3D model reconstruction using photoconsistency

Kirk Michael Joubert

Submitted to the Faculty of Engineering and the Built Environment, University of Cape Town in fulfilment of the requirements for the degree of Master of Science in Engineering

August 2007

Declaration

I know the meaning of plagiarism and declare that all work in the document, save for that which is properly acknowledged, is my own.

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

Signed

Signed by candidate

KM Joubert

31 August 2007

Acknowledgments

I would like to thank:

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

Prof. Gehard de Jager, Co-Supervisor, for his efforts in obtaining funding for this work.

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

Abstract

Model reconstruction using photoconsistency refers to a method that creates a *photohull*, an approximate computer model, using multiple calibrated camera views of an object. The term photoconsistency refers to the concept that is used to calculate the photohull from the camera views. A computer model surface is considered photoconsistent if the appearance of that surface agrees with the appearance of the surface of the real world object from all camera viewpoints.

This thesis presents the work done in implementing some concepts and approaches described in the literature. A photoconsistent voxel based method was used to generate the photohull. An algorithm based on this method calculates the geometry of the visual hull by removing inconsistent voxels from a initial spherical volume until the resultant appearance of the volume is consistent with all the camera views. A photoconsistency cost function is used to determine the consistency of a voxel. This cost function is based on the colours of the pixels of the camera views that correspond to the portion of the surface that a particular voxel is representing. The primary cost function used in this thesis is the maximum RMS error between the colour of the voxel, determined by the mean of all the pixel colours from all camera views that can see the voxel, and the pixel colours obtained from each camera view. A threshold is used to determine whether the photoconsistency error of a voxel indicates if it is consistent or inconsistent. An estimation algorithm is used to determine an approximation to the threshold that would correspond to the best model reconstruction results.

The accuracy of the constructed photohull is determined by comparing a rendering of the obtained photohull and a camera image that was not used in the reconstruction process. A silhouette is used to remove background from the images.

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 are 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: This term refers to a state where the appearance of the surface of a computer generated geometrical shape is consistent with the camera views of the corresponding surface of the real world object.

Projection Matrix: The projection matrix is a mathematical construct, 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 originating from the centre 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. The origin of the world reference frame is arbitrary. In this work, the units are in mm and 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 **P** refers to the projection matrix. Lower-case bold letters refer to vectors. For example, the letter **t** refers to the translation vector of a camera.

t: Translation vector of a camera indicating its position in the world reference frame.

R: Rotation matrix indicating the orientation of a camera (or the camera reference frame) in the the world.

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 to a single incident light ray.

ρ: The BRDF of a surface.

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

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

Chapter 1

Introduction

The creation of a computer model of a real world object hereafter referred to as the target object, can be accomplished using various computational techniques.

Some techniques use active lighting to determine the geometry of the target object. An example is the laser line scanner. This procedure involves the projection of a laser line or stripe onto the surface of the target object. The appearance of the laser line on the surface is dependent on the geometry of the surface. A camera obtains an image of the surface with the reflected laser line and using this image, the distortion of the laser line can be calculated. The surface geometry of the target object is determined by combining the degree of distortion of the laser line and the calibration information of the camera, the information that relates pixels in the camera image to regions in the real world. The laser line scans the surface to obtain a "point cloud", or a set of coordinates that correspond to the points on the surface of the target object. This "point cloud" can then be use to construct a computer model representation of the target object.

There are passive methods of creating a model of an object. An example is silhouette based model reconstruction. This method obtains multiple calibrated views of an object and segments these images to produce silhouettes. A geometrical shape called the visual hull is calculated so as to be consistent with all the silhouettes. If such a geometrical shape can be found, then the shape is an approximate model of the target object.

What is meant by 3D model reconstruction using photoconsistency? In the context of this thesis, it refers to the creation of a computer model, or photohull, of a real world object using multiple camera views of that object. The reconstruction process is based on the reasoning of photoconsistency. Photoconsistency refers to a state where the *appearance* of the surface of a geometrical shape is consistent with the camera views of the target object. When such a state

is achieved, the resulting photohull should be an approximate model of the target object.

Seitz and Dyer describe a method in their paper [1] which uses the surface colour 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 an approximate model that is consistent with the available views. It is this method that forms the basis of the work done in this thesis.

1.1 Objectives of the thesis

The objectives of this thesis are threefold.

- A review of some of the literature available concerning the field is presented in order to obtain a starting point for the development of the research.
- To develop and implement a computer algorithm based on a method from the literature that can construct a computer model.
- 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 contains 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 presented to accomplish such a reconstruction. Chapter 4 discusses the concept of model reconstruction using photoconsistency. Chapter 5 discusses 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 relates points in a world reference frame to corresponding points in an image reference frame. Most of the information presented about intrinsic parameters comes from [2]. Normal cameras use lenses not pinholes, but a pinhole camera can approximate within limitations, the characteristics of a camera that uses a lens.

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.

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 principal axes, X and Y. The origin of this reference frame is located in the top left hand corner of the camera image plane.

The camera reference frame is simply a right-handed three dimensional Cartesian space. The pinhole is located on the origin of the space and the 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 set-up of a pinhole camera is illustrated in Figure 2.1.

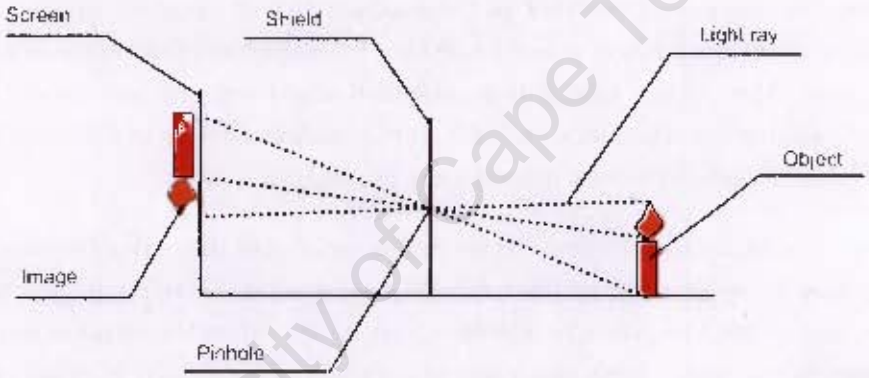


Figure 2.1: Diagram illustrating the concept of the pinhole camera and its operation.

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. The operation of the pinhole camera can be defined mathematically. The screen and pinhole are defined in the camera reference plane. Figure 2.2 illustrates this concept.

The optical axis shown in Figure 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 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:

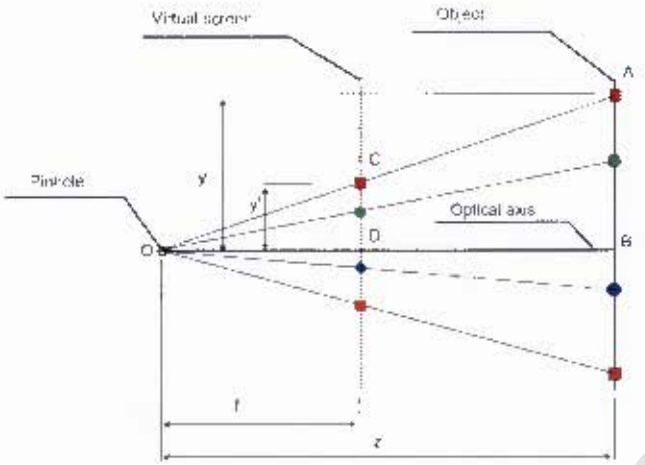


Figure 2.2: Diagram illustrating the concept of a virtual image obtained by placing a virtual screen in front of the pinhole.

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

where 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 same reasoning applies to a three dimensional pinhole camera even though the diagram Figure 2.2 is only a two dimensional representation. If the coordinates of a point in the camera image plane are (x, y) and the coordinates of the point in the camera reference frame are (X', Y', Z') and the camera is aligned so that it looks down the z axis then the equations linking the two are as follows:

$$x = \frac{f}{Z'}X' \quad \text{and} \quad y = \frac{f}{Z'}Y' \tag{2.2}$$

The above equations assume that the origin $(0, 0)$ of the image plane is centred along the optical axis of the camera but this may not always be the case. For instance, the origin of the digital images stored in computer format is 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 centreline intersects the image plane is called the principal point and has coordinates (P_x, P_y).

The equations are modified to become:

$$x = \frac{f}{z}X' + P_x \quad \text{and} \quad y = \frac{f}{z}Y' + P_y \tag{2.3}$$

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*. The equations apply only to cameras that have square pixels without skew. 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 millimetres 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 parameters: 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 corresponding to a vector $\mathbf{X}$ in the world reference frame, is simply:

$$\mathbf{X}' = \mathbf{R}\mathbf{X} \tag{2.4}$$

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

$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{t} \tag{2.5}$$

The inverse mapping is:

$$\mathbf{X} = \mathbf{R}^{-1}\mathbf{X}' - \mathbf{R}^{-1}\mathbf{t} \tag{2.6}$$

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

$$\mathbf{X} = \mathbf{R}^T\mathbf{X}' - \mathbf{R}^T\mathbf{t} \tag{2.7}$$

2.3 The projection matrix

The sections discussing intrinsic and extrinsic parameters define 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 (2.7) can be rewritten in matrix form as:

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.8}$$

The intrinsic equations (2.3) are non-linear but they can be converted into matrix form:

$$Z'.x = f.X' + Z'.P_x \quad \text{and} \quad Z'.y = f.Y' + Z'.P_x \tag{2.9}$$

$$\begin{bmatrix} tx \\ ty \\ t \end{bmatrix} = \begin{bmatrix} f & 0 & P_x \\ 0 & f & P_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \tag{2.10}$$

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

$$\begin{bmatrix} tx \\ ty \\ t \end{bmatrix} = \begin{bmatrix} f & 0 & P_x \\ 0 & f & P_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.11}$$

$$\begin{bmatrix} tx \\ ty \\ t \end{bmatrix} = \mathbf{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.12}$$

The matrix $\mathbf{P}$ is called the *projection matrix*.

This matrix equation with associated scaling 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 is, 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 behaviour 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 of an object 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 point light source. Figure 3.1 depicts a view of an object and the corresponding silhouette.

Figure 3.1: Image of object and corresponding silhouette.

These silhouettes are obtained by segmenting the image into two regions, foreground and

background. The foreground represents the projection of the object while the background represents the regions that are not inside the object boundary. The methods available to automatically segment the image into suitable foreground and background regions is beyond the scope of this work. The images in the datasets used in this work (those that were not obtained from external sources) were segmented manually with an image editor.

3.2 The visual hull

The visual hull is the polyhedral shape that is consistent with the projections of the silhouettes of the object [3].

The diagram Figure 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 are consistent with the image silhouette.

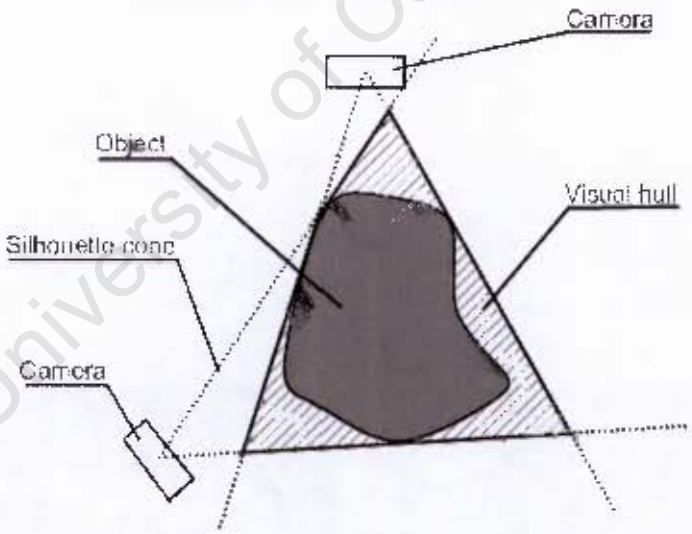


Figure 3.2: Concept of the visual hull.

As Figure 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.

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.

3.3 Reconstruction using voxels

A voxel 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. Dyer [4] describes a number of approaches that can be used to create a volumetric model of an object.

A simple method is used in this work to create the visual hulls of objects. First a voxel grid, a cube built up of a number of voxels, is defined. The voxel grid is situated in the world so that the entire volume of the object to be constructed is contained. Each voxel is then projected into the image planes of the cameras. In this case, only the center of a 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. 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 [5].

4.1 Photoconsistency and application to voxels

Photoconsistency describes a state whereby an object's surface colour 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 volume of the model is composed of voxels. The photoconsistency of a voxel is based on the photoconsistency of the set of pixels that are enclosed by the projections of that voxel into the camera image planes. The definition of whether this set of pixels is considered to be photoconsistent depends on the photoconsistency cost function that is used. In this thesis, the colour of the pixels is used to define the similarity of the set.

The motivation behind this method is that if the pixels are completely dissimilar then the voxel cannot lie on the surface of the model and must be removed from the overall model con-

struction. The converse is not necessarily true. There can be situations, such as a uniformly coloured surface or similar surface texture, where a voxel that maps to similar pixels could still be incorrect. A similarity measure that takes the orientation of the pixels into account will provide additional information that can decrease the chances of a voxel being incorrectly labelled as consistent.

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 [6] and [7].

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* cross-section of the travelling rays of light. In other words, it is the rate of flow of energy through a cross-sectional

area of light. The unit of flux density is the 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.

Radiant Intensity

Radiant intensity defines the flux contained in a single ray of light. The ray is modelled as an extremely small cone that extends from the source outwards. The unit 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 modelled as a extremely small cone that extends from the source outwards. The unit 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 have 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 modelled as rays or straight lines with an origin at the point on the surface and a direction expressed in spherical coordinates. The diagram Figure 4.1 illustrates this concept.

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 and it gives no indication of the density or number of rays present, while 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.

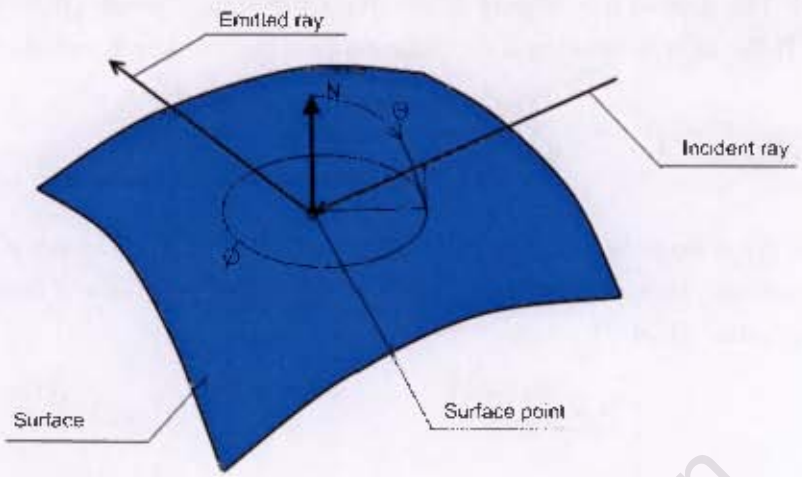


Figure 4.1: Diagram illustrating the concept of rays entering and leaving a point on an arbitrary surface. The direction of the incoming and outgoing rays are defined in spherical coordinates, ϕ and θ, where ϕ is the azimuth and θ is the elevation of the ray. N is the surface normal at the point. The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

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 than it actually is (Figure 4.2).

Since a smaller cross-section of light rays are needed to irradiate the surface patch when the incident angle changes, the amount of energy flowing to 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

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

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

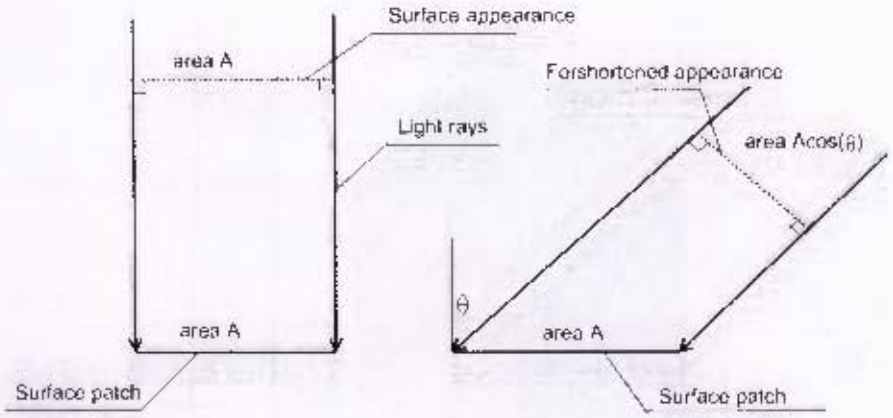


Figure 4.2: The change in appearance of a surface as the viewing angle is changed. This change has the effect of reducing the cross-section of the light rays needed to irradiate the surface patch.

a strong peak in that direction. A "matte" surface reflects light more or less equally in all directions, so the BRDF will be a smooth, uniform function.

The term *Lambertian* refers to a surface that has a constant BRDF. In other words, incoming rays are scattered in such a manner that they are emitted in a uniform hemisphere from the surface. Matte surfaces are reasonably approximated by the theoretical Lambertian surface.

If the lighting is diffuse so there is no strong component of light from any particular direction then the emitted radiance from a Lambertian surface will be equal for all viewing angles. 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.

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

As Figure 4.3 demonstrates, visibility information is needed to reject camera views that cannot

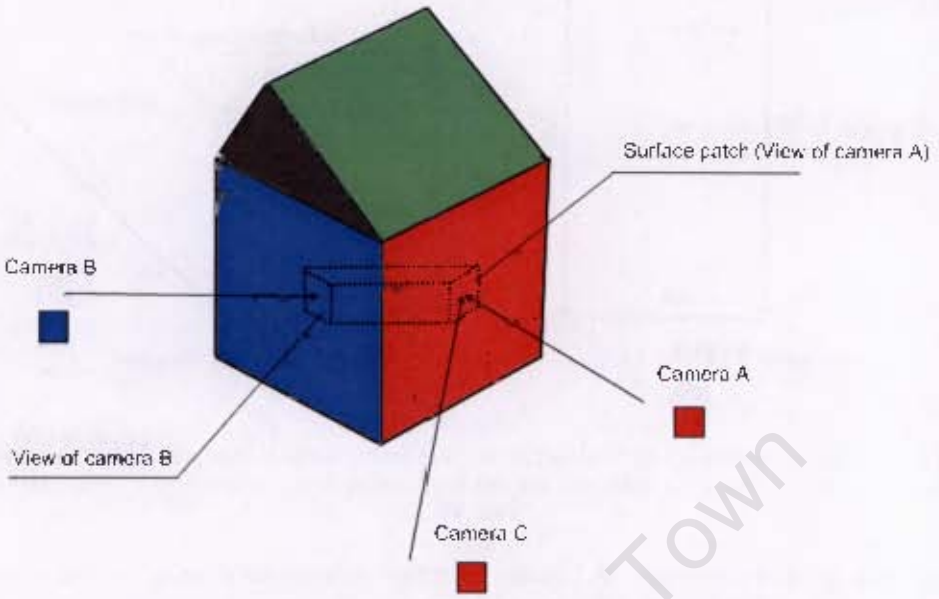


Figure 4.3: Diagram illustrating the consequences of not having visibility information. The surface region designated "surface patch" lies on the surface of the object and should be photoconsistent. However, due to the lack of visibility information, the surface region is labelled as inconsistent due to the disagreement between camera B and the other two cameras A and C.

see the surface of the object and therefore provides invalid information.

As stated, the visibility information is vital to the accurate reconstruction of the target object. The only information available to the reconstruction algorithm 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: 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. The drawback to using the visual hull is that the silhouettes are required and the visibility information degrades the further the actual surface is from the visual hull. The algorithms presented in this work use an iterative procedure to calculate visibility.

4.4 Photoconsistency measures

This section lists a number of photoconsistency measures, some of which are mentioned in [5].

If a surface region of the target object is projected into a camera view, the set of intensity values of the pixels that it encloses is defined for the purposes of this text to be p_i, where i is the number of the corresponding camera.

Assume that the set N is the set of all cameras used in the reconstruction and the subset V contains all the cameras that have direct, unobstructed line of sight of the same region of surface.

Assume that a set P is defined to be the union of all the sets p_i where i is an element of V. In other words, P is the set of all the pixel intensities from all the cameras that have an unobstructed view of the same surface region of the target object.

$$P = \bigcup p_i \quad \forall i \in V$$

4.4.1 Standard Deviation

Given that $P(k)$ represents the k^{th} element of P, the standard deviation measure σ is calculated by

$$\sigma = \sqrt{\frac{1}{\mid P \mid} \sum_{k=1}^{|P|} (P(k) - E\{P\})^2} \tag{4.3}$$

$$E\{P\} = \frac{1}{\mid P \mid} \sum_{k=1}^{|P|} P(k) \tag{4.4}$$

where $\mid P \mid$ is the number of elements in the set.

This measure assumes that if the spread of the colours in the set is large, then the voxel cannot be consistent and the standard deviation will be high. 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 suffers a drawback. If the surface of the object is highly textured and the surface patch is large (as it might be when using the voxel representation), then the spread of the colours in the set will likely be large, and therefore the measure will be high.

A voxel is labelled 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 thresholding is that regions 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 adaptive measure τ is then given by

$$\tau = \sigma - \alpha \frac{1}{|\ V\ |} \sum_{i=1}^{|V|} \sigma_i \tag{4.5}$$

$$\sigma_i = \sqrt{\frac{1}{|\ p_i\ |} \sum_{k=1}^{|p_i|} (p_i(k) - E\{p_i\})^2} \quad \forall i \in V \tag{4.6}$$

$$E\{p_i\} = \frac{1}{|\ p_i\ |} \sum_{k=1}^{|p_i|} p_i(k) \quad \forall i \in V \tag{4.7}$$

where σ is the standard deviation as calculated by (4.3), $p_i(k)$ is the k^{th} element of the set p_i and α is a weighting factor.

As with the standard deviation measure, the voxel is considered consistent if the value of τ is below a certain threshold.

The disadvantage of this measure is that it requires a predetermined threshold and weighting factor.

4.4.3 Histogram

The histogram method calculates consistency based on the intersection of colour histograms. For each set p_i a colour 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 which determines whether the pairs of histograms are a match or not.

For example, [8] 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 non-empty.

4.4.4 Root mean squared

The RMS measure was developed to address an apparent shortcoming of the standard deviation measure. Photoconsistency reasoning assumes that a surface region is not consistent if any one of the set of cameras that can see that surface does not agree with any other camera in that set.

The standard deviation measure operates by determining the spread of the pixel colours obtained from the set of cameras V. Experiments indicate that the measure may be affected by the number of cameras used in the set.

In the experimental procedure, two variables X and Y corresponding to two normal distributions around two different means are defined. The two variables represent the pixel intensities obtained from various cameras. A surface is defined to be consistent if the pixel intensities are similar. This scenario is simulated by forming a set of values composed of purely variable X or variable Y. The scenario where the surface is inconsistent is simulated by forming a set composed of a mixture of variables X and Y.

Assume a scenario where a single set of pixel intensities extracted from the set of camera images indicate that the surface is inconsistent. As the number of camera images with similar pixel intensities is added, the overall standard deviation *decreases*. This corresponds to a reduced photoconsistency error. This is illustrated in Figure 4.4.

This experiment indicates that if there is a large proportion of similar pixel intensities to non-similar pixel intensities then the error is reduced. This is unacceptable as this effect can cause a surface region to be incorrectly classified.

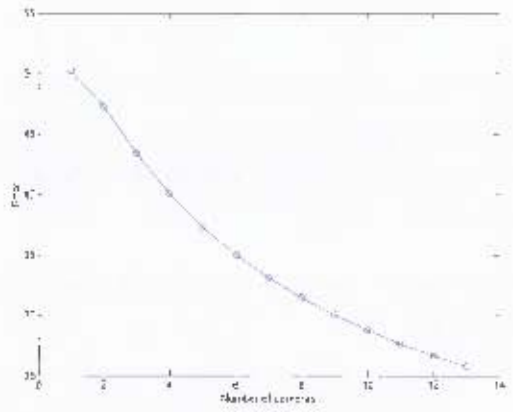


Figure 4.4: Results of the experiment showing the decrease of the photoconsistency error, calculated using standard deviation, as the number of cameras used by the measure is increased.

The RMSE error measure was developed as an alternative. This particular measure calculates the maximum root mean squared error of the elements in each set $p_i, \quad i \in V$ from the assumed colour of the voxel undergoing the test. The colour of the voxel is assumed to be the average of all the colour 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:

$$E_{rms} = \max \left\{ \sqrt{\frac{1}{\mid p_i \mid} \sum_{k=1}^{\mid p_i \mid} \left(p_i(k) - E\{P\} \right)^2} \right\} \quad \forall i \in V \tag{4.8}$$

$$E\{P\} = \frac{1}{\mid P \mid} \sum_{k=1}^{\mid P \mid} P(k) \tag{4.9}$$

where $\mid p_i \mid$ is the number of elements in each set p_i and $p_i(k)$ is the k^{th} element of the set p_i.

This measure was defined based on the following assumptions. If a voxel is consistent, then the average colour 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 declared 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 the voxel is more likely to be labelled as inconsistent.

Similar experiments as conducted on the standard deviation measure, shows that the error

increases as the number of cameras is increased. This is illustrated in Figure 4.5. An increase in error is not optimal as the measure is still dependent on the number of cameras, however, it is preferable to having the error decrease.

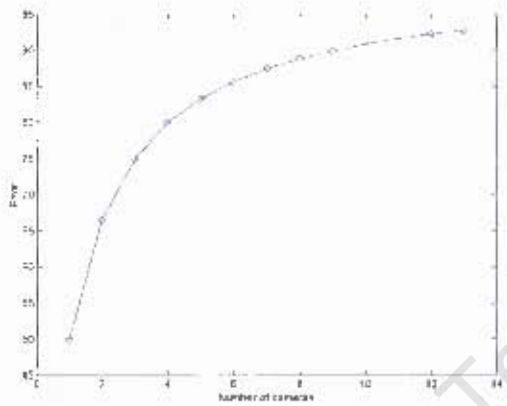


Figure 4.5: Results of the experiment showing the increase of the photoconsistency error, calculated using the RMSE photoconsistency measure, as the number of cameras used by the measure is increased.

There is a possible reason as to why the error increases as the number of cameras increases. The error is calculated relative to the mean of all the pixel intensities of all the cameras in set V. As the number of similar pixel intensities is added, the mean converges to the average of these similar pixel intensities. This results in an increased discrepancy between the mean and the dissimilar pixel intensities resulting in a large photoconsistency error.

An improvement on the measure is to calculate the expected colour of the surface, not from the mean of all the pixel intensities, but from the midpoint between the lowest and highest mean of each of the sets $p_i \in V$. This should have the effect of fixing the mean regards of the number of cameras present and therefore ensuring that the error is independent of the number of cameras used. Therefore, the revised RMSE equations are:

$$E_{rms} = \max\left\{\sqrt{\frac{1}{|p_i|}\sum_{k=1}^{p_i}(p_i(k) - f(P))^2}\right\} \quad \forall i \in V \tag{4.10}$$

$$f(P) = \frac{1}{2}\left[\max\left\{\frac{1}{|p_i|}\sum_{k=1}^{|p_i|}p_i(k)\right\} + \min\left\{\frac{1}{|p_i|}\sum_{k=1}^{|p_i|}p_i(k)\right\}\right] \tag{4.11}$$

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 by using a technique known as graph cuts. This approach finds a solution that has the minimum possible photoconsistency error given the available camera views. Two papers, using this approach, describe methods to create a voxel-based model [9] and a stereo depth map [10].

The graph cut technique operates by finding a particular state, in a massive system of interconnected elements, referred to as a graph, which minimizes the combined errors associated with the interconnections between these elements. Methods of how to construct a graph from certain energy functions are described in [11].

The technique can be applied to voxel based reconstructions by assigning an element or node of the graph to each voxel. The photoconsistency errors associated with the voxels are encoded into the interconnections between the nodes.

The technique operates by dividing the components of the graph 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 node known as the *source node* and all the nodes in the sink set are linked to a node known as the *sink node*.

Therefore, the constraints of the system can be encoded into this framework using values associated with the interconnections between the nodes and 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 the basis of finding the cut is the max-flow/min-cut theorem of set theory.

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. The capacities of the pipes are related to the value of the connections between the nodes. 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 pipes that are saturated, or which carry their maximum capacity of flow.

Chapter 5

Obtaining datasets

This chapter provides an overview of 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 modelled, hereafter referred to as the target object. The number of viewpoints do vary for each dataset, but the viewpoints were chosen in such a manner as to view as much of the of the target object as possible 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. A minimum of 13 camera views were obtained.

The lighting was chosen to be as diffuse as possible and the target objects were chosen for the surface optical qualities. Objects with highly specular surfaces such as glazed pottery and transparent objects such as glass were not used. Some of the objects are made of plastic and the surfaces tend to be non-Lambertian. Therefore, harsh or spot lighting was avoided so as to minimize the specular highlights that would result from such lighting.

The images are calibrated using the Caltech Matlab Calibration Toolbox, which at the time of writing, is available freely from the Caltech website [14]. 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 but only three were used in the evaluation process. The toy lion dataset, the toy animals dataset and the brick fragment dataset. Figure 5.1 shows one image from each of the datasets. The rest of the images are presented in Appendix A.

Once the camera calibration information was obtained, the silhouettes of the objects were

Figure 5.1: A single image from the toy lion dataset, the toy animals dataset and the brick fragment dataset.

created by manually segmenting the images using a image editing tool. These silhouettes are not used in the reconstruction process but are used to evaluate the accuracy of the reconstructions.

Chapter 6

Description of the developed algorithms

This chapter provides an overview of the various algorithms that were developed to implement the task of photoconsistency based model reconstruction.

The first section of this chapter describes the set-up of the reconstruction process. This involves defining the voxel space and loading the camera parameters needed to perform the reconstruction. The second section describes the estimation algorithm. The estimation algorithm attempts to estimate the best photoconsistency threshold for a given dataset which 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 according to the camera geometry provided. The rendered views allow 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 set-up

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

- Load the camera views and corresponding camera parameters.
- Define an origin point of the world that projects as closely as possible to the center of

all the camera image planes.

- Create a voxel grid centred 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 but limits the resolution of the images that can be stored. The higher the resolution, the greater 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 colour bands is approximately 1MB.

The initial shape of the computer model was chosen to be a sphere. A cube shaped model 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, a cubic shape could cause voxels to be incorrectly classified as some camera views that should see the same region of surface are occluded. A spherical shape allows the surface of the voxel 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 chapter 2, 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 arbitrarily set to the camera reference frame of one of the cameras. Once this is defined, the model will lie 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 voxel grid. The radius is set manually and the center point is estimated. The radius of the initial sphere is selected to exclude as much of the background as possible. This was done because large amounts of consistent background causes the estimation algorithm to fail due to certain assumptions breaking down and also results in incorrectly labelled voxels in the the final reconstruction. If the background is consistent then it may be possible to automatically segment the views to obtain the silhouettes.

The center point of the voxel grid is set to the estimated 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 centred in the camera view.

6.2 Estimation algorithm

The purpose of the estimation algorithm is to attempt to automatically find the best photo-consistency threshold. The algorithm operates using the voxel representation of the object surface. The algorithm begins by scanning the voxel grid, using no visibility restrictions, and determines the photoconsistency error for each voxel. These photoconsistency errors are stored as a vector. The mean and standard deviation of the photoconsistency errors are calculated and the photoconsistency threshold is set to three standard deviations below the mean.

The assumption behind the estimation algorithm is as follows. A voxel is labelled as part of the model if two conditions hold:

- The voxel is located on the surface of the model.
- The occlusion reasoning is correct.

Since visibility was not taken into account in this instance, most of the measures returned from the scan of the voxel grid will correspond to a value of a voxel that is *not* part of the model due to one of the aforementioned reasons. Therefore, the photoconsistency errors obtained should represent the spread of values that are associated with voxels that are not part of the model. Histograms of the photoconsistency errors indicate that the spread of values is Gaussian in nature. The mean and standard deviation are used to model this spread of values. A value is considered to be outside the distribution if it is more than three standard deviations from the center of a 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, for example, a single colour backdrop. Then the assumption that the voxels return an inconsistent value fails as it is probable that a large number of voxels will project into this consistent background and thus be labelled as consistent.

Some datasets were evaluated to determine whether the assumption that the scan of the voxel grid, without taking visibility into account, returns values corresponding to the inconsistent state holds true. Two histograms of the returned photoconsistency values under different scenarios were generated. The first scenario involves a pass through the entire voxel grid with no occlusion reasoning. The second scenario involves a pass only through the voxels contained in the visual hull.

The expected results of the two different scenarios are as follows. In the first case, the values returned by the photoconsistency measures should correspond to 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 case, the visual hull should be a reasonable approximation to the actual target object and the photoconsistency values returned should correspond to a consistent state. The resulting histogram should model the distribution of photoconsistency values that are associated with a consistent voxel.

By comparing these two histograms, an estimate of the validity can be made of the assumption that the pass through the voxel grid without taking visibility into account, returns values that correspond to inconsistency. Figure 6.1 shows the comparison between the two histograms obtained from the toy lion dataset, the toy animals dataset and the brick fragment dataset.

6.3 Blob renderer

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

The voxel centres, sizes and colours are passed to the renderer, along with the projection matrix of 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, the 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 modelling is as follows:

If the voxel is closer to the center of the camera then it is 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, resulting in a rendered image where portions of the model that are occluded are not rendered.

6.4 Occlusion mask algorithm

The occlusion mask algorithm determines which voxel can be seen by which camera. It 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 to each camera center is

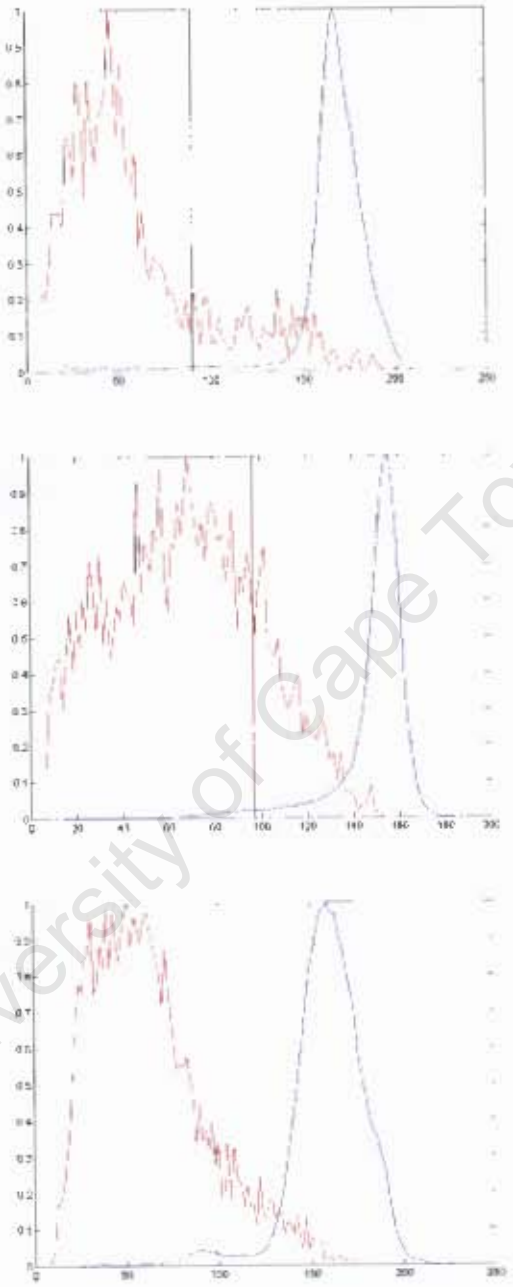


Figure 6.1: Comparison of the histograms obtained from a pass through the entire voxel grid (blue) and through the voxel grid enclosed by the visual hull (red) for three different datasets. Both histograms in each case have been normalized to the same scale for ease of comparison. The black line represents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

calculated in world coordinates. 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 in Figure 6.2.

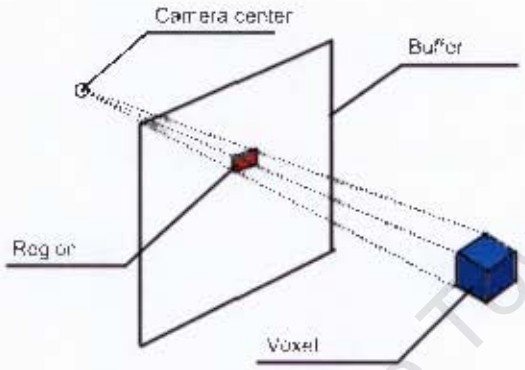


Figure 6.2: Illustration of the process to create the occlusion buffer.

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 smaller, then the voxel is marked as occluded for that camera. If the value in the buffer is greater, then the voxel is marked as visible and the buffer is overwritten with the smaller distance.

Two passes are made through the voxel grid. The first pass finds the voxels closest to the camera and the second pass labels all other voxels that are further away along the same optical ray as occluded.

The limitations of this algorithm are 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, voxel size is halved in the calculations. This forms small gaps in between the voxel grid allowing voxels one layer deep to be classified as visible. Figure 6.3 illustrates the effect of halving the voxel size on the visibility labelling.

This procedure does have the unfortunate side effect of labelling some occluded voxels as visible. However, this side effect should not penetrate deeper than two voxel layers.

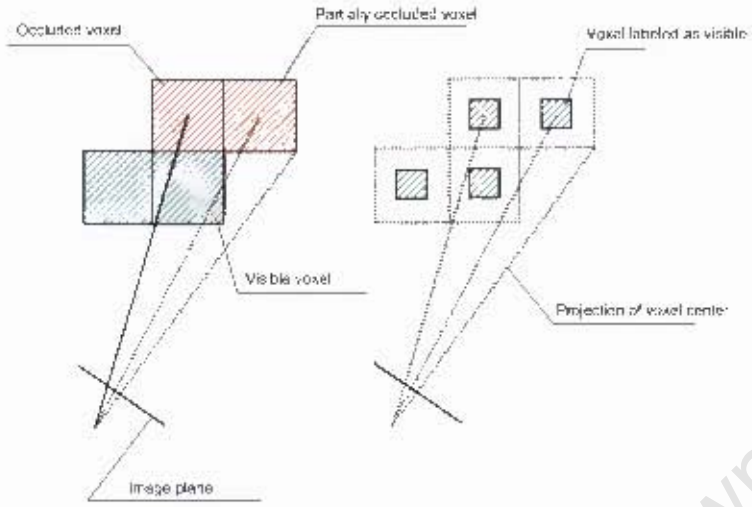


Figure 6.3: Diagram illustrating the effect, on the visibility labelling of voxels, of halving the voxel size.

6.5 Photoconsistency-based reconstruction

This section describes the algorithm used in the thesis to generate models without using silhouettes as a starting point.

To calculate visibility, 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 based on the assumption that it will converge to an acceptable solution whereby both the conditions of visibility and photoconsistency are satisfied.

There is an algorithm described in [13] that 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 re-evaluated with the changed visibility. Rather than using a table, the algorithm used in this thesis simply recalculates the visibility of all the voxels in each iteration. This is a simpler but more computationally expensive process.

This algorithm uses a layer-by-layer approach to reconstruct the target object. The occlusion estimate of the outermost layer of the current model estimate of the object is found using the occlusion algorithm. Once this occlusion estimate is obtained, the vertices of the voxels in the outermost layer are projected into the camera image planes, taking visibility into account. The corresponding image projection is approximated by a rectangle. All the pixels colours

that are contained in the rectangle are stored in a set for each colour band: red, green and blue.

Once this process is complete, each voxel will have 3 sets of pixel colours for each camera. These pixel colours are processed by the photoconsistency measure and a result returned. The 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, some of the voxels may have been removed from the model, resulting in some of the voxels in the next layer to be exposed. Therefore, the occlusion estimate is recalculated to take into account the changed visibility of the newly exposed voxels.

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

Results

This chapter presents the results of various experiments conducted to evaluate the performance of the algorithms described in chapter 6. The first section of this chapter describes the experimental procedure used to generate the results. The second section presents the results obtained from the experiments and the final section demonstrates the effect that a large amount of consistent background has on the estimation algorithm.

7.1 Experimental procedure

Four experiments were performed on each of three datasets, to evaluate the performance of the algorithms. Each dataset contains a number of camera views of an object and the camera calibration information.

First, a model of the object is created using a 121x121x121 voxel grid. In a "leave-one-out" procedure, a single camera view is excluded from the dataset, hereafter referred to as the reference image, so that an evaluation of the accuracy of the generated model can be obtained. By not using this camera view in the dataset, a better estimation of the accuracy of the reconstruction can be obtained as the information provided by that camera view is not used in the reconstruction process. The estimation algorithm is used to calculate the photoconsistency threshold.

The size of the voxel grid was set manually to the smallest value that still ensured that the voxel grid encompasses the entire volume that the computer model of the object occupies. The smallest value was chosen to maximize the resolution of the computer model and not to include too much of the background.

Once the computer model is generated, an image of the reconstruction is rendered, from the same camera viewpoint as the reference image, using the blob renderer algorithm. The reference image and the rendered image are compared by using an accuracy measure.

The accuracy measure is based on root mean squared error (RMSE). This measure determines the error between two images by calculating the mean squared error of the difference between corresponding pixel intensities in the two images. If the images are exact, then the error will be zero. The measure is defined as

$$E_{rms} = \sqrt{\frac{1}{|I_1|}\sum_{x=0}^{|I_1|}(I_1(x) - I_2(x))^2} \tag{7.1}$$

The pixel value vectors I_1 and I_2 are obtained by extracting pixel values from the reference image and the rendered image respectively.

The silhouettes were used purely to remove parts of the reference camera image that are not part of the object itself. This prevents large errors caused by the background being present in the reference image but not being part of the reconstructed model. Using the silhouettes 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 target object. The value of zero was chosen because the value of the blob renderer canvas is initially set to zero. Any pixels that do not have a value of zero forms part of the projection of the computer model.

For example, objects placed on a calibration grid will cause both the object and the calibration grid to be reconstructed, since the calibration grid is 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.

An error map is generated indicating the distribution of the pixel errors and can be used to evaluate the accuracy of the computer model. The silhouettes are not used to remove the background so that a better visual estimate of the reconstruction can be made.

The second experiment evaluates the operation of the estimation algorithm by generating a number of computer models using different voxel grid resolutions. The resolutions span from a 21x21x21 voxel grid to a 121x121x121 voxel grid. Six computer models were created for each resolution, with a different reference image left out of the dataset in each case. The same

RMS error measure was used in each case to evaluate the accuracy of the reconstruction. The results of these experiments are tabulated in the reconstruction results section for each colour band.

The third experiment calculates the RMS error of a reconstruction, using the second variant of the RMSE photoconsistency measure, as the photoconsistency threshold is changed. The results of this experiment demonstrate what the best photoconsistency threshold actually is. The accuracy of the reconstruction using this threshold can be compared to the accuracy results of the second experiment, where the threshold was selected using the estimation algorithm, in order to determine how effective the estimation algorithm is.

The fourth experiment demonstrates the difference in performance between the standard deviation photoconsistency measure and both variants of the RMS photoconsistency measure. The voxel grid resolution used in this experiment is 61x61x61. The threshold is stepped from a high to a low value. At each step, the accuracy of the resulting computer model reconstruction is evaluated. The results are presented in the next section.

A seperate experiment demonstrates the detrimental effect that a large amount of consistent background has on the estimation algorithm and also provides a possible solution to lessen this effect.

7.2 Object reconstruction results

7.2.1 Reconstruction of a toy lion.

This subsection describes the results obtained from the experiments done using the toy lion dataset.

Camera views

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

Views of reconstructed model

Figure 7.2 illustrates some of the views taken of one of the reconstructed models. The resolution of this model is 121x121x121 voxels. The camera view set aside for measurement

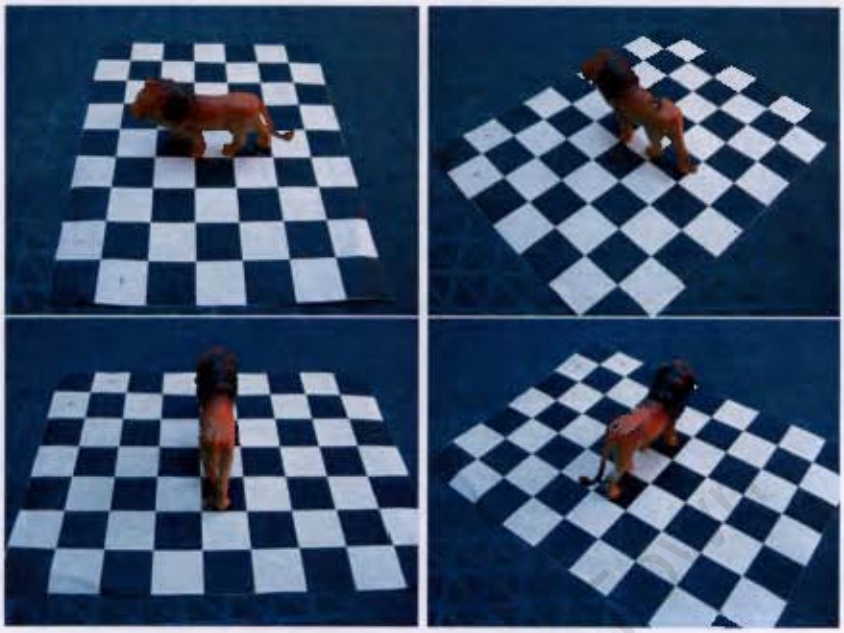


Figure 7.1: Selected camera views of the toy lion.

purposes was view 1. The remaining 12 views were used in the reconstruction process.

Figure 7.2: Selected views of the reconstructed lion.

Comparison images

Figure 7.3 shows the two images used to determine the accuracy of the reconstruction.

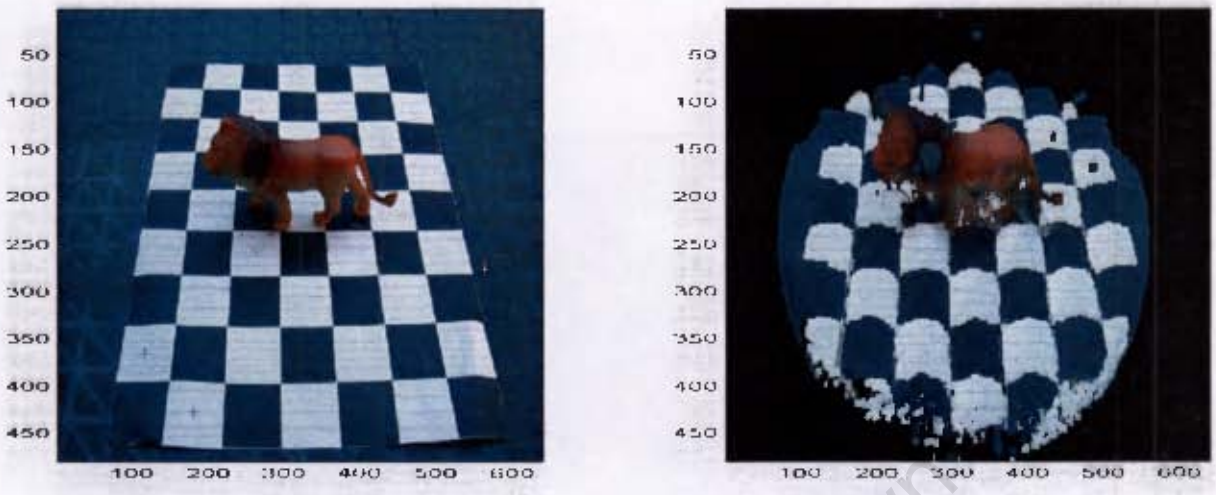


Figure 7.3: Images used in the comparison process. The image on the left is the actual camera view of the toy lion while the image on the right is the rendered view of the reconstruction from the same camera viewpoint.

Error map

Figure 7.4 is the map of the RMS error between the pixels in the comparison image and the rendered model image. Dark areas indicate low error while light areas indicate high error. The silhouettes are not used to remove the background.

Performance of the estimation algorithm

Tables 7.1, 7.2, and 7.3 list the RMSE values for each colour band obtained by comparing the reconstructed model with the view that was kept aside. These are the results of the second experiment using the toy lion dataset.

Table 7.1: Red colour band RMSE values for different resolution reconstructions using the toy lion dataset.

View / Resolution	1	2	3	4	5	6	Mean
21x21x21	21.76	23.60	17.06	21.95	19.60	24.83	21.47
41x41x41	17.91	17.37	14.92	17.74	16.79	19.32	17.34
61x61x61	17.91	17.37	14.92	17.74	16.79	19.32	17.34
81x81x81	25.62	24.82	21.76	21.87	23.70	24.66	23.74
101x101x101	26.47	25.09	21.48	21.50	23.28	25.32	23.86
121x121x121	25.10	23.72	19.83	21.28	22.09	25.21	22.87

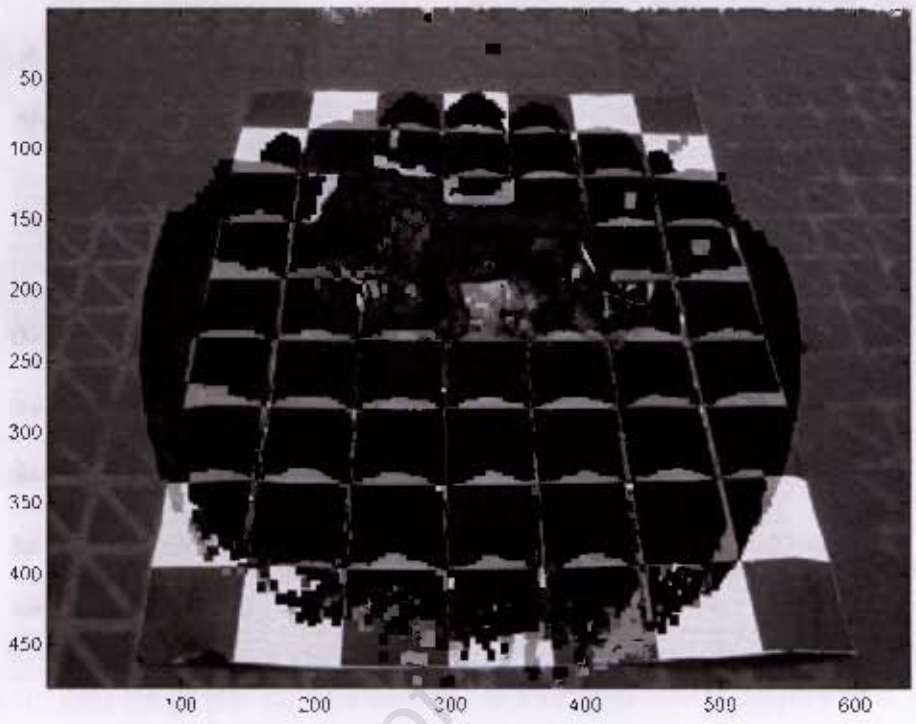


Figure 7.4: Red colour band RMS error map of the toy lion reconstruction.

Table 7.2: Green colour band RMSE values for different resolution reconstructions using the toy lion dataset.

Resolution \ View	1	2	3	4	5	6	Mean
21x21x21	19.89	22.68	14.84	20.34	18.17	20.84	19.46
41x41x41	16.86	15.36	12.03	16.02	14.30	16.45	15.17
61x61x61	18.72	20.74	14.83	18.15	16.69	17.58	17.78
81x81x81	23.98	22.70	20.12	20.63	21.56	21.78	21.79
101x101x101	24.83	23.12	19.64	20.16	21.23	22.32	21.88
121x121x121	23.49	21.85	18.09	19.91	19.99	22.01	20.89

Table 7.3: Blue colour band RMSE values for different resolution reconstructions using the toy lion dataset.

Resolution \ View	1	2	3	4	5	6	Mean
21x21x21	19.90	22.87	15.05	20.23	18.63	20.76	19.57
41x41x41	16.85	15.28	11.62	15.88	13.81	16.57	15.00
61x61x61	18.44	19.94	14.14	17.59	16.22	17.44	17.30
81x81x81	23.24	21.71	18.98	19.86	20.73	21.17	20.95
101x101x101	23.98	22.02	18.47	19.35	20.42	21.76	21.00
121x121x121	22.69	20.87	17.11	19.04	19.21	21.39	20.05

Table 7.4: Computational time required by the reconstruction process using the toy lion dataset (in seconds).

Resolution \ View	1	2	3	4	5	6	Mean
21x21x21	30.34	29.86	29.58	29.53	30.06	29.56	30s
41x41x41	74.42	73.16	73.53	72.66	74.70	72.14	1m 13s
61x61x61	173.06	165.88	170.03	168.64	172.66	165.38	2m 49s
81x81x81	411.66	408.74	416.56	424.63	401.53	398.45	6m 50s
101x101x101	988.34	961.88	964.98	992.88	1007.10	1147.30	16m 51s
121x121x121	1922.00	2110.20	1984.00	1906.90	1808.20	2105.60	32m 53s

Accuracy verses Photoconsistency threshold

Figure 7.5 illustrates how the accuracy of the reconstructed model changes as the photoconsistency threshold is changed. The resolution of the voxel grid used in the reconstruction process is 61x61x61.

Comparison of different photoconsistency measures

Figure 7.6 shows the difference in the performance of three different photoconsistency measures, the standard deviation measure and both variants of the RMSE photoconsistency error. The thresholds have been normalized for better representation. The resolution of the reconstruction used to obtain these graphs is 61x61x61. The accuracy results are obtained from the RMS error of the comparison images in the red colour band.

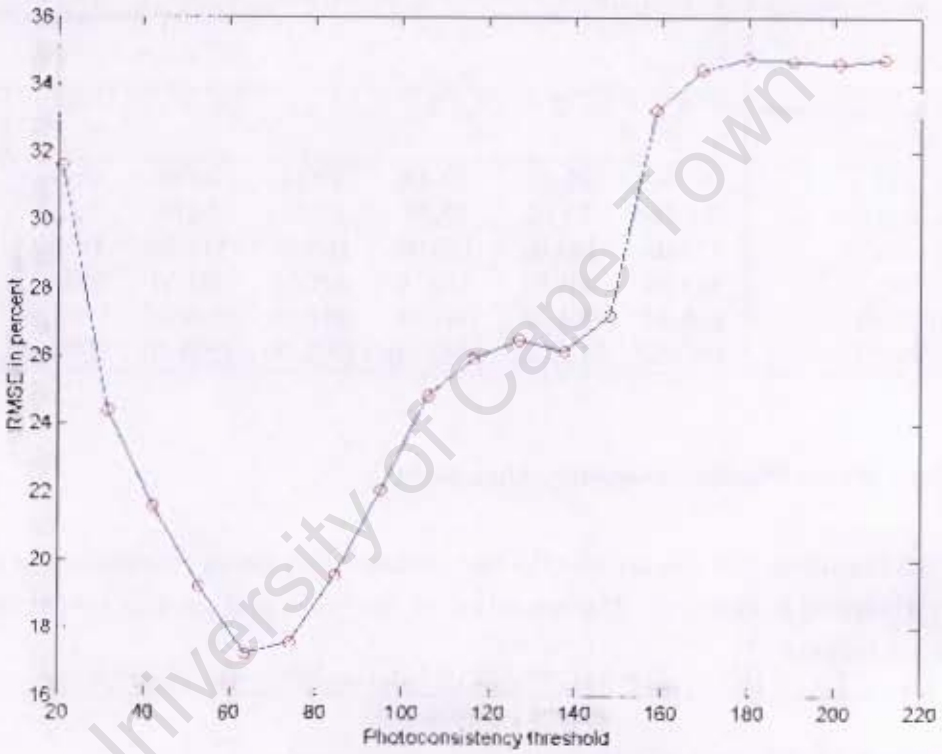


Figure 7.5: Plot of accuracy verses photoconsistency threshold using the toy lion dataset

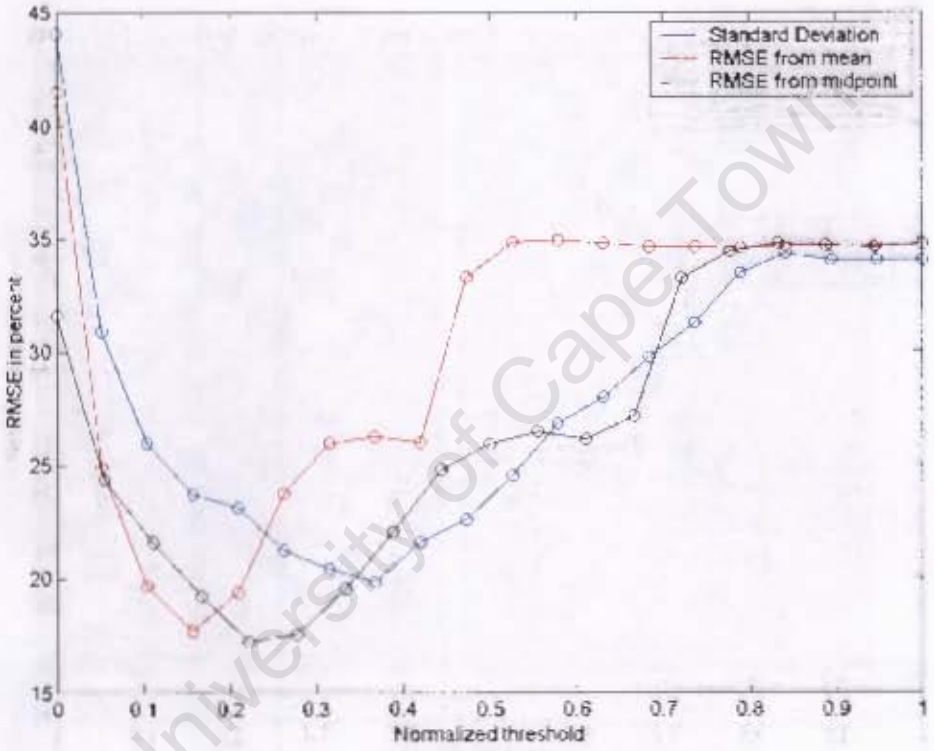


Figure 7.6: Graph illustrating the difference in performance between three photoconsistency measures used to create a computer model of the toy lion.

7.2.2 Reconstruction of a collection of toy animals

This section presents the results obtained from the experiments using the toy animals dataset.

Camera views

Figure 7.7 illustrates some of the camera views used to generate the computer model of the object.

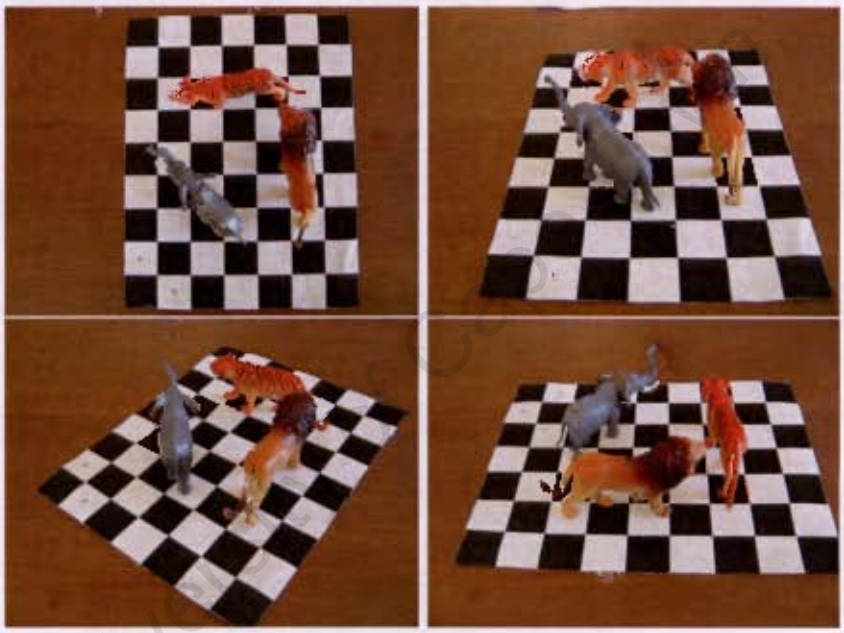


Figure 7.7: Camera views of the toy animals dataset.

Views of reconstructed model

Figure 7.8 shows three views of the reconstructed collection of toy animals.

Comparison images

Figure 7.9 show the two images used to determine the accuracy of the reconstruction.

Figure 7.8: Selected views of the reconstruction using the toy animals dataset.

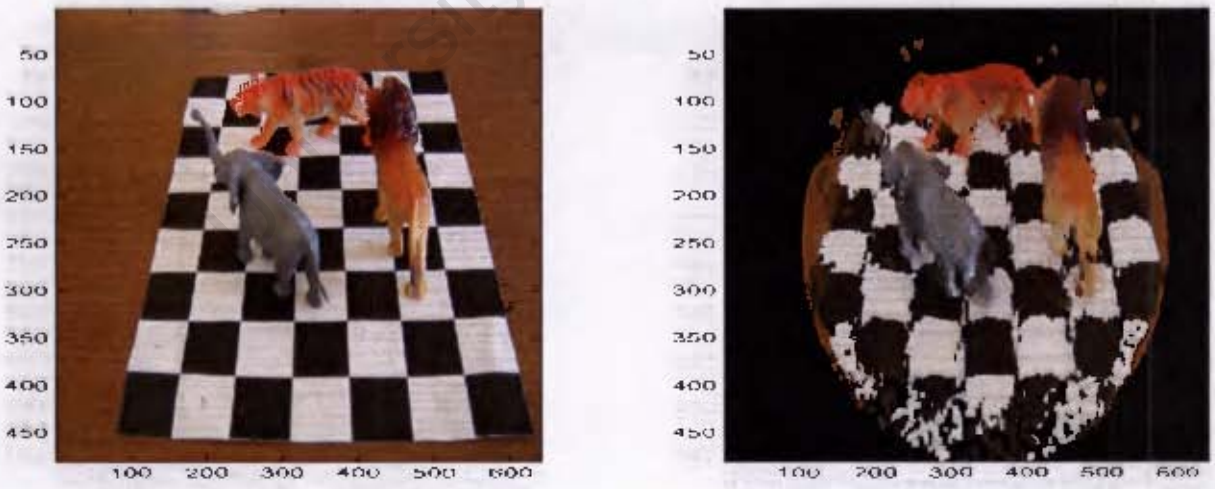


Figure 7.9: Images used in the comparison process. The image on the left is the actual camera view of the toy animals while the image on the right is the rendered view of the reconstruction from the same camera viewpoint.

Error map

Figure 7.10 shows the error map obtained by using the RMS measure on the comparison images.

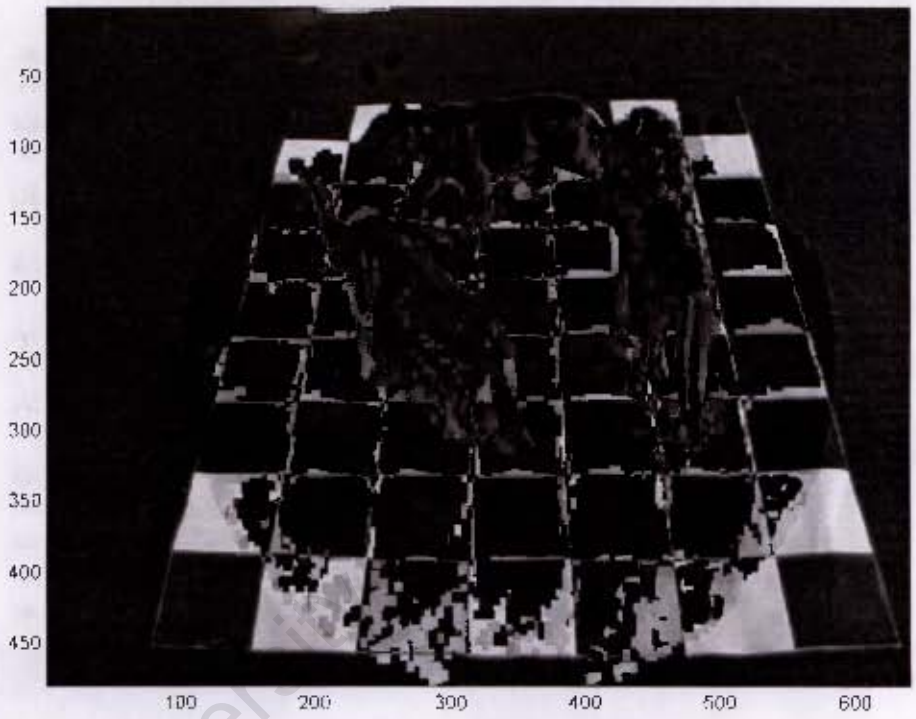


Figure 7.10: Red colour band RMS error map of the reconstruction of the toy animals.

Performance of the estimation algorithm

Tables 7.5, 7.6 and 7.7 present the results obtained from the accuracy measurements of the different model reconstructions of the toy animals dataset. The average computational time required by the reconstruction process is presented in Table 7.8.

Table 7.5: Red band RMSE values for different resolution reconstructions of the toy animals dataset.

View / Resolution	1	2	3	4	5	6	Mean
21x21x21	22.22	20.57	23.67	23.05	25.34	22.46	22.89
41x41x41	18.01	18.75	18.42	18.02	18.70	17.80	18.28
61x61x61	22.40	20.63	19.63	20.24	21.85	20.30	20.84
81x81x81	23.58	21.91	19.42	19.95	22.06	21.14	21.35
101x101x101	22.84	22.17	19.34	19.99	21.38	20.79	21.08
121x121x121	22.06	22.46	19.80	20.59	21.58	21.22	21.28

Table 7.6: Green band RMSE values for different resolution reconstructions of the toy animals dataset.

View / Resolution	1	2	3	4	5	6	Mean
21x21x21	19.23	18.33	18.62	20.43	20.15	19.66	19.40
41x41x41	16.99	18.25	16.61	17.93	17.40	17.07	17.38
61x61x61	22.36	20.62	19.89	20.11	22.08	19.99	20.84
81x81x81	23.74	21.86	19.69	19.82	22.49	21.03	21.44
101x101x101	22.89	22.06	19.28	19.91	21.55	20.51	21.03
121x121x121	22.00	22.31	19.77	20.25	21.56	20.99	21.15

Table 7.7: Blue band RMSE values for different resolution reconstructions of the toy animals dataset.

View / Resolution	1	2	3	4	5	6	Mean
21x21x21	20.76	20.77	20.10	23.27	20.56	21.91	21.23
41x41x41	18.98	20.41	18.91	21.14	19.53	19.77	19.79
61x61x61	24.46	22.54	23.45	22.58	24.83	22.42	23.38
81x81x81	26.02	23.76	23.00	22.40	25.37	23.72	24.04
101x101x101	25.00	23.86	22.43	22.45	24.19	23.17	23.52
121x121x121	24.06	24.10	22.90	22.58	24.17	23.74	23.59

Table 7.8: Computational time required by the reconstruction process using the toy animals dataset (in seconds).

View / Resolution	1	2	3	4	5	6	Mean
21x21x21	28.69	29.58	28.61	29.02	28.88	29.44	29s
41x41x41	78.20	75.63	77.59	76.19	78.28	76.02	1m 17s
61x61x61	214.14	205.78	208.89	203.53	217.50	205.01	3m 29s
81x81x81	632.30	592.38	611.94	638.95	694.13	586.70	10m 26s
101x101x101	1137.20	1159.60	1153.80	1168.30	972.02	1181.10	18m 49s
121x121x121	1892.70	1963.20	1695.00	1991.10	1711.40	2005.50	31m 16s

Accuracy verses Photoconsistency threshold

Figure 7.11 illustrates how the accuracy of the reconstructed model changes as the photoconsistency threshold is changed.

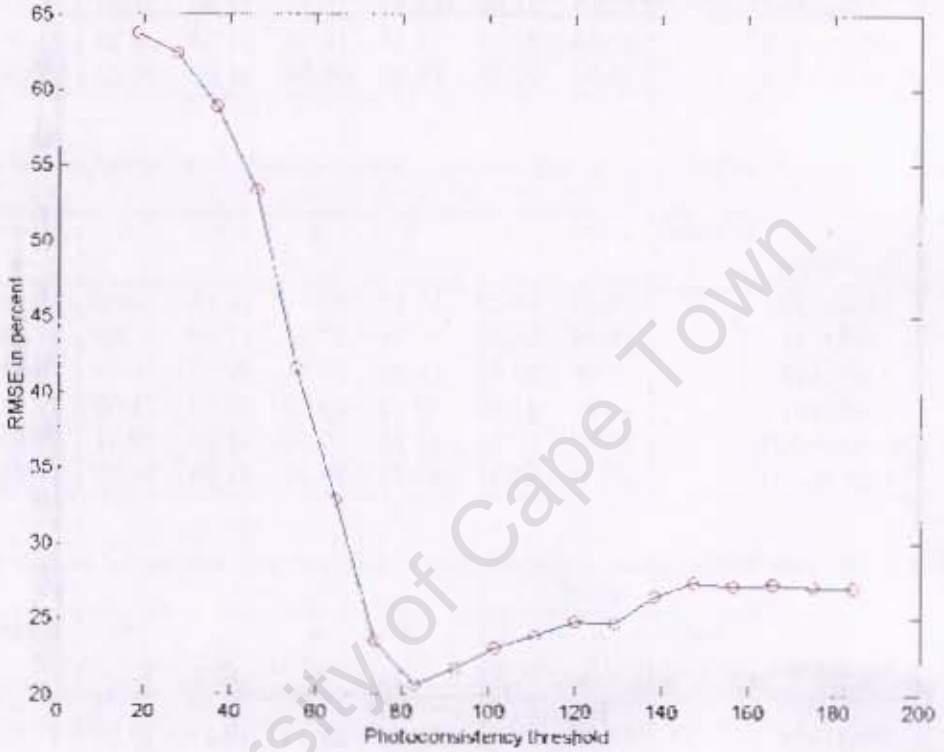


Figure 7.11: Plot of accuracy verses photoconsistency threshold using the toy animals dataset.

Comparison of different photoconsistency measures

Figure 7.12 shows the difference in the performance of three different photoconsistency measures, the standard deviation measure and both variants of the RMSE photoconsistency error. The thresholds have been normalized for better representation. The resolution of the reconstruction used to obtain these graphs is 61x61x61. The accuracy results are obtained from the RMS error of the comparison images in the red colour band.

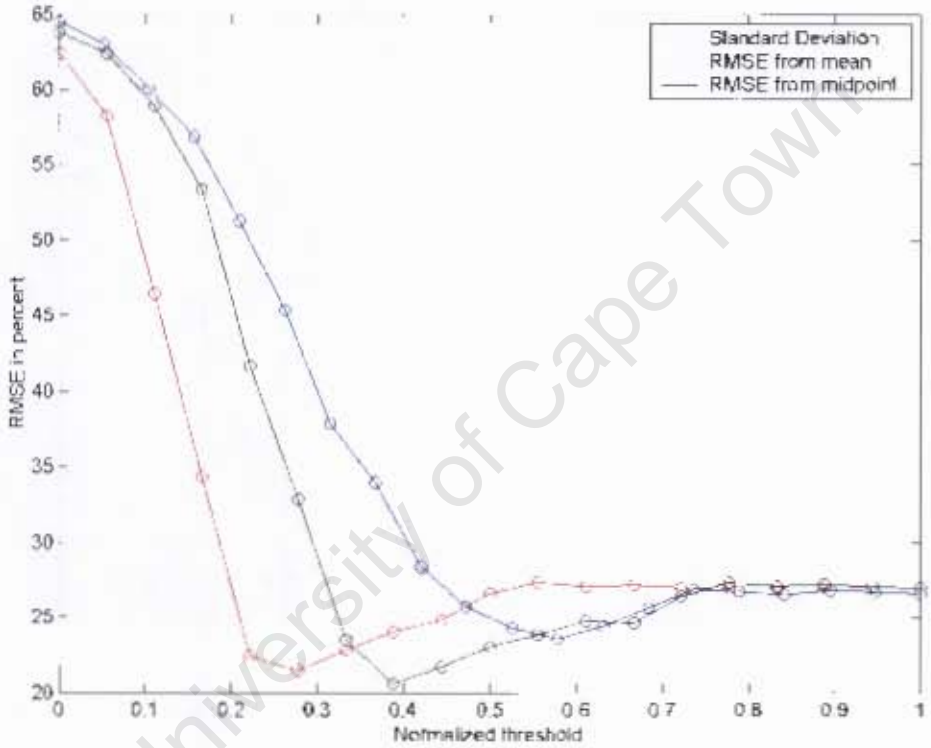


Figure 7.12: Graph illustrating the difference in performance between three photoconsistency measures used to create a computer model of the toy animals.

7.2.3 Reconstruction of a brick fragment

This section presents the results obtained from the experiments using the brick fragment dataset.

Camera views

Figure 7.13 presents some of the camera views used in the reconstruction.

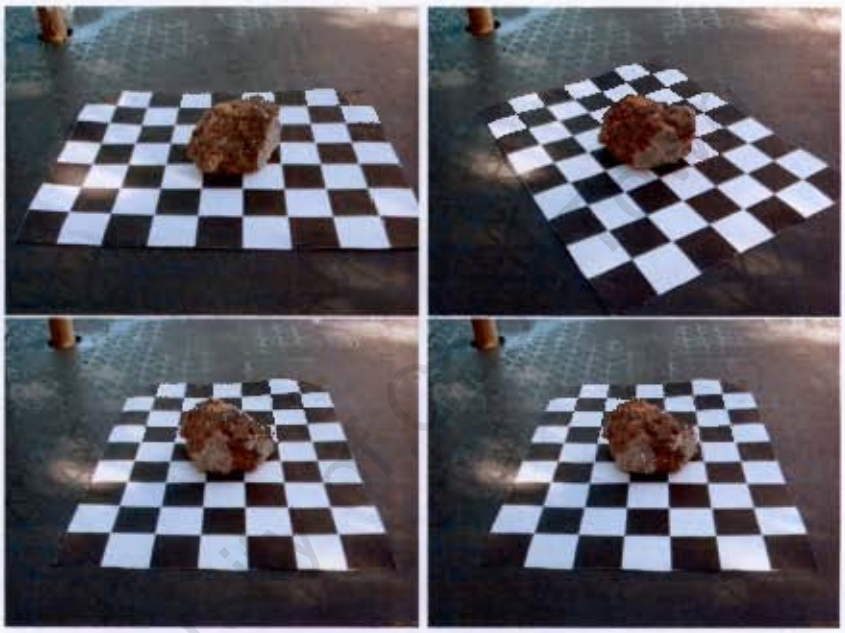


Figure 7.13: Selected camera views of a brick fragment.

Views of reconstructed model

Following the same format as before, Figure 7.14 presents some selected views of the reconstructed brick fragment.

Reconstruction accuracy results

Tables 7.9, 7.10 and 7.11 present the reconstruction accuracy results for different resolution voxel grids for the brick fragment dataset. Table 7.12 presents the computational time required by the reconstruction process.

Figure 7.14: Selected rendered views of the brick fragment model.

Table 7.9: Red colour band RMSE values for different resolution reconstructions using the brick fragment dataset.

View / Resolution	1	2	3	4	5	6	Mean
21x21x21	19.44	18.42	19.06	17.84	21.28	21.15	19.53
41x41x41	12.85	13.89	14.15	13.87	17.51	15.80	14.68
61x61x61	11.95	12.78	15.48	14.42	16.55	14.75	14.32
81x81x81	12.13	14.02	20.88	17.50	18.29	18.62	16.91
101x101x101	14.89	16.98	21.92	19.35	18.91	21.86	18.99
121x121x121	16.20	17.61	21.66	19.86	19.07	24.73	19.85

Table 7.10: Green colour band RMSE values for different resolution reconstructions using the brick fragment dataset.

View / Resolution	1	2	3	4	5	6	Mean
21x21x21	19.81	21.43	21.14	22.36	22.03	23.52	21.71
41x41x41	13.34	15.20	15.60	16.40	19.87	17.37	16.30
61x61x61	12.42	13.62	16.98	16.64	19.04	15.77	15.75
81x81x81	13.00	15.09	23.66	20.20	20.86	20.62	18.90
101x101x101	16.33	18.17	24.79	22.29	21.52	24.65	21.29
121x121x121	18.00	18.94	24.35	22.78	21.82	29.13	22.50

Table 7.11: Blue colour band RMSE values for different resolution reconstructions using the brick fragment dataset.

View / Resolution	1	2	3	4	5	6	Mean
21x21x21	19.54	21.43	21.49	24.25	21.50	25.01	22.20
41x41x41	13.17	14.56	15.51	16.87	20.43	18.24	16.46
61x61x61	12.29	13.11	17.10	17.10	19.93	16.21	15.96
81x81x81	13.17	14.84	24.40	20.89	21.81	21.64	19.46
101x101x101	18.16	18.79	26.03	23.22	22.95	27.02	22.69
121x121x121	20.21	19.90	25.89	24.08	23.64	32.80	24.42

Table 7.12: Computational time required by the reconstruction process using the brick fragment dataset (in seconds).

View / Resolution	1	2	3	4	5	6	Mean
21x21x21	41.50	41.81	42.08	41.77	41.84	41.20	42s
41x41x41	97.80	94.19	97.20	94.99	97.16	95.27	1m 36s
61x61x61	204.28	202.49	205.59	199.66	209.25	208.50	3m 25s
81x81x81	446.88	423.95	452.83	428.69	435.76	440.95	7m 18s
101x101x101	875.01	852.70	849.19	906.25	1007.30	991.66	15m 14s
121x121x121	1894.70	1939.20	1491.60	1676.50	1544.50	1754.50	28m 37s

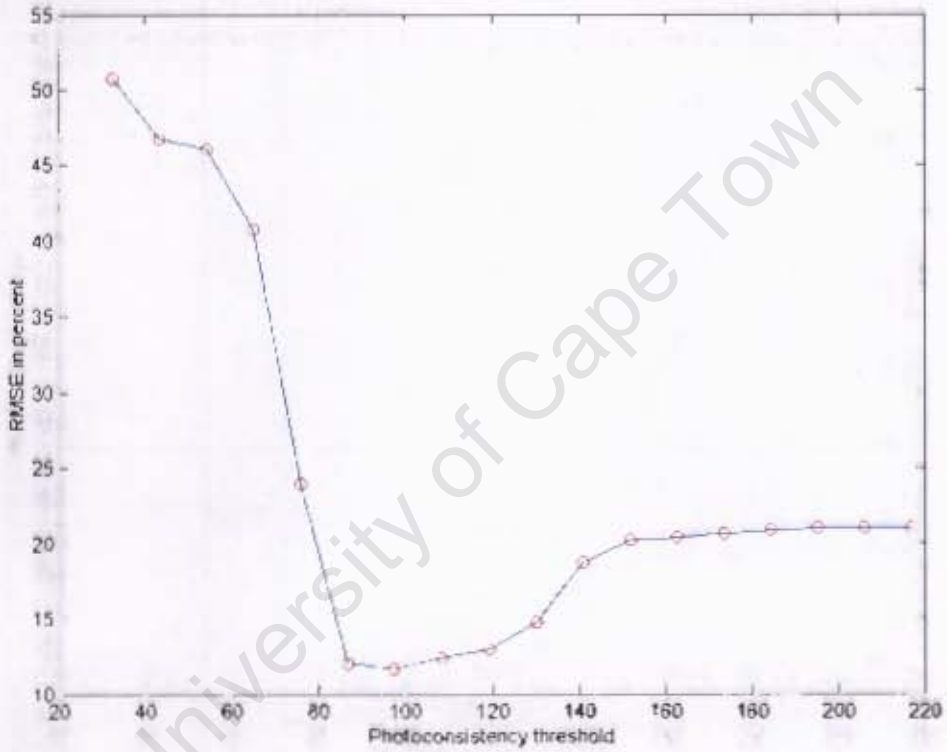


Figure 7.15: Plot of accuracy verses photoconsistency threshold using the brick fragment dataset.

Comparison of different photoconsistency measures

Figure 7.16 shows the difference in the performance of three different photoconsistency measures, the standard deviation measure and both variants of the RMSE photoconsistency error. The thresholds have been normalized for better representation. The resolution of the reconstruction used to obtain these graphs is 61x61x61. The accuracy results are obtained from the RMS error of the comparison images in the red colour band.

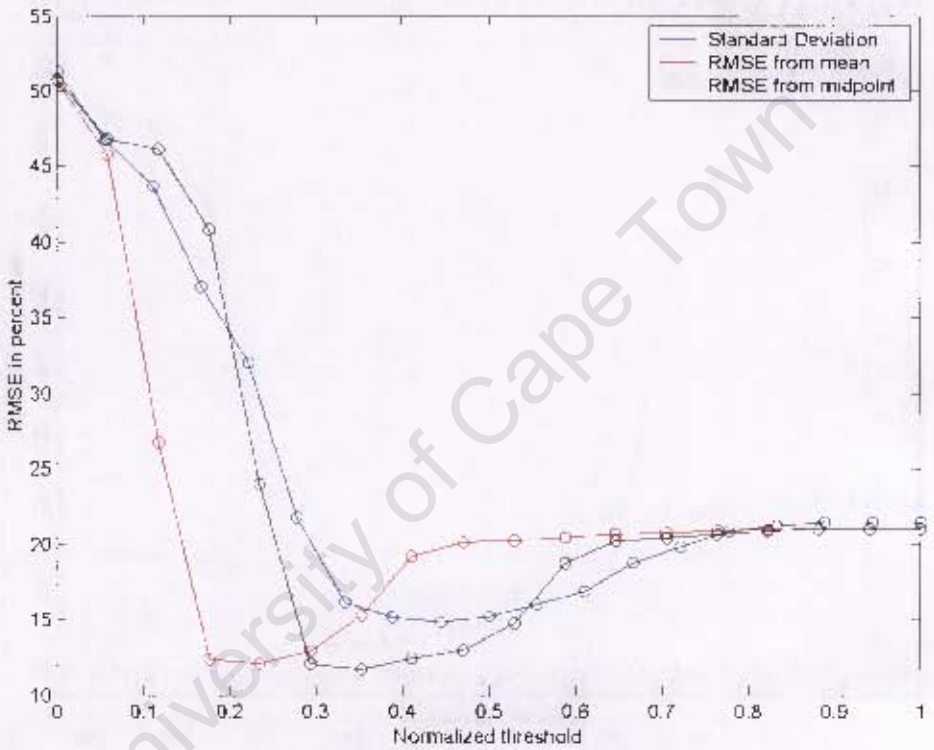


Figure 7.16: Graph illustrating the difference in performance between three photoconsistency measures used to create a computer model of the brick fragment.

7.3 Effect of consistent background

The following reconstruction was performed using a dataset from Web sources [13]. The dataset consisted of 24 views of a movie action figure. Some of the camera views of the object are shown in Figure 7.17.

Figure 7.17: Some camera views of the action figure.

Notice that the background is essentially uniform in colour. This highly consistent background results in the estimation histogram that contains a large percentage of values corresponding to consistency. The histogram of the photoconsistency errors from the pass through the initial voxel grid and the histogram obtained from the pass of the voxels contained in the visual hull, shown in Figure 7.18, illustrates this large amount of overlap.

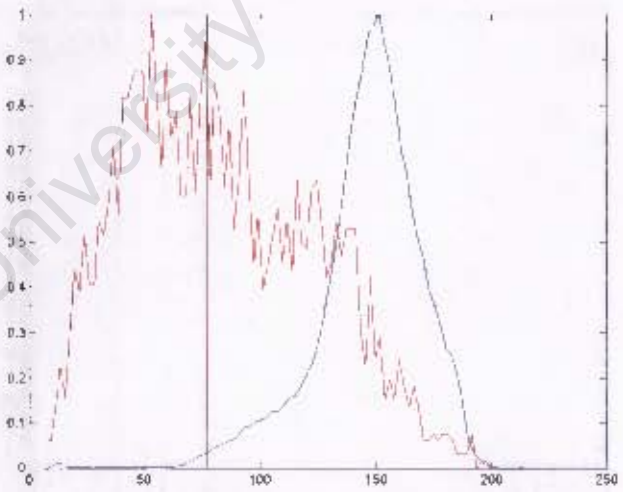


Figure 7.18: Graph illustrating the overlap between the histogram of the photoconsistency measures obtained from the estimation pass through the voxel grid (blue) and the histogram obtained from the a pass through the visual hull (red). The black line indicates the inaccurate estimated threshold.

The overlap of the two histograms results in an inaccurate model of the photoconsistency

values that correspond to inconsistency. Therefore, the estimated threshold is very different from the best value. The result is an inaccurate and incomplete model reconstruction (Figure 7.19).

Figure 7.19: Inaccurate reconstruction of the action figure due to the presence of large amounts of uniformly coloured background.

It is also possible that the photoconsistency measure does not have the distinguishing power needed to create a model with a large amount of consistent background. If the surface colour of the model is close to the colour of the background then the measure may fail to distinguish between the background and the model.

Chapter 8

Conclusions and possible future work

The objectives of this thesis is to perform a 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 create a computer model of an object. The literature indicates that there are better methods such as graph cut based reconstruction that are likely better in creating accurate models than the iterative procedure described in this thesis.

A reconstruction 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. However, the resolution of the recreated model is highly dependent on the resolution of the voxel grid used to perform the reconstruction. Coarse grids allow for a fast reconstruction with small memory usage while finer resolutions require finer grids and more processing time. The cost of having finer resolution increases by a cubic factor.

The limitations of physical computer memory pose a problem. In order to use higher resolution images and voxel grids, more physical memory is needed. Eventually, the limit is reached and other memory management methods need to be used. Perhaps breaking the model into sections, processing each section separately, recording the results to a physical disk and recombining the sections afterwords may resolve the memory limitations.

The results show that the algorithm performs agreeably. However, the estimation algorithm

needs refinement as it is influenced by large amounts of uniformly coloured background.

The algorithms attempt to perform a reconstruction without the need of silhouettes but the reconstructions model the scene rather than a specific object. The user needs to define what part of the reconstructed scene is the desired model. Therefore, either the images or the reconstructed scene needs to be segmented into desirable and non-desirable elements. Perhaps, segmentation can be performed using a recognizable background, be it a uniform colour or a defined pattern. Perhaps the image silhouettes can be generated by segmenting the images based on colour and whether the resulting silhouettes form a consistent visual hull.

The use of voxels as a representation of the object is useful in that it is versatile and can represent any kind of structure. Nevertheless, voxels are not inherently smooth and the resulting reconstructions are "blocky". The voxel is only an approximation to the actual geometry of the surface and the real world object's surface may have any orientation or facing. The voxel representation will cause difficulties when further refinements to the model are desired, such as surface reflectance and lighting effects.

Perhaps deformable meshes can be used to represent the reconstructed model. Meshes can form a better approximation to a surface than voxels but do not have the same flexibility and ease of use as voxels. For instance, voxels can reconstruct a model of an object with distinct separate sections. Multiple meshes would be required to recreate the same object. An algorithm would have to take separateness into account.

The photoconsistency measures presented in this thesis are based purely on the content or distribution of pixel colour values. The measures take no account of the spatial relationship between the pixels. When using voxels, a perfect surface orientation is not known therefore these measures are a simple way of determining whether that particular voxel is photoconsistent or not, yet these measures can return false positives if the colour distributions are similar in nature. For instance, a particular texture may have a certain distribution of colour elements. This distribution may be similar all over the surface of the target object. This can result in voxels being incorrectly classified as part of a surface because each view of that voxel may project onto a *similar* looking surface but not necessarily the *same* region of surface. Therefore, a more accurate result would probably be obtained by using a measure that performs actual spatial image *matching* rather than *consistency*. The difficulty in using spatial measures is that the surface geometry of that particular region needs to be known or estimated beforehand.

Bibliography

[1] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1067–1073, 1997.

[2] David. A. Forsyth and Jean Ponce. Cameras. In *Computer Vision*, pages 3–6. Prentice Hall, 2002.

[3] Andrea Bottino and Aldo Laurentini. The visual hull of smooth curved objects. In *Transactions on Pattern Analysis and Machine Intelligence*, pages 1622–1632. IEEE Computer Society, 2004.

[4] C. Dyer. Volumetric scene reconstruction from multiple views. In L.S. Davis, editor, *Foundations of Image Understanding*, pages 469–489. Kluwer, Boston, 2001.

[5] Ulaş Yılmaz Oğuz Özün and Volkan Ataly. Comparison of photoconsistency measures used in voxel coloring. In *ISPRS Workshop in conjunction with ICCV 2005*, 2005.

[6] David. A. Forsyth and Jean Ponce. Radiometry. In *Computer Vision*, pages 27–42. Prentice Hall, 2002.

[7] James. M. Palmer. Radiometry and photometry FAQ. 1999. http://www.optics.arizona.edu/Palmer/rpfaq/rpfaq.htm.

[8] T. Malzbender M.R. Stevens, B. Culbertson. A histogram-based color consistency test for voxel coloring. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 4, pages 118–121, 2002.

[9] G. Vogiatzis, P. Torr, and R. Cippola. Multi-view stereo via volumetric graph-cuts, 2005.

[10] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part III*, pages 82–96. Springer-Verlag, 2002.

[11] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part III*, pages 65–81, London, UK, 2002. Springer-Verlag.

[12] Camera calibration toolbox for MATLAB. http://www.vision.caltech.edu/bouguetj/calib_doc.

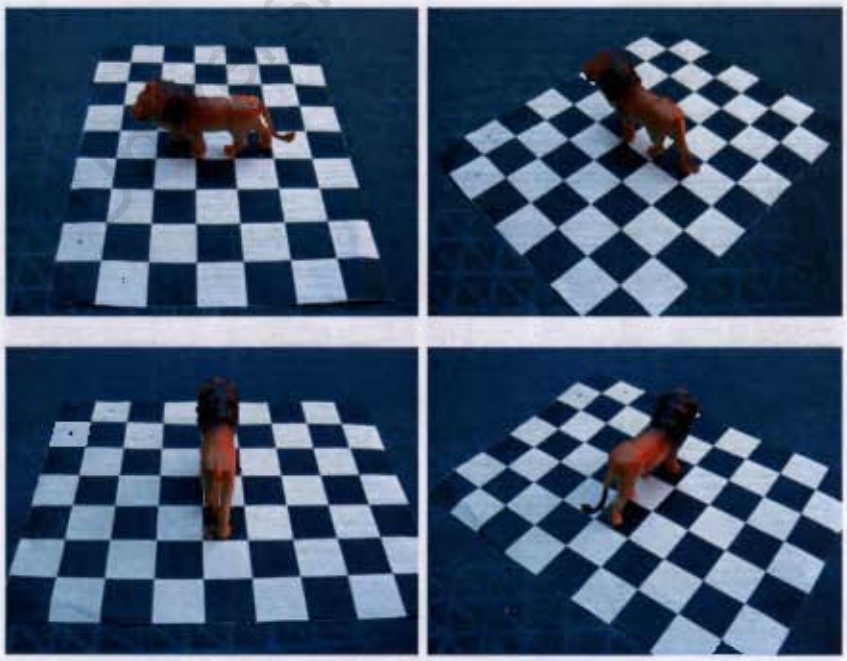
[13] T. Malzbender W. Culbertson and G. Slabaugh. Generalized voxel coloring. In *ICCV Vision Algorithms Workshop*, number 1883 in Lecture Notes in Computer Science, pages 100–115. Springer-Verlag, 1999.

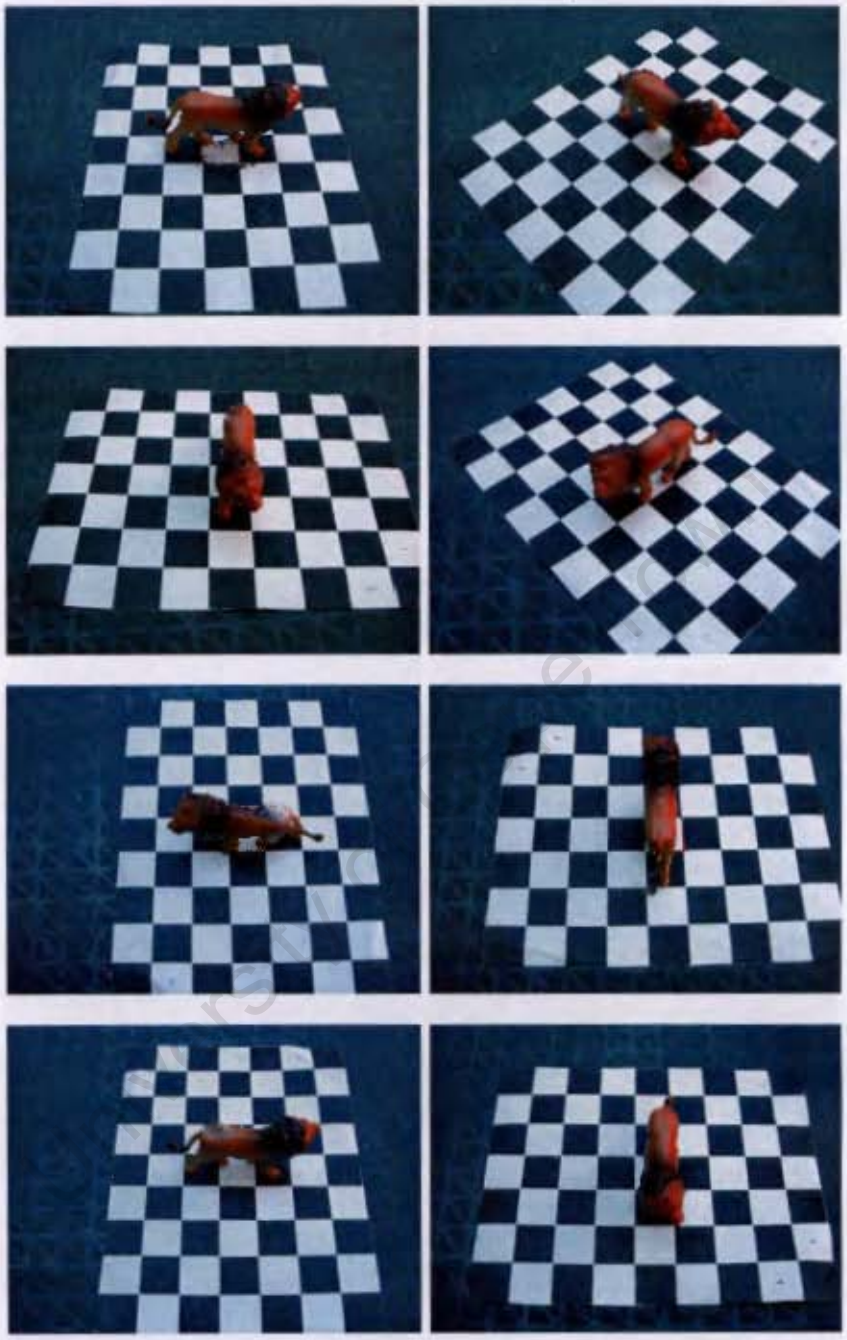
[14] Jean Ponce Yasutaka Furukawa. 3D photography dataset. http://www-cvr.ai.uiuc.edu/ponce_grp/data/mview.

Appendix A

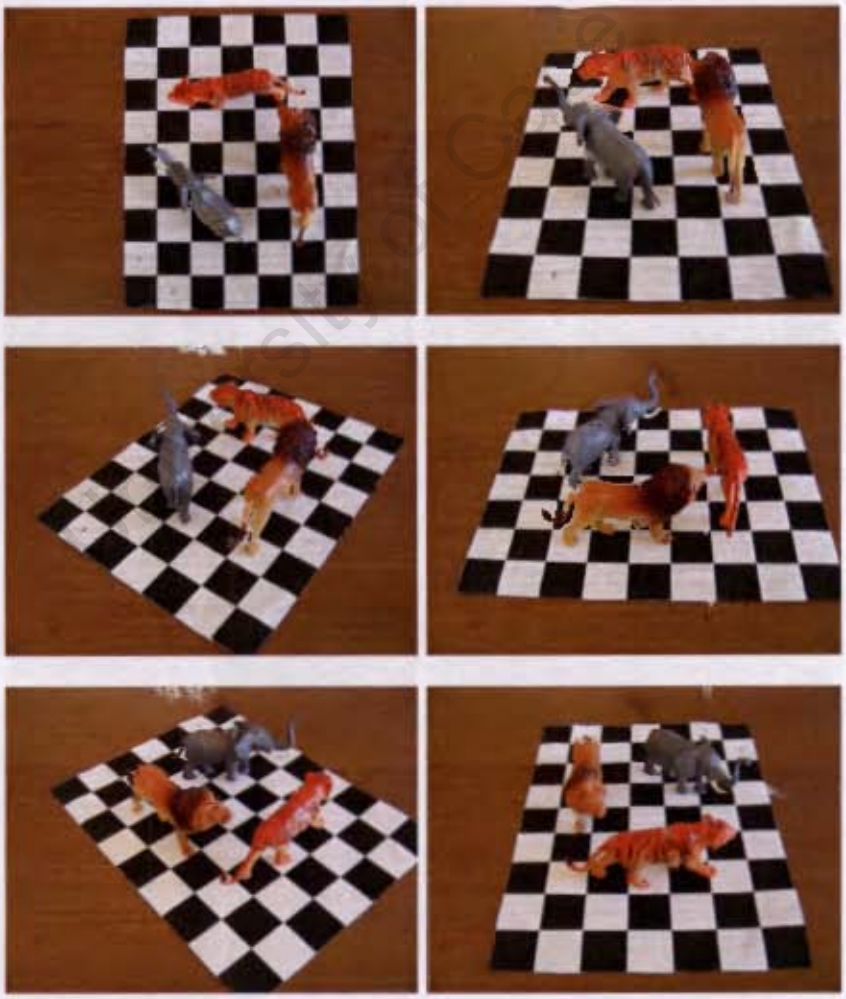
Dataset images

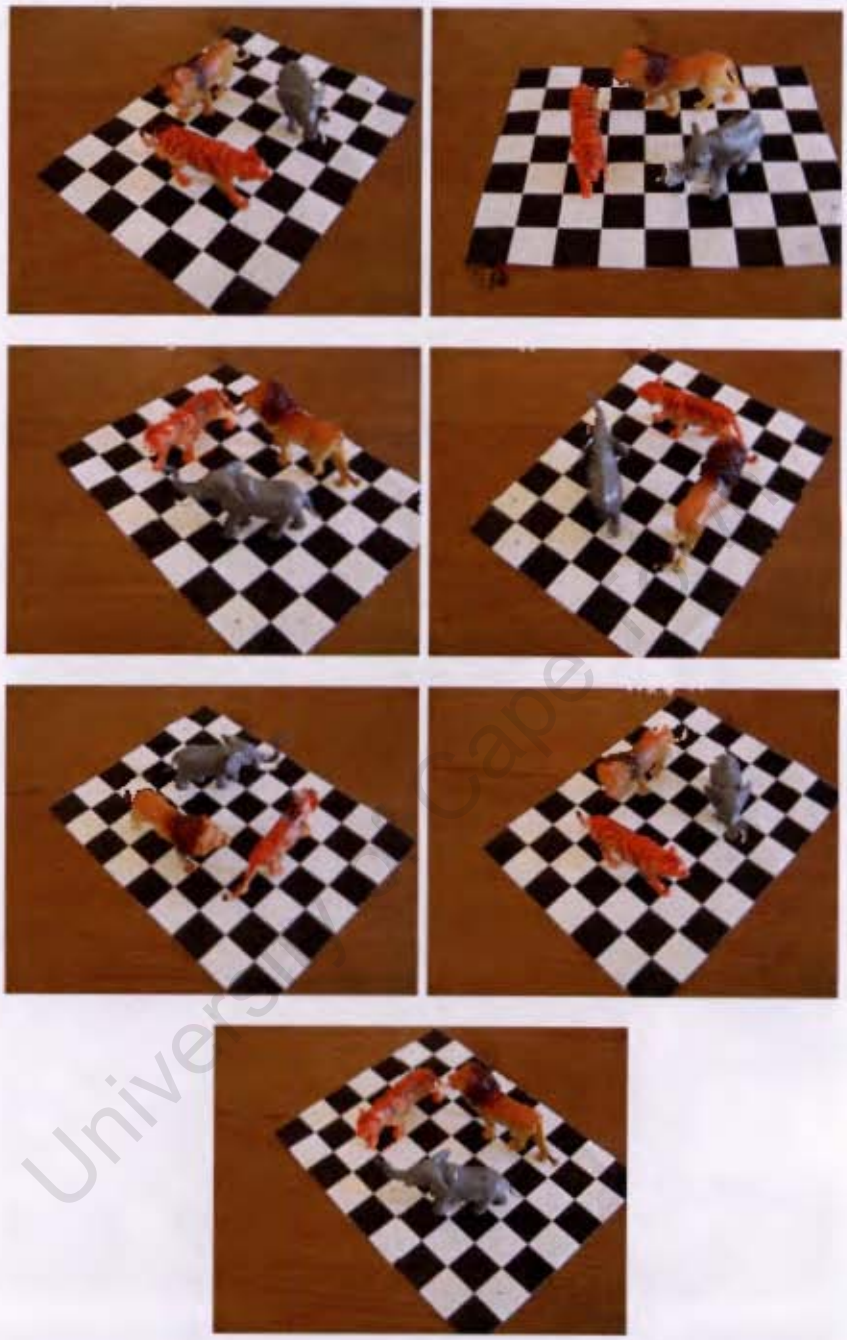
A.1 Lion dataset





A.2 Figurine dataset





A.3 Brick fragment dataset