pixel
Undistorted image plane error	0.01923 pixel	0.01055 pixel	0.06454 pixel
Object space error	0.00359 mm	0.00200 mm	0.01255 mm
Normalized calibration error	0.04710 mm	0.02585 mm	0.15809 mm
Projector (128 points)	Mean	Standard Deviation	Max
Distorted image plane error	0.01023 pixel	0.00536 pixel	0.02981 pixel
Undistorted image plane error	0.00727 pixel	0.00398 pixel	0.02290 pixel
Object space error	0.00771 mm	0.00404 mm	0.02401 mm
Normalized calibration error	0.01780 mm	0.00974 mm	0.05610 mm

Table 9.10: Calibration statistics from the single plane calibration object.

Z (mm)	Mean (mm)	Min (mm)	Max (mm)	Std Dev (mm)	N
-50	0.24497	0.01277	0.52733	0.13457	549
-40	0.19413	0.00914	0.41663	0.10500	538
-30	0.14831	0.00407	0.32213	0.07922	540
-20	0.09966	0.00519	0.21913	0.05140	527
-10	0.05085	0.00151	0.11280	0.02596	524
0	0.00435	0.00028	0.01341	0.00230	511
10	0.05538	0.00264	0.11473	0.02587	512
20	0.11028	0.00621	0.22595	0.04861	485
30	0.16878	0.01068	0.34806	0.07475	436
40	0.22157	0.00919	0.44731	0.09790	359
50	0.28532	0.01517	0.59713	0.12959	314

Table 9.11: Volume accuracy of the camera at different Z values with the single plane calibration object.

tively for the cylinder, hollow cylinder and the two plane calibration objects. The projector's normalized calibration error for the single plane calibration object is 0.0178 mm whereas it is 0.02944 mm for the cylinder, 0.02268 mm for the hollow cylinder and 0.05894 mm for the two planes calibration object.

The volume accuracy of the camera obtained with the single plane calibration object is very similar to the one obtained with the two planes calibration object (compare Table 9.11 with Table 9.8). It is minimal at $z = 0$ and maximum at both extremities. The minimum error is smaller than the one of the two planes calibration, however the maximum errors are twice as big as the two planes calibration object. The standard deviation is also proportional with the distance from

		Horizontal Lines				Vertical Lines			
Z (mm)	N	Mean (mm)	Min (mm)	Max (mm)	Std Dev (mm)	Mean (mm)	Min (mm)	Max (mm)	Std Dev (mm)
-50	238	0.18679	0.00073	0.43465	0.11457	0.18385	0.00096	0.53075	0.14617
-40	247	0.15751	0.00214	0.35778	0.09437	0.13476	0.00216	0.38174	0.10527
-30	260	0.12340	0.00063	0.27473	0.07253	0.09334	0.00056	0.27623	0.07119
-20	277	0.09007	0.00019	0.20459	0.04956	0.06050	0.00009	0.17214	0.04365
-10	291	0.05445	0.00106	0.11756	0.02438	0.03074	0.00001	0.10277	0.02204
0	307	0.02480	0.00001	0.17093	0.02696	0.00954	0.00001	0.05067	0.00664
10	320	0.06481	0.00016	0.22613	0.03400	0.03040	0.00009	0.07066	0.01710
20	331	0.11937	0.00396	0.31279	0.05477	0.06176	0.00031	0.13209	0.03494
30	309	0.17466	0.00116	0.35670	0.07944	0.09360	0.00005	0.18471	0.05280
40	273	0.23318	0.00110	0.38897	0.10317	0.12423	0.00462	0.24103	0.07241
50	238	0.29659	0.00381	0.48939	0.12947	0.15666	0.00013	0.30092	0.09047

Table 9.12: Global volume accuracy at different Z values with the single plane calibration object.

the plane at $z = 0$. The standard deviation for the single plane calibration is the worst of all calibration objects. Its value peaks at 0.13457 mm at $z = -50mm$ whereas the closest results obtained was 0.06973 mm of standard deviation for the hollow cylinder. While the other calibration objects were able to obtain a standard deviation which remained mostly constant, the one obtained with the single plane calibration varies with the value of z.

The global volume accuracy obtained with the single plane is shown in Table 9.12. It is also very accurate with a mean value of 0.0890 mm obtained with the

vertical lines compared to the 0.0904 mm obtained with the cylinder calibration object and like the volume accuracy with the camera, it varies with the value of z (see Figure 9.14). This is the situation also observed with the two plane calibration. The global volume accuracy obtained with the horizontal lines is equivalent to the one obtained by the two planes method. This global volume accuracy varies from 0.2966 mm to 0.0248 mm and the global volume accuracy of the horizontal lines for the two planes calibration object varies from 0.2550 mm to 0.1447 mm. The global volume accuracy obtained with the vertical line is worse than the two planes method as it varies from 0.0095 mm to 0.1839 mm with a mean of 0.0890 mm as opposed to 0.0295 mm to 0.0612 mm with a mean of 0.0409 mm obtained with the two planes calibration object. This mean value is slightly smaller than the one obtained with the cylinder calibration object (0.0904mm) but it is 2.86 times more precise than the average of 0.2545 mm obtained with the hollow cylinder.

The dispersion of the errors is dependent on the point's distance to the projector for both the $z = 50mm$ and $z = -50mm$ planes. This situation is observed in Figure 9.15.

9.4 Results with a Synthetic Image

The structured light is projected on the famous Utah teapot [3]. The Utah teapot was one of the first free 3-D models based on B-spline. It is also an interesting object to test with. It is instantly recognizable. It has a complex topology. It self-shadows, has hidden surface issues and both convex and concave surfaces. The Utah teapot is defined with B-splines patches. This simplifies the process of analyzing the error between the known Utah teapot and the surface generated with the

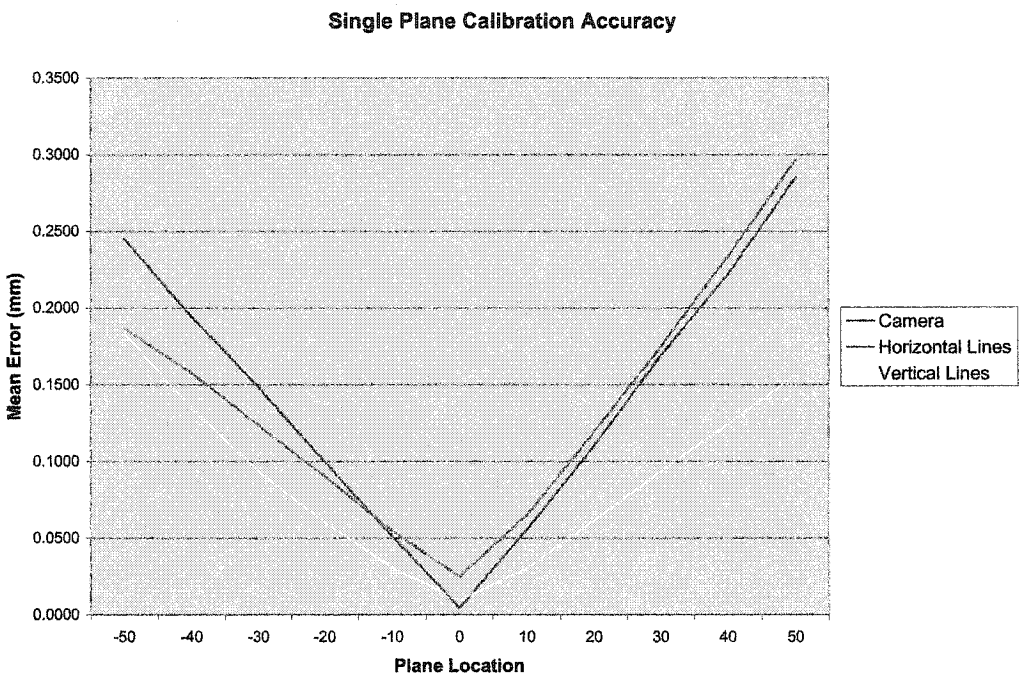


Figure 9.14: Volume accuracy at different Z value with the single plane calibration object.

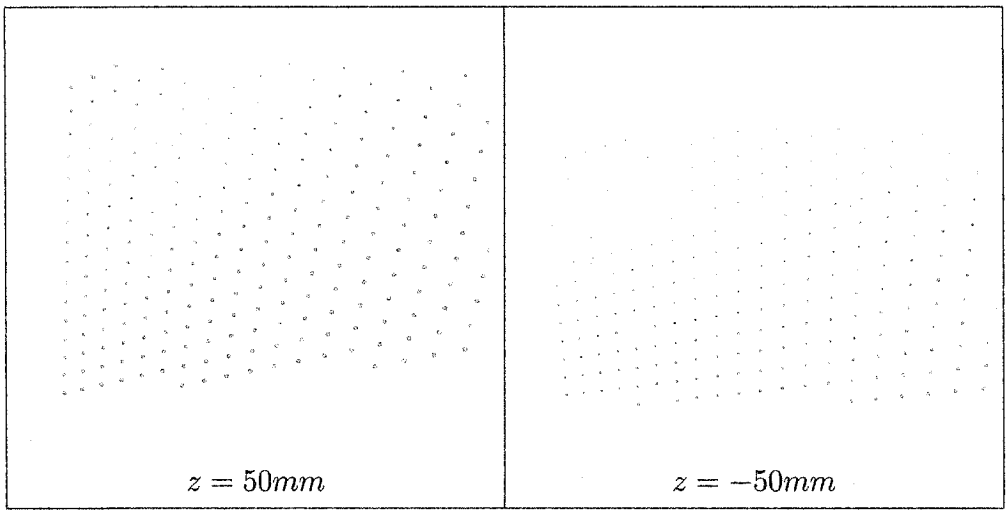
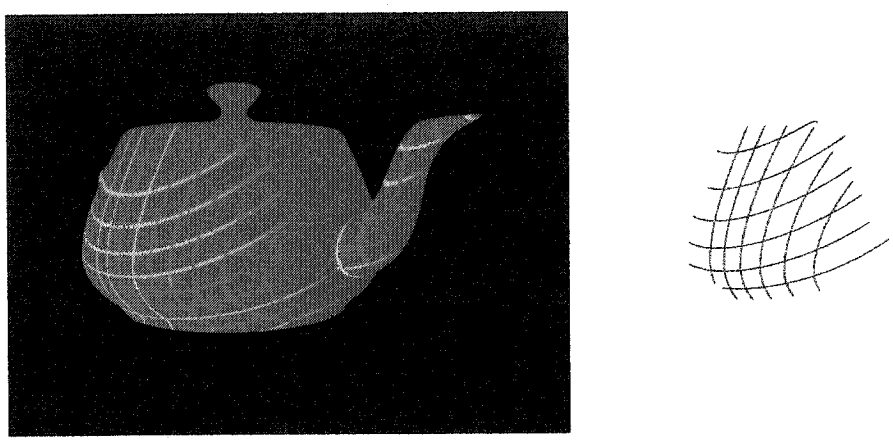


Figure 9.15: Global volume accuracy at 50 mm and -50 mm with the single plane calibration object. The radius of a circle is the z-error $\times$ 10 with the horizontal line and vertical line errors using the blue and green colour respectively.

methodology proposed for 3-D reconstruction. Using the calibration parameters obtained with the two plane calibration object, the 3-D lines are then projected into 3-D space to create a view of the object. This view is limited to the grid lines which have sufficient lighting conditions.

The data to generate a raytracing of the Utah teapot is a set of B-splines patches. This set of B-spline patches can be compared with the set of NURBS surfaces generated by the methodology presented in thesis in order to compute the error in 3-D reconstruction. The results obtained with various stages of the methodology are shown in Table 9.13.

Definition 9.2. *The error between the Utah teapot and a NURBS curve or NURBS surface is the distance between a point on the NURBS curve or surface and the point on the Utah teapot that is closest to it.*



(a) (b)

Figure 9.16: (a) The Utah teapot illuminated by the structured light. (b) The 2-D lines detected by the methodology.

The Table 9.13 shows results for the vertical and horizontal grid lines projected in the 3-D world coordinate system. It also shows the errors when someone uses skinning from the vertical lines or the horizontal lines. Finally, the error obtained when using the Gordon surface is also shown. The results were all obtained by using the image shown in Figure 9.16(a) as the input image, then extracting the lines generated (see Figure 9.16(b)) and the final 3-D result is shown with two different views in Figure 9.17.

The Z value of the lines varies between 25 mm and 40 mm. The results obtained with the NURBS vertical grid are consistent with the results obtained inside Table 9.9 for the region near $z = 40$. However, the results for the horizontal NURBS grid are better than those of Table 9.9 as the average error obtained is 0.1532 mm at 40 mm whereas the NURBS curve obtained an average error of 0.05386 mm on the range 25 mm to 40 mm.

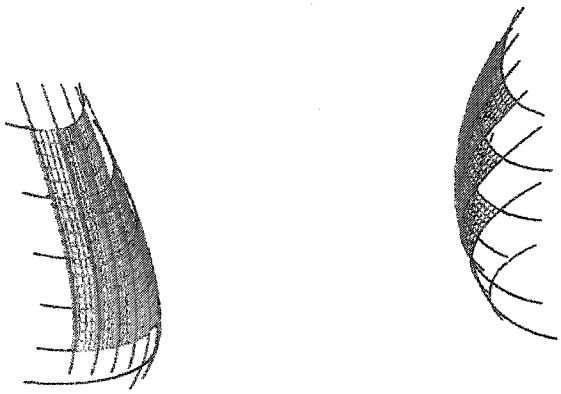
Object	Mean (mm)	Std Dev (mm)	Min (mm)	Max (mm)
Horizontal Grid Lines	0.110745	0.0986030	0.00179233	0.465082
Vertical Grid Lines	0.0538629	0.025974	0.00384417	0.110581
Horizontal Skin Surface	0.0940525	0.0792902	0.000346476	0.558296
Vertical Skin Surface	0.121947	0.0884404	0.000663184	0.449805
Gordon Surface	0.0305820	0.0391224	0.0000217415	1.05252

Table 9.13: Accuracy with the Utah teapot.

Definition 9.3. *The error between a NURBS surface and the Utah teapot is computed by taking 100 samples across that surface and then finding the closest distance between those samples and the Utah teapot surface. The error for that sample is the average of these closest distances.*

The results obtained with skinning highlight the fact that if the surface is smooth then skinning obtains efficient results which are close to the error obtained with the grid lines: 0.110745 mm for the horizontal grid lines compared to 0.0940525 mm for the horizontal skin surface.

The maximum error obtained with the Gordon surface generated is localized to a single surface patch and not to the entire set of surfaces that are required to cover the surface. The average error obtained with the Gordon surface is smaller than the one obtained with simply the grid lines. It should be noted that although the vertical and horizontal lines projection do not match perfectly, the Algorithm 7.3 accounts for this and allows the system to obtain a better surface representation than simple skinning could obtain.



(a) (b)

Figure 9.17: Generated views of the Utah teapot. The view in (a) is from the left and the view in (b) is from the right of the object.

9.5 Results with Camera

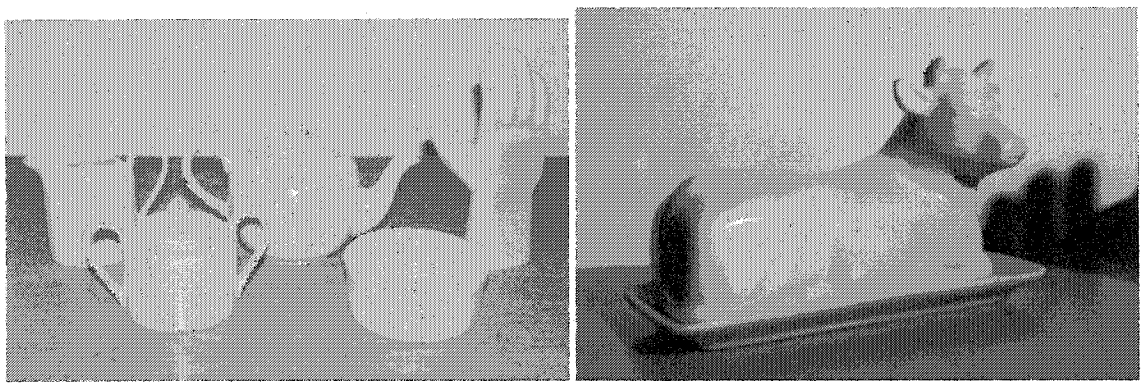
The test images are taken with a Canon Digital Rebel D300 camera. The effective size of the pictures taken are 1024×1536 pixels. An overhead projector made by In Focus Systems (model Overview 360P) is used to project the grid on the object of interest. The grid is printed on a transparency sheet using an inkjet printer (HP 5550). The total cost of the system is below 2000$ excluding the computer cost.

Vibration can be a factor with a dynamic stereo system. However, the sensibility of the system to vibrations is negligible due to the fact that the camera's aperture speed is much slower than the ambient vibrations which are present in a room. Due to the low light conditions of the set-up, an aperture speed of 1/60 of a second was used in all the experiments (except for the hand object). At that relatively slow speed, the camera will average most or all the high-frequencies vibrations. The precision will not be affected as subpixel detection is actually more robust when the step-edge is not a true step-edge [16].

Various objects were used to test the methodology: a hand generated with a stereo lithography (SLA) system, a teapot, a cup, a sugar bowl, a milk holder and a cow shaped butter plate. Except for the hand, the objects have a glazed ceramic finish. The objects can be seen in Figure 9.18 except for the cow shaped butter plate. The sections which follow analyze the results obtained with each object.

9.5.1 Digital Light Processing Projector

The first experiments were made using a digital light processing (DLP) projector. Those kinds of projectors are common. They offer good precision and they open up some possibilities in terms of projecting structured information such as using



(a) (b)

Figure 9.18: (a) Hand, teapot, cup, milk holder and sugar bowl objects used to test the methodology. (b) Cow shaped butter holder to test the methodology.

temporal projection to increase the accuracy of the technique or to get more coverage of a scene. However, they work pixel by pixel. This causes some problem when the system analyzes the information contained within the picture.

Figure 9.19 shows a zoomed out view of an image taken with a DLP. In that picture, it is easy to see that artificial artifacts are added by the DLP and that they can hinder the ability of the system to retrieve the edge definition.

It is difficult to perform the edge extraction on an image projected by a DLP. It is possible to apply Gaussian blurring techniques to remove some of that noise, however since there is an inherent problem to the way the structured light is projected. Consequently, it was found unsatisfactory for the purposes of this thesis. Instead of a DLP projector an overhead projector was used.

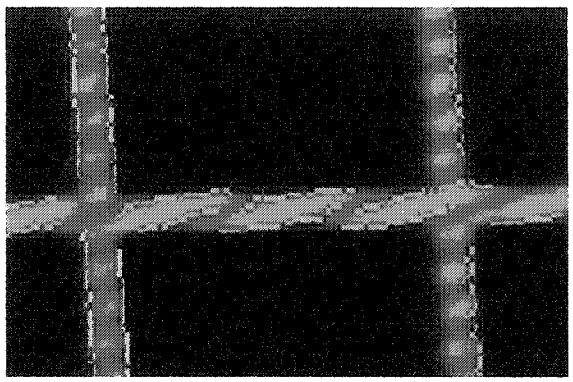


Figure 9.19: Structured light projected by a DLP projector.

9.5.2 Line Detection

The Focus overhead projector and an inkjet printer allow the system to project a structured light containing less artifacts than the one obtained with a DLP. It should be noted that this is because the colour chosen (cyan, magenta, yellow, green and red) are saturated colours and the inkjet can simply use 100% of the ink on every spot of the grid. When the colors are not saturated, dithering will occur and create artifacts when projected on a surface.

The line detection proposed in Chapter 6 has several advantages. The first is that it doesn't require any smoothing operator to be applied to the image before edge detection. This keeps edges sharp. However, it's efficiency varies depending on the type of material used since the light is being absorbed or reflected differently for each material.

The above is true after a raw image obtained from the Canon digital camera is halved in size. The raw image has a size of 2048×3072 pixels, and the images used in this thesis are 1024×1536 pixels. The halving process uses bi-linear

mapping to compute the new value of a pixel. This removes the noise caused by the low light conditions inside the images. The line detection algorithm was unstable with the unprocessed raw images obtained from the Canon digital camera but its behaviour was very solid once a smaller picture size was used. Other means of noise removal could be used in future work to allow the methodology to work with full size images.

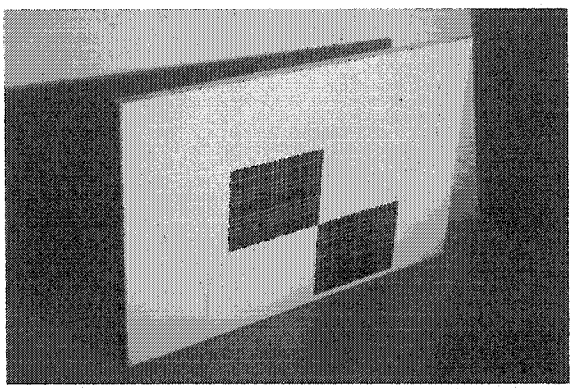
9.5.3 The Calibration

The calibration for the camera was done by printing a colour coded grid using a dye-sub printer (Olympus P-400). That grid was taped to a board and then photographed with the camera. This setup is not the most precise but it can be representative of real world usage of the methodology presented in this thesis.

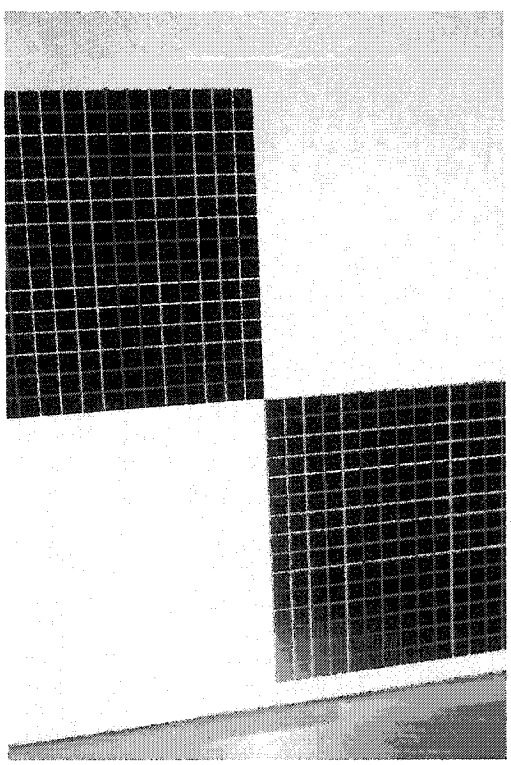
Figure 9.20 shows three subfigures: the calibration object from a distance, the image used to calibrate the camera and the image used to calibrate the projector.

The photos in Figure 9.20(a) and Figure 9.20(b) are used to calibrate the system. The calibration results obtained from the camera and listed in Table 9.14 may lead one to believe that great accuracy can't be achieved. However, the normalized precision of the overall calibration is of 0.08901 mm which corresponds to 3.5 tau ($\frac{1}{1000}$ inch). The volume of interest covers a $300 \times 300 \times 100$ mm^3 zone which means that the error in one direction is 0.02967% of the total range being covered.

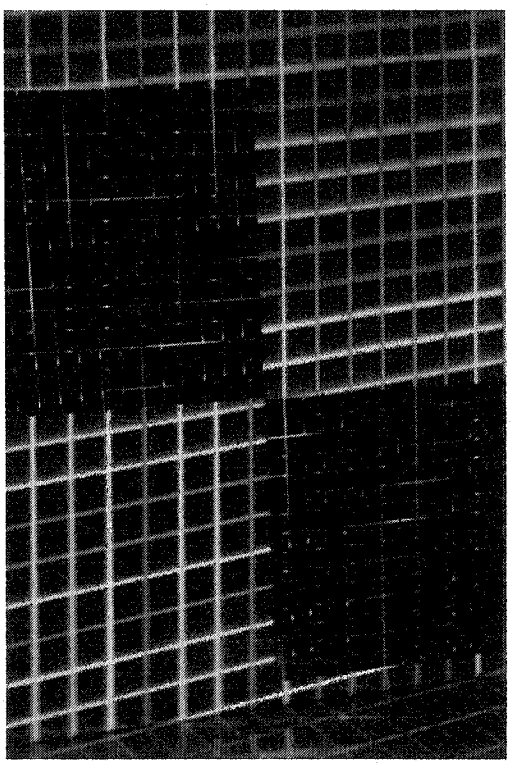
The initial high level of error when calibrating the camera is partially due to the nature of a grid. The grid is constructed by first painting the horizontal lines then the vertical lines. This implies that if a line which resides on top of another is of a different hue, it is possible that the system will view the difference between the



(a)



(b)



(c)

Figure 9.20: (a) Image of the calibration object. (b) Image for camera calibration. (c) Image for projector calibration.

Camera (325 points)	Mean	Standard Deviation	Max
Distorted image plane error	1.74785 pixel	1.69141 pixel	9.90764 pixel
Undistorted image plane error	0.43621 pixel	0.25457 pixel	1.50518 pixel
Object space error	0.03518 mm	0.01833 mm	0.11159 mm
Normalized calibration error	1.06848 mm	0.62357 mm	3.68693 mm
Projector (105 points)	Mean	Standard Deviation	Max
Distorted image plane error	0.32076 pixel	0.19119 pixel	0.72064 pixel
Undistorted image plane error	0.03634 pixel	0.02091 pixel	0.08379 pixel
Object space error	0.08408 mm	0.04846 mm	0.18897 mm
Normalized calibration error	0.08901 mm	0.05121 mm	0.20524 mm

Table 9.14: Calibration results with real images.

hue of one line and the other lines indicating the presence of an edge. Figure 9.21 depicts this situation.

The presence of bad intersection points (or outliers) explain the high maximum error seen in the Table 9.14. These outliers can be removed by using a bucketing technique such as the one proposed in Section 6.2.

It should be noted that this situation mostly affects the image used to calibrate the camera. For the projector calibration image and the images of the object being scanned, the projected light saturates the image and creates an even distribution of values between all the grid lines.

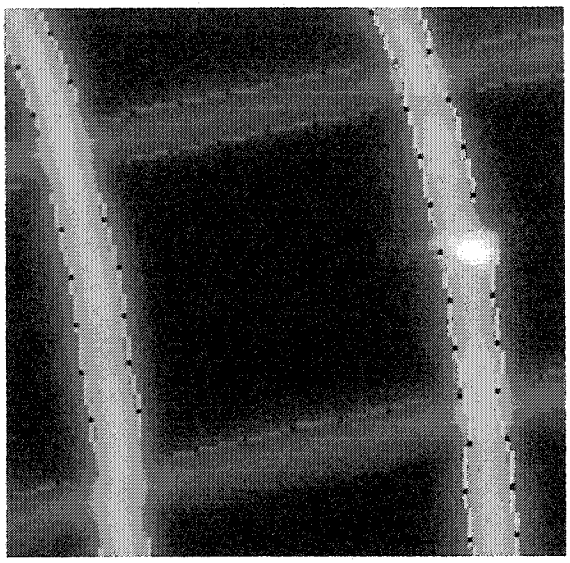


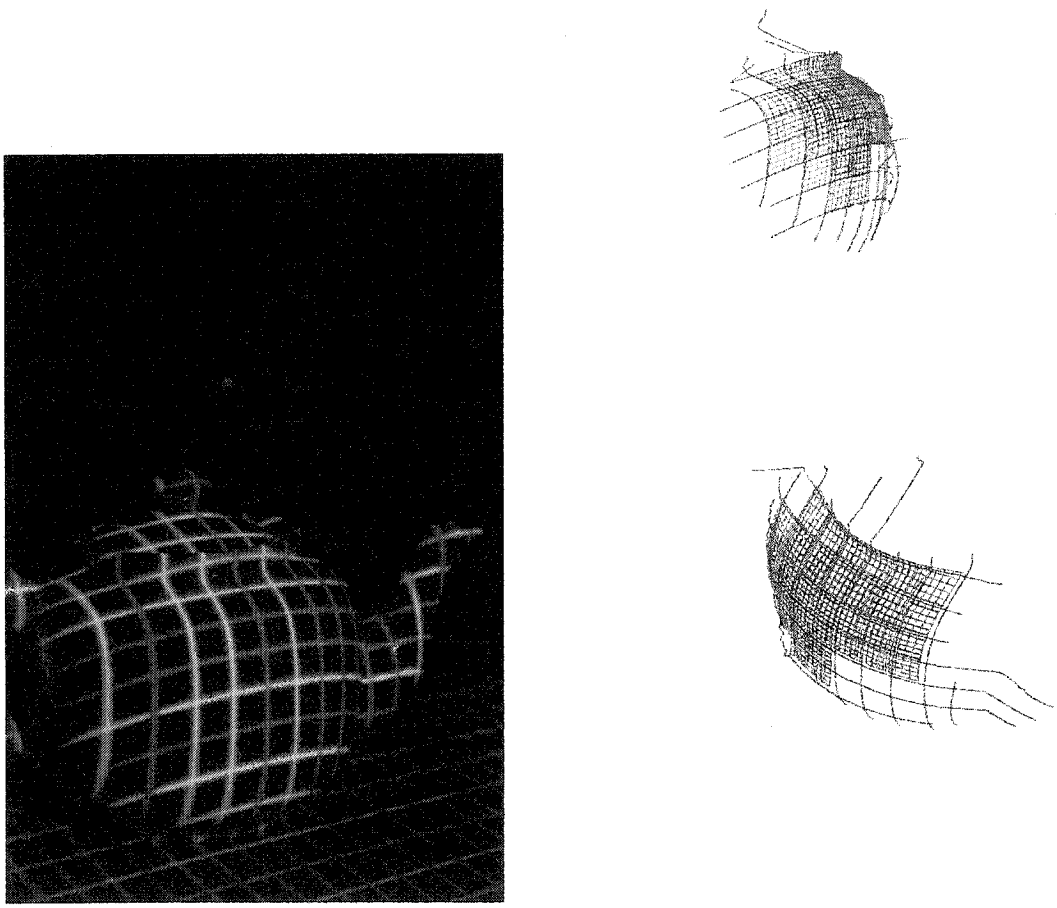
Figure 9.21: Image of structured light and detected edges.

9.5.4 The Teapot

The teapot object, as seen in Figure 9.22(a), measures 220 mm from spout to handle. Its height is 120 mm and its diameter is 130 mm. The 3-D reconstruction of this object is shown in Figure 9.22(b).

In the reconstruction, one can observe on the upper left corner a line which extrudes from the teapot. This is caused by two factors. The first is that the system failed to properly estimate the colour of the line as yellow instead of cyan. The second is that the system failed to detect that the segment is a vertical segment. Figure 9.23 illustrates this. The NURBS surface generation process is resilient to such problems.

The NURBS surface process relies on contiguous sets of intersection points lying on the network of NURBS curves. Trouble segments such as the one seen



(a) Input image

(b) Generated surface from two different view points.

Figure 9.22: Input image and the 3-D representation of the teapot object.

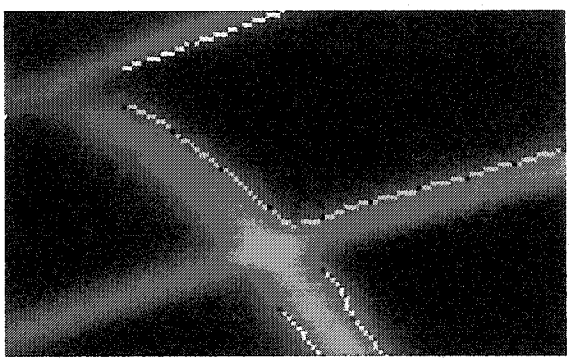


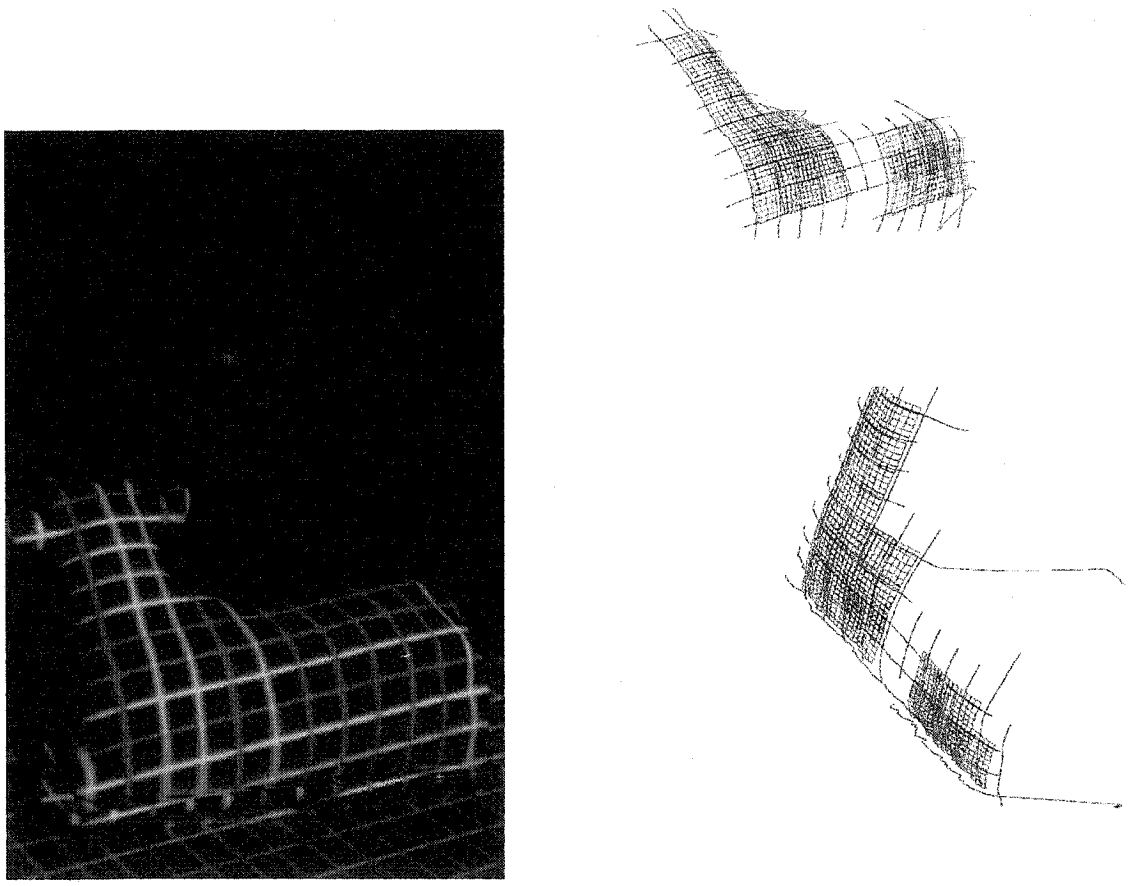
Figure 9.23: Error in edge detection.

inside Figure 9.23 are therefore always excluded from the surface generation algorithm. This really increases the quality of the generated surface since only segments with strong correlation (i.e. part of a contiguous region) affect the generation of the NURBS surface.

9.5.5 The Cow Butter Holder

This object is a butter holder in the shape of a cow (see Figure 9.24(a)). Like the teapot object, it is made of white ceramic and it reflects light in a similar manner. It is relatively smooth. Figure 9.24(b) shows the results obtained when generating this object.

There is no surface being generated in the middle of the cow butter holder object. This is because the process needs a contiguous region to work from and in this case, the middle section doesn't contain enough contiguous intersection points.



(a) Input image

(b) Generated surface from two different view points.

Figure 9.24: Input image and the 3-D representation of the cow butter holder object.

9.5.6 The Milk Holder

The milk holder is also made from white ceramic (see Figure 9.25(a)). It is a smooth 3-D surface except near the beak where there is a smooth edge (see Figure 9.25(b)). The smooth edge is properly captured by the algorithm and to illustrate this, a section of the image is zoomed (see Figure 9.26(b)) and its corresponding 3-D representation is shown Figure 9.26(a).

9.5.7 The Tea Cup

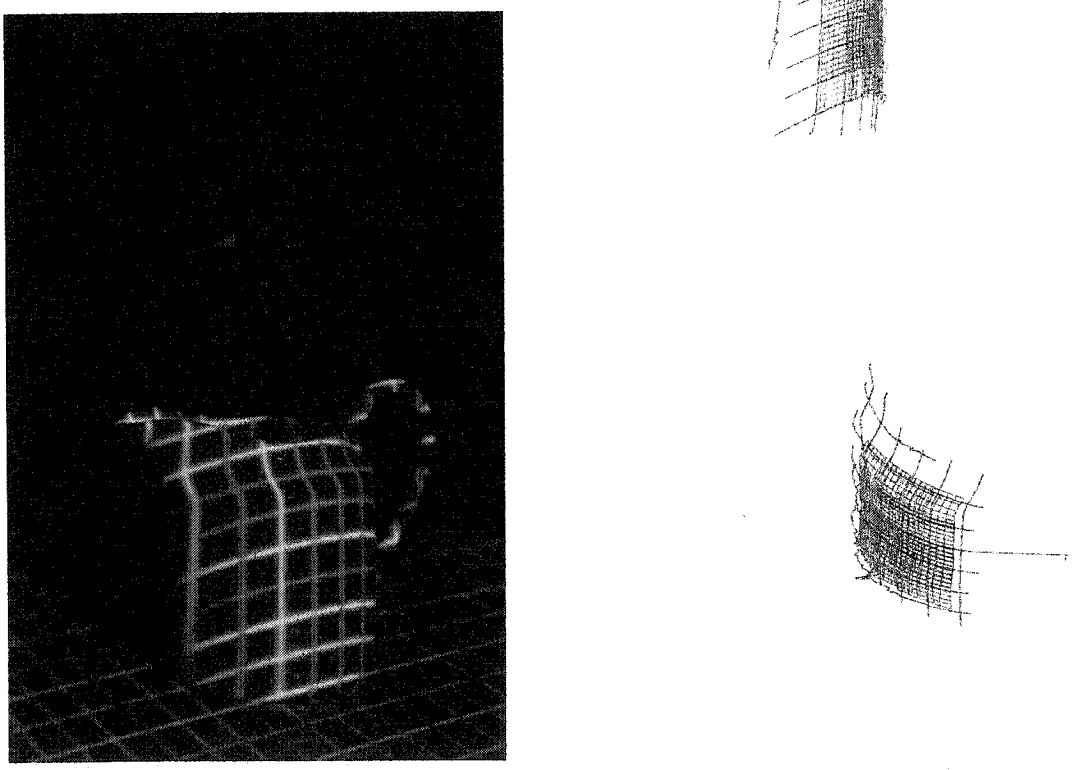
The tea cup is made from white ceramic. The bottom of it is smooth and the top part contains some flower patterns in 3-D. The flower pattern lies between grid lines (see Figure 9.27(a)) and only the sections that are on a grid line are affecting the 3-D surface generation. The results are shown in Figure 9.27(b).

9.5.8 The Sugar Bowl

The sugar bowl is a smooth white ceramic object which contains a hard edge (see Figure 9.28(a)). As Figure 9.28(b) depicts, this hard edge is properly captured by the system.

9.5.9 The Egg Bowl

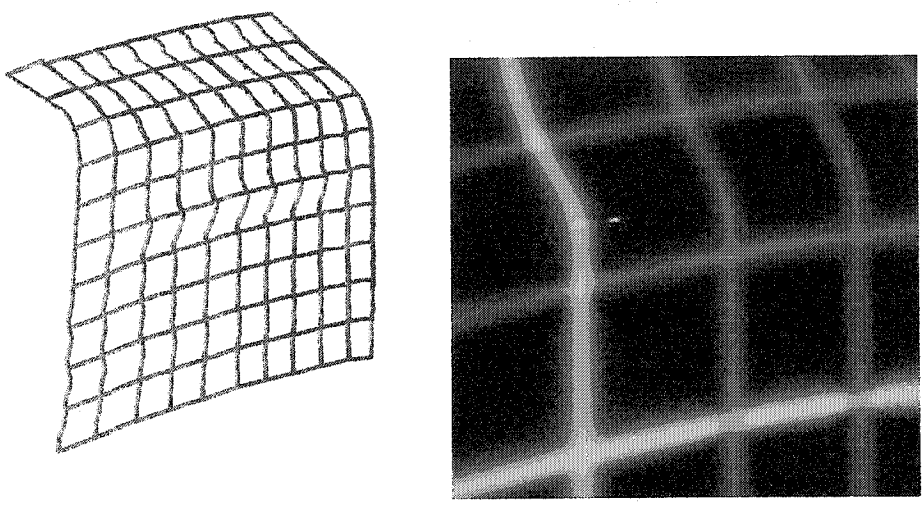
Figure 9.29(a) shows the bottom of the egg bowl. It is a slightly convex surface with a smaller concave surface in the middle. It is smooth, however the material is porous and not glossy. The system has no problem handling this porous material and captures the 3-D content correctly (see Figure 9.29(b)).



(a) Input image

(b) Generated surface from two different view points.

Figure 9.25: Input image and the 3-D representation of the milk holder object.



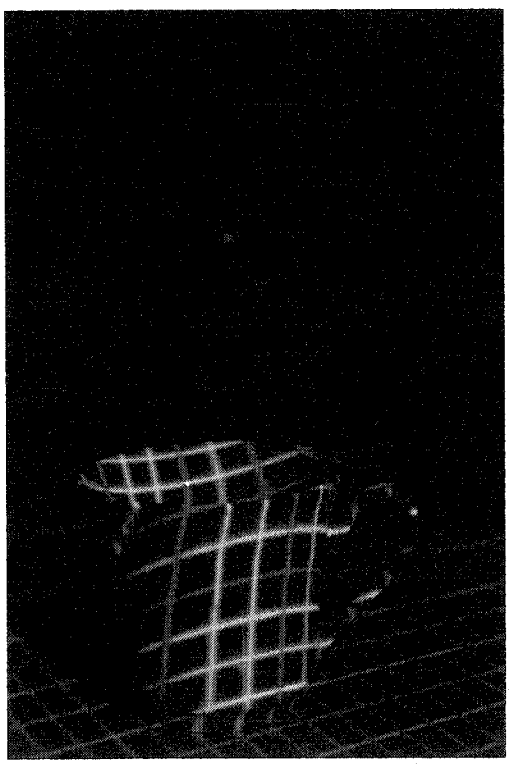
(a) 3-D NURBS Surface (b) Section of the input image.

Figure 9.26: Generation of a 3-D surface near the edge.

9.5.10 The SLA Hand Object

The hand object is made by a stereo lithography (SLA) system. It is a non reflective surface that slightly absorbs the light. In order to be able to register the edges properly with that object the shutter speed of the camera had to be increased. This was the only change made to the camera and it does not affect the calibration parameters. Figure 9.31 depicts the image at a shutter speed of $\frac{1}{60}$ of a second and also at $\frac{1}{125}$ of a second.

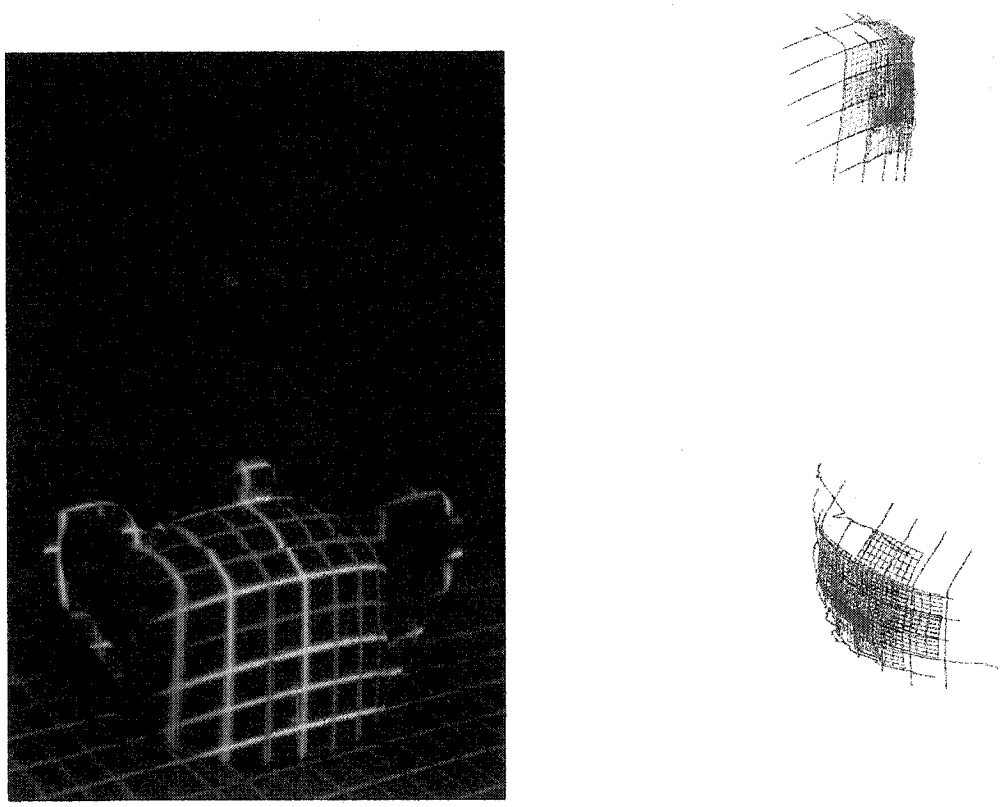
The system is able to capture properly all the contiguous areas found on the hand object as it is apparent in Figure 9.30. The thumb area is also properly captured by the NURBS curves, however due to the angle of the projector a discontinuity is created that prevents the system from having a contiguous area for a surface reconstruction near that part of the hand.



(a) The input image

(b) The generated surface from two different view points.

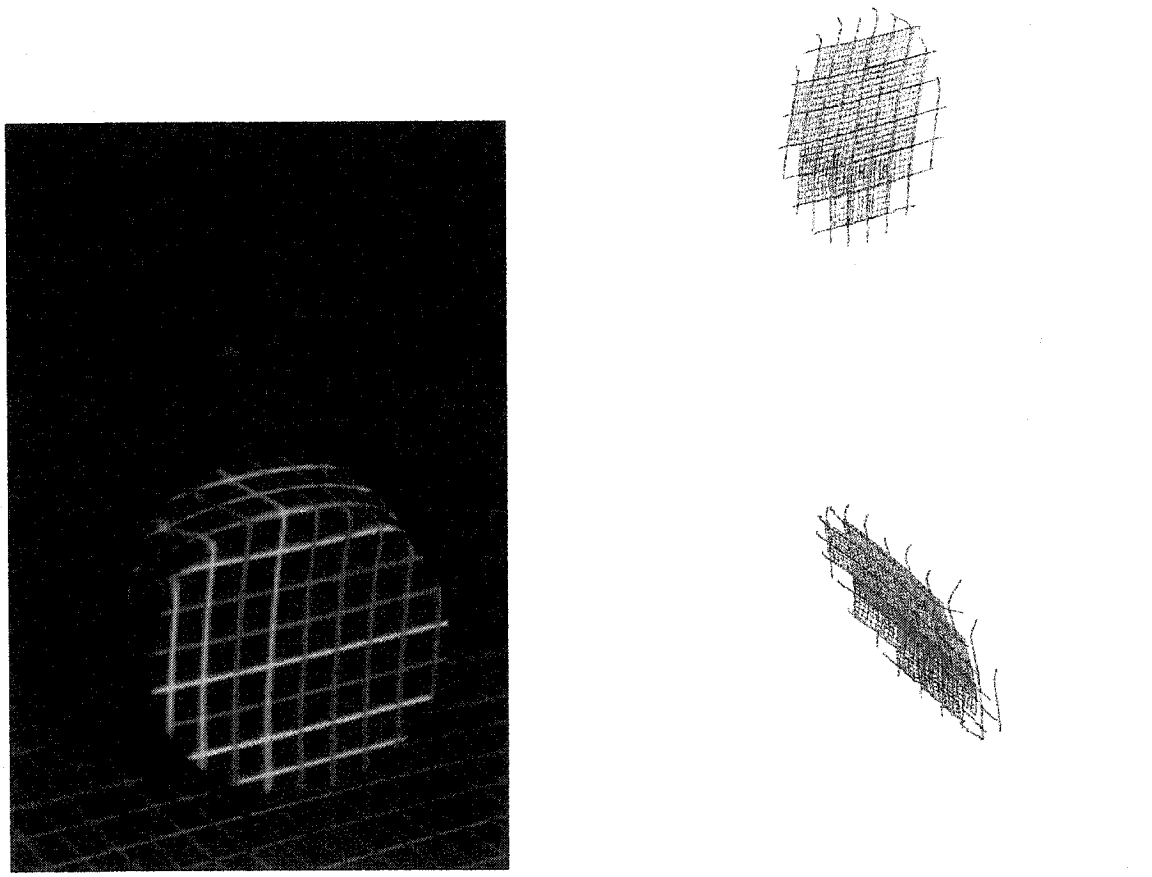
Figure 9.27: Input image and the 3-D representation of the tea cup object.



(a) Input image

(b) Generated surface from two different view points.

Figure 9.28: Input image and the 3-D representation of the sugar bowl object.



(a) Input image

(b) Generated surface from two different view points.

Figure 9.29: Input image and the 3-D representation of the egg bowl object.

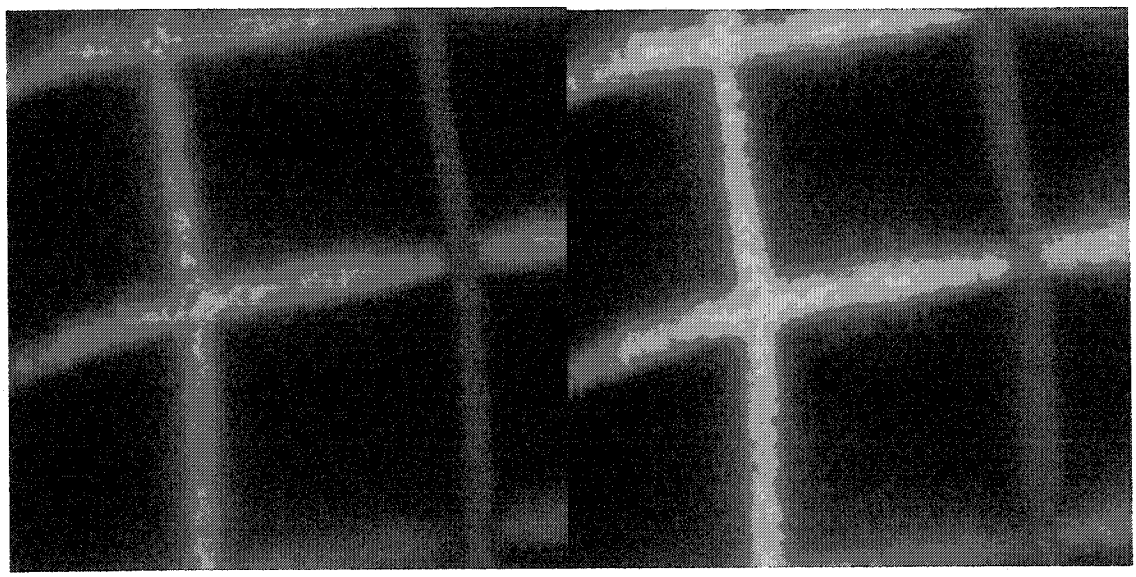
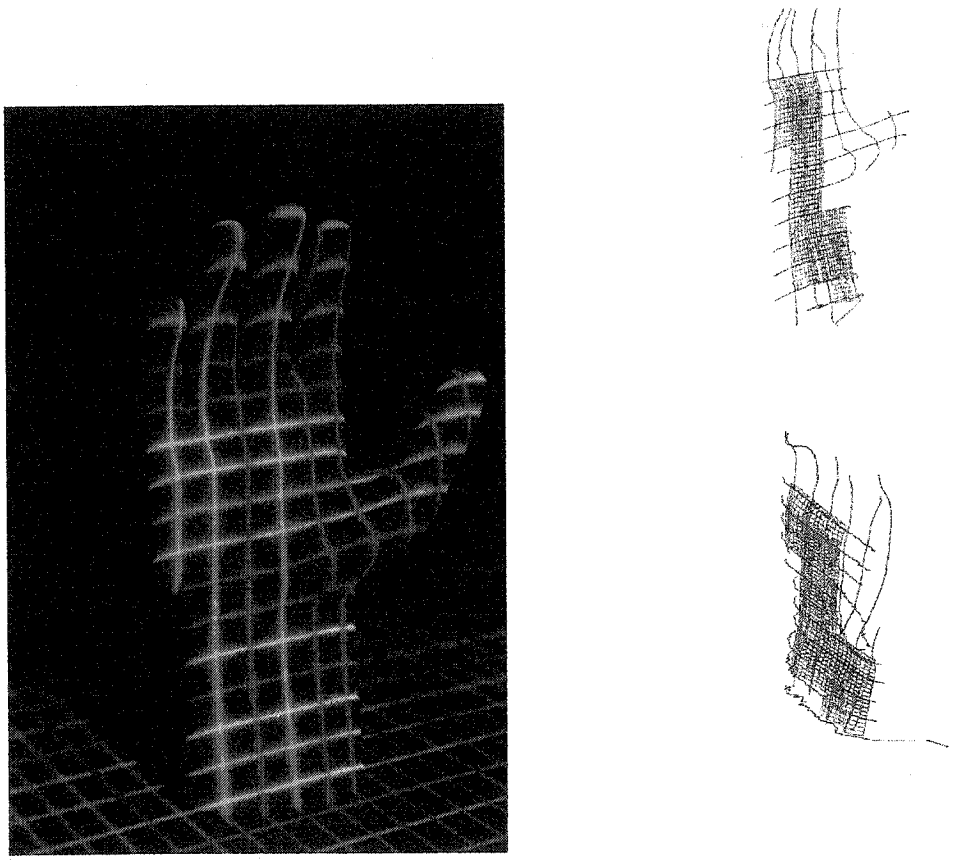


Figure 9.30: Image taken with a shutter speed of $\frac{1}{125}$ second in left image and $\frac{1}{60}$ second in the right image.



(a) Input image

(b) Generated surface from two different view points.

Figure 9.31: Input image and the 3-D representation of the hand object.

9.6 Conclusion

The accuracy of the methodology and the techniques proposed in this work is very high both for the real objects observed (0.08408 mm object space error) and the synthetic objects used (0.0306 mm NURBS surface error). The results obtained also highlight that to generate a NURBS surface, a certain contiguous area needs to be present to obtain enough intersection points and allow the generation of a NURBS surface. This fact makes the system resilient to errors in determining if a segment is vertical or horizontal as shown by the result with the teapot object.

Other methodologies focus on point location and are, therefore, more sensitive to errors in localizing the points inside the image. The algorithm proposed in this thesis solves the problem of needing to remove bad information as most of the NURBS surface information is valid. An entire contiguous region of size $(d+1) \times (d+1)$ is required before a NURBS surface is generated.

Four calibration objects are analyzed in this chapter. Although the use of a cylinder to perform the validation is sound theoretically, tests indicate that the classic two calibration planes or even a single calibration plane outperforms this object as a calibration object. The author of this thesis recommends the use of a two planes calibration object for industrial applications where the calibration object can be manufactured with great precision. The use of a simple single plane is also highly recommended as it is the easiest to construct and it also generate good results: 0.2966mm mean error at $z = 50$ and 0.0248mm mean error at $z = 0$.

The line detection algorithm is always strong when synthetic images are used. For real images, the digital camera had to capture an image in a low light environment. This low light environment added noise to the images which required

pre-processing before being used by the methodology in the form of halving the image. Other noise removal techniques could be used but they may have impacted the quality of the edge detection.

Chapter 10

Conclusion and Future Work

10.1 Conclusion

This thesis creates a new methodology for an end-to-end 3-D object reconstruction based on a single image of an object obtained with a regular digital camera. The use of NURBS curves and surfaces enable the system to be consistent throughout all the steps needed for the 3-D object reconstruction. Furthermore, the methodology and techniques described in this thesis are validated by the results obtained with both synthetic and real objects.

The tests with real objects were made with very low cost equipment and with little regard to optimizing the accuracy of the system. For instance, the calibration is done by fixing a printed grid to a board using scotch tape. This was done intentionally to illustrate that the methodology proposed is robust and can generate results with high precision even though the setup is not near any industrial standard. With this kind of set-up, the calibration obtained was on average at an error of 0.08901 mm with a standard deviation of 0.05121 mm. These values are

obtained with a single view of the object since multi-view registration is not in the scope of the current work presented in this thesis.

Most 3-D reconstruction systems rely on an epipolar constraint. The proposed 3-D reconstruction methodology imposes no such constraint and in practise, the quality of the calibration improves when the camera and the projector have very different viewing angles. This gives more freedom as to the placement of the camera and the projector. A user could easily find a position that minimizes glare and other potential lighting issues.

10.2 Contributions

The major contributions of this work are:

- The first contribution of this thesis is the calibration process. The calibration process in most techniques normally uses a different methodology than the methodology used for the 3-D reconstruction process to acquire feature points. The approach proposed in this thesis has high precision in terms of feature extraction where high precision benefits both the calibration and the 3-D reconstruction processes [39].

  The features used for the calibration process are the intersection points between the horizontal and vertical grid lines. The grid lines are defined with NURBS curves which are constructed by interpolating points obtained at sub-pixel accuracy. This approach allows the system to be calibrated with an object space error of 0.08408 mm with real objects. This approach is also the same first steps which are required to generate a set of NURBS surfaces that describe the object in 3-D.

- The pseudo-random encoding technique proposed in this thesis used to map a structured light pattern is unique. It is a grid composed of lines which are colour coded using a pseudo random multivariate sequence (PRMVS). The PRMVS removes ambiguities when registering information contained in the images viewed from the camera and the projector. This form of encoding allows the methodology to match the whole line as opposed to only matching interest points such as intersections or other meaningful feature points located on an image.

- The 3-D reconstruction process doesn't generate a cloud of points but instead generates a 3-D NURBS surface. This kind of surface is more easily manipulated by a computer than a cloud of points and it conveys more meaning information. If multiple views are to be used, then a network of 3-D NURBS curves is also a lot more meaningful in terms of 3-D information than a cloud of points. Finally, a network of 3-D NURBS curves can be viewed as way to compress the 3-D information obtained from the image.

  The methodology proposed in this thesis uses the algorithm proposed by Gordon to generate a NURBS surface based on the information contained inside a bi-directional network of NURBS curves.

- Feature extraction with the algorithm proposed in this thesis does not rely on single points but instead relies on the information contained on a grid line. Furthermore, the feature extraction uses a parametric representation which adapts to any shape, from linear line to free hand drawing. Most methods rely on the matching of points between the left and right images to create a depth value at that point. This limits their ability to work at sub-

pixel accuracy. To work at sub-pixel accuracy, those methods must rely on multiple views registration. Multiple views registration is not a substitute for the real sub-pixel feature extraction which is proposed in this thesis.

- The approach proposed in this thesis is global. It accounts for all the necessary steps to go from a single image of an object to its absolute 3-D world coordinate NURBS surface model. Several approaches do not use 3-D world coordinates and instead rely on relative coordinates. By doing so, they minimize the issues related to going from the relative to the absolute world.

  The steps of the approach described in this thesis include: the capture of the edge points to sub-pixel accuracy, the generation of a bi-directional network of 2-D NURBS curves, the calibration of the system, the generation of a bi-direction network of 3-D NURBS curves and the generation of a 3-D NURBS surface inside the world coordinate system.

Multiple minor contributions are also generated by the proposed algorithm.

- The proposed structured light also enables the calibration process. Most structured lights cannot be used when performing calibration since they rarely encode in 2-D. This characteristic is shared with some previous work that was done with pseudo-random encoded binary arrays. However, the accuracy of the current method is higher.

- Sub-pixel accuracy is possible with the proposed approach since it doesn't rely on the grid lines themselves but on the edges which constitute these lines. Techniques which extract to sub-pixel accuracy when performing

edge detection are common. This thesis shows a method which is very accurate and offers sub-pixel accuracy when doing line detection. However, that method relies on interpolation techniques which are error-prone and are less accurate than edge detection.

- This thesis does not rely on the line definition itself, but on the accurate location of the edges on both sides of the line. This not only improves the accuracy of the edge detection in 2-D but also generates a more accurate 3-D reconstruction of the object. The 3-D reconstruction is improved because a line's true center is different in the 2-D view perceived by the camera than it is on the 3-D object illuminated by the projector.

10.3 Future Work

The methodology proposed in this thesis materialized in a framework on which one can build and improve. For instance, one possible area of future work would be to optimize the Tsai's calibration method based on the fact that an intersection point projected to the horizontal line or the vertical line should be at the same 3-D location. Although Tsai's calibration method offers accurate results, it could be improved upon by leveraging this unique constraint.

Another potential source of improvement regarding the calibration methodology proposed in this thesis is based on the observation that since the geometry of the lines is known, one could find a NURBS curve which fits the model instead of a set of points. With this approach, a linear NURBS curve would be used to represent the grid lines and the end points would be moved to optimize the fit with the data points in the least square sense.

The surfaces are presently being generated from a set of curves which form a matrix of $(d+1) \times (d+1)$ grid nodes. The Gordon surface method requires a rectangular $m \times n$ region to generate a surface. It would be worthwhile to solve the optimization problem that would minimize the number of NURBS surfaces required to cover a grid of NURBS curves.

Multi-view registration of the same object could also be a future direction of this system. The integration would allow for more complete information about the object being scanned while keeping the high level value of NURBS surfaces to help CAD applications manipulate the end-result. Current methodology focus on cloud of points or on triangular meshes to perform the registration. With a NURBS surface, a cloud of points can be generated along with their tangent information. This would allow effective registration between two views.

Since the network of NURBS curve is still available inside both views, once they are registered, a new network may be created by merging the two network of NURBS curve. From that new network, a new combined surface could then be generated. This method could be iteratively applied until all occluded areas of the object are removed.

As earlier work indicated [38], it is possible to use morphological operators to extract the line information inside the grid network. It is also possible to use a watershed or a skeleton operation to extract the center of the lines. Depending on the noise level, these operators can extract the line information properly.

In the case of this thesis, the vertical lines and the horizontal lines have to be extracted individually and then the edges associated with those lines are extracted at subpixel accuracy. The morphological operators do not operate at sub-pixel accuracy. Furthermore to extract an horizontal line a closing operator would have to

be used with a proper mask. Since the lines observed are always curved, the union of multiple morphological operators would be required to get an approximation of the lines. The usefulness of these morphological operation would be limited to guiding the line extraction algorithm.

The morphological operators are better suited at finding the non-illuminated squares inside the grid. The location of those squares could then be leveraged to estimate more easily where intersection points are located and also where the grid lines are. The line detection algorithm presented in this thesis would still be required but it could leverage this new knowledge to more easily determine the edge points at sub-pixel accuracy.

Other areas of future work include the inclusion of trimmed NURBS surfaces or the use of triangular NURBS patches to describe the generated 3-D object representation.

Bibliography

[1] N. Ayache and B. Faverjon. Efficient registration of stereo images by matching graph descriptions of edge segments. *Int. J. Comput. Vision*, pages pages 107–131, 1987.

[2] H. H. Baker and T. O. Binford. Depth from edge and intensity based stereo. In *Proc. 7th Joint Conf. Artificial Intell.*, pages 631–636, Vancouver, Canada, Aug. 1981.

[3] Steve Baker. A brief history of the utah teapot. http://sjbaker.org/teapot/.

[4] S.T. Barnard and W.B. Thompson. Disparity analysis of images. *IEEE Trans. Pattern anal. Machine Intel.*, vol. 2(no. 4):pages 333–340, July 1980.

[5] Stephen T. Barnard and Martin A. Fischler. Computational stereo. *ACM Computing Surveys*, vol. 14(no. 4):pages 553–572, Dec. 1982.

[6] S.N. Bernstein. Démonstration du théorème de Weierstrass fondée sûr le calcul dès probabilités. *Commun. Soc. Math. Khrakow*, Vol. 12(No. 2):pages 1–2, 1912.

[7] P.J. Besl and N.D. Mckey. A method for registration of 3-D shape. *IEEE Trans. Pattern anal. Machine Intel.*, Vol.PAMI–14:pp 239–256, 1992.

[8] K.L Boyer and A.C. Kak. Color-encoded structured light for rapid active ranging. *IEEE Trans. Pattern anal. Machine Intel.*, vol. 9(no. 1):pages 14–28, Jan 1987.

[9] Zen Chen, Shinn-Ying Ho, and Din-Chang Tseng. Polyhedral face reconstruction and modeling from a single image with structured light. *IEEE Trans. on Syst., Man and Cybern.*, vol. 23(no. 3):pages 864–872, May/June 1993.

[10] Steven D. Cochran and Gèrard Medioni. 3-D surface description from binocular stereo. *IEEE Trans. Pattern anal. Machine Intel.*, vol. 14(no. 10):pages 981–994, Oct. 1992.

[11] M. G. Cox. The numerical evaluation of B-splines. *Journal of Institutional Mathematical Application*, Vol. 10:pages 134–149, 1972.

[12] H. B. Curry and I.J. Schoenberg. On spline distributions and their limits: the pòlya distribution functions. *Bulletin of the American Mathematical Society*, Vol. 53:page 109, 1947. Abstract 380t.

[13] Cyra. Cyra's cyclone pc software. http://www.neigps.com/products_cy_cyclone.php, 2004.

[14] C. De Boor. On calculating with B-splines. *Journal of Approximation Theory*, Vol. 6:pages 50–62, 1972.

[15] C. De Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 1978.

[16] Frederic Devernay. A non-maxima suppresion method for edge detection with sub-pixel accuracy. Technical Report no. 2724, Institut National de Recherche en Informatique et en Automatique, November 1995.

[17] Umesh R. Dhond and J.K. Aggarwal. Structure from stereo — a review. *IEEE Trans. on Systems, Man and Cyber.*, vol. 19(no. 6):pages 1489–1510, Nov./Dec. 1989.

[18] Kaichang Di and Rongxing Li. Cahvor camera model and its photogrammetric conversion for planetary applications. *Journal of Geophysical Research*, Vol. 109(E04004), April 2004. doi:10.1029/2003JE002199.

[19] Geoffrey Egnal and Richard P. Wildes. Detecting binocular half-occlusions: Empirical comparisons of five approaches. *IEEE Trans. Pattern anal. Machine Intel.*, Vol. 24(No. 8):pages 1127–1133, August 2002.

[20] James D. Foley, Andries van Dam, Steven K. Feiner, and AJohn F. Hughes. *Computer Graphics Principles and Practice.* The Systems Programming Series. Addison-Wesley Publishing Company, second edition edition, 1992.

[21] Frank Forster, Manfred Lan, and Bernd Radig. Real-time range imaging for dynamic scenes using colour-edge based structured light. In *IEEE Proceeding 16th International Conference on Pattern Recognition*, volume vol 3, pages pages 645–648, 2002.

[22] D. B. Gennery. Object detection and measurement using stereo vision. In *Proc. ARPA Image Understanding Workshop*, pages 161–167, College Park, MD, Apr. 1980.

[23] Jens Gühring. Dense 3-D surface acquisition by structured light using off-the-shelf components. In SPIE, editor, *Photonics West, Videometrics VII*, volume Volume 4309, pages pages 220–231, SAn Jose, USA, 2001.

[24] W. Gordon. Spline-blended surface interpolation through curve networks. *Journal of Mathematics and Mechanics*, vol. 18(no. 10):pages 931–952, 1969.

[25] J. J. Guerrero and C. Sagüés. Estimating the motion direction from brightness gradient on lines. *IEEE Transactions on Systems, Man. and Cybernetics-Part C*, 2001.

[26] Olaf Hall-Holt and Szymon Rusinkewicz. Stripe boundary codes for real-time structured-light range scanning of moving objects. In *Proceedings of the Eighth IEEE International Conference on Computer Vision*, volume Vol. 2, pages pages 359–366, 7–14 July 2001.

[27] Olaf Hall-Holt and Szymon Rusinkiewicz. Stripe boundary codes for real-time structured-light range scanning of moving objects. In *Eighth International Conference on Computer Vision*, pages pages 359–366, 2001.

[28] M. J. Hannah. Bootstrap stereo. In *Proc. ARPA Image Understanding Workshop*, pages 201–208, College Park, MD, Apr. 1980.

[29] R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision*. Addison Wesley Publishing Company, 1993.

[30] Robert M. Haralick. Zero-crossing of second directional derivative edge operator. In *SPIE symposium on robotic vision*, page 23 pages, 1982.

[31] D.Q. Huynh, R.A. Owens, and P.E. Hartmann. Calibrating a structured light stripe system: A novel approach. *International Journal of Computer Vision*, Vol. 33(No. 1):pages 88–86, 1999.

[32] Raindrop Geomagic Inc. geomagic studio: Reverse engineering software. http://www.geomagic.com/products/studio/, 2004.

[33] Inc. INUS Technology. Rapidform 3D scanning software. http://www.rapidform.com/product/productX2.htm, 2004.

[34] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.

[35] R. A. Jarvis. A perspective on range finding techniques for computer vision. *IEEE Trans. Pattern anal. Machine Intel.*, vol. 5(no. 2):pages 122–139, March 1983.

[36] Y.C. Kim and J.K. Aggarwal. Positioning 3-D objects using stereo images. *IEEE J. Robotics and Automation*, vol. 3(no. 4):pages 361–373, Aug. 1987.

[37] Philippe Lavoie. 3D object reconstruction using structured light and two 2d images. Master's thesis, University of Ottawa, December 1995.

[38] Philippe Lavoie, Dan Ionescu, and Emil Petriu. A high precision 3D object reconstruction method using a color coded grid and nurbs. In *IEEE Proceedings International Conference on Image Analysis and Processing*, pages pages 370–375, 1999.

[39] Philippe Lavoie, Dan Ionescu, and Emil Petriu. Camera and projector calibration with nurbs. Submitted to IEEE Transactions on Instrumentation and Measurements, 2004.

[40] Philippe Lavoie, Dan Ionescu, and Emil M. Petriu. 3-D object reconstruction using color coded grids and nurbs. In *Proceedings ET&VS-IM/97 IEEE Workshop on Emergent Technologies and Virtual Systems for Instrumentation Measurement*, pages 99–104, Niagara Falls, Canada, 1997.

[41] Philippe Lavoie, Dan Ionescu, and Emil M. Petriu. 3-D object model recovery from 2-d images using structured light. *IEEE Transactions on Instrumentation and Measurement*, Vol. 53(No. 2):pages 437–447, April 2004.

[42] J. J. Le Moigne and A. M. Waxman. Structured light patterns for robot mobility. *IEEE J. Robotics Automat*, vol. 4(no. 5):pages 541–548, Oct 1988.

[43] E.T.Y Lee. Rational quadratic bézier representaion for conics. In *Geometric Modeling: Algorithms and New Trends*, pages 3–19. SIAM, 1987.

[44] F.J. MacWilliams and N.J.A. Sloane. Pseudo-random sequences and arrays. *Proc. IEEE*, vol. 64(no. 12):pp 1715–1729, Dec. 1976.

[45] Henri Maître and Wei Luo. Using models to improve stereo reconstruction. *IEEE Trans. Pattern anal. Machine Intel.*, vol. 14(no. 2):pages 269–277, Feb. 1992.

[46] D. Marr, G. Palm, and T. Poggio. Analysis of a cooperative stereo algorithm. *Biol. Cybern.*, vol. 28:pages 223–229, 1978.

[47] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, vol. 194:pages 283–287, 1976.

[48] G. Medioni and R. Nevatia. Segment-based stereo matching. *Comput. Vision, Graphics, Image Processing*, vol. 31:pages 2–18, 1985.

[49] H. P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. 5th Int. Joint Conf. Aritificial Intell.*, page 584, 1977.

[50] E. Mouaddib, J. Batlle, and J. Salvi. Recent progress in structured light in order to solve the correspondence problem in stereovision. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pages pages 130–136, Albuquerque, New Mexico, April 1997.

[51] Yuichi Ohta and Takeo Kanade. Stereo by intra- and inter-scanline search. *IEEE Trans. Pattern anal. Machine Intel.*, vol. 7(no. 2):pages 139–154, March 1985.

[52] Jordi Pagès, Joaquim Salvi, and Carles Matabosch. Implementation of a robust coded structured light technique for dynamic 3D measurements. In *IEEE International Conference on Image Processing*, volume Volume 1, pages 1073–1076, Barcelona, Spain, September ICIP 2003.

[53] Gonzalo Pajares and Jesús de la Cruz. The non-parametric parzen's window in stereo vision matching. *IEEE Transactions on Systems, Man. and Cybernetics-Part B*, 2001.

[54] Johnny Park, Guilherme N. DeSouza, and Avinash C. Kak. Dual-beam structured light scanning for 3-D object modeling. In *IEEE Proceedings Third In-*

ternational Conference on 3-D Digital Imaging and Modeling, pages 65–72, 2001.

[55] Emil M. Petriu, Z. Sakr, H.J.W. Spoelder, and A. Moica. Object recognition using pseudo-random color encoded structured light. In *Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference*, volume Vol. 3, pages pages 1237–1241, May 2000.

[56] Lee Piegl. A technique for smoothing scattered data with conic sections. *Computers in Industry*, vol. 9:pages 223–237, 1987.

[57] Lee Piegl. On nurbs: A survey. *ÏEEE Computer Graphics and Applications*, vol. 10(no. 1):pages 55–71, 1991.

[58] Les Piegl and Wayne Tiller. *The NURBS book.* Springer, Berlin, 1995.

[59] J.L. Posdemer and M.D. Altschuler. Surface measurement by space-encoded projected beam systems. *Comput. Graphics Image Processing*, 18:pages 1–17, 1982.

[60] POV-Team. Persistence of vision raytracer. http://www.povray.org, July 2004.

[61] L. Ramshaw. Blossoming: A connect-the-dots approach to splines. Technical Report 19, Digital, System Research Center, Palo Alto, CA, 1987.

[62] A. Rosenfeld, R.A. Hummel, and S.W. Zucker. Scene labeling by relaxation operation. *IEEE Trans. Syst. Man Cybern.*, vol 6:pages 420–423, June 1976.

[63] S. Roy and J. Meunier. Stereoscopic analysis of multiple images. submitted to the International Journal of Computer Vision, August 1993.

[64] RSI. Rsi 3D systems & software. http://www.rsi.gmbh.de/innovativee.htm, 2004.

[65] Janez Perš and Stanislav Kovčič. Nonparametric, model-based radial lens distortion correction using tilted camera assumption. Technical report, University of Ljubljana, Ljubljana, 2002.

[66] J. Salvi, X. Armangue, and J. Batlle. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, 35(7):pages 1617–1635, July 2002.

[67] Joaquim Salvi, Jordi Pagès, and Joan Batlle. Pattern codification strategies in structured light systems. *IEEE Trans. Pattern anal. Machine Intel.*, Volume 37(Issue 4):827–849, April 2004.

[68] Gregory C. Sharp, Sang W. Lee, and David K. Wehe. Multiview registration of 3D scenes by minimizing error between coordinate frames. *IEEE Trans. Pattern anal. Machine Intel.*, Vol. 26(No. 8):pp 1037–1050, August 2004.

[69] Luciano Silva, Olga R.P. Bellon, and Kim L. Boyer. Enhanced, robust genetic algorithms for multiview range image registration. In *Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM'03)*, volume vol 3, 2003.

[70] I. E. Sobel. *Camera Models and Machine Perception*. Electrical engineering department, Stanford University, Stanford, CA, 1970.

[71] H. J. W. Spoelder, F. M. Vos, F. C. A. Groen, and E. M. Petriu. Some aspects of pseudo random binary array based surface characterization. In R. Hunter

and E. M. Petriu, editors, *Proc.IMTC/98, St Paul May 18-21 1998, IEEE Press no 98CH36222*, pages 2–7, 1998.

[72] Wayne Tiller. Rational b-splines for curve and surface representation. *IEEE for Computer Graphics and Applications*, vol. 3(no. 6):pages 61–66, 1983.

[73] Wayne Tiller. Knot-removal algorithms for nurbs curves and surfaces. *CAD*, vol. 24(no. 8):pages 445–453, 1992.

[74] N. Trif. Model-based visual recognition of 3-D objects using pseudo-random grid encoding. Master's thesis, University of Ottawa, May 1993.

[75] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, vol. 3:pages 323–344, 1987.

[76] D.-C. Tseng and Z. Chen. Computing location and orientation of polyhedral surfaces using a laser-based vision system. *IEEE Trans. Robotics Automat.*, vol. 7:pages 842–848, Dec. 1991.

[77] P. Vuylsteke and A. Oosternlink. Range image acquisition with a single binary encoded light pattern. *IEEE Trans. Pattern anal. Machine Intel.*, vol. 12(no. 2):pages 148–164, Feb. 1990.

[78] E. J. Watson. *Mathematics of Computation*, vol. 16:pages 368–369, 1962.

[79] Eric W. Weisstein. Circle-line intersection. http://mathworld.wolfram.com/Circle-LineIntersection.html, 2004. From MathWorld–A Wolfram Web Resource.

[80] Eric W. Weisstein. Relatively prime. http://mathworld.wolfram.com-/RelativelyPrime.html, October 2004. From MathWorld–A Wolfram Web Resource.

[81] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Trans. Pattern anal. Machine Intel.*, Vol. 14(No. 10):pages 965–980, October 1992.

[82] P.M. Will and K.S. Pennington. Grid coding: A preprocessing technique for robot and machine vision. In *Proc. 2nd Int. Joint Conf. Artificial Intell.*, pages 66–68, Sep. 1971.

[83] Reg Willson. Tsai camera calibration software. http://www-2.cs.cmu.edu/afs/cs.cmu.edu/user/rgw/www/TsaiCode.html, October 28 1995.

[84] Li Zhang, Brian Curless, and Steven M. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *The 1st IEEE International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 24–36, June 2002.

[85] Li Zhang, Brian Curless, and Steven M. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 367–374, June 2003.

[86] Z. Zhang, R. Deriche, Olivier Faugeras, and Q.T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Technical Report RR–2273, INRIA, May 1994.

[87] O. Zuniga and Robert M. Haralick. Integrated directional derivative gradient operators. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 17:pages 508–517, 1987.

[88] O. Zuniga and Robert M. Haralick. Correction to integrated directional derivative gradient operators. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 18, 1988.