CERTIFICATION OF APPROVAL

2D Photo Converter: Modeling 3D Objects from 2D Photos Using OpenGL

by

Anis binti Ismail

A project dissertation submitted to the
Information Communication Technology Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirements for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION COMMUNICATION TECHNOLOGY)

Approved by,

(Assoc. Prof. Dr Abas Md Said)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
June 2006

2D Photo Converter: Modeling 3D Objects from 2D Photos Using OpenGL

by

Anis binti Ismail

Project Dissertation submitted in partial fulfillment of the requirements for the
Bachelor of Technology (Hons)
(Information Communication Technology)

June 2006

Universiti Teknologi PETRONAS
Bandar Sri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

ANIS BINTI ISMAIL

ABSTRACT

The concept of modeling a 3D object based on 2D photos has indeed been widely discussed and researched among the computer vision professionals and virtual reality technologists. However, regardless of the many researches going on and the rapid technological development in 3D modeling world, the best method to render a 3D model that satisfies the requirement of minimum image pre-processing, maximum model-realism and minimum error percentage is yet to be studied. This report will lay out another technique of modeling 3D objects using a 2D photo by analyzing the possibility and accuracy of light intensity evaluation towards the model.

The main objective of this study is to propose an alternative solution to 3D modeling techniques by using the information from a 2D photo. It is hoped that by applying the proposed solution, the constraint of costs and time in the current 3D modeling system could be reduced. The research focuses on bitmap photos and it applies the principles of light intensity and distance relativity in estimating the depth volume of the model. The application is built by using Microsoft Visual C++ 6.0 and utilizes OpenGL Application Programming Interface (API) in its code.

However, the results of the experiments conducted in this research study shows that the formula used in the application might not be the best method to produce a 3D model from a 2D photo. Nonetheless, the idea of using light intensity valuations in producing 3D models could be the new solution in 3D modeling technology. The framework design and the ideas could be the base research for further development in the 3D modeling research and analysis study.

ACKNOWLEDGEMENTS

بسم الله الرحمن الرحيم

Alhamdulillah, first and foremost, I would like to thank the Almighty Allah for giving me the strength, wisdom, and good health throughout this research study.

My highest gratitude and thanks to my supervisor, Assoc. Prof. Dr Abas Md Said, for his immense contribution, guidance and encouragement towards producing a quality research project. Throughout the whole duration of this work, I have learnt much from him, not only in terms of technical knowledge but also the finer points of being a better researcher in the future.

Special thanks to both my beloved friends, Jamilia Laila Abd Rahim and Mohammad Ashraf Ahmat Amin, for your time and helpful moments with me throughout the completion of my final year project. Not forgetting to all my wonderful friends, thank you for sharing your enormous experiences and ideas.

To my beloved family, *Mama*, *Abah*, my sisters and brothers, thank you all for your faith, love and encouragement in me at all time.

Last but not least, my appreciation to those who had contributed in this research project in one way or another. Thank you all…

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND OF STUDY

This study aims to propose a new alternative method for 3D image modeling by using a single 2D-photo as its main model reference. The major focus of this study is to find the best method and algorithm in extracting the depth value of an object based on the raw image data contained in a 2D photo. The concept of modeling a 3D object based on 2D photos has indeed been widely discussed and researched among the computer vision professionals and virtual reality technologists. However, regardless of the many researches going on and the rapid technological development in 3D modeling world, the best method to render a 3D model that satisfies the requirement of minimum image pre-processing, maximum model-realism and minimum error percentage is yet to be studied. The challenge is in finding out the most precise and accurate depth-value estimation of the targeted model based on the data that could be extracted solely from 2D photos.

By achieving the main objective of this study, the algorithm could then be further applied in the various knowledge fields that require the use of 3D models and environment. Through the preceding researches done, it has been discovered that the demand for an effective 3D modeler is rising and that it has the potential of becoming a highly usable resource in the process of research and development of real situation simulations and human imitations. The 3D modeler software could be used by professionals and technologists in various areas such as to[1]:

- Create drawings and measurements for process and plant engineers
- Create drawings for architects and historic preservationists
- Model sets, objects, people and vehicles for animators and film/video producers
- Measure and create drawings of buildings, excavations & artifacts for archaeologists
- Measure and model anatomical morphology for anthropologists and medical practitioners
- Reverse-engineer mechanical parts and assemblies for manufacturing engineers
- Survey complex 3D shapes, structures and volumes for civil engineers and surveyors
- Model objects for 3D databases for virtual reality builders

The study started by analyzing the information that could be extracted from 2D photos. Since a 2D illustration could only give the estimation of x and y value, further research is done by focusing on the methods to extract and estimate the z value of the object. The z value could then be manipulated and used in the 3D model rendering process as the depth value of the model. As the core of this research, focus has been put on the manipulation of light intensity data and pixels values that could be obtained directly from a 2D photo. By implementing the concept of light intensity and distance relativity, the required depth value is calculated for each and every pixel of the 2D image to give the illusion of a 3D model. The raw data extraction and manipulation will be further described in the subsequent chapters of this report.

1.2 PROBLEM STATEMENT

1.2.1 Demand for a simpler, faster, yet effective method of modeling 3D objects

It has been realized that the use of 3D models and virtual environment could become powerful tools in many research areas and real-life applications. However, the analysis and design process in producing a highly accurate virtual objects/environment through lengthy codes and complicated formulas requires a comprehensive and thorough study of the targeted designs. It is also extremely time-consuming and consumes high resources of the workstations used. Parallel to the rapid technological advances, it is logical that users demand for simpler, faster, yet accurate tools to produce highly acceptable virtual objects/environments.

1.2.2 Costly existing 3D Modeler software, tools and devices

However, there also exist several 3D Modeler tools and software in the market today. And due to the high demand of such product, these software and tools are priced at a very competitive price. This has become a problem for individuals who might require the use of such advanced tools to do their researches and studies and might not have high budget allocated for it. Furthermore, in order for the users to use the software, they are also required to purchase additional hardware and devices that are required by the software providers. Among the common devices that could be used with the existing software are the 3D scanners and specially designed cameras which are also expensive to be used by individuals.

1.2.3 Complicated analysis and computations for objects with complex structures

Objects with complex structures such as human face, bushes, or trees are among the most complicated objects to be drawn in a 3D world. A human face for an example consists of complex curves and the depth for each and single point on a face might vary. The uniqueness of these objects makes it difficult for 3D developers to obtain a precise and accurate 3D model. The use of exhaustive formula and multiple polygons could consume a lot of the processor resources which then could lead to low rendering rate.

1.3 OBJECTIVES OF THE RESEARCH STUDY

The project was initiated to achieve a few identified objectives in the area of virtual reality which include:

1.3.1 To analyze the best method in rendering a 3D model based on 2D photos

The main objective of this project would be to come up with the most effective solutions to 3D modeling techniques by using the information from a 2D photo. The proposed method is hoped to accommodate the market demand for a simpler, faster, cheaper, yet effective and powerful 3D modeler tools.

1.3.2 To study on the effects of photo-image enhancement on the 3D models

The research was also conducted to analyze and test on the effects of photo-image enhancement of the input photo towards the final outcome. Several factors were considered in finding the most accurate 3D model which has lowest error percentage and highest model realism.

1.3.3 To study the best method to manipulate the raw data obtained from a 2D photo in achieving the main objective.

The study focused on extracting raw data from a bitmap photos and manipulating raw data to for estimate the model depth and structure. By using the proposed method, a 3D model could be rendered just by referring to single photo thus reduce the complexity of the 3D modeling process.

1.4 SCOPE OF THE RESEARCH STUDY

This research study focused on the development of 3D models based on a single 2D photo as its main input source. However, this research was only conducted by using the data from bitmap photos (*.bmp) as its input. The feature of a bitmap photo which consists of pixel by pixel data makes the data extraction manipulation process much easier compared to other photo file formats.

As for the experiment objects used in the testing and prototype progress, initially, simple objects are used as the based-models for this study. Among the objects that were used in the experiments are *ping pong* balls (sphere) and drink cans (cylinder). Then, the scope of the study was widen to several other more complex objects such as a human face and toy miniatures to test on the accuracy of the depth formula obtained from the based research.

The whole application was built by using Microsoft Visual C++ 6.0 and OpenGL as the main programming language to extract the raw data, manipulate the information gathered, and render the 3D model.

CHAPTER 2
LITERATURE REVIEW AND THEORY

Through out the process of completing this research study, some extensive reviews were done on existing researches that are related to the 3D modeling methods and applications. From the research, it is discovered that there are various possibilities and methods that could be used to produce 3D models using 2D photos and the research findings have contributed much in understanding the concept of 2D and 3D images. Below are some of the findings of the literature reviews:

2.1 BASIC CONCEPTS

2.1.1 2D Computer Graphics and Design

Two-dimensional (2D) graphics are the creation, display and manipulation of objects in the computer in two dimensions. Drawing programs and 2D CAD programs allow objects to be drawn on an *x-y* scale as if they were drawn on paper. Although 3D images can be drawn in 2D programs, their views are static. They can be scaled larger or smaller, but they cannot be rotated to different angles as with 3D objects in 3D graphics programs. They also lack the automatic lighting effects of 3D programs. Any desired shadows must be created by the artist using color fills or gradients.[2]

2D computer graphics are mainly used in applications that were originally developed upon traditional printing and drawing technologies, such as typography, cartography, technical drawing, and advertising. In those applications, the two-dimensional images are not just a representation of real-world objects, but an independent artifact with added semantic value. However, since the models used in 2D computer graphics usually do not provide for three-dimensional shapes, or three-dimensional optical phenomena such as lighting, shadows, reflection and refraction which are useful factors in visual reality and animation applications, 3D graphics are more preferable.

2.1.2 3D Computer Graphics and Design

Three-dimensional graphics on the other hand is the creation, display and manipulation of objects in the computer in three dimensions. 3D CAD and 3D graphics programs allow objects to be created on an *x-y-z* scale (width, height, depth). As 3D entities, they can be rotated and viewed from all angles as well as be scaled larger or smaller. They also allow lighting to be applied automatically in the rendering stage. 3D computer graphics are works of graphic art that were created with the aid of digital computers and specialized 3D software. In general, the term may also refer to the process of creating such graphics, or the field of study of 3D computer graphic techniques and its related technology. [3]

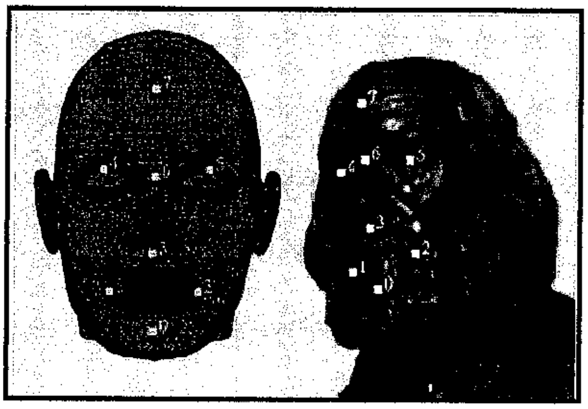


Figure 2.1 - 3D Imitations of a Human Face

However, as simple as the explanation might be, the process of creating a 3D model requires a lot of comprehensive and tedious task especially if it involves objects with complex structures. The process of creating a 3D model could be generally subdivided into three main stages[3] which are *modeling, scene layout setup and rendering process*. This project mainly focused on the main stage of creating the 3D model that is the modeling stage.

The modeling stage could be described as shaping individual objects that are later used in the scene. Modeling processes may also include editing object surface or material properties such as color, luminosity, diffuse and specular shading components, as well as adding textures, bump-maps and other features.

Modeling may also include various activities related to preparing a 3D model for animation. Normally, modeling can be performed by means of a dedicated program, an application component or some scene description language.

In this project, a new method of modeling a 3D model is to be discovered, that is by using and manipulating just a single 2D photo of the targeted model. The process of transforming a 2D photo into a 3D model that is realistic and accurate requires some data manipulation on the information that are extracted from the 2D photo. These processes are crucial to obtain the illusions of a 3D model, that is by adding the z-depth value for each of the pixels of the 2D image thus creating the depth volume for the 3D model.

2.1.3 Bitmap Photos

Bitmaps are defined as a regular rectangular mesh of cells called pixels, each pixel containing a color value. They are characterized by only two parameters, the number of pixels and the information content (color depth) per pixel. There are other attributes that are applied to bitmaps but they are derivations of these two fundamental parameters.[4]

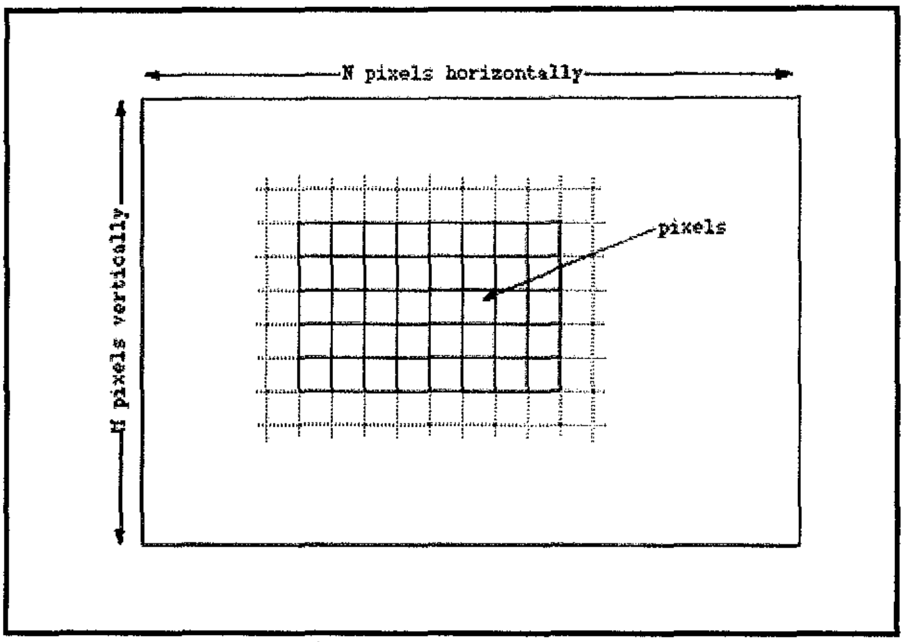


Figure 2.2 - Bitmap Images

Bitmaps are always orientated horizontally and vertically and pixels should be considered square although they may have other aspect ratios in practice.

In the majority of situations bitmaps are used to represent images on the computer. Each pixel in a bitmap contains certain information, usually interpreted as color information. The information content is always the same for all the pixels in a particular bitmap and amount of color information could be whatever the application requires but there are some standards.

2.1.4 Color Depth Standards

2.1.4.1 1 bit (black and white)

This is the smallest possible information content that can be held for each pixel. The resulting bitmap is referred to as monochrome or black and white. The pixels with a 0 are referred to as black and pixels with a 1 are referred to as white. However, even though only two states are possible they could be interpreted as any two colors, 0 is mapped to one color, and 1 is mapped to another color. [4]

2.1.4.2 8 bit greys

In this case each pixel takes 1 byte (8 bits) of storage resulting in 256 different states. If these states are mapped onto a ramp of greys from black to white the bitmap is referred to as a greyscale image. By convention 0 is normally black and 255 white. The grey levels are the numbers in between, for example, in a linear scale 127 would be a 50% grey level. [4]

Figure 2.3 - 8 Bits of Greyscale Range

In any particular application the range of grey values can be anything, it is most common to map the levels 0-255 onto a 0-1 scale but some programs will map it onto a 0-65535 scale.

2.1.4.3 24 bit RGB

This is the next step from 8 bit grey, of which there is 8 bits allocated to each red, green, and blue component. In each component the value of 0 refers to no contribution of that color, 255 refers to fully saturated contribution of that color. Since each component has 256 different states there are a total of 16777216 possible colors. [4]

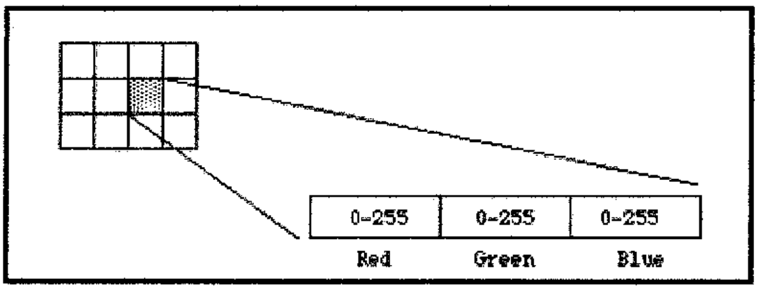


Figure 2.4 - 24 Bits RGB Color Distribution

2.2 THEORIES

2.2.1 3D Modeling Technology

Even though there exists quite several number of commercialized software that could assist technologists in doing such tedious task, these software and tools still require high level of 3D image modeling knowledge and comprehensive analysis from the users before they are able to fully utilized its functionalities. Therefore, it is the main motivation of this study to explore more improved and enhanced alternatives that could be used in rendering 3D models based on 2D photos.

According to Johnny and Guilherme[5]

> In the last few decades, constructing accurate three-dimensional models of real-world objects has drawn much attention from many industrial and research groups. Earlier, the 3D models were

used primarily in robotics and computer vision applications such as bin picking and object recognition. The models for such applications only require salient geometric features of the objects so that the objects can be recognized and the pose determined. Therefore, it is unnecessary in these applications for the models to faithfully capture every detail on the object surface. More recently, however, there has been considerable interest in the construction of 3D models for applications where the focus is more on visualization of the object by humans. (p. 1)

The theory of using 2D photos in rendering 3D models are further supported by Paul (1999)[8]. According to him

While photorealistic renderings can now be achieved with traditional computer graphics, they generally do not come easily. First, creating realistic models is a time and talent-intensive task. With most software, the artist needs to build a detailed model piece by piece, and then specify the reflectance characteristics (color, texture, specularity, etc.) for all the various surfaces in the scene. Second, generating photorealistic renderings requires advanced techniques such as radiosity and global illumination, which tend to be very computationally intensive. In brief, modeling tends to be hard, and rendering is slow.

For applications such as interior design, architectural planning, simulation, virtual museums and cultural heritage, the objects and environments that one wishes to render using the computer often exist within the real world. In such cases, the modeling and rendering challenges increase considerably: since people are familiar with what the results should look like, they will have higher expectations of the quality of the results. However, objects and environments in the real world have a very useful property as well: they can be photographed. Additionally, a great deal of research has been done in the last several years to model the geometry and appearance of objects and scenes from photographs.

2.2.2 Related Work and Current Software

For virtual reality systems, modeling of 3D objects and scenes is important and challenging. Sujin and Zhiyong (1996, p. 1), in their paper, have explored the use of images in interactive modeling systems to achieve the automation. In particular, the use of only one image is addressed. On one side, unlike the common fully interactive modeling framework, the users are not required to specify some low level details interactively which can be derived automatically from the image. On the other side, it still requires human interactions to do some high level tasks that the algorithms are difficult to perform automatically. [6]

In a brief summary, the major features of their method are:

- It requires human interaction. By applying human interaction, the use of some complicated procedures of computer vision such as camera calibration and procedures of computational geometry such as polygonization of unorganized points can be avoided.
- It has automatic process. A user can start from a good initial 3D model. It is automatically derived from the image of the object.
- It is not a framework of the stereo vision. Only one image is used. Thus, it is not necessary to solve the correspondence between two images. As one image can not determine a unique shape in 3D, the modeling result is always an approximation of the real object.

However, for virtual reality systems, it meets the visual requirement.

- It is not model-based and does not require a generic 3D model. So the shapes to be modeled are not constrained by the initial generic 3D models as in.
- It is general though they have not implemented modeling of the shapes with holes.

According to Angel Osorio (1998)[7]

> Though there are many modeling software packages on the market, most of them are aimed at geometrical modeling: they use the full geometrical approach and contain only vertex, surface and texture characteristics in the kernel. Furthermore, and mostly for marketing reasons, only a few graphics software packages take into account the very fast evolution of computers, and even these do not make full use of the specific characteristics of today's computers. The problem of making a realistic 3D model is very complicated because of the protocols and interactions involved. (p. 1)

Sujin and Zhiyong (1996, p. 2)[6] also points out that the ease of creating 3D models is crucial for the success of any virtual reality systems. Although much progress has been made on geometric modeling systems, they are mainly used in computer aided design (CAD) and yet to be used to create geometric models for VR systems, usually with irregular shapes such as animals and plants.

Additionally, according to Sujin and Zhiyong (1996, p. 2)[6]

> 'Geometric modeling can be roughly divided into forward and reverse approaches. In the forward approach, the real model of the object is not available or not directly used in the modeling process. The shape of the object is usually modeled by an iterative bottom up manner. Oppositely, in the reverse approach, images or the real measurements of the object are directly used. The purpose of the reverse approach is to achieve automation.
>
> The most popular forward methods are based on CSG/Brep framework, i.e., the framework of constructive solid geometry (CSG) and boundary representation (B-rep). They use geometric primitives to hierarchically build up through successive shape operations and transformations. Recent work includes SKETCH which introduces new interfaces for rapid modeling of CSG-like models consisting of simple primitives. This approach is most suitable for CAD systems where shapes of objects are relatively regular and require precise representation.
>
> Other work includes Teddy where new interfaces and interactive techniques are introduced for rapid designing 3D freeform objects. In particular, Teddy allows a first-time user to model a moderately complex object within minutes. A more recent work presents a virtual environment that is created by a direct input of a hand-drawn perspective sketch to computer.
>
> Another type of forward modeling methods are implicit surface based. The user specifies the skeleton of the intended model and the system constructs smooth, natural-looking surfaces around it.

The most popular reverse methods are 3D digitizing (range image) based. Using such devices as the Cyberware scanner, a dense mesh of 3D points can be derived from a set of laser-ranged images of a real model. Then triangulation techniques of computational geometry are applied to recover the topology and geometry of the object. An early work is on human body modeling, while recent work includes accuracy incrementing reconstruction by multi-round digitizing.

Currently, 3D digitizing is widely used in the entertainment industries such as video games and film production. A different type of reverse modeling methods are digital image (intensity image), with some well known research topics such as camera calibration, shape from motion, and stereo vision. In computer graphics, it is widely used in facial modeling and animation. The approach takes the advantage that digital images are much easier to get than the range images.

The forward approach emphasizes on human interaction, while the reverse approach on automation. Combining both approaches is expected to achieve better performance. This idea was explored in the Facade system where a hybrid approach, with automation achieved from the use of images, is proposed to construct large buildings. The results are very impressive. The best use of human interaction and automation together is the central concern of the hybrid approach.

CHAPTER 3
METHODOLOGY

3.1 PROJECT CYCLE

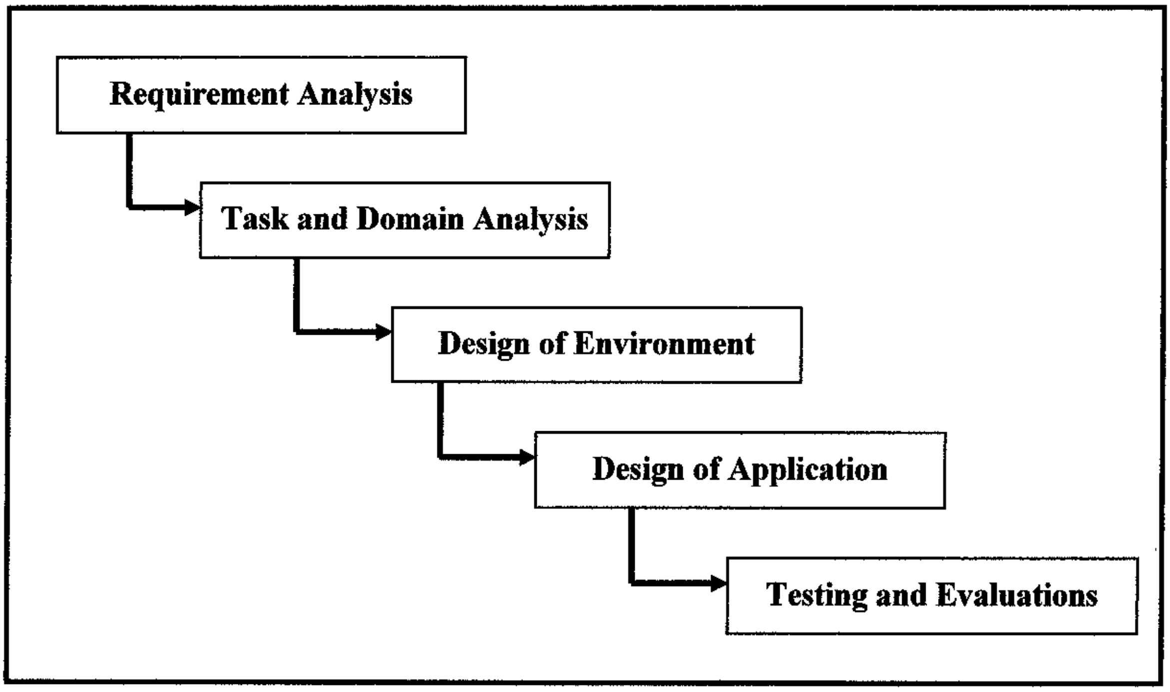


Figure 3.1 – Project Cycle Overview

3.1.1 Requirement Analysis

The initial motivation of the project came from the realization of the importance of such system in the virtual reality systems. Therefore, before a deeper research was conducted, some problem identifications analysis was performed. The requirement analysis acts as an important stage of a project because it defines the main project motivational factors and the main objectives of the project. By defining these key factors, the project would have clear and comprehensible targets. The problem statements were identified through some basis research on the current development of 3D modeling applications and the flaws of the readily available products in the market. Through these problems, several solutions alternatives were considered and later became the basis of the project objectives.

3.1.2 Task and Domain Analysis

After all the objectives were clearly lined out and the direction of the project was clearly identified, the next stage involves deeper research and study on other several scholars' articles. The main stages of the project were identified and the project outline was planned. The literature reviews conducted covers all sorts of area that are related to the projects which includes some study on bitmaps data, image-based modeling techniques, as well as the factors that could affect the 3D modeling process.

The task and domain analysis stage also include the identification of the possible objects to be used in the tools testing and evaluation process. As part of the research, simple objects were used to test the accuracy and precision of the designed formula. The initial objects used during the early development phase were a ball and a can. Since the experiment showed outstanding results, further experiments were conducted by using more complex and complicated objects to compare the findings during the testing and evaluation stage.

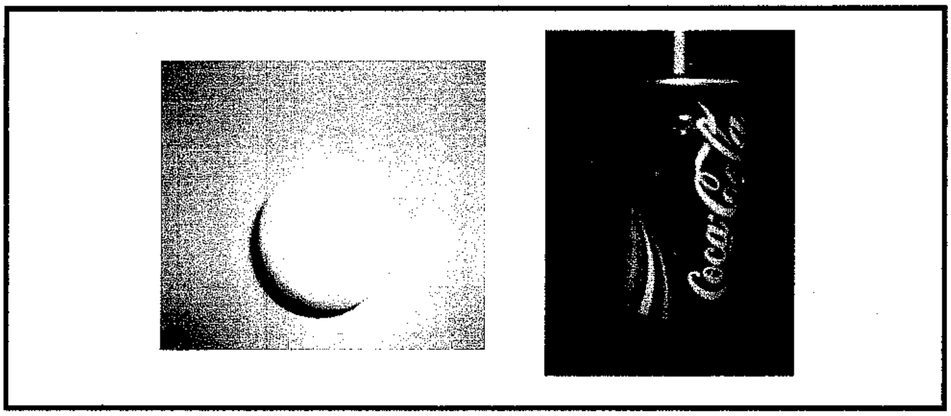


Figure 3.2 – Initial Objects Used: a Ball and a Can

3.1.3 Design of Environment

Based on the plans outlined in the previous stage, the project development process took place. The design of the whole application started of by analyzing the possible algorithms and program flows of the system. Since the target input file format of this project is a Bitmap photo, the image processing flow was planned based on the concept and attributes of bitmap data. The basic algorithm for the application would be:

```
For each pixel in photo
        Extract RGB value
        Calculate the light intensity
        Calculate the z-depth estimation value
End for

For each point in the photo dimension (height*width)
        Render model based on pixel information
        (x,y,z-depth)
        Add on RGB values
End for
```

At first, a suitable API was chosen to act as the core of the program codes. OpenGL is one the common APIs in the 3D modeling subject and can be used to create great real-time graphics. OpenGL follows C/C++ programming standard thus it is easier to use it for the graphics modeling process. The application used to build the code is Microsoft Visual C++

6.0 since it can easily calls C++ functions and APIs in the same manner as C/C++ programming language.

Then a suitable environment to take the photographs of the experiment subjects was designed. To get the best result of a 3D model, the photo used for the application has to meet some criteria, which includes:

- In a dark surrounding to minimize background noise
- Use only ONE light source (camera flash)
- Persistent camera angle and distance
- Surface of object should NOT be glossy to reduce light reflection

In order to ensure the fulfillment of the above criterion, a "mini studio" was used to take the photos of the sample images. Built by using non-reflecting materials and dark background, the photos taken were ensured of minimum disturbance or image distortion.

Other than the surrounding for the photos, the object itself has to go through few processes prior to the experiment. The objects used in the experiment need to be covered by non-glossy material as to reduce light reflection and increase its light intensity accuracy on the overall object. Furthermore, during the base experiment conducted in the previous stage, it has also been discovered that colored objects resulted in a less accurate 3D model. The reason being is because the depth volume calculated in this application is based on the light intensity and RGB attributes of each pixel of the image. To overcome such problem, the objects used on the experiment are more preferably to be in white as to get a black and white photo of the experiment subjects.

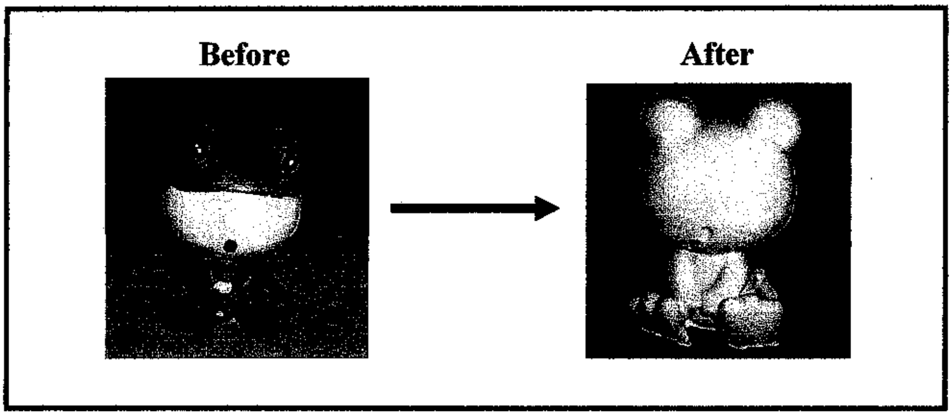


Figure 3.3 – Object Transformation

The camera that was used in the experiment is Casio Exilim EXZ55 5MP Digital Camera with 3x Optical Zoom. The camera was equipped with several Best Shot features which could ensure that the photos taken were clear and the level of details was maximized. Other important feature of the camera includes:

- High-resolution images captured by the CCD with 5 million effective pixels for photo-quality prints up to 13x17 inches
- Large 2.5-inch digital interface TFT LCD screen
- Zoom: 3x optical zoom; 4x digital zoom (12x total when optical and digital zooms are used in combination)
- Approximate focus range: 40cm to infinity
- White balance: Auto Fixed (6 modes) or Manual
- Sensitivity: Auto Fixed ISO 50, 100, 200, 400
- Built-in flash: Auto Flash Modes, Flash On, Flash Off, Red-Eye Reduction with approximate range of 0.4 to 2.6 meters wide
- Other features: Histogram display and grid display

3.1.4 Design of Application

Based on earlier research and studies on how the data of a 2D image could be manipulated to produce a 3D model, the design of the application took place. The application was built using Microsoft Visual C++ 6.0 and calls for OpenGL API.

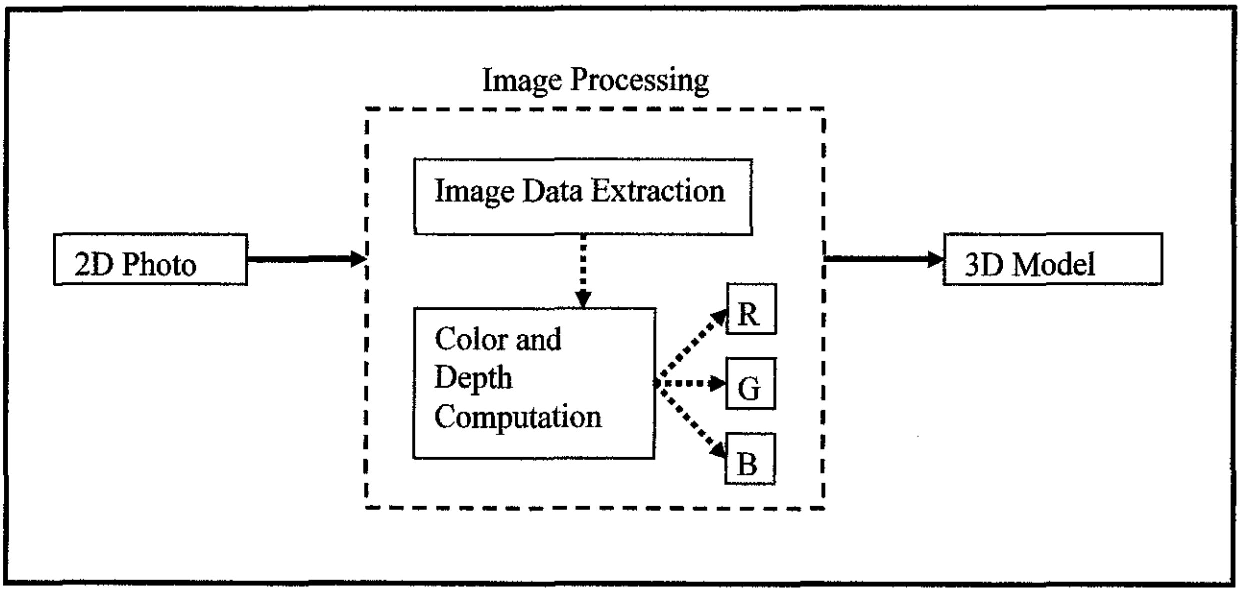


Figure 3.4 – Framework Overview

Initial code development focused on extracting the color information of the 2D image and storing the extracted data into a temporary array buffer. Since the size of buffer depends on the photo dimension (height x width) in pixels, the image size was reduced to 256 x 256 pixels. Otherwise, the data extraction process would be slow and might exhaust the workstation resources.

The code continues on to manipulate the information gathered and an estimation of the model depth volume is calculated. The formula used to calculate the depth values applies the principles of light intensity and distance relativity. In other words, in the area of which its light intensity is the highest, the depth volume would be high as well, and vice versa.

The next step was to program the navigation keys for the experiments and application prototype. To ensure the simplicity of the program, the navigation keys programmed are the common keys of a computer and users could conduct the experiments smoothly. Below is the list of navigation keys programmed:

Table 3.1 - List of Navigation Keys

Key	Action
Up arrow	Rotate model at x-axis (clockwise)
Down arrow	Rotate model at x-axis (counter-clockwise)
Right arrow	Rotate model at y-axis (clockwise)
Left arrow	Rotate model at y-axis (counter-clockwise)
F1	Decrease depth factor
F2	Increase depth factor
F3	Model → Ping Pong Ball
F4	Model → Ping Pong Ball (Enhanced Image)
F5	Model → Cylinder
F6	Model → Human Face
F7	Model → Frog Miniature

3.1.5 Testing and Evaluations

During this stage, the application was tested on identified experiment objects and the results were evaluated. The experiments conducted were divided into two. The first one covers for different objects and evaluate the effects of the object's behavior and the effects of the photo characteristics. The second set of experiments tested on the effects of different depth formula used in producing the 3D model.

However, during this stage, the development of the application is still an ongoing process. In other words, as the experiments were conducted, any defects detected or if the model produced did not meet the acceptable level, further modifications were made. Detailed discussion on the testing results will be further described in the next chapter of this report.

CHAPTER 4
RESULTS AND DISCUSSION

During the testing and evaluation stage of the project, several experiments were carried out to verify the accuracy of the final 3D output. The experiments conducted throughout the project development aims to study the effects of several key factors on the 3D model. Basically, the experiments were divided into two; the first one caters for several sample subjects while the second one caters for different depth formulas. All sample photos were taken in the "mini studio" which was built to produce best photos of the sample objects.

4.1 EXPERIMENTS ON SEVERAL OBJECTS

4.1.1 White objects

Since the concept behind the depth calculation of the 3D models rely on the light intensity and color distribution of the 2D photo, the experiments started of by using objects with bright surface (white). The surface was ensured to have minimum light reflectivity or other disturbance to ensure its smoothness.

4.1.1.2 Experiment 1: Ping Pong Ball (Sphere)

First experiment used a Ping Pong ball to resemble a sphere. This is because a sphere is the simplest object that is built of continuous curves and zero sharp edges. Due to its unique characteristics, a sphere could be made as the reference object for the formula verification. And the comparison of the actual object with the rendered 3D model could also be done base on the radius of the sphere.

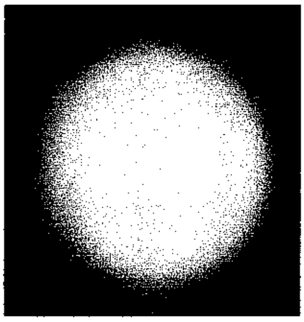
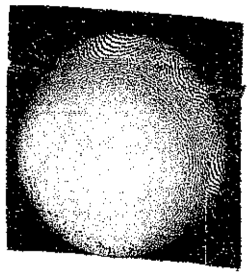
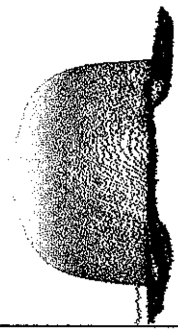
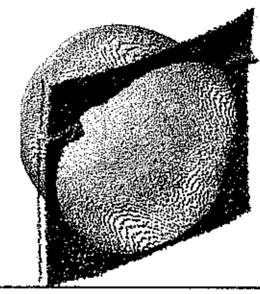
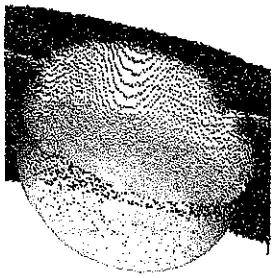


Figure 4.1 – 2D Photo of the Ping Pong Ball

Table 4.1 - Experiment 1 Result

Angle/View	**3D Model**
Front	
Side	
Back	

Top	

4.1.1.2 Experiment 2: Drink Can (Cylinder)

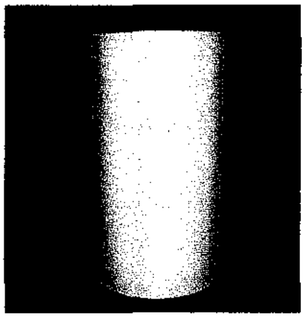
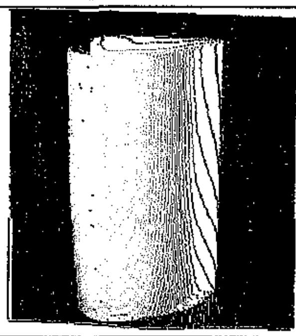
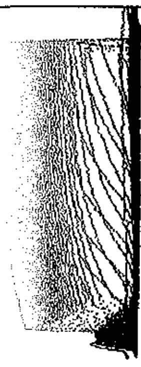
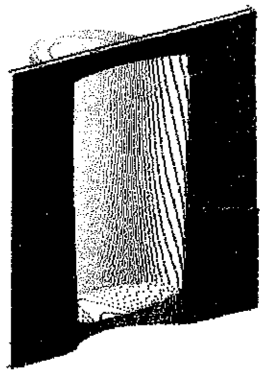
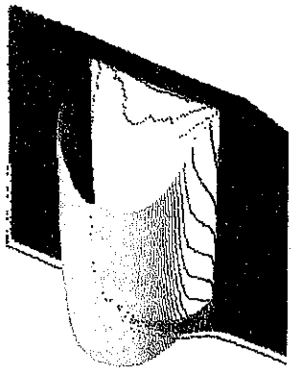


Figure 4.2 – 2D Photo of a Can

Table 4.2 - Experiment 2 Result

Angle/View	3D Model
Front	
Side	

Back	
Top	

4.1.1.3 Experiment 3: Frog Miniature

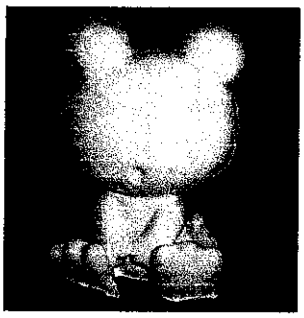
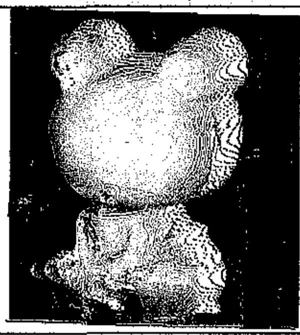
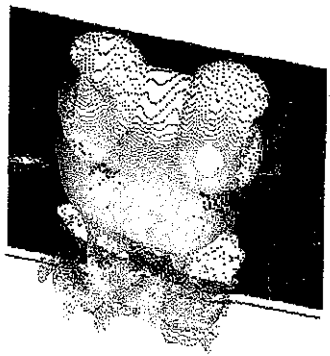
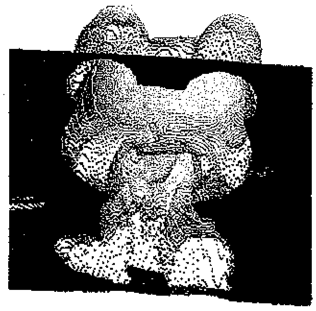


Figure 4.3 – 2D Photo of a Frog Miniature

Table 2.3 - Experiment 3 Result

Angle/View	3D Model
Front	
Side	
Back	
Top	

Bottom	

4.1.2 Colored objects

To test on the consequences of color distribution on the final output of the 3D model, several colored photos were used. These experiments were conducted without considering the external factors of the photos used such as the background color, environment noise (multiple light source) and object's characteristics.

The difference between these two sections of experiment could be clearly seen and the result confirms the theory of color influence on the final output.

4.1.2.1 Experiment 4: Drink Can

The result showed a lot of image distortion thus producing a low quality 3D model compared to the can in white and black photo. This may be caused by the characteristic of the can surface which is made of aluminum and has high reflectivity. And since the can is built of colorful designs on its surface, it also affects the final outcome of the 3D model.

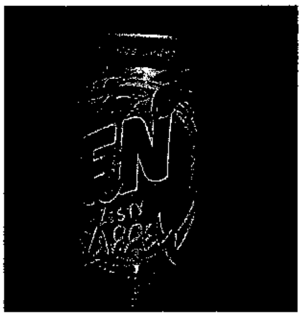
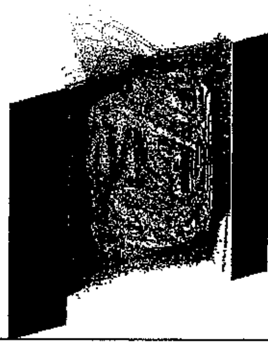


Figure 4.4 – 2D Photo of a Can

Table 4.4 - Experiment 4 Result

Angle/View	**3D Model**
Front	
Side	
Back	

Top	

4.1.2.2 Experiment 5: Frog Miniature

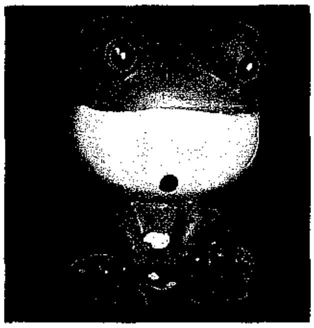


Figure 4.5 – 2D Photo of a Frog Miniature

Table 4.5 - Experiment 5 Result

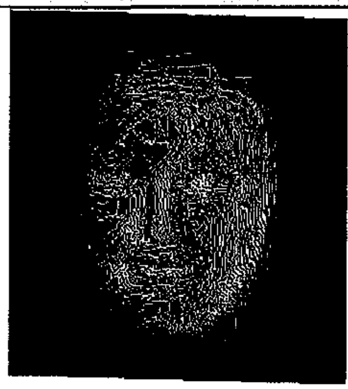
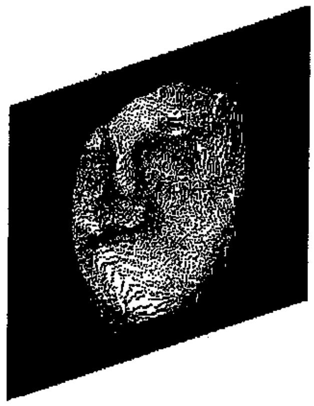
Angle/View	3D Model
Front	

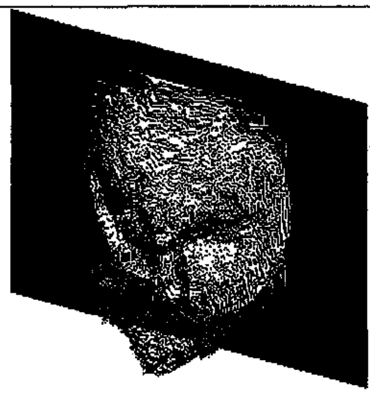
Side	
Back	
Top	

4.1.2.3 Experiment 6: Human Face

Figure 4.6 – 2D Photo of a Human Face

Table 4.6 - Experiment 6 Result

Angle/View	3D Model
Front	
Side	
Back	

Top	

4.2 EXPERIMENTS USING DIFFERENT DEPTH FORMULAS

The formula used to produce the 3D model was formed based on the idea that light intensity of a pixel could be derived from its RGB values. As a start, the experiment was conducted by using the average light intensity calculation to observe the final outcome of the formula used. However, it has also been discovered that the 3D model produced by this application is also affected by the denumerator of the formula. Therefore, the next experiments were conducted to test on the affects of different denumerators used towards the depth volume.

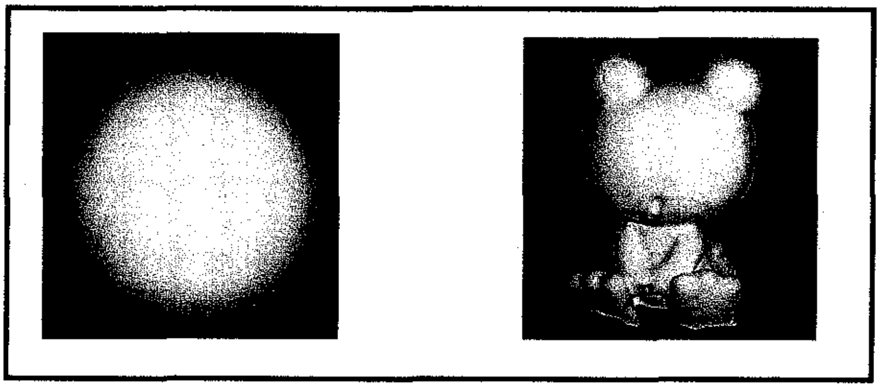
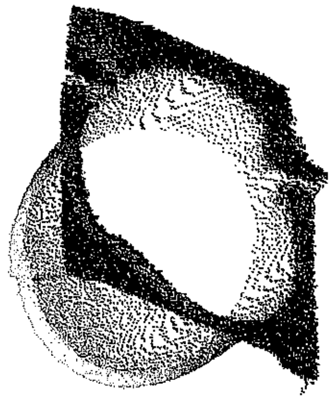
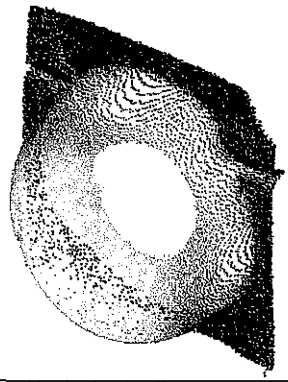
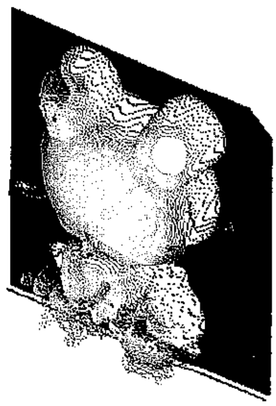
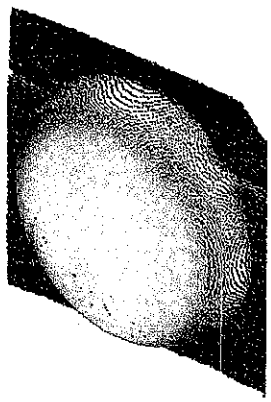


Figure 4.7 – 2D Photo of the Experiment Subjects

Table 4.7 - Depth Volume Experiments Result

Formula Used	3D Model
Depth = (R+G+B) / 3	 
Depth = (R +G+B) / 5	 
Depth = (R +G+B) / 10	 

Depth = (R +G+B) / 1		

4.3 DISCUSSION AND ANALYSIS

The results showed that the formula used in the application might not be the best method to produce a 3D model from a 2D photo. However, the idea could be further analyzed and further attempt could be made to repair the final outcome of this method. The framework design and the ideas could be the base research for further development in the 3D modeling research and analysis study.

However, good results could be obtained if the photos used in this application follow all the set of laws stated previously. As shown in the first experiments result section, the photos of objects which were covered with non reflective material, white in color and were photographed in a dark environment, produced highly acceptable results.

In comparison to the results of experiments that used colored objects and neglect the rules of disclosed and dark surrounding, the models are greatly distorted and at some point, the models are almost unrecognizable. There are two possible explanations behind the distorted models. First, the distortion may be caused by the high light reflectivity in the photos and the fact that the photos were photographed with multiple light sources. Second, the distortion might also be caused by the color characteristics of the objects. Since the light intensity formula was derived from the summation of the RGB values of each pixel, the colors would also be a factor for the 3D model. That is why, in areas which was colored with darker colors might have lower depth volume, and in areas which was colored with brighter color have larger depth volume.

The next section of the experiments was conducted to test on the affects of different depth factor used in the formula towards the final outcome of the application. Based on the results obtained, an average light intensity does not truly produce an accurate and precise 3D model. Nevertheless, by changing the denumerator of the formula to other values, some improvement could be seen in the illusion of the 3D models. Based on the author's observation and judgment, by dividing the RGB values by 4.5–5.5, the result of the 3D model seems to be mostly acceptable.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

Conclusively, all the objectives of the project have been achieved. Although the outcome of the 3D models are not as fine and as smooth as anticipated, the idea of using light intensity in calculating depth value of a pixel was proven to be possible. The solution proposed also meets the target to find out simpler methods of 3D modeling from 2D photos. It does not involve a lot of computational processes and users do not have to identify any reference points on the photos like most of the current software requires.

Nonetheless, the results obtained from the experiments showed that the formula used in the application might not be the best method to produce a 3D model from a 2D photo. Among the factors that were considered in the study are the effects of photo lighting, surrounding environment, attributes of the light source, contrast factor of the image, image brightness and light ambience, as well as the factor of the camera angles. The idea could be further analyzed and further attempt could be made to repair the final outcome of this method.

Below are a few recommendations to the project:

Base research to 3D solid objects remodeling applications

As stated above, this project could be made as a base research for further development in 3D objects modeling studies. Based on the results obtained, there is plenty of space for improvement in the application. The solution proposed does not produce a solid and smooth 3D model and this might reduce its realism and appearance. Further modifications could be done on the codes so that the final outcome would look much more realistic and precise. It could have been done by implementing use of polygons to render the model instead of disjoint points.

Capability to render multiple views of the 3D objects

The proposed solution only renders the front view of the image and output it as a 3D model based on the estimation of z-depth values. To further improve this solution, enhancement towards the 3D models would increase the final output quality and interaction. However, there might be some possible complexity in the code to produce a complete dimension of 3D models just by referring to a single 2D photo. Another option would be to use multiple views of 2D photos and based on the photos' overlaid points, the 3D model could be rendered. Hence, it might be possible to produce a full dimension 3D model.

Further research on the effects of real object's sizes, color, texture and other attributes that might contribute to the precision of the final output

Based on the research findings, a lot of factors need to be considered in producing high quality 3D models based on 2D photos. A deeper research on these matters might be able to improve the result and might even give a way to better solutions of modeling 3D objects from 2D photos.

REFERENCES

[1] PhotoModeler, Measuring and Modeling the Real World. <http://www.photomodeler.com>

[2] Wikipedia, 2D Computer Graphics. <http://www.wikipedia.com>

[3] Wikipedia, 3D Graphics. <http://www.wikipedia.com>

[4] Paul Bourke November 1993, *A Beginners Guide to Bitmaps*,

[5] Johnny Park and Guilherme N. DeSouza, *3-D Modeling of Real-World Objects Using Range and Intensity Images*.

[6] Sujin Liu and Zhiyong Huang 1996, *Interactive 3D Modeling Using Only One Image*, ACM SIGGRAPH '96: 163-170

[7] Angel Osorio 1998, *Making 3D models: A challenge in man-machine communication*, Proc. 2nd Euro. Conf. Disability, Virtual Reality & Assoc. Tech., Skövde, Sweden

[8] Paul E. Debevec 1999, *Image-Based Modeling and Lighting*, in *Applications of Computer Vision to Computer Graphics*, ACM SIGGRAPH, Vol. 33 No. 4, November 1999

[9] Byong Mok Oh, Max Chen, Julie Dorsey and Fr´edo Durand 2001, *Image-Based Modeling and Photo Editing*, ACM SIGGRAPH 2001, 12-17 August 2001, Los Angeles, CA, USA

[10] Douglas Fidaleo, G´erard Medioni, Pascal Fua and Vincent Lepetit, *An Investigation of Model Bias in 3D Face Tracking*

[11] Paul E. Debevec, Camillo J. Taylor and Jitendra Malik 1996, *Modeling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach*, ACM SIGGRAPH '96

[12] Virgilius Stănciulescu, *3D Tomography: Computerized Modeling of 2D Scanned Images. Results. Advantages*

[13] I. J. Grimstead and R. R. Martin, *Creating solid models from single 2D sketches*, University of Wales College of Cardiff

[14] Won-Sook Lee and Nadia Magnenat Thalmann, *Head Modeling from Pictures and Morphing in 3D with Image Metamorphosis based on triangulation*, MIRALab, CUI, University of Geneva, Switzerland

[15] Reinhard Klette August 1997, *Image Analysis and Object Surfaces*, Computer Science Department of The University of Auckland

[16] Ismail Oner Sebe, *Real-Time Image-Based Modeling*, Integrated Media Systems Center, University of Southern California, Los Angeles

[17] Holly Rushmeier, Jose Gomes, Laurent Balmelli, Fausto Bernardini and Gabriel Taubin, *Image-Based Object Editing*, IBM T.J. Watson Research

[18] Muralidhara Subbarao, *Efficient Depth Recovery through Inverse Optics*, Stony Brook

[19] Greg Turk and James F. O'Brien 2002, *Modeling with Implicit Surfaces that Interpolate*, ACM Transactions on Graphics, Vol. 21, No. 4, October 2002

[20] Muralidhara Subbarao July 1989, *Determining Distance from Defocused Images of Simple Objects*, IEEE Trans. on PAMI

APPENDICES

Application Code

Below is the part of code to extract the RGB values from the 2D photo. The code calls for *bitmap.c* and *bitmap.h* external files which enable the application to read the information in bitmap files:

```
.
.
.
for(i=0;i<=3*size*size;i++)
        myPixel[i]=(int)TexBits[i];

    n=0;
    for(i=0;i<size;i++)
        for(j=0;j<size;j++) // pixel values stored as GBR
        {
            zG[i][j] = (float)myPixel[n]/255.0;
            zB[i][j] = (float)myPixel[n+1]/255.0;
            zR[i][j] = (float)myPixel[n+2]/255.0;

            n = n + 3;
        }
.
.
.
```

Below is the code built to render back the 2D photo and turn it into a 3D model:

```
.
.
.
n=0;
        glBegin(GL_POINTS);
        for(i=0;i<size;i++)
                for(j=0;j<size;j++)
                {
                        // color back the pixels
                        glColor3f(zR[i][j],zG[i][j],zB[i][j]);

                        zDepth = (zR[i][j]+zG[i][j]+zB[i][j])/dFact;

                        // x,y,z position

                        glVertex3f((float)j/(float)size,(float)i/(float
                        )size, zDepth);

                }
glEnd();
.
.
.
```

Full C++ Code

```
#include <GL/glut.h>
#include <windows.h>

#include <math.h>
#include "bitmap.c"

#define size 256

float N=1, xRot = 0, yRot = 0, zDepth, dFact=3.0;
int i, j, n=0;

BITMAPINFO *TexInfo;  //Texture bitmap information
GLubyte    *TexBits;  //Texture bitmap pixel bits


int    myPixel[3*(size+1)*(size+1)];
float zR[size+1][size+1], zG[size+1][size+1], zB[size+1][size+1];

void Draw(void)
    {

       glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );

    glEnable(GL_CULL_FACE);
       glCullFace(GL_BACK);

       glPushMatrix();
       glRotatef(xRot,1,0,0);
       glRotatef(yRot,0,1,0);

       for(i=0;i<=3*size*size;i++)
               myPixel[i]=(int)TexBits[i];

       n=0;
       for(i=0;i<size;i++)
               for(j=0;j<size;j++) // pixel values stored as GBR
               {
                       zG[i][j] = (float)myPixel[n]/255.0;
                       zB[i][j] = (float)myPixel[n+1]/255.0;
                       zR[i][j] = (float)myPixel[n+2]/255.0;

                       n = n + 3;
               }

       n=0;
       glBegin(GL_POINTS);
       for(i=0;i<size;i++)
               for(j=0;j<size;j++)
               {
                       // color back the pixels
                       glColor3f(zR[i][j],zG[i][j],zB[i][j]);

                       zDepth = (zR[i][j]+zG[i][j]+zB[i][j])/dFact;

                       // x,y,z position
                       glVertex3f((float)j/(float)size,(float)i/(float)size, zDepth);

               }
       glEnd();

       glPopMatrix();

       glutSwapBuffers();
       }

void myInit(void)
```

```
    {
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-N,N,-N,N,-N,N);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

        glClearColor(1,1,1,1);
    }

void SpecialKeys(int key, int x, int y)
        {
        if(key == GLUT_KEY_UP)
                xRot -= 5.0 ;
        if(key == GLUT_KEY_DOWN)
                xRot += 5.0 ;
        if(key == GLUT_KEY_LEFT)
                yRot -= 5.0 ;
        if(key == GLUT_KEY_RIGHT)
                yRot += 5.0 ;

        if(key == GLUT_KEY_F1)
        {dFact -= 0.5;
        printf("DEPTH FACTORIAL: %f \n", dFact);}

        if(key == GLUT_KEY_F2)
        {dFact += 0.5;
        printf("DEPTH FACTORIAL: %f \n", dFact);}

        if(key == GLUT_KEY_F3)
    {TexBits = LoadDIBitmap("pp.bmp", &TexInfo);
        glutSetWindowTitle("2D Photo Converter - Ping Pong Ball (Original Image)");}

        if(key == GLUT_KEY_F4)
    {TexBits = LoadDIBitmap("pingpong2_edit.bmp", &TexInfo);
        glutSetWindowTitle("2D Photo Converter - Ping Pong Ball (Enhanced Image)");}

        if(key == GLUT_KEY_F5)
    {TexBits = LoadDIBitmap("silinder4.bmp", &TexInfo);
        glutSetWindowTitle("2D Photo Converter - Drink Can");}

        if(key == GLUT_KEY_F6)
    {TexBits = LoadDIBitmap("humanface.bmp", &TexInfo);
        glutSetWindowTitle("2D Photo Converter - Face");}

        if(key == GLUT_KEY_F7)
    {TexBits = LoadDIBitmap("froggie2.bmp", &TexInfo);
        glutSetWindowTitle("2D Photo Converter - Miniature of a Frog");}


        // Refresh the Window
        glutPostRedisplay();
        }

void main(void)
    {
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(512, 512);
    glutCreateWindow("2D Converter");
        myInit();

    glutDisplayFunc(Draw);
        glutSpecialFunc(SpecialKeys);

        puts("To view the model from different angles:");
        puts("Use the arrow cursors to rotate the model. \n");
        puts("Press F1 to decrease the depth factor.");
        puts("Press F2 to increase the depth factor.");
        puts("Press F3 for SPHERE model");
        puts("Press F4 for SPHERE model (Enhanced Image)");
        puts("Press F5 for SILINDER model");
```

```
    puts("Press F6 for a FACE model");
    puts("Press F7 for FROG model \n\n");

    printf("DEPTH FACTORIAL: %f \n", dFact);

   TexBits = LoadDIBitmap("humanface.bmp", &TexInfo);
   glutMainLoop();
}
```