3D model reconstruction using photoconsistency

Kirk Michael Joubert

Submitted to the Department of Electrical Engineering, Faculty of Engineering and the Built Environment, University of Cape Town in fulfillment of the requirements of the degree of Master of Science in Engineering

August 2007

Declaration

I declare that the unreferenced contents of this dissertation is my own work. Any concepts and data that has been obtained from external sources has been referenced to the best of the my ability.

This dissertation has been submitted to the Faculty of Engineering and the Built Environment, UCT, for the degree of Master of Science in Engineering and has not been submitted for any other degree at any other University.

I hereby grant the University of Cape Town free license to reproduce the above thesis in whole or in part, for the purpose of research.

Signed

KM Joubert

31 August 2007

Acknowledgments

I would like to thank:

Dr Fred Nicolls, Supervisor, for his patience and support.

Dr Gehard de Jager, Co-Supervisor, for his efforts in making sure this work was well funded.

The National Research Foundation and De Beers Technical Group for their financial assistance.

Abstract

Computer models of real world objects can be created using methods that involve the use of cameras. Active methods such as laser line scanning can determine the surface geometry of the object by passing a laser line over the surface of the object and using a camera to record the distortion of the line and using this, calculate the contours of the object. Passive methods such as silhouette based reconstruction uses the outline or silhouette of an object. A number of silhouettes obtained from different viewpoints of the object can be used to calculate a model of the object. Photoconsistency based methods use the surface color or texture to create a *photohull*, or an approximation of the object that is consistent with the available views of that object. This thesis presents the work done in implementing concepts and algorithms presented in the literature. The model reconstruction is based on voxels. The thresholds used by the photoconsistency measures, used to determine the photoconsistency of a reconstructed surface, are determined using an estimation procedure that uses histograms of the photoconsistency values of the voxels. The algorithm calculates the visibility and photoconsistency of voxels in an iterative cycle.

Contents

List of Figures

List of Tables

Glossary

BRDF or Bidirectional Reflectance Distribution Function: This is the function that describes the reflectance properties of a surface. For more information, see the section on Radiosity.

Camera reference frame: This refers to the three dimensional Cartesian space that defines the local world of the camera. The space is defined so that the Z axis of the space lies along the optical axis of the camera. The origin of the space is located where the pinhole of a pinhole model camera would be located.

Image reference frame: This refers to the two dimensional Cartesian space that can be used to approximate the photosensitive area of a camera. The distance units of the image reference frame is normally defined to be in pixels.

Occlusion: This term refers to the visibility state of a region of surface of an object. The surface is occluded if the direct line of sight of that surface is blocked by either another region of surface or another object.

Photoconsistency: In the context of this thesis, this term refers to whether the color of a region of surface, of an object undergoing computer reconstruction, is similar or consistent with the colors of the pixels, of all the camera images that correspond to the available viewpoints that have clear view of that region of surface, that are enclosed by the projection of that region of surface into the image plane.

Projection Matrix: The projection matrix is a mathematical construct, i.e a matrix, that defines the mathematical mapping of Cartesian points in a three dimensional camera reference frame to points in a two dimensional image reference frame.

Steradian: Like the radian, which is used as a measure of the angle between two lines, the steradian is used to measure the solid angle of a cone origi-

nating from the center of the sphere. The *radian* is defined as the length of the arc subtended by the two lines, divided by the radius of the circle containing that arc. The *steradian* is defined as the surface area of a sphere subtended by a cone, divided by the square of the radius of the sphere.

Visibility: This term refers to whether a region of surface is visible or not occluded when viewed from a particular viewpoint. This term is also used to describe what cameras can see a region of surface.

Voxel: This term refers to a small cubic region of space in three dimensions. This concept is similar to pixels in two dimensions.

World reference frame: This refers to the three dimensional Cartesian space that defines the physical world. The units of the world reference frame are normally chosen to conveniently match the scale of the physical world under consideration. In this thesis, the units are in mm. The origin of this space is arbitrary. In this thesis, the world reference frame is normally locked to one of the camera reference frames.

Symbol definitions

Bold capital letters refer to matrices. For example, the letter $\mathbf{P}$ refers to the projection matrix. Lowercase bold letters refers to vectors. For example, the letter $\mathbf{t}$ refers to the translation vector of a camera.

$\mathbf{t}$: Translation vector of a camera indicating its position in the world reference frame.

$\mathbf{R}$: Rotation matrix indicating the orientation of a camera (or the camera reference frame) in the the world.

$\mathbf{P}$: The projection matrix which maps points from the camera reference frame into the image reference frame.

L: The radiance of a light ray.

E: The irradiance of a surface due a single incident light ray.

ρ: The BRDF of a surface.

P: The set that contains all the pixel colors, from all the cameras that can see the same region of surface, corresponding to that region of surface.

p: The set of pixel colors from one camera corresponding to a region of surface of an object.

Chapter 1

Introduction

What is meant by 3D model reconstruction? In the context of this thesis, it refers to the creation of a computer model of a real world object using various computational techniques.

Techniques exist that can create a computer model of an object using methods that affect the appearance of the target object. Laser line scanning is one of these.In this method, a laser line is projected onto the surface of the source object. The reflection of the laser line is distorted by the object's surface geometry. Using either a single camera or a stereo pair, measurements of the distortion of the laser line are made. These measurements can then be used to estimate the geometry of the surface and thus a model of the object can be created. Related to laser line scanning is the use of adaptive structured light. This method uses a structured light pattern, such as a light grid, that is projected onto the surface of the object. If the object is non-planar then the projected light grid will appear distorted. The projected light pattern is altered until a recognized reflected pattern is obtained. The object geometry can then be estimated by the changes made to the structured light grid.

These methods all require that the object under construction be affected in some manner. There are techniques that do not require this. One technique uses multiple silhouettes of the object and the camera geometry to construct an approximation of the object. This method is limited in that it can only, at most, construct the outermost boundary of the object.

Seitz and Dyer describe another method in their paper [1] which uses the surface color information to reconstruct the geometry of the object. Basically, this method attempt to find a surface that when viewed from the different camera orientations, matches the corresponding camera view of the

target object. This surface then defines the model that is the most consistent with the available views. It is this method that formed the basis of the work done in this thesis.

1.1 Objectives of the thesis

The objectives of this thesis are threefold. First, a review of some of the literature available concerning the field in order to obtain a starting point for the development of the research. The second objective is to develop and implement a computer algorithm based on a method from the literature that can construct a computer model. The final objective is to evaluate the reconstruction performance of the developed algorithms.

1.2 Structure of the thesis

The thesis is divided into chapters according a particular aspect of the work done. The beginning of each chapter will contain a summary or overview of the chapter as a whole.

Chapter 2 presents a brief overview of the mathematics used to describe the operation of a simple camera. These mathematical equations are needed to relate the object in the real world to the corresponding image views of the object and are vital to the operation of the reconstruction algorithms. Chapter 3 describes a method of creating computer models using the silhouettes of the object under construction. A simple method is also presented to accomplish such a reconstruction. Chapter 4 discusses the concept of model reconstruction using photoconsistency. Chapter 5 provides an overview of the datasets used to create the models presented in the thesis and how they were obtained. Chapter 6 describes the operation of the algorithms developed in the thesis. Chapter 7 presents some results that were obtained from the use of the various algorithms. Finally, Chapter 8 provides the conclusions of the author concerning the results that were achieved.

Chapter 2

Mathematics of pinhole cameras

This chapter presents a brief overview of the mathematics of a pinhole camera. Essentially, the mathematics relate points in a world reference frame to corresponding points in an image reference frame. Most of the information presented below about intrinsic parameters comes from [2]. Normal cameras use lenses not pinholes, but a pinhole camera can approximate, within limitations, the char

The parameters that are used to describe the characteristics of the pinhole camera are divided into two groups, *intrinsic* and *extrinsic*. *Intrinsic* parameters define the optical properties of the camera such as focal length. The *extrinsic* parameters define the location and orientation of the camera in the world. Both these sets of parameters are needed to relate points in the camera image to points in the real world.

First, three reference frames need to be defined. The *world reference frame*, the *camera reference frame* and the *image reference frame*. The image reference frame refers to the structure and definition of the image plane. The image plane is two dimensional, therefore the reference frame is also two dimensional. The distances in image plane are defined to be in pixels and there are two principle axes, X and Y. The origin of this reference frame is located in the top, left hand corner.

The camera reference frame is simply a right handed three dimensional Cartesian space. The pinhole is located on the origin of the space and optical axis is located along the positive Z axis. The units in this reference frame are defined to be the same as the world reference frame.

The world reference frame is also a right handed three dimensional Cartesian space. This reference frame relates to the physical world. The origin and orientation of this reference frame are arbitrary. The units are defined to be a measurement that is most convenient for the scale required.

The following sections provide a basic overview of the extrinsic and intrinsic parameters of a pinhole camera.

2.1 Intrinsic parameters

The pinhole camera is a device that recreates an image of an object from the light emitted from the object. The basic setup of a pinhole camera is illustrated in Fig. 2.1.

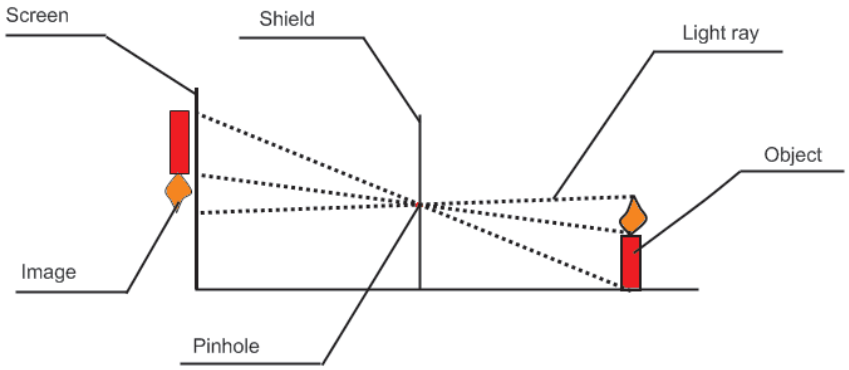


Figure 2.1: Diagram illustrating the concept of the pinhole camera and its operation

Essentially, the pinhole camera isolates single light rays that travel a line from the object through the pinhole to the screen. Without the pinhole, multiple light rays from different parts of the object would all converge on the same point on the screen and no discernible image would be formed. For simplicity, it is convenient to imagine a screen in front of the pinhole instead of behind. A virtual image is formed on this imaginary screen. Now the operation of the pinhole camera can be defined mathematically. The screen and pinhole are defined in the camera reference plane. Fig. 2.2 illustrates this setup.

The optical axis shown in Fig. 2.2 refers to the central axis of the camera. The location of the projected point in the image y' and the location of the

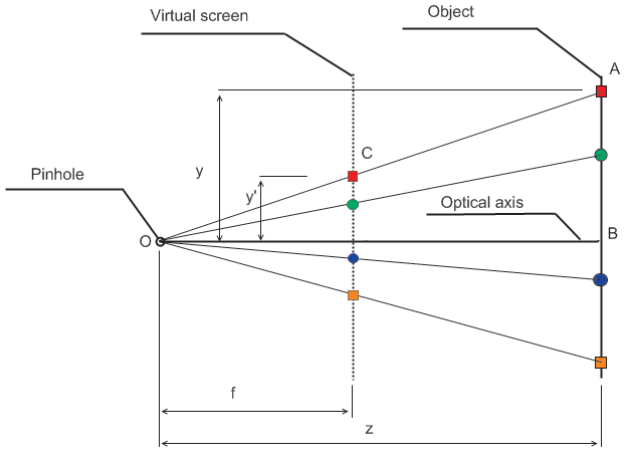


Figure 2.2: Diagram illustrating the concept of a virtual image obtained by placing a virtual screen in front of the pinhole.

point in the camera world y are linked by a simple equation calculated by looking at the triangle formed by OAB and OCD. The equation is given by:

$$\frac{y'}{f} = \frac{y}{z} \tag{2.1}$$

y' represents the height of the projected point in the camera image plane, y is the actual height in the camera world, f is the distance of the virtual screen from the pinhole and z is the distance of the camera world point from the pinhole.

The diagram Fig. 2.2 is only two dimensional representation. However, the same principle applies to a three dimensional pinhole camera. If the coordinates of a point in the camera image plane are (x', y') and the coordinates of the point in the real world are (x, y, z) then the equations linking the two are as follows:

$$x' = \frac{f}{z}x \tag{2.2}$$

$$y' = \frac{f}{z}y \tag{2.3}$$

The above equations assume that the origin $(0, 0)$ of the image plane is centered along the optical axis of the camera. This may not always be the case. For instance, the origin of the digital images stored in computer format are located in the uppermost left hand corner. If this is the case, all the image points need to be shifted by a certain amount to take this into account. The point at which the centerline intersects the image plane is called the principal point and has coordinates (P_x, P_y). The equations are modified to become:

$$x' = \frac{f}{z}x + P_x \tag{2.4}$$

$$y' = \frac{f}{z}y + P_y \tag{2.5}$$

These equations define a mapping of the points in the three dimensional *camera reference frame* to the corresponding points in the two dimensional *image reference frame*. Note that no units are specified. It is up to the user to define the parameter f to match camera world units to image units. In this thesis, the camera world units are defined to be in millimeters and the image units are defined as pixels. Therefore, the unit of f is [pixels/mm].

2.2 Extrinsic parameters

The preceding section describes the parameters needed to map points in the camera world to points in the image plane. The extrinsic parameters relate the points in the actual real world to points in the camera world. In other words, the extrinsic parameters map points from the *world reference frame* to the *camera reference frame*.

There are two basic extrinsic parameter vectors, translation and rotation. The translation vector $\mathbf{t}$ describes the location of the camera (or the camera reference frame) relative to the world reference frame. The rotation matrix $\mathbf{R}$,describes the orientation of the camera in the world (or the orientation of the camera axes relative to the world axes).

The points describing the object in the real world lie in the world reference frame. These points need to be mapped into the camera reference plane so that they can be further mapped into the image plane, the objective of the camera mathematics.

It is simpler to first find the mapping from the camera reference plane to the world reference plane. Assume a point P defined by a vector $\mathbf{x}$ relative to the camera reference frame. Since the orientation of the axes of the camera frame relative to the world frame is given by $\mathbf{R}$, the orientation of the vector $\mathbf{x}$ in the camera frame corresponds to a vector in the world reference frame, given by $\mathbf{x'}$, is simply:

$$\mathbf{x'} = \mathbf{R}.\mathbf{x} \tag{2.6}$$

The origin of the camera axis lies at a point given by the translation vector $\mathbf{t}$. Therefore, the origin of the vector $\mathbf{x'}$ in the world is simply $\mathbf{t}$. The mapping of the point P, described by the vector $\mathbf{x}$ in the camera reference frame, to the point P' described by the vector $\mathbf{x'}$ in the world reference frame is simply:

$$\mathbf{x'} = \mathbf{R}.\mathbf{x} + \mathbf{t} \tag{2.7}$$

However, this is the mapping from the camera frame to the world frame. The inverse mapping can be found by:

$$\mathbf{R}^{-1}\mathbf{x'} = x + \mathbf{R}^{-1}\mathbf{t}x = \mathbf{R}^{-1}\mathbf{x'} - \mathbf{R}^{-1}\mathbf{t}$$

Due to the mathematical properties of the rotation matrix (it is orthogonal), the inverse is simply $\mathbf{R}^T$. Therefore the mapping from the world reference frame to the camera reference frame is:

$$x = \mathbf{R}^T\mathbf{x'} - \mathbf{R}^T\mathbf{t} \tag{2.8}$$

2.3 The projection matrix

The sections discussing intrinsic and extrinsic parameters defines the equations and parameters needed to map points in the world reference frame to the camera reference frame, and from the camera reference frame to the image reference frame. These equations can be combined in a fashion to produce the *projection matrix* which can be used in the various computer algorithms. The extrinsic equation 2.8 can be rewritten in matrix form as:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.9}$$

The intrinsic equations 2.4 and 2.5 are nonlinear, however they can be converted into matrix form to a limited extent:

$$z.x' = f.x + z.P_x$$

$$z.y' = f.y + z.P_x$$

$$\begin{bmatrix} a \\ b \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & P_x \\ 0 & f & P_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.10}$$

$$x' = \frac{a}{w} \tag{2.11}$$

$$y' = \frac{b}{w} \tag{2.12}$$

These matrix representations can be combined into one representation as follows:

$$\begin{bmatrix} a \\ b \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & P_x \\ 0 & f & P_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.13}$$

$$\begin{bmatrix} a \\ b \\ w \end{bmatrix} = \mathbf{P} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.14}$$

$$x' = \frac{a}{w} \tag{2.15}$$

$$y' = \frac{b}{w} \tag{2.16}$$

The matrix **P** is called the *projection matrix*.

This matrix equation with associated non-linear equations defines the mapping of the points in the world reference frame to the image reference frame.

Normal cameras do not use pinholes. Pinholes are inefficient in that they block a large amount of the reflected light from an object. This leads to dark indistinct images. In order for a focused image to be formed, a pinhole has to be small. The smaller the pinhole, the better the image definition and quality but the lower the amount of light. Normal cameras use lenses. The lenses are shaped to focus the light rays originating from a point on the object surface back to a point on the screen. This behavior is similar to the operation of the pinhole, but much more light from the object is being utilized.

For simple thin lenses, these projective equations can also be used.

Chapter 3

Reconstruction using silhouettes

This chapter presents a method that uses the silhouettes or shadows of a model to perform a reconstruction. The first section describes what is meant by silhouettes and how they are obtained from the various views of the object. The second section briefly describes the concept of the visual hull, a construct that is formed from the use of the silhouettes. Finally, the last section describes a simple method of using silhouettes to generate an approximate computer model of the object.

3.1 Silhouettes and image segmentation

What is meant by the silhouette of an object? The silhouette is comparable to the shadow that the object would cast on a screen when placed in front of a light source. The silhouette consists of a region that is bounded by the boundary of the object when seen from a particular viewpoint. Fig. 3.1 depicts a view of an object and the corresponding silhouette.

These silhouettes are obtained by segmenting the image into two regions, foreground and background. The foreground represents the actual model while the background represents the regions that are not included in the object boundary. The methods available to automatically segment the image into suitable foreground and background regions is beyond the scope of this thesis. In the datasets used in the thesis, those that were not obtained from external sources, the images were segmented manually with an image editor.

Figure 3.1: Image of object and corresponding silhouette.

3.2 The visual hull

The visual hull is the polyhedral shape that is consistent with the projections of all possible silhouettes of the object under observation. In the context of this thesis however, the term visual hull is used to describe that shape which is consistent will the available silhouette views.

The diagram Fig. 3.2 demonstrates this concept of the visual hull. The solid region in the diagram represents an arbitrary object while the shaded region represents the shape that is formed by the intersection of the silhouette cones. The silhouette cones are the back projections of the silhouettes in the world. In other words, the silhouette cone forms the boundary of all possible shapes in the world that, when projected into the image plane, are consistent with the image silhouette.

As Fig 3.2 illustrates, the visual hull is only an approximation to the actual object. Even with all possible views, the visual hull will not model the concavities in the object that are not visible in any of the silhouettes. This is the limitation of the visual hull. For example, the visual hull will not model the inside of a bowl or a cup as these concavities are located inside the outer boundary of the object.

However, the visual hull is useful as a starting point for other reconstruction algorithms as it defines the outermost boundary of the object and thus removes irrelevant background information.

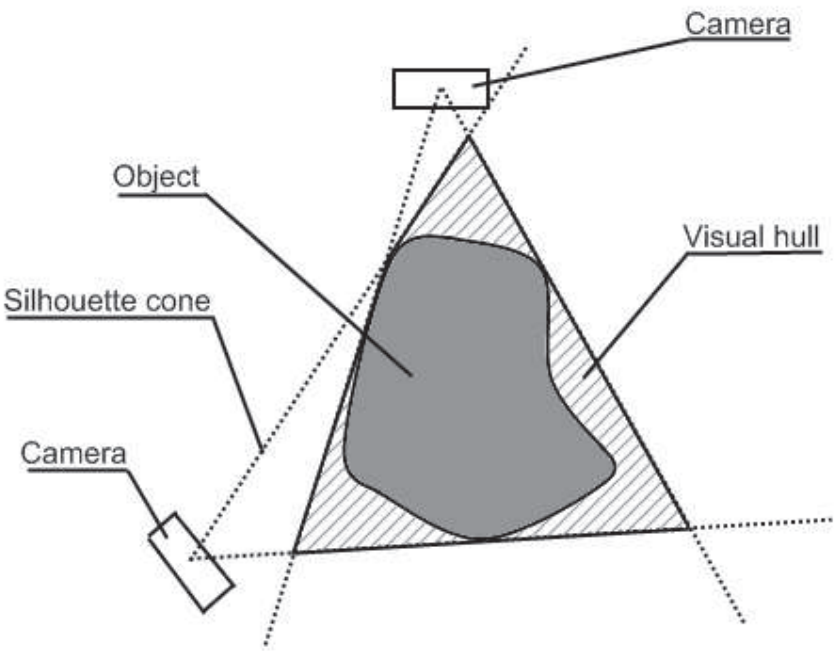


Figure 3.2: Concept of the visual hull

3.3 Reconstruction using voxels

This section describes a simple method of reconstructing an object using voxels. A voxel, as used in this thesis, is defined to be a small volume in the world. A number of voxels can be used to build up an object. This process can be likened to building a structure using toy blocks.

First a voxel grid, a cube built up of a number of voxels, is defined. The voxel grid is situated in the world in such a manner as to encompass the entire volume of the object to be constructed. Each voxel is then projected into the image planes of the available cameras. In this case, only the center of the voxel is projected into the image planes. This center point is checked to see if it lies inside the silhouettes of all the available views. If this is the case, the voxel is marked as consistent. However, if the projected center lies outside of any of the silhouettes views, it is marked as inconsistent and is discarded. The end effect of this process is to carve away voxels that are not consistent with the silhouettes. The voxels remaining should be an approximation of the visual hull.

Chapter 4

Reconstruction using photoconsistency

This chapter discusses the main concept behind the work done in this thesis. The chapter begins with an explanation of what photoconsistency is and how it is applied to voxel based model reconstruction. Next, a brief description of **Radiometry** is provided. After that section, a discussion of visibility and its effects on photoconsistent reconstruction is presented and finally, some photoconsistency measures are described. These measures were taken mostly from [4].

4.1 Photoconsistency and application to voxels

Photoconsistency describes a state whereby an object's surface color or texture is consistent with all available camera views of that surface. In other words, the appearance of the reconstructed model matches the available camera images of the target object.

In the case of voxel based model reconstruction, the model is composed of voxels and the surface is a subset of these voxels. A voxel is photoconsistent if its projections, into the relevant camera views, overlaps a set of pixels which forms a set that is defined to be similar. The definition of whether this set of pixels is considered to be similar depends on the photoconsistency cost function that is used. In this thesis, the color of the pixels is used to define the similarity of the set.

The motivation behind this method is that is the pixels are completely

dissimilar, the voxel cannot lie on the surface of the model therefore it is incorrectly placed and must be removed from the overall model construction. However, the converse is not necessarily true. A voxel that maps to similar pixels could still be incorrect due to the fact that this method does not take the relative orientation of the pixels into account. It is not a spatial measure. Therefore, a scenario may arise that regions of the surface of the target object have similar coloring or texture but are located in different places thus leading to a poor reconstruction.

4.2 Radiometry

This thesis makes the assumption that the surface of the object under construction is *Lambertian* and that the lighting is diffuse. To aid in understanding this concept, this section briefly touches on Radiometry. The principal source of this information is [3].

Radiometry is the science concerning the measurement of light. It defines the characteristics and measurement units of light and describes its physical properties under certain conditions. Photometry is the science of the perception of light. It is similar to radiometry, but must take into account the non-linear aspects of the human eye.

Radiometry can be used to describe the optical properties of a surface. For instance, how does one describe a "matte"surface or a "shiny" surface mathematically?

To begin, some definitions of concepts used in radiometry are presented in the following text.

Radiant energy

Radiant energy is the total amount of energy conveyed by the photons of the light being measured in a particular interval of time. The unit of radiant energy is the Joule (J).

Flux

Flux is the radiant power or rate of energy conveyed by the photons of light. The unit of flux is the Watt (W equivalent to $J.s^{-1}$).

Flux density

Flux density is the amount of radiant power or flux in a *perpendicular* crossection of the traveling rays of light. In other words, it the amount of energy flowing through an area per second. The unit of flux density is Watt per square meter ($W.m^{-2}$ equivalent to $J.s^{-1}.m^{-2}$). If the light is entering a surface, then the flux density is called irradiance. Essentially, irradiance is the power entering a surface.

Radiant Intensity

Radiant intensity defines the Flux contained in a single ray of light. The ray is modeled as an extremely small cone that extends from the source outwards. The units of radiant intensity is Watts per steradian ($W.st^{-1}$)

Radiance

Radiance defines the Flux density of a single ray of light. The ray is modeled as a extremely small cone that extends from the source outwards. The units of radiance is defined as Watts per square meter per steradian ($W.m^{-2}.st^{-1}$ equivalent to $J.s^{-1}.m^{-2}.st^{-1}$)

Radiosity

Radiosity refers to the flux density of the light emitted from a point on the surface. Radiosity can be calculated by integrating over the radiance from all viewing angles. Radiosity has units Watts per square meter ($W.m^{-2}$). This measure is useful for surfaces that has a nearly uniform radiance from all viewpoints.

The optical properties of a surface can be defined as the relationship between the radiance of the incoming light striking the surface and the radiance of the light leaving a surface. The mathematical function linking the magnitude of these two parameters is called the BRDF or Bidirectional Reflectance Distribution Function.

The incoming and outgoing light from a point on the surface is modeled as rays, or straight lines with an origin at the point on the surface and a direction expressed in spherical coordinates. The diagram Fig 4.1 illustrates this concept.

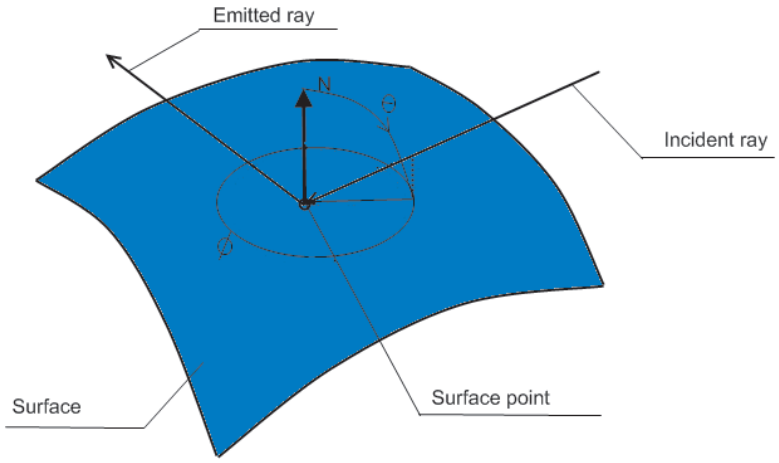


Figure 4.1: Diagram illustrating the concept of rays entering and leaving a point on an arbitrary surface. The direction on the incoming and outgoing rays are defined in spherical coordinates, ϕ and θ. ϕ is the azimuth and θ is the elevation of the ray and N is the surface normal at the point. The azimuth is measured along the surface tangent plane defined by N and the elevation is measured starting from N

There is a difference between Radiant Intensity and Radiance. Radiant Intensity simply refers to the amount of power contained in a single ray of light. However, it gives no indication as to the density or number of rays present. Radiance accounts for the distribution of the rays.

There is a subtle difference between *radiance* and*irradiance*. The irradiance of a patch is not necessarily the radiance of the rays striking that patch.

The irradiance of the light falling on the point can be found from the radiance of the incoming rays of the light at that point. Assume that the point on the surface can be represented by a small patch tangent to the surface. Then, the *irradiance* E of the patch is related to the *radiance* L_i of the incoming light by **Lambert's Cosine Law**.

$$E = L_i \cos(\theta_i) \tag{4.1}$$

where θ_i is the elevation of the incoming ray of light referenced to the surface normal at the point on the surface. The cosine term comes from the fact that a patch, when viewed at an angle, seems smaller that it actually is as the diagram, Fig 4.2 illustrates.

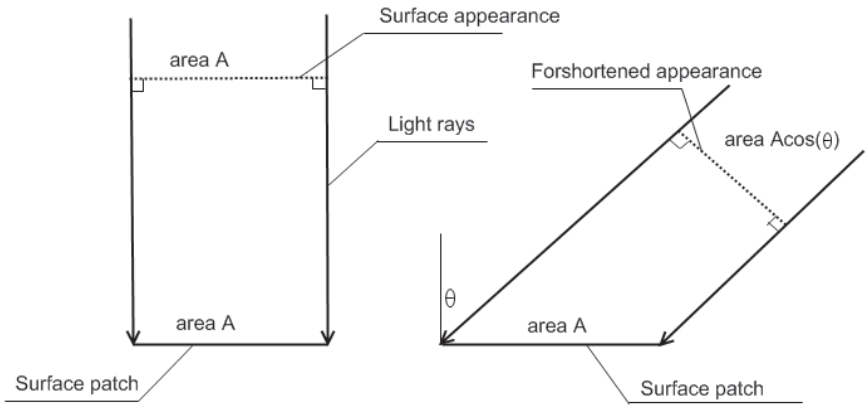


Figure 4.2: Diagram illustrating the change in appearance of a surface as the viewing angle is changed. This change has the effect of reducing the required crosssection of the light rays needed to irradiate the surface patch.

Since a smaller crossection of light rays are needed to irradiate the surface patch as the incident angle is changed, the amount of energy falling on that patch reduces by a cosine factor.

The BRDF is defined as the ratio of the radiance of the emitted ray to the irradiance due to the radiance of the incoming ray. The BRDF is defined as

$$\rho\left(\phi_i, \theta_i, \phi_o, \theta_o\right) = \frac{L_o\left(\phi_o, \theta_o\right)}{L_i\left(\phi_i, \theta_i\right)\cos\left(\theta_i\right)} \tag{4.2}$$

The BRDF can describe the optical properties of the surface. For instance, a "shiny" or *specular* surface reflects light strongly in a particular direction therefore the BRDF will have a strong peak while a "matte" surface reflects light more or less equally in all directions therefore the BRDF will be a smooth, uniform function.

Now, the term *Lambertian* refers to a surface that has a constant BRDF. In other words, incoming rays are scattered in a manner so that they are emitted in a uniform hemisphere from the surface. Matte surfaces are a reasonable approximation the theoretical Lambertian surface.

If the lighting is diffuse so that there is no strong component of light from any particular direction, then the emitted radiance from a Lambertian surface will be uniform from all viewing angles and the surface will thus look the same from different viewpoints.

Essentially, the assumption used in this thesis is that the lighting and the optical properties of the surface are such that it looks the same from different viewpoints.

4.3 Visibility

This section describes the effect of visibility in the reconstruction process. Visibility, as defined in the context of this thesis, refers to the ability of a camera to see a particular portion of the surface of the target object. In this case, the portion is approximated by a voxel.

The visibility information is required to determine the photoconsistency of the surface of the model. This is illustrated by Fig. 4.3 which demonstrates why visibility information is needed.

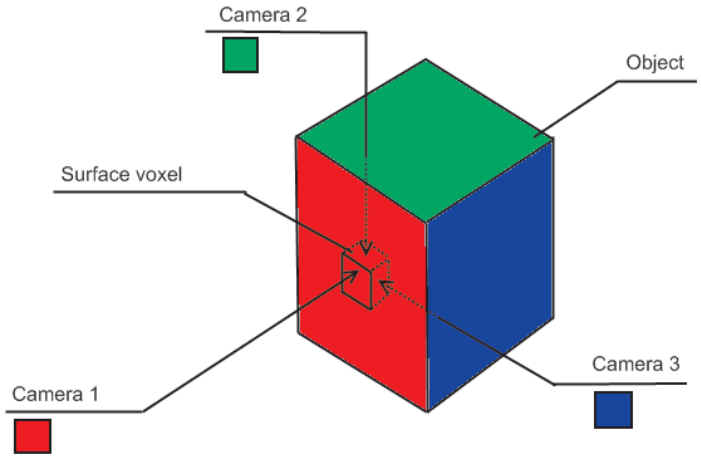


Figure 4.3: Diagram illustrating the consequences of not having visibility information. The voxel designated 'surface voxel' lies on the surface of the object and is therefore should be photoconsistent. However, due to the lack of visibility information, cameras 2 and 3 which cannot normally see the voxel injects erroneous information and causes the voxel to be labeled inconsistent.

As Fig. 4.3 describes, the visibility information is needed to reject camera views that cannot see the surface of the object and therefore may provide erroneous information.

As stated, the visibility information is vital to the accurate reconstruction of the target object. However, the only information available are the camera parameters and viewpoints of the target object. To calculate the visibility information accurately requires a model of the target object as a starting point. Herein lies the difficulty in reconstructing objects using photoconsistency as the model is required to determine the visibility and the visibility is required to determine the model. Therefore, the surface structure and visibility need to estimated at the same time during the reconstruction process.

It is possible to estimate the visibility information from the visual hull, since the visual hull is an approximation to the surface of the object. However, this requires the use of silhouettes and the visibility information degrades the further the actual surface is from the visual hull.

In this thesis, an iterative process is used whereby the visibility of a starting model is calculated and using this information, the photoconsistency of the surface voxels are evaluated. A new model is formed based on the photoconsistency results and the visibility recalculated. The procedure is iterated in the assumption that it will converge to an acceptable solution whereby both the conditions of visibility and photoconsistency are satisfied. Further details are described in the algorithms chapter of this thesis.

4.4 Photoconsistency measures

This section lists a number of photoconsistency values some of which are mentioned in [4].

If a voxel is projected into a camera view, the set of colors of the pixels that it overlaps is defined for the purposes of this text to be p_i where i is the number of the corresponding camera if the images are in grayscale.

Assume that a set P is defined to be the union of all the sets p_i from set of all the relevant camera views designated N. That is

$P = \bigcup_N p_i$

4.4.1 Standard Deviation

This measure is simple and straightforward. The measure σ is calculated by

$$\sigma = \sqrt{\frac{1}{M}\sum_{i=1}^{n}(P - \overline{P})^2} \tag{4.3}$$

where M is the number of elements in the set and $\overline{P}$ is the mean of the set.

This measure assumes that if the spread of the colors in the set is large, then the voxel cannot be similar and the standard deviation will be a high value. Conversely if the spread is small then the pixels are similar, the standard deviation is small and the voxel is photoconsistent.

In order to discriminate between a consistent and inconsistent set, a threshold needs to be defined whereby any value below the threshold is considered consistent and every value above the threshold is inconsistent.

This measure does suffer a drawback. If the surface of the object is highly textured, then the spread of the colors in the set will likely be high and therefore the measure will be high.

Therefore, the voxel will be consistent if the standard deviation σ is less than a specified threshold.

4.4.2 Adaptive threshold

The adaptive threshold measure is designed to compensate for the drawback of the standard deviation measure. The reasoning behind adaptive threshold is that region which have high texture will have a greater standard deviation. Therefore, the standard deviation of each set p_i designated σ_i is also calculated. The average of these standard deviations is calculated to give an indication of the spread of the pixel values due to the texture of the surface.

The value of the measure τ is then given by

$\tau = \sigma - \alpha\overline{\sigma_i}$

where σ is the standard deviation as calculated by 4.3 and α is a threshold.

As before, the voxel is considered consistent if the value of τ is below a certain threshold.

The disadvantage of this measure is that it requires two preset thresholds.

4.4.3 Histogram

The histogram method calculates consistency based on the intersection of color histograms. For each set p_i a color histogram is generated. Pairs of sets are extracted and the histograms compared. If the histograms are *not* a match then the procedure is halted and the result returned is that of inconsistency.

In this method, a matching function needs to be defined. This matching function determines whether the pairs of histograms are a match or not.

[6] describes a histogram based photoconsistency measure of which the match measure declares a histogram to be a match if any two corresponding histogram bins are nonzero.

4.4.4 Root mean squared

This measure was not explicitly extracted from the literature and was defined by the author and is the measure used in the reconstruction algorithms mentioned in this thesis.

This particular measures calculates the maximum root mean squared error of the elements in each set p_i from the assumed color of the voxel undergoing the test. The color of the voxel is assumed to be the average of all the color elements in the set P. The maximum root mean squared error is retained and compared to a set threshold in order to determine whether the voxel under consideration is consistent. That is

$$E = max\left\{\sqrt{\tfrac{1}{m_i}\sum\left(p_i - \overline{P}\right)^2}\right\}$$

where m_i is the number of elements in each set p_i and $\overline{P}$ is the average of the combined sets.

This measure was defined based on the following assumptions. If a voxel is consistent, then the average color of the voxel will be close to the individual elements in the sets p_i provided that the surface of the target object is not highly textured. Therefore the error measurement will be low and the voxel is defined to be consistent. If the voxel is not consistent, then one of the sets should return a larger error value. This larger error value prevails and therefore the voxel is more likely to be labeled as inconsistent.

4.5 Reconstruction using Graph Cuts

There is an approach in the literature which claims to be fast and claims to produce a good solution to the reconstruction problem. This approach finds a solution that has the minimum possible photoconsistency error given the available camera views. [8] describes a method to construct a voxel based model and [9] describes a method to generate a stereo depth map, both employing this approach.

This method, like other methods, also requires visibility information and is usually estimated from the visual hull of the object.

This method operates by finding a particular state, of a massive system of interconnected elements, that minimizes the combined errors associated with the interconnections between these elements. A complete description of how to minimize certain energy functions using graph cuts is described in [7].

This massive interconnected system is called a graph, and the means to determine the solution is called a graph cut. The individual elements are called nodes. The graph is designed in such a manner so that the graph cut corresponds to the best possible computer model that matches the available camera views.

In the case of construction using voxels, each voxel becomes an node in the graph.

The graph nodes can be separated into two sets, known as the *source* and the *sink* set. Thus, the state of the voxel, whether part of the model or node, can be determined by whether the node corresponding to that voxel is part of the source or sink set. All the nodes in the source set are linked to a virtual node known as the *source node* and all the nodes in the sink set are linked to a virtual node known as the *sink node*. The photoconsistency error associated with the voxel is usually encoded in some manner in the interconnections between the ordinary nodes and the virtual source and sink nodes.

Therefore, the constraints of the system can be encoded into this framework using values associated with nodes and values associated with the interconnections between the nodes. The solution can be extracted by determining whether a node is part of the source set or part of the sink set. The graph cut is the boundary or border that determines which set a node is part of. This graph cut can be calculated in a number of ways but its basis is the Max Flow-Min Cut theorem.

If the graph can be thought of as a system of pipes connected to each other with water flowing from the source node and draining at the sink node, and that the capacity of the pipes are related to the value of the interconnects between the nodes then the max flow-min cut theorem states that the graph cut corresponding to the minimum error in the system can be calculated by finding the interconnections that are saturated or carries its maximum capacity of flow.

Chapter 5

Obtaining datasets

This chapter provides a brief overview of where and how the datasets used in this thesis were obtained.

The datasets consists of a number of camera images or views of the object to be modeled, hereafter referred to as the target object. The number of viewpoints do vary for each dataset, however the viewpoints were chosen in such a manner as to view the entire of the target object and to ensure that each point on the surface of the target object can be viewed by at least two cameras. The target object was placed on a calibration grid or pattern. This pattern is used to determine the focal length and relative orientation of the camera for that corresponding view.

The lighting was chosen to be as diffuse as possible and the target objects were chosen for the surface optical qualities. See the section on Radiosity for the reasons for these choices.

A minimum of 13 views were obtained. This number was chosen based on the estimated number of cameras needed to cover a shape such as a cube and that each point on the surface of the cube needs to be seen by at least two cameras in order to pinpoint it in the real world. The cube was chosen because it has a large surface area to volume ratio and bad geometry, in that its shape leads to more occlusions than something simpler like a sphere. Therefore, it can approximate objects with a large surface area and geometry that causes a large number of occlusions. However, if the target object has very bad geometry then more views may be required.

The images are calibrated using the Caltech Matlab Calibration Toolbox, which at the time of writing, is available freely from the Caltech website. This toolbox can determine the focal length and relative orientation of the

camera if provided with the corners of the calibration pattern and the origin of the calibration pattern. This is a manual process.

A number of datasets were generated by three were used in the evaluation process, the lion figurine dataset, the animal figurines dataset and the brick fragment dataset.

Once the camera calibration information was obtained, the silhouettes of the objects were created by manually segmenting the images using a image editing tool. These silhouettes are not using the reconstruction process but are used to determine the accuracy of the reconstructions.

Chapter 6

Description of the developed algorithms

This section of the thesis provides an overview of the inner workings and the methods used of the various algorithms that were developed to implement the task of photoconsistency based model reconstruction.

The chapter begins with a description of the setup of the reconstruction process. This involves defining the voxel space and loading the camera parameters needed to perform the reconstruction. The chapter continues to describe the estimation algorithm. This algorithm attempts to estimate the best photoconsistency threshold for a given dataset. This threshold is then used in the reconstruction process.The second algorithm described is the blob renderer. This was developed so that different views of the reconstructed model can be rendered depending on the camera geometry provided. The rendered views allows comparisons to be made between the model reconstruction and the actual camera views in order to determine the accuracy of the reconstruction.The final algorithm discussed in this section creates the computer model from the camera views using photoconsistency.

6.1 Process setup

The reconstruction process needs a mathematically defined space in which to perform the reconstruction. This is achieved in a number of steps. The steps are listed below:

- Load the camera views and corresponding camera parameters.

- Define an origin point of the world that projects as closely to the center of all the camera views.
- Create a spherical voxel grid centered on the on the world origin point.

The first step is to load the camera images and the corresponding camera parameters. This information is stored in computer memory for quick retrieval. Storing the images in computer memory lowers the computational time. However, the amount of computer memory limits the resolution of the images that can be stored at a given time. The higher the resolution, the grater the amount of memory required. The resolution of the images used in the reconstruction process is 640 by 480 pixels. Using this resolution, the amount of memory required to store an image in all three color bands is approximately 1MB.

The voxel grid was chosen to be spherical rather than a cube. A cube grid has the disadvantage of hiding camera views due to its shape. This poses no problem if the reconstruction uses the silhouettes of the object under reconstruction. However, if the algorithm uses photoconsistency, the shape of the cube could cause voxels to be incorrectly classified as some camera views that may be relevant are hidden. A sphere allows the surface of the grid to be seen by as many cameras as possible if the cameras are uniformly arranged around the object.

The camera images are views of an object that lies in the physical world. As stated in the description of the world reference frame in the section on Pinhole Cameras, the origin and orientation of the world reference frame is arbitrary. In the case of camera parameters generated using the calibration grid measure, the world reference frame is set to the camera reference frame of one of the cameras used. Once this is defined, the model will lies somewhere in this world space. The generated voxel grid should completely enclose the space occupied by the model. This is accomplished by setting the radius and the center point of the spherical voxel grid. The radius is set manually and the center point is estimated. The radius of the spherical grid is set in such a manner so as to exclude as much of the background as possible. This was done due to the fact that large amounts of consistent background (background of similar color) causes the estimation algorithm to fail due to the assumptions made by the estimation algorithm and also results in patches of erroneous background colored mass in the final reconstruction. However, if the background is that consistent then it may be possible to automatically segment the views to obtain the silhouettes. If the background is different in each camera view then this is not such a problem.

The center point is set to the approximate center point of the model in the world. This center point is estimated by finding a point that most closely projects to the center of each image in all the available camera views. This estimate is based on the assumption that the cameras are all pointing in the direction of the target object and that the target object is centered in the camera view.

6.2 Estimation algorithm

The purpose of the estimation algorithm is to attempt to automatically find the best photoconsistency threshold. The algorithm begins by assuming no occlusion reasoning. A scan of the voxel grid is made and the results from the photoconsistency measure for each voxel is stored. A histogram is generated from the measures and a Gaussian is fitted to the histogram. The threshold is selected to be the highest threshold that lies 3.5 standard deviations from the center of the fitted Gaussian.

The assumption behind the estimation algorithm is as follows. A voxel is labeled as part of the model if two conditions hold. (1) The voxel is located on the surface of the model and (2) the occlusion reasoning is correct. Keeping in mind that there is no occlusion reasoning in this instance, most of the measures returned from the scan of the voxel grid should be associated with the value of a voxel that is *not* part of the model due to one of the aforementioned reasons. Therefore, the histogram generated will indicate the spread of values that are associated with voxels that are not part of the model. The Gaussian that is fitted to the histogram should model this distribution. A value is considered to be outside the distribution if it is more than three standard deviations from the center of the Gaussian. Hence, the highest threshold chosen outside this distribution should be the upper limit of the values associated with a correct voxel. Any value below this threshold indicates a consistent voxel.

This method fails in the case of images having a consistent background, e.g a single color backdrop. Then the assumption that the voxels return an inconsistent value will fail as it is probable that a large number of voxels will project into this consistent background and thus be labeled as consistent.

To determine whether this assumption holds true, some datasets were evaluated. Two histograms of the returned photoconsistency values under different scenarios were generated. The first scenario involved a pass of the entire voxel grid with no occlusion reasoning. In the second scenario, the visual hull of the target was generated first from silhouettes. A pass was

then made through the voxels contained in the visual hull.

The expected results of the two different scenarios are as follows. In the first scenario, the values returned by the photoconsistency measures should be those corresponding to a state of inconsistency according to the assumption made above. Therefore, the histogram generated should model the distribution of the values associated with an inconsistent voxel. In the second scenario, the visual hull should be a reasonable approximation to the actual target object, therefore the photoconsistency values returned should correspond to a consistent state. The second scenario histogram should model the distribution of phtoconsistency values that are associated with a consistent voxel.

By comparing these two histograms, an estimation to the validity of the assumption can be made. Fig. 6.1 shows the comparison between the two histograms obtained from the Lion figurine dataset. Fig. 6.2 shows the comparison between the two histograms obtained from the animal dataset and Fig. 6.3 shows the comparison between the two histograms obtained from the brick fragment dataset.

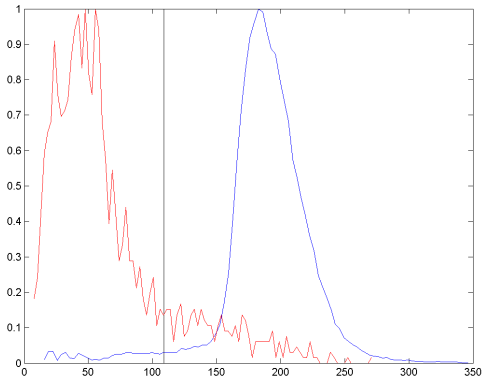


Figure 6.1: Comparison of the histograms obtained from a pass through the entire voxel grid (blue) and through the voxel grid enclosed by the visual hull (red). Both histograms have been normalized to the same scale for ease of comparison. The black line represents the location of the estimated threshold.

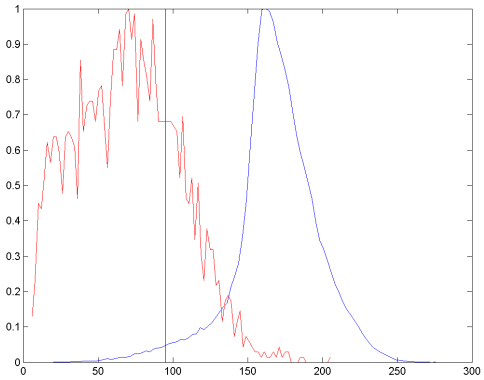


Figure 6.2: Comparison of the histograms obtained from a pass through the entire voxel grid (blue) and through the voxel grid enclosed by the visual hull (red). Both histograms have been normalized to the same scale for ease of comparison. The black line represents the location of the estimated threshold.

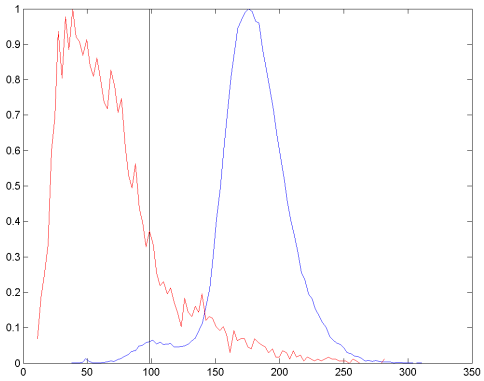


Figure 6.3: Comparison of the histograms obtained from a pass through the entire voxel grid (blue) and through the voxel grid enclosed by the visual hull (red). Both histograms have been normalized to the same scale for ease of comparison. The black line represents the location of the estimated threshold.

6.3 Blob renderer

The blob renderer is an algorithm that generates an image of the reconstructed model from a specific camera viewpoint using the corresponding projection matrix. The renderer is designed to be used with the voxel representation of an object.

The voxel centers, size and color are passed to the renderer as well as the relevant projection matrix for the desired viewpoint. The renderer projects all the voxel vertices into the image plane. These vertices define a polygon in the image plane. For simplicity, this polygon is approximated by a rectangle. To model occlusion, a buffer is kept in memory listing the distance of each voxel center from the location of the camera in the world reference frame. The reasoning behind this occlusion modeling is as follows. If the voxel is closer to the center of the center of the camera then it is approximately closer to the virtual image plane of that camera. Therefore, voxels that project to a similar location in the image plane are ordered by distance from the center of the camera. This allows the surface that can be seen from that viewpoint to be rendered.

6.4 Occlusion mask algorithm

This algorithm determines which voxel can be seen by which camera. This allows occlusion reasoning to be incorporated into the reconstruction of the target object.

The operation of the algorithm is simple. The distance of each voxel, in the world, to each camera center is calculated. The voxels are then projected into the virtual image plane of each camera. This image plane is referred to as the buffer. The region that the voxel projects to in the buffer is approximated by a rectangle and then marked with the distance of the voxel to the camera center. This step of the process is illustrated by Fig. 6.4

If the center of the region in the buffer already has a value, then this value is compared to the calculated distance of the voxel from the center of the camera. If the value in the buffer is less, then the voxel is marked as occluded from the perspective of that camera. If the value in the buffer is more, then the voxel is marked as visible and the buffer is overwritten with the smaller distance.

This procedure is repeated twice. The first pass finds the voxels closest to the camera and the second pass marks all other voxels that are further away

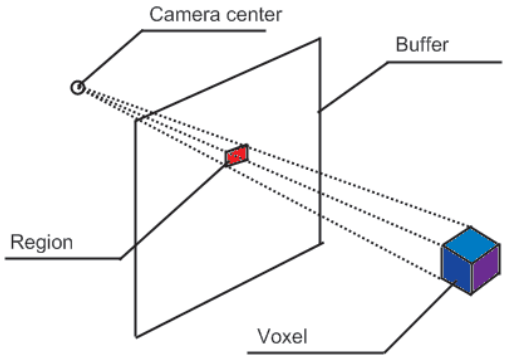


Figure 6.4: Illustration of the process to create the occlusion buffer

along the same optical ray as occluded.

The limitations of this algorithm is that it marks voxels as either occluded or visible. This can lead to holes being formed in the voxel grid where partially occluded voxels are marked as fully occluded and are not considered in the reconstruction process. In an attempt to compensate for this, the voxel size is divided by two so that partially occluded voxels are marked as visible.

6.5 Photoconsistent based Reconstruction

This section describes the algorithm used in the thesis to generate models without using silhouettes as a starting point. This algorithm is an iterative process that determines both the visibility and photoconsistency of voxels in each step.

There is an algorithm describes in [5] which uses a table linking the visibility and consistency state of voxels. If the voxel state is changed from consistent to inconsistent, the table is used to determine which voxels should be reevaluated now that the visibility information has changed. Rather than using a table, the algorithm used in this thesis simply recalculates the visibility of all the voxels at each iteration. This is a simpler, although more computationally expensive process.

This algorithm uses a layer by layer approach to reconstruct the target object. The occlusion reasoning of the outermost layer is found by using the occlusion algorithm. Once this occlusion reasoning is obtained, the vertices

of the voxels in that outermost layer are projected into the relevant camera image planes. That corresponding image projection is approximated by a rectangle. All the pixels colors for each color band, RED, GREEN and BLUE that are contained in that rectangle are stored as a set.

Once this process is complete, each voxel will have 3 sets of pixel colors for each relevant camera. These pixel colors are processed by the photo-consistency measure and a result returned. This result determines whether the voxel is consistent or not. If the voxel is consistent then it is retained otherwise it is discarded from the voxel grid.

Once this process is completed, the next layer in the spherical voxel grid may be exposed, therefore the occlusion recalculated.

This process is repeated, layer by layer, until there are no more inconsistent voxels. The final set of voxels should define the photohull of the object.

Chapter 7

Reconstruction results

This chapter presents some reconstructions of different objects to demonstrate the effectiveness of the algorithm. The chapter begins with a description of the error measure used to evaluate the accuracy of the reconstruction. The three sections, after the accuracy measure description, present the results from three different datasets. The final part of the section illustrates the results obtained when the assumptions behind the estimation algorithm are violated.

7.1 Reconstruction accuracy measure

The accuracy measure is based on Normalized Cross Correlation. This measure determines the similarity of two images. The measure is simply statistical cross correlation of two vectors, but normalized by the magnitude of the values present in the vectors so as to establish an accuracy index between -1 and 1.

The definition of normalized cross correlation is

$$E_{ncc} = \frac{\sum \left(i_1\left(x\right) - \bar{i_1}\right).\left(i_2\left(x\right) - \bar{i_1}\right)}{\sqrt{\sum \left(i_1\left(x\right) - \bar{i_1}\right)^2 . \sum \left(i_2\left(x\right) - \bar{i_2}\right)^2}} \tag{7.1}$$

where E_{ncc} is the NCC value between -1 and 1, i_1 and i_2 are the vectors of the pixel colors obtained from the two images under comparison.

The pixel value vectors i_1 and i_2 are obtained by extracting pixel values

from a camera image that was not used in the reconstruction process and a rendering of the reconstruction model, using the blob renderer, as seen from the corresponding camera viewpoint using the silhouettes as a mask to eliminate background.

The silhouettes were used purely to remove parts of the reference camera image that are not part of the object itself. This is to prevent large errors due to the background present in the reference image but not part of the reconstructed model. This has the side effect of removing pixels from the rendered view that may be due to erroneous voxels in the reconstruction. Therefore, any extra pixels outside the silhouettes in the rendered view that do not have a value of zero are compared to the reference image to determine if it is in fact consistent with the scene even though it is not part of the object under construction.

For example, objects placed on a calibration grid will result in both the object and the calibration grid to be reconstructed as the calibration grid would be consistent in all the views. However, the silhouettes would normally be constructed so as to isolate the actual object. Therefore, the reconstruction algorithm should not be penalized for reconstructing the calibration grid even though it is not part of the desired model.

7.2 Photohull reconstruction without silhouettes

7.2.1 Reconstruction of a toy lion figurine

This object is a simple plastic model of a lion. The dataset was obtained using the calibration grid method. Each view will contain the calibrated grid therefore the reconstructions will also contain the grid. This dataset consists of 13 camera views of the object with the cameras distributed in such a manner as to have photographed as much of the object surface as possible. Different voxel grid resolutions were used in each reconstruction.

The threshold was selected using the estimation algorithm.

A single view was left out of the reconstruction dataset for accuracy measurement purposes.

Camera views

Fig 7.1 illustrates some of the camera views used to generate the computer model of the object.

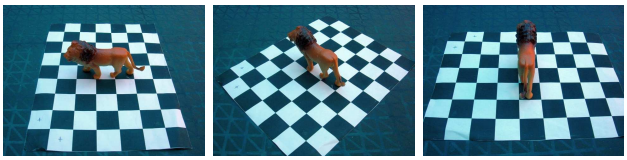


Figure 7.1: Selected camera views of the lion figurine.

Views of reconstructed model

Fig 7.2 illustrates some of the views taken of the one of the reconstructed models. The resolution of this model is 121x121x121 voxels. The camera view set aside for measurement purposes was view 1. The remaining 12 views were used in the reconstruction process.

Figure 7.2: Selected views of the reconstructed lion.

Reconstruction accuracy results

Table 7.1 lists the normalized cross correlation percentages for each color band of the comparison views. The percentage will indicate the match between the reference image and the rendered image. The closer the match, the more positive the percentage and the more likely the reconstructed model is to be accurate.

Table 7.1: Red band NCC values for different resolution reconstructions.

View / Resolution	1	2	3	4	5	6	Mean
60	70.88	72.98	82.35	80.25	76.81	72.21	75.91
50	68.34	71.86	80.33	79.16	77.34	72.10	74.85
40	67.53	70.35	79.08	80.03	74.34	72.72	74.01
30	72.94	73.09	79.90	80.31	75.84	78.19	76.71
20	82.57	83.99	88.09	84.05	84.77	82.15	84.27
10	62.87	54.39	78.96	69.68	69.84	61.06	66.13

Table 7.2: Green band NCC values for different resolution reconstructions.

View / Resolution	1	2	3	4	5	6	Mean
60	73.73	76.62	84.86	82.37	80.08	76.52	79.03
50	71.64	75.07	83.04	81.32	80.28	76.13	77.91
40	70.31	74.27	82.18	81.93	77.40	76.20	77.05
30	75.23	76.98	83.04	82.22	78.66	81.40	79.59
20	84.42	86.67	91.91	86.67	87.97	85.24	87.15
10	59.86	43.15	76.79	70.99	70.32	67.45	64.76

Table 7.3: Blue band NCC values for different resolution reconstructions.

View / Resolution	1	2	3	4	5	6	Mean
60	75.95	78.08	85.29	83.12	81.70	77.24	80.23
50	74.17	76.45	83.55	82.08	81.77	77.00	79.17
40	72.72	76.01	82.81	82.25	79.19	77.17	78.36
30	77.09	78.39	83.60	82.72	80.30	81.67	80.63
20	85.73	86.79	92.17	86.81	89.18	85.15	87.64
10	65.28	50.72	75.83	72.94	72.81	70.82	68.07

Accuracy vs Photoconsistency threshold

Fig. 7.3 illustrates how the accuracy of the reconstructed model changes verses the photoconsistency threshold. The resolution of the voxel grid used in the reconstruction process is 61x61x61.

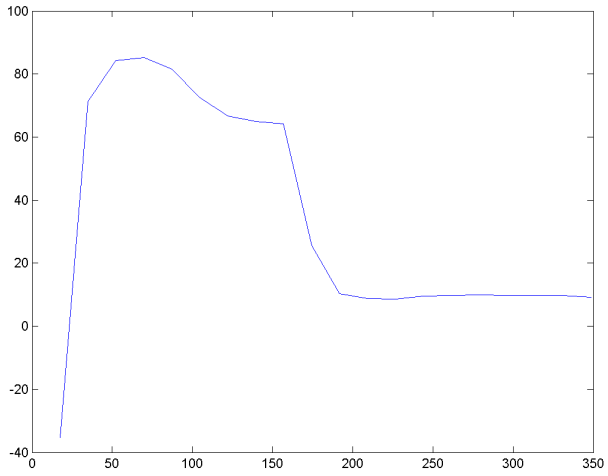


Figure 7.3: Plot of accuracy vs photoconsistency threshold.

7.2.2 Reconstruction of a figurine set

This dataset contains a number of toy models. This dataset was also obtained using the calibration grid method. This dataset contains 13 views of the figurine set.

Camera views

Fig. 7.4 illustrates some of the camera views used to generate the computer model of the object.

Figure 7.4: Camera views of the figurine set.

Views of reconstructed model

Fig. 7.5 below shows two views of the reconstructed figurine set.

Figure 7.5: Selected views of the reconstructed figurine set

Reconstruction accuracy results

Table 7.4: Red band NCC values for different resolution reconstructions.

View / Resolution	1	2	3	4	5	6	Mean
60	57.59	52.40	60.64	60.28	60.49	50.69	57.02
50	58.76	55.00	64.82	62.84	62.85	57.69	60.33
40	60.58	57.90	67.04	66.07	65.63	60.63	62.97
30	56.51	59.18	68.30	67.33	61.61	66.67	63.27
20	67.62	65.13	65.95	66.56	65.14	67.62	66.34
10	56.85	57.94	49.50	52.20	46.09	47.68	51.71

Table 7.5: Green band NCC values for different resolution reconstructions.

View / Resolution	1	2	3	4	5	6	Mean
60	56.23	51.66	60.02	58.88	56.72	49.36	55.48
50	56.51	54.15	63.11	61.03	59.26	56.70	58.46
40	57.62	56.12	64.86	64.08	61.22	58.27	60.36
30	52.76	57.47	64.75	64.71	56.22	63.22	59.85
20	63.99	61.57	61.77	61.34	59.47	65.31	62.24
10	49.78	52.22	47.24	45.66	37.91	39.59	45.40

Table 7.6: Blue band NCC values for different resolution reconstructions.

View / Resolution	1	2	3	4	5	6	Mean
60	56.82	54.00	60.41	60.27	56.41	52.73	56.77
50	57.29	56.33	62.32	61.58	59.22	59.55	59.38
40	58.26	58.49	63.81	64.71	60.66	59.38	60.88
30	54.04	60.16	62.05	64.74	56.34	63.65	60.16
20	64.35	61.50	61.15	61.75	59.96	66.49	62.53
10	53.57	50.85	53.69	47.54	52.47	50.04	51.36

Accuracy vs Photoconsistency threshold

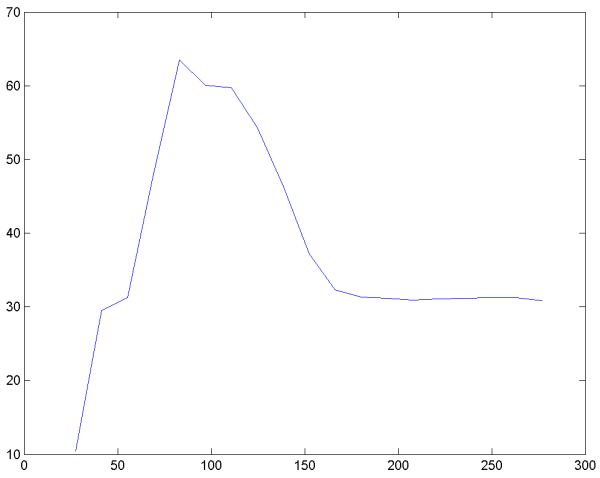


Figure 7.6: Plot of accuracy vs photoconsistency threshold.

7.2.3 Reconstruction of a brick fragment

Camera views

Fig. 7.2.3 below presents three of the camera views used in the reconstruction.

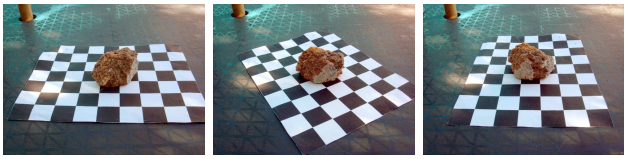


Figure 7.7: Selected camera views of a brick fragment.

Views of reconstructed model

Following the same format as before, Fig 7.2.3 below presents some selected views of the reconstructed brick fragment.

Figure 7.8: Selected rendered views of the brick fragment model.

Reconstruction accuracy results

Table 7.7: Red band NCC values for different resolution reconstructions.

View / Resolution	1	2	3	4	5	6	Mean
60	85.15	71.56	51.93	66.79	60.26	61.09	66.13
50	85.34	74.67	51.06	66.87	61.12	61.47	66.76
40	85.02	81.16	54.67	66.16	66.30	63.88	69.53
30	84.89	85.99	77.52	78.66	70.69	75.70	78.91
20	80.21	82.50	74.40	80.49	70.05	74.35	77.00
10	-6.04	4.52	16.57	18.20	5.32	10.06	8.11

Table 7.8: Green band NCC values for different resolution reconstructions.

View / Resolution	1	2	3	4	5	6	Mean
60	87.89	76.00	57.15	70.84	65.27	66.69	70.64
50	88.06	78.89	56.01	70.32	65.72	67.37	71.06
40	88.28	84.68	59.87	69.39	70.33	70.05	73.77
30	90.25	89.24	82.10	81.98	77.72	82.55	83.97
20	88.36	86.93	83.35	85.17	78.33	82.06	84.03
10	-7.00	10.95	39.08	30.04	-1.10	5.64	12.94

Table 7.9: Blue band NCC values for different resolution reconstructions.

View / Resolution	1	2	3	4	5	6	Mean
60	90.04	79.93	63.33	75.18	70.32	72.27	75.18
50	90.30	82.65	62.59	74.64	70.85	73.25	75.71
40	90.70	87.83	66.34	74.03	75.37	75.69	78.33
30	93.14	91.92	86.11	85.67	83.13	87.16	87.86
20	92.44	90.43	88.96	89.14	84.39	86.94	88.72
10	-5.51	22.53	55.98	38.90	-6.00	3.04	18.16

Accuracy vs Photoconsistency threshold

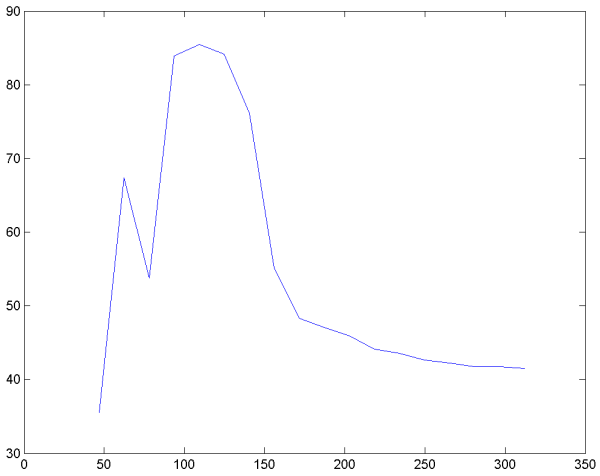


Figure 7.9: Plot of accuracy vs photoconsistency threshold.

7.2.4 Reconstruction of a figurine with large amounts of consistent background

The following reconstruction was performed using a dataset from Web sources. The dataset consisted of 21 views of a movie action figure. Some of the camera views of the object are shown in Fig. 7.10

Notice that the background is essentially uniform. This highly consistent background results in the estimation histogram containing a large percentage of consistent values. The estimation histogram and the histogram obtained

Figure 7.10: Some camera views of the action figure

from the silhouette, shown in Fig. 7.11 illustrates this large amount of overlap.

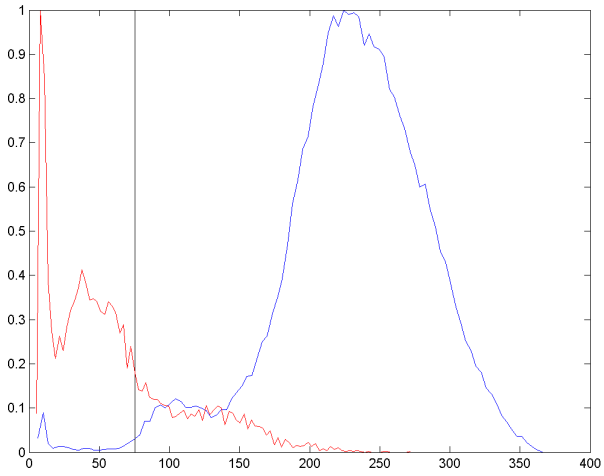


Figure 7.11: Graph illustrating the overlap between the histogram of the photoconsistency measures obtained from the estimation pass through the voxel grid (blue) and the histogram obtained from the a pass through the visual hull (red). The black line indicates the inaccurate estimated threshold.
Graph illustrating the overlap between the histogram of the photoconsistency measures obtained from the estimation pass through the voxel grid (blue) and the histogram obtained from the a pass through the visual hull (red). The black line indicates the inaccurate estimated threshold.

The overlap of the two histograms results in an inaccurate Gaussian fit to the inconsistent photoconsistency value peak and therefore a poor model of the situation. Therefore, the estimated threshold is far from the best value. The result of this is an inaccurate and incomplete model reconstruction. This is illustrated in Fig. 7.12

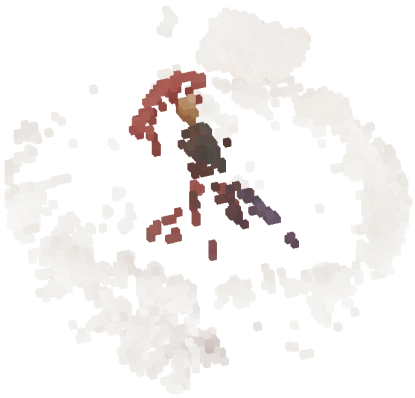


Figure 7.12: Inaccurate reconstruction of action figurine due to the presence of large amounts of consistent background.

In an attempt to compensate for the large amount of photoconsistent values, the following procedure was used. Although both consistent and inconsistent photoconsistency values are present, it may be likely that there will be more inconsistent values than consistent values. Therefore, the histograms are modified by subtracting the RMS constant of the histogram and removing all resulting negative terms. This has the effect of suppressing smaller peaks and curves that correspond to the consistent set of values. Then the Gaussian curve is fitted to the modified histogram.

The result of this procedure is presented in Fig. 7.13. The estimation histogram is displayed without the modifications. Notice that the estimated threshold is now closer to the edge of the consistent set of values.

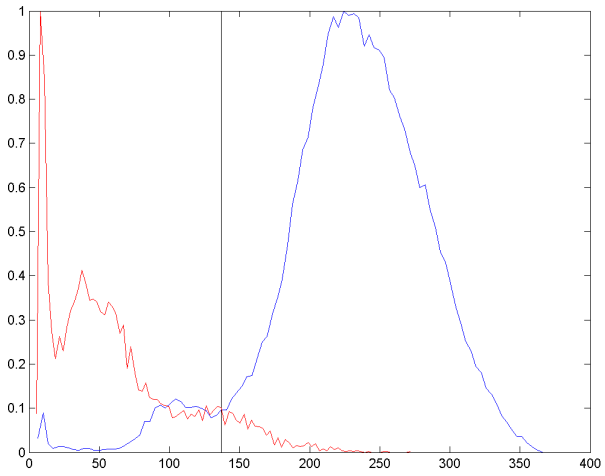


Figure 7.13: Graph comparing the histograms obtained using the estimation algorithm (blue) and the visual hull (red). The black line indicates the estimated threshold.

A view of the resulting reconstruction is shown in Fig. 7.14

Figure 7.14: A view of the compensated reconstruction of the action figurine

Chapter 8

Conclusions

To reiterate, the objectives of this thesis is to perform as review of the literature pertaining to model reconstruction using photoconsistency based methods. To develop an algorithm that can perform the reconstruction and to evaluate its performance.

Some background has been provided on the necessary mathematics and concepts required to perform a reconstruction. The literature indicates that there are better methods such as graph cut based reconstruction that are most likely more efficient in creating accurate models that the iterative procedure described in this thesis.

An algorithm was developed based on the use of voxels to represent the model. The algorithm computes the photoconsistency and the visibility of voxels in an iterative cycle in the hope of a convergent solution.

The results performed on the datasets indicate that the algorithm does perform agreeably although more work is needed to eliminate problems due to consistent backgrounds.

Bibliography

[1] Steven. M. Seitz, Charles. R. Dyer, 1997, Photorealistic Scene Reconstruction by Voxel Coloring, *In Proc. Computer Vision and Pattern Recognition Conf.*, pp. 1067-1073

[2] David. A. Forsyth, Jean Ponce, Cameras, *Computer Vision*, Prentice Hall, ISBN 0-13-085198-1

[3] David. A. Forsyth, Jean Ponce, Radiometry, *Computer Vision*, Prentice Hall, ISBN 0-13-085198-1

[4] Oğuz Özün, Ulaş Yılmaz, Volkan Ataly, *Comparison of Photoconsistency Measures used in Voxel Coloring*

[5] W. Bruce. Culbertson, Thomas Malzbender, 1999, *Generalized Voxel Coloring*

[6] Mark. R. Stevens, Bruce Culbertson, Tom Malzbender, *A Histogram-based Color Consistency Test for Voxel Coloring*

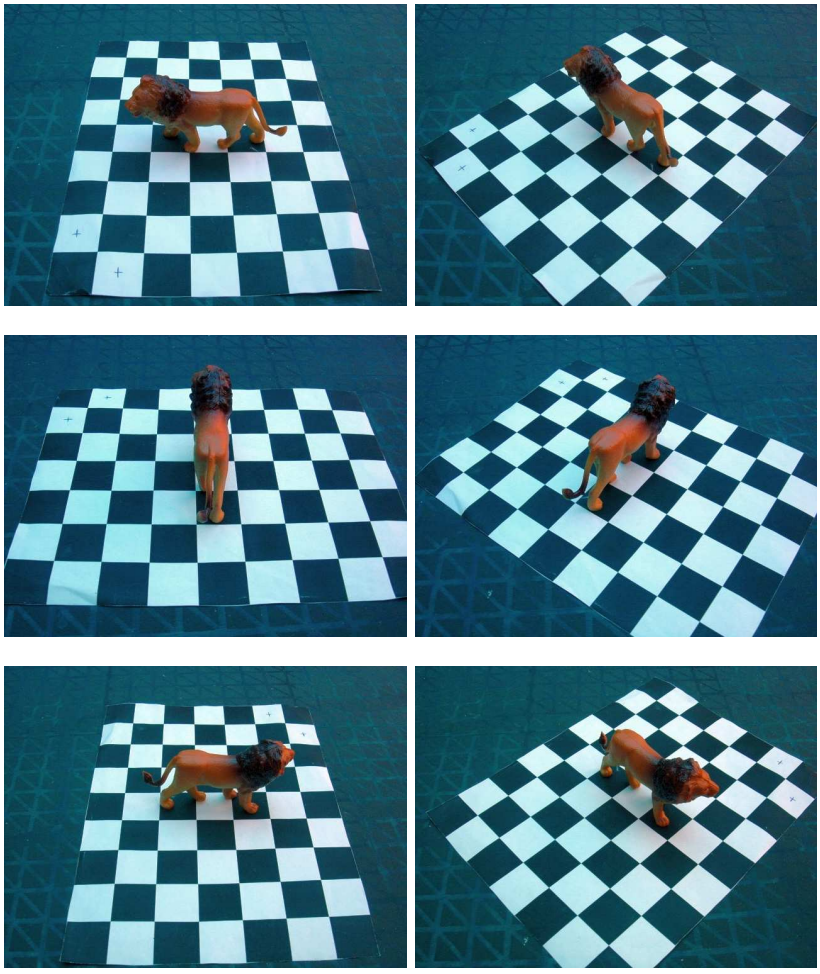
[7] Vladimir Kolmogorov, Ramin Zabih, 2004, What Energy Functions Can Be Minimized via Graph Cuts, *IEEE transactions on pattern analysis and machine intelligence, Vol 26, no 2, February 2004*

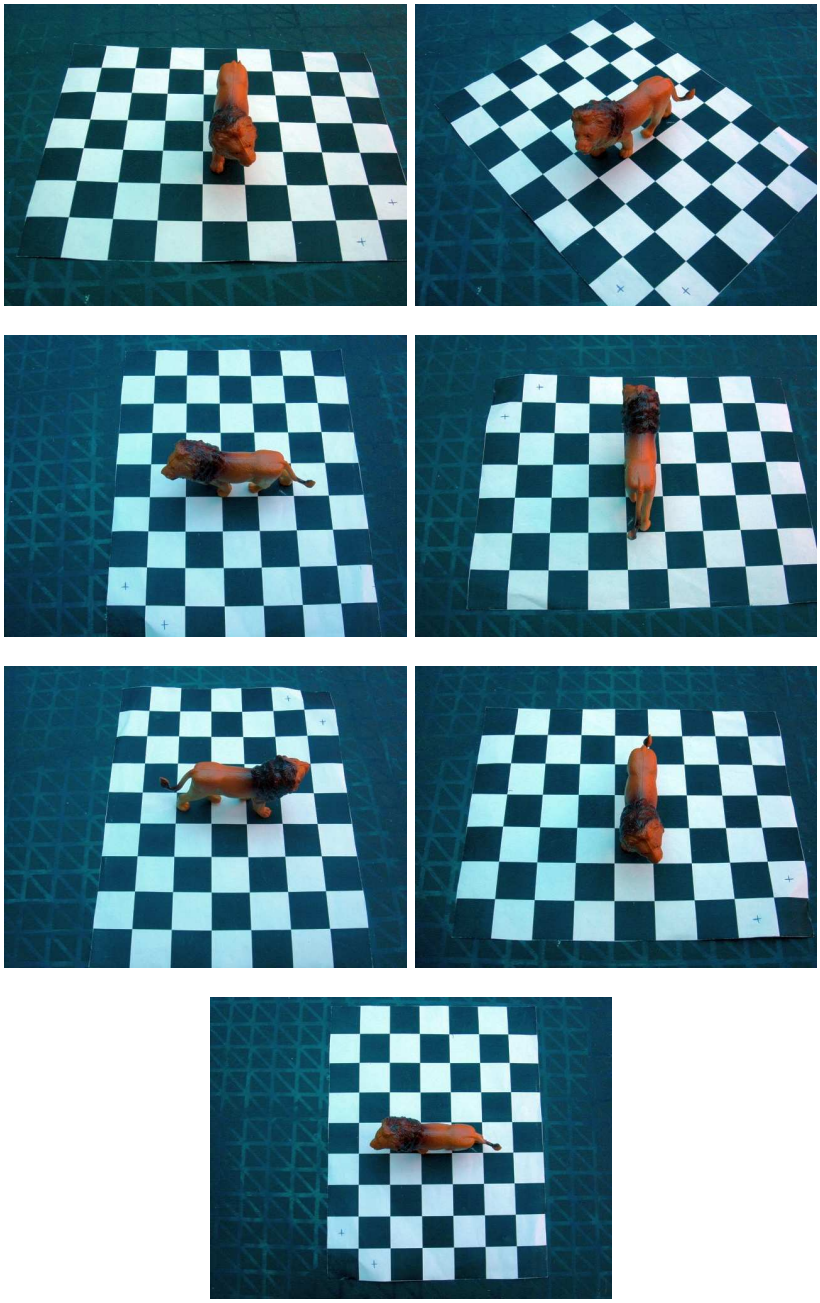
[8] G. Vogiatzis, P. H. S. Torr, R. *Cipolla, Multi-view Stereo via Volumetric Graph Cuts*

[9] Vladimir Kolmogorov, Ramin Zabih, *Multi-camera Scence Reconstruction via Graph Cuts*

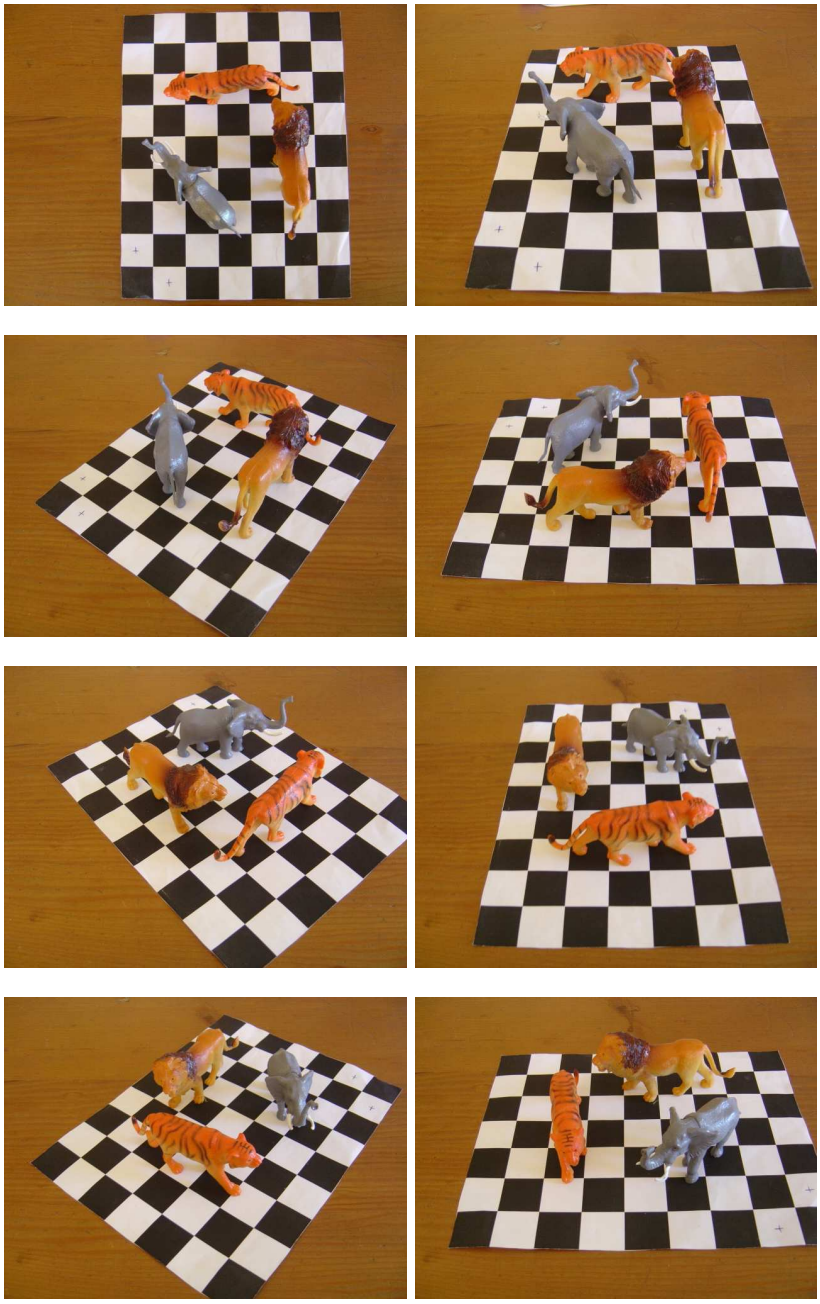
Appendix A - Dataset images

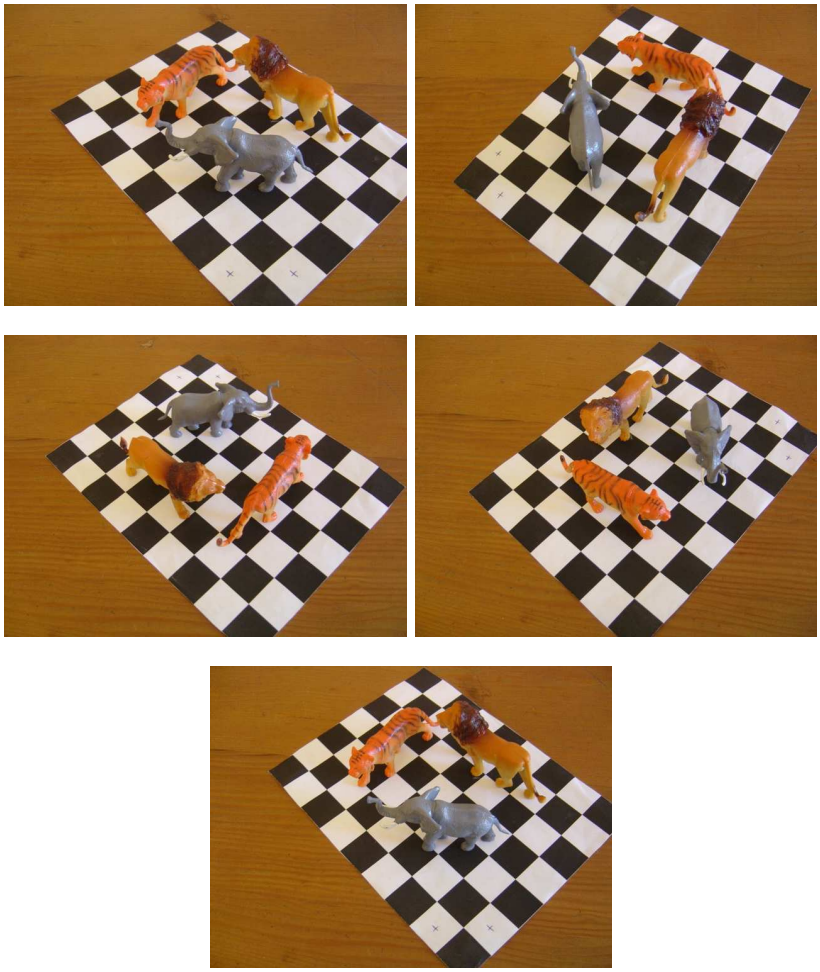
Lion dataset





Figurine dataset





Brick fragment dataset