

# 國立交通大學

資訊科學系

博士論文

環物影片產生及應用技術之研究



New Techniques for Production and Application of Object Movies

研究生：蔡玉寶

指導教授：施仁忠 教授

洪一平 教授

中華民國九十六年六月

環物影片產生及應用技術之研究

New Techniques for Production and Application of Object Movies

研究生：蔡玉寶

Student：Yu-Pao Tsai

指導教授：施仁忠教授

Advisors：Dr. Zen-Chung Shih

洪一平教授

Dr. Yi-Ping Hung



Submitted to Department of Computer and Information Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer and Information Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

# 環物影片產生及應用技術之研究

學生：蔡玉寶

指導教授：施仁忠教授  
洪一平教授

國立交通大學資訊科學學系（研究所）博士班

## 摘 要

由於環物影片(Object Movies)在建置上相當簡單以及能產生相片品質的成像結果，因此環物影片目前已是一種非常通用的方法來表現可互動的三維物體。雖然環物影片已成功地被應用在不同的領域，但關於產生高品質環物影片的技術以及更具有開創性應用的技術都有待研究開發。

本論文提出一些方法來產生高品質的環物影片以及展示一些利用環物影片來表現三維物體的應用。首先，我們提出一個環物影片拍攝架的校正方法並提供一個視覺化的界面幫助使用者根據計算結果來調整拍攝架。實驗顯示利用此校正方法，只需取得12張校正物的影像便可以得到任一視角的精確相機參數。接著，我們提出環物影片背景去除的方法。該方法最主要的好處是使用者的介入非常地少。更確切地說使用者只需從自動去背景後的影像中選擇足夠好的結果，該方法便能自動地將這些正確的資訊傳遞到其它的影像並修正去背結果。在傳遞的過程中，該方法利用選擇的影像重建該三維物體的三維幾何形狀，再投影到其它的影像上產生物體的剪影以協助自動去背方法產生更好的去背結果。實驗顯示，透過這剪影資訊可以明顯地降低去背結果的誤差並能有效地抵抗雜訊的影響。另外，一個新的三維重建方法被提出來從環物影片之中重建出該物體的三維資訊。相較之前的方法，該方法所產生的三維模型可以保留住更多的細節部份而且其剪影會和原始輸入的剪影影像相吻合。最後，我們延續之前環場環物整合技術的研究，提出可將立體環物整合到立體環場的方法。使用者可以透過所提供的系統可以整合立體環

場環物建置出一個非常逼真的互動環境，並可直接瀏覽該虛擬環境以及觀看立體環物。  
經由這樣的互動環境使用者可以透過立體視覺體驗到更真實的3D感受。



# New Techniques of Production and Application of Object Movies

Student: Yu-Pao Tsai

Advisors: Dr. Zen-Chung Shih  
Dr. Yi-Ping Hung

Department ( Institute ) of Computer and Information Science  
National Chiao Tung University

## Abstract

Object movie (OM) is a conventional approach for modeling and rendering interactive 3D objects because of its simplicity in production and its photorealistic presentation of objects. Although OMs have been successfully adopted in many applications, the techniques for production and application of OMs must still be enhanced if high-quality and efficient OMs are desired.

This work proposes some methods for generating high quality OMs, and demonstrates some applications using generated OMs to present the 3D objects. First, a method for calibrating the motorized object rig is presented, and a visual tool is introduced to adjust the axes of the motorized object rig. The distances among all the three axes of the motorized object rig can be minimized after adjustment, and more reliable camera parameters can be obtained after the calibration process. Experimental results indicate that highly accurate parameters can be obtained from only 12 images. Second, an image segmentation method is proposed to remove the backgrounds of OMs. The major advantage of the proposed method is it can propagate the successful segmentation results from some selected images to the whole OM. The

new OM segmentation method extracts a 2D shape from the reconstructed 3D model and uses the 2D shape to remove the background from the foreground object. This work demonstrates that the proposed method can significantly improve OM segmentation. Third, a novel approach is proposed to reconstruct high quality 3D models from OMs. The silhouettes and detail features of reconstructed 3D model are successfully preserved. Finally, previous work on augmented panoramas is extended to augmented stereo panoramas. This work develops an interactive system that allows the user to integrate stereo OMs into a stereo panorama, and interactively browses the augmented stereo panorama. To generate stereo OMs, the 3D models reconstructed from monocular OMs are rendered. The proposed method takes less than half of the processing time, including acquisition and segmentation, than traditional approaches, which take two separate sets of OMs. The proposed interactive system provides the users with two approaches to determine the reference frames where the object is inserted in a stereo panorama. The left view and the right view are rendered separately after determining the reference frames. For each view, the background layer is first rendered, followed by the shadow and the object layers. A user can directly rotate and translate the stereo object movie of interest by browsing the augmented panorama. The augmented stereo panoramas provide users with more persuasive interaction with better depth perception.

# Content

摘 要 .....	I
ABSTRACT .....	III
CONTENT .....	V
LIST OF FIGURES .....	VII
LIST OF TABLES .....	XII
<b>CHATPER 1 INTRODUCTION .....</b>	<b>1</b>
1.1. MOTIVATION .....	1
1.2. REVIEW OF RELATED WORK .....	4
1.3. ORGANIZATION OF THIS THESIS .....	8
<b>CHATPER 2 OBJECT RIG CALIBRATION.....</b>	<b>9</b>
2.1. ESTIMATION OF CAMERA PARAMETERS.....	10
2.2. COMPLETELY AND PARAMETER CONTINUOUS (CPC) KINEMATIC MODEL.....	11
2.3. KINEMATIC CALIBRATION USING THE CPC MODEL.....	13
2.4. EXPERIMENTAL RESULTS OF CALIBRATION.....	18
<b>CHATPER 3 BACKGROUND REMOVAL.....</b>	<b>22</b>
3.1. INITIAL LABELING.....	25
3.2. LABEL UPDATING WITH MOTION VECTORS.....	28
3.3. LABEL UPDATING WITH SHAPE PRIORS .....	32
3.4. EXPERIMENTAL RESULTS OF BACKGROUND REMOVAL .....	36
<b>CHATPER 4 OBJECT MOVIE-BASED 3D RECONSTRUCTION .....</b>	<b>44</b>
4.1. VOLUMETRIC GRAPH CUTS.....	44
4.2. OUR APPROACH .....	49
4.3. EXPERIMENTAL RESULTS OF 3D RECONSTRUCTION .....	55
<b>CHATPER 5 AUGMENTED STEREO PANORAMAS .....</b>	<b>61</b>
5.1. GENERATION OF STEREO PANORAMAS .....	61
5.2. GENERATION OF STEREO OBJECT MOVIES .....	63
5.3. AUGMENTING STEREO PANORAMAS WITH STEREO OMS .....	64
5.4. EXPERIMENTAL RESULTS OF AUGMENTED STEREO PANORAMAS .....	66
<b>CHATPER 6 CONCLUSION AND FUTURE WORK .....</b>	<b>68</b>

**REFERENCES**..... 71

**APPENDIX** ..... 76

SINGULARITY-FREE LINE REPRESENTATION ..... 76





# List of Figures

Fig. 1. Motorized object rig – AutoQTVR. ....	2
Fig. 2. Motorized object rig – Kaidan Magellan™ 2500. ....	2
Fig. 3. Processing Flowchart of Calibration .....	10
Fig. 4. The calibration object, called physical control cube (PCC), and the extracted feature points used to obtain intrinsic camera parameters.....	11
Fig. 5. The schematic of motorized object rig.....	14
Fig. 6. The OM of the toy shark before calibration. (a) shows the estimated relation among 3 axes, and (b) shows the OM of the toy shark. The cross markers indicate the center of images.....	20
Fig. 7. The result of the toy shark experient. (a) shows some images of the OM of the toy shark after calibration, while (b) shows that after centralization. (c) shows the Visual Hull of the shark, and (d) shows the estimated axes after calibration. ....	21
Fig. 8. Part of the two different equi-tilt sets before applying the OM segmentation method. Except for leftmost two images in the figure, the remainder of the images in this thesis are cropped in order to show more examples. ....	23
Fig. 9. The flowchart of segmentation method.....	25
Fig. 10 The top row shows a portion of the input image sequence taken from an equi-tilt set of the pottery owl OM. For all the images in the middle and bottom rows, the black pixels correspond to the classified background regions. The foreground regions are colored white, and the unknown regions are colored gray. The middle row shows the corresponding result during the <i>B</i> -labeling for each image. Notably, to filter out the incorrectly classified pixels and obtain the global background mask used during <i>F</i> -labeling, label consistency and mathematical morphology are used as shown in Fig. 11. Finally, the bottom rows shows the generated trimap for each image that is used to activate the graph cut image segmentation.....	27
Fig. 11. (a) The result including the label consistency concept is included; (b) The global background mask obtained by applying the mathematical morphology on (a).....	28
Fig. 12. 2D spatial graph construction example. ....	29
Fig. 13. The worms of the uncertain pixels $i_1$ and $i_2$ .....	32
Fig. 14. The process flowchart of the new system. ....	35
Fig. 15. $C_1$ , $C_2$ , and $C_4$ denote the views adopted to built the visual hull. Notably, the true surface of the object is assumed to be between the base and inner surfaces. Although the segmentation results of $C_3$ and $C_5$ are poor, they can be improved by incorporating	

the projection of the reconstructed model into the graph cut image segmentation algorithm.....	36
Fig 16 The top row shows a portion of the input image sequence taken from an equi-tilt set of the pottery owl OM. For all the images in the middle and bottom rows, the black pixels correspond to the classified background regions. The foreground regions are colored white, and the unknown regions are colored gray. The middle row shows the corresponding result during the B-labeling for each image. Finally, the bottom rows show the generated trimap for each image that is used to activate the graph cut image segmentation.....	39
Fig. 17. The results of the automatic initial segmentation corresponding to the image sequence shown in Fig 16. The three images on the left show the segmentation results that should be selected for the 3D reconstruction, while the others shows results that should be excluded and refined in the next run. The red circles denote the noticeable segmentation errors in each image. ....	40
Fig. 18. The top row shows a portion of an equi-tilt set for the toy house OM. The middle row shows the trimap labeling result for each image. Finally, the bottom row shows the results of the automatic initial segmentation. The red circles indicate the noticeable segmentation errors in each image, to be rectified in the next run.....	40
Fig. 19. The rectification of the segmentation errors for the pottery owl in Fig. 17 and the toy house in Fig. 18. Trimaps (top row): the projection of the reconstructed model is colored white, and serves as the foreground hard constraints together with the previously generated trimap. Refinement (bottom row): the refinement of the segmentation result is shown for each image.....	41
Fig 20. The first row shows six consecutive images in an equi-tilt set of the pottery cat OM. The second row shows the result of trimap labeling. The third row shows the result of the automatic initial segmentation. In the fourth row, the projection of the reconstructed 3D model provides the information on regions that is quite difficult to obtain by the methods based on color and contrast alone. The last row shows the refinement of the segmentation result by using shape priors. ....	41
Fig. 21. The Armadillo that is the 3D model adopted to generate the synthetic data. ....	42
Fig. 22 Mean segmentation errors on the synthetic data. The image size is 800 x 600. In the experiments, the 3D shape was reconstructed by randomly selecting ground truth images. The shape prior 1 was learnt by using 10 images, and the shape prior 2 was learnt by 20 images.....	42
Fig. 23. The first row shows six consecutive images in an equi-tilt set of the Armadillo OM. The second row shows the result of trimap labeling. The third row shows the result of the automatic initial segmentation. In the fourth row, the projection of the reconstructed 3D model provides the information on regions that is quite difficult to	

obtain by the methods based on color and contrast alone. The fifth row shows the refinement of the segmentation result by using shape priors. The last row shows the comparison between the segmentation results produced by the proposed method and the ground truth. The red solid lines denote the contours of the ground truth, and the green dot lines denote the segmentation results produced by the proposed method.....43

Fig. 24. Illustration of volumetric graph cuts algorithm. (a) Graph cuts algorithm is used to find the  $S_{min}$  surface between  $S_{base}$  and  $S_{in}$  in volumetric graph cuts. (b)  $x_i$  and  $x_j$  are the neighbor voxels. The edge weight between these two voxels is represented as  $w_{ij}$  and the edge weight between voxels and source node is represented as  $w_b$ .  $h$  means the length between two voxels. ....47

Fig. 25 Two cases that cause errors may occur in volumetric graph cuts. Because of the shorter cut property of volumetric graph cuts, Concavity-Convex feature will be flattened in volumetric graph cuts. ....48

Fig. 26. Silhouette images. (a) is the input silhouette image for 3D reconstruction. (b) is the silhouette image generated from the reconstruction model using volumetric graph cuts algorithm.....48

Fig. 27. The comparison between silhouette images shown in Fig. 26. The unmatched regions are colored in red and green.....49

Fig. 28. The broken ears is caused by not considering the silhouette information in volumetric graph cuts. ....49

Fig. 29. The flowchart of our approach. This approach contains two phases. In the first phase, a silhouette-preserved model is generated by a silhouette-preserved volumetric graph cuts algorithm. Then, the result of phase 1 is refined by gradient descent in phase 2. ....50

Fig. 30. Silhouette-preserved volumetric graph cuts algorithm. Orange circle: the 3D object to be reconstructed. Purple grid: The voxels labeled "IN Object" after volumetric graph cuts. Red grid: The voxels have to increase edge weights to match the silhouettes. ....53

Fig. 31. A silhouette image projected from the reconstructed 3D model using the silhouette-preserved volumetric graph cuts.....53

Fig. 32 Comparison between silhouette images shown in Fig. 31 and Fig. 26 (a).....53

Fig. 33. The reconstructed 3D model after phase 1. Notably, The broken ears are fixed. ....54

Fig. 34 The meaning of symbols in (50).....55

Fig. 35. The reconstructed model of the toy house by using the volumetric graph cuts algorithm without imposing the DMA constraint. The ballooning term is increased gradually from left to right. The figure indicate that reconstructing the toy house is a difficult task without the DMA constraint.....57

Fig. 36. Visualization and comparison of the 3D reconstruction algorithm. Both (b) and (c) are taken from a cross-section of the visual hull for the toy house, which is shown in (a). The golden voxels correspond to the base surface in all three images. The cyan voxels denote the inner surface, which is parallel to the base surface. Additionally, the voxels in VA are also colored cyan in (c). The photo-consistency scores between the base and inner surfaces are shown, where the darker region indicates a better photo-consistency. Additionally, the line within the base and inner surfaces represents the reconstructed surface of the object. In (b), without the DMA constraint, although the reconstructed surface passes through the worse photo-consistency regions, the integral of the energy on the entire surface is lower. Consequently, the protrusive part (i.e., the tower of the house) is flattened incorrectly. The image in (c) shows the correctly reconstructed surface for the same portion of the object with the DMA constraint. ....57

Fig. 37. Image (a) shows the visual hull generated from the available silhouettes of the toy house to act as the base surface in the algorithm; (b) the DMA of the visual hull that is considered to be an approximate DMA of the toy house. Images (c)-(h) show the reconstructed model from three different viewpoints of the toy house, together with the images captured at similar viewpoints. ....58

Fig. 38. (a) The visual hull of the pottery owl. (b) The DMA of the visual hull. (c) An example image of the pottery owl MVI. (d) The reconstructed model of the pottery owl by using our method. ....58

Fig. 39 The reconstructed owl models. (a) The result of phase one. (b) The result of phase two. ....59

Fig. 40 The reconstruction results of the bunny model. (a) The result of traditional volumetric graph cuts algorithm (b) The result of phase one (c) The result of phase two (b) The ground truth. ....59

Fig. 41 The reconstructed buddha models. (a) The result of traditional volumetric graph cuts. (b) The result of phase 1. (c) the result of phase 2 (d) The ground truth..... 59

Fig. 42 Comparison between our result and original image. Left is the original image and the right is the result reconstructed by our method. .... 60

Fig. 43. A diagram shows the idea to create a stereo panorama using a video camera..... 62

Fig. 44 (a) The original OM images. (b) Our rendering results of binocular views..... 64

Fig. 45. The UI allows users to integrate stereo OMs into a stereo panorama in 3D mode..... 66

Fig. 46. Illustration of casting shadow for an object movie. .... 66

Fig. 47: Stitching result of a stereo panorama. .... 67

Fig. 48: Result of the augmented panorama with a stereo OM. (a) shows the rendered left view, and (b) shows the right view..... 67

Fig. 49 Rotating the OM in the augmented stereo panorama. (a) and (c) are the left views. (b) and (d) are the right views. ....67



# List of Tables

Table 1. Processing time and accuracy of the calibration processing ..... 19



# Chapter 1

## Introduction

### 1.1. Motivation

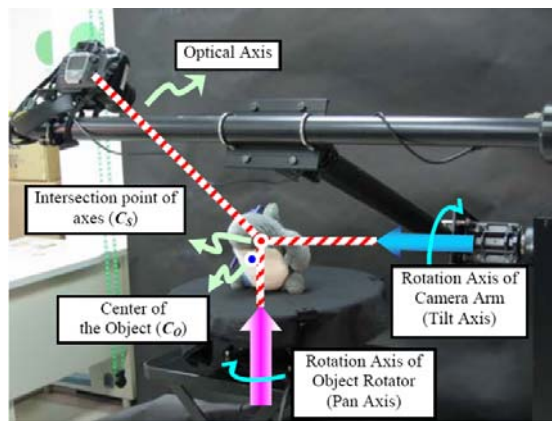
Modeling and rendering photorealistic 3D objects are significant tasks in computer graphics. Two conventional techniques are the geometry-based and the image-based approaches. The geometry-based approach first constructs 3D models of real world objects then generates the results by rendering the 3D models with attached textures. This approach provides good interactivity, but the 3D models to be constructed, which is a tedious process. In contrast, the image-based approach uses real images for interactive displaying and browsing. It provides photo-realistic visual effects, and its rendering speed is independent of the complexity of the scenes or objects.

Many methods have been proposed for image-based modeling and rendering. Object movie (OM) proposed by Apple Inc. is a conventional approach because of its simplicity in acquisition and its photorealistic ability to present the 3D objects. OM has recently been widely adopted in many applications, e.g., e-commerce, digital archive, digital museum [32], etc. An OM is a set of images taken from different perspectives around a 3D object. Fig. 1 and Fig. 2 show two different motorized object rigs. The OM can be treated as an interactive video of the 3D object after acquisition. Each image in an OM is associated with a pair of distinctive pan and tilt angles of the viewing direction, allowing a particular image to be chosen and shown on screen according to the user's viewing direction, which is generally specified by controlling mouse motion. Users can thus interactively rotate the virtual artifacts arbitrarily, and freely

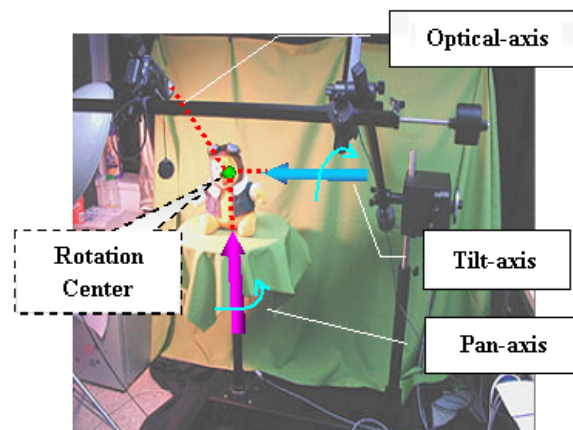
manipulate the object.

Although OMs have been successfully used in many applications, the techniques for producing OMs still need to be improved if high-quality and efficient OMs are desired. This work investigates the methods of production and applications of high-quality OMs.

The motorized object rig, AutoQTVR, developed by Texnai Inc., was adopted to acquire the OMs. The motorized object rig is a computer-controlled 2-axis omniview shooting system, as shown in Fig. 1. It has two rotary axes, the pan-direction object rotator and the tilt-direction camera arm rotator. For convenience, these rotation axes of the rotators are called the tilt and the pan axes, respectively.



**Fig. 1. Motorized object rig – AutoQTVR.**



**Fig. 2. Motorized object rig – Kaidan Magellan™ 2500.**

In OM acquisition, the center of an object should be placed at the crossing point of the two



rotation axes and the optical axis of the camera, as shown in Fig. 1. Otherwise, the acquired OM would have a bizarre rotation effect when it is browsed. Consequently, to acquire high quality OMs that rotate smoothly, the three axes should first be made to intersect at a common point,  $C_s$ . However, since the optical axis of the camera is invisible, aligning these three axes is inherently a difficult problem. This work develops a method for calibrating a motorized object rig to facilitate the acquisition of OMs, and for improve the accuracy of camera parameters, which can be used for different subsequent tasks, e.g., 3D reconstruction, background removal, and stereo OM generation.

To enhance the rendering results, or to integrate OMs into a new background, the background must be efficiently and effectively removed from the foreground object. However, this is a challenging task. Additionally, another task requires OM segmentation is 3D reconstruction using captured OMs. However, as is well known, OM segmentation is a more tedious and expensive task than the acquisition of the OM as mentioned above. In our experience, segmenting the images manually would take more than 30 man-hours, because an OM generally contains hundreds of images. Additionally, the OM segmentation task can become very time-consuming and burdensome for stereo object movies [9].

Yielding two distinct foreground and background color distributions can obviously mitigate the difficulty of OM segmentation. Blue-screen and green-screen matting have been widely adopted in movie production to achieve this purpose. However, a black screen is preferable for acquiring the OM to prevent the object from reflecting the blue or green light, particularly in the domain of digital archives and digital museums. A black screen frequently results in ambiguously shadowed regions that can significantly increase the difficulty of OM segmentation, such that even a patient expert might become tired of the segmentation. Therefore, the usability of the designed OM segmentation method should be examined in terms of computational expense, accuracy of the segmentation result and amount of user intervention. This work devises a new image segmentation method to help the user obtain a quality OM

segmentation result in less than one man-hour.

Many applications require 3D geometry model to perform 3D processing, e.g., shadow generation, collision detection, lighting and novel view generation, for to enhance rendering results and visual effects. This work investigates how to reconstruct 3D models from OMs to improve the applications of OMs. Given the camera parameters and silhouette images, some methods [28][49][62][22] have been proposed to recover the 3D model from multi-view images. To the best of our knowledge, the graph cuts based methods [62][22] can produce the better results than other methods[28][49]. However, the graph-cut-based methods do not preserve either concavity-convex features or silhouettes are not preserved. To improve the 3D model, a two phase approach is proposed to deal with these problems in this thesis.

Since binocular vision provides the human depth perception of 3D objects, with stereo vision, the viewer can see where objects are in relation to them with high precision, especially when those objects are moving toward or away from them. To benefit from human binocular visions, this work extends the work on augmented panoramas [25] to augmented stereo panoramas. Once the 3D modes are reconstructed, stereo OMs can be generated from monocular OMs with the help of the 3D model. After producing high-quality OMs, this work develops an interactive system that allows the user to integrate stereo OMs into a stereo panorama, and to interactively browse the augmented stereo panorama. A user can directly browse the stereo OMs of interested by navigating in the augmented stereo panorama with a stereoscopic display. With augmented stereo panoramas, the user can enjoy more persuasive interaction with better depth perception.

## **1.2. Review of Related Work**

Virtual reality systems involve two major classes of technique, i.e., geometry-based and image-based rendering. In geometry-based methods, a complete 3D model of the environment,

including all the objects within the virtual world, is constructed and rendered to simulate the virtual world. Conversely, image-based methods, collections of images taken from different viewpoints of the environment are used to generate novel views of the virtual world. Both approaches have their own advantages and weaknesses. However, image-based methods have become increasingly popular, because they can easily be applied to construct high-quality and photorealistic environments. Shum et al. [56] performed a thorough survey of image-based rendering techniques, and classified the techniques into three categories according to the amount of geometric information used: rendering without geometry[12][29][54], rendering with implicit geometry (i.e. correspondence)[11][19] and rendering with explicit geometry (either with approximate or accurate geometry)[7][51]. Light Field Rendering [29] and Lumigraphs [19] are two famous methods, but their large memory requirements make them impractical for real applications, especially those requiring Internet transmission. Conversely, OM has a smaller storage requirement than those methods. The OM approach can be classified into the first class, rendering without geometry, because it does not need 3D information when rendering OMs. OM has recently become the most popular approach to modeling and rendering the 3D objects, and has been adopted in many applications. This work investigate the techniques for producing high quality OMs including object movie rig calibration, OM segmentation, stereo OMs generation, and 3D reconstruction. The related work is discussed as follows.

As described in Section 1.1, the aim of the rig calibration is to ensure that the pan-, the tilt- and the optical- axes intersect at a common point,  $C_s$ . Since a camera is mounted on the object movie rig, it can be used to perform the calibration, which can be considered as a pose estimation problem. The problem is widely studied in robotic motion and automatic industry [35][42]. A camera can be adopted in a robot system to determine the robot pose from the camera extrinsic parameters, as is well known. Camera calibration is widely discussed. Calibration methods fall into two categories. The first category is self-calibration, in which the

camera parameters are estimated without any reference object, by moving a camera in a static scene [21][39]. However, many parameters need to be estimated, making reliable results hard to obtain. The other calibration methods are estimation with a reference object. Calibration is performed by observing a calibration object whose geometry in 3D space is known with very high precision [58]. In this thesis, the motorized object rig is formulated with the kinematic model. Denavit and Hartenberg [16] developed a notation for assigning orthonormal coordinate frames to a pair of adjacent links in an open kinematic chain. However, parameter jumps occur when two consecutive joint axes change from parallel to almost parallel. Zhuang *et al.* [69] proposed a complete and parametrically continuous (CPC) kinematic model to avoid this situation.

To our knowledge, OM segmentation is currently performed entirely by the artists. These experts mainly manipulate some industrial interactive tools (e.g., magic wand and intelligent scissors from Adobe Photoshop [1]) to remove the backgrounds of each image individually. The work flow does not utilize any information between images captured in neighboring viewing directions, and consequently is very expensive. Unfortunately, background removal in the OM has not been widely investigated, so OM segmentation is an obstacle to the spreading of image-based objects.

Interactive background removal tools have been developed for many years because of their practical importance. Such tools include magic wand [1], intelligent scissors, [40][41][26] Bayesian matting [13], graph-cut-based image segmentation [6][47][31][17][11], and interactive matting based on belief propagation [64]. The color information (e.g., foreground and background color model) and contrast information (e.g., gradient and edge strength) are usually exploited to achieve the goal. The most popular of these methods is probably graph-cut-based image segmentation. The remaining of the image are automatically classified as the foreground or background immediately after a user manually provides foreground and background hard constraints on the image. These approaches are often quite successful for

single-image segmentation, but hard to apply to the OM segmentation due to the endless drudgery of manually specifying hard constraints on each image of the OM individually.

OM background removal is a specific type of video object segmentation. Some automatic methods for video object segmentation have been proposed [44][36], but are not always able to extract the desired video objects. Some researchers have proposed semi-automatic methods that allow user interaction to improve the accuracy of results [20][36][38][67]. Although many approaches have been proposed to deal with video object segmentation, none of these are devoted to object movie segmentation.

Generating stereo OMs from monocular ones is a novel view generation problem, which can be intuitively solved by image morphing [4][48]. Since it does not consider any 3D information, it may produce unexpected effects. View morphing [50] utilizes additional 3D information, such as epipolar geometry and camera parameters, to eliminate the unexpected effects. Moreover, image morphing and view morphing require corresponding features on the original images. However, obtaining good corresponding features is also an open problem. Another approach tries to reconstruct a geometric model of the object according to the consistency with the image information. A calibrated laser projector and a calibrated camera can be used to reconstruct 3D surface [59]. However, laser scanner devices are expensive. Another methods [18][61][37], photometric stereo, can recover high-quality 3D models. To utilize these methods, the lights must be conscientiously and carefully controlled, which is impractical for many applications. Passive methods have been developed for more practical purposes. Laurentini [28] proposed a stable method, called visual hull, to reconstruct a 3D surface using silhouette information. However, his method cannot recover the concavity features of the 3D objects. Seitz and Dyer [49] proposed an improved method that considers voxel colors from different views in order to carve the voxels outside of the true surface. However, the method has a problem in that the surface points are dispersed. Vogiatzis *et al.* [62] recently proposed a graph-cut-based method, called volumetric graph cuts, to solve this

problem. Because graph cuts algorithm prefers *shortest cut*, the volumetric graph cuts has the problems that concavity-convex features and silhouettes cannot be preserved. Tran and Davis [57] tried to solve these problems with silhouette constraints. Their method sets hard constraints on some verified surface voxels. It works well for some cases, but does not solve the problems completely.

### **1.3. Organization of this Thesis**

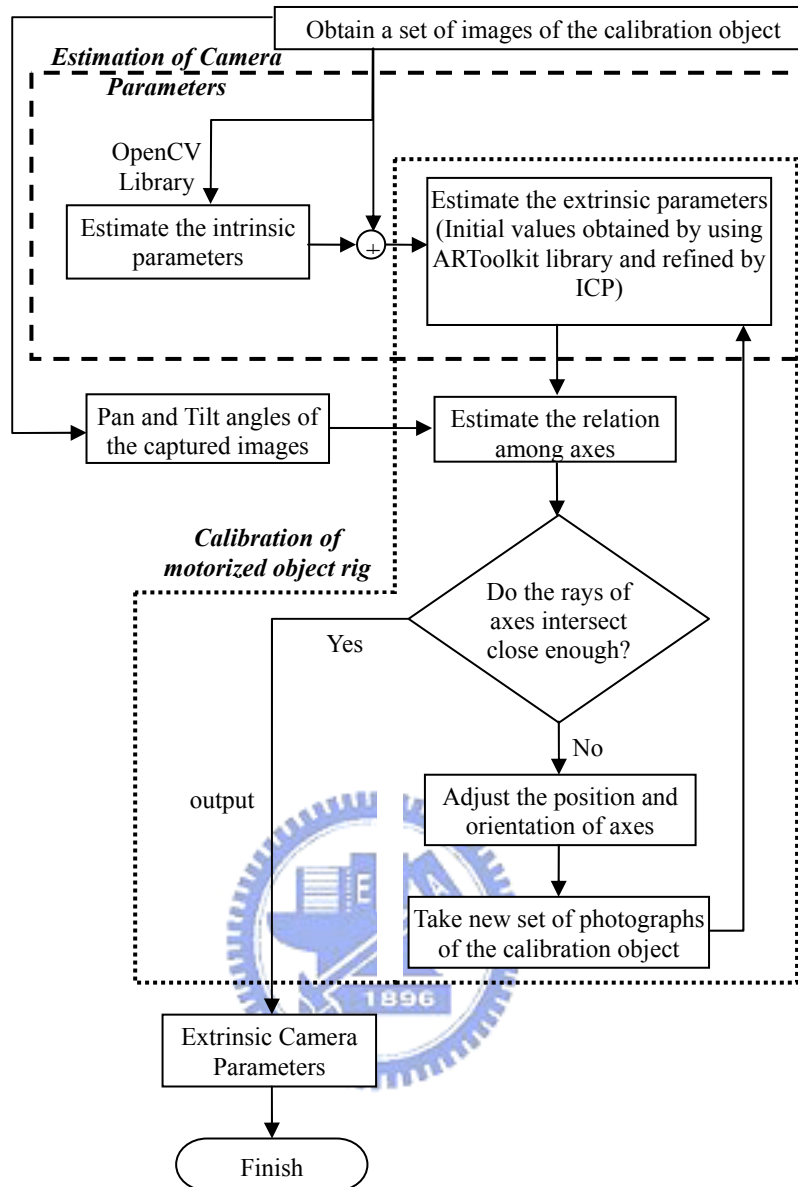
This thesis investigates the techniques of producing high-quality OMs, including object movie rig calibration, OM segmentation, and stereoscopic OMs generation. Chapter 2 presents a calibration method for object movie rigs to help users to acquire high quality OMs, and to obtain camera parameters. Chapter 3 describes two segmentation methods for removing the backgrounds of OMs. The objective of the proposed segmentation method is to minimize the user intervention. The first method utilizes motion vectors to propagate the corrected information to other frames containing segmentation errors. It works well for most cases, but requires much user intervention for some cases due to error motions. Therefore, the second method is proposed to propagate the corrected information efficiently by previously learning shape priors. Chapter 4 presents a novel 3D reconstruction approach to obtain high-quality 3D models from OMs. Chapter 5 describes a novel method, called augmented stereoscopic panoramas, to augment stereo panoramas with stereo OMs. With augmented stereo panoramas, the user can enjoy more persuasive interaction with better depth perception. A conclusion is given in Chapter 7.

# Chapter 2

## Object Rig Calibration

In this chapter, we describe a method for assisting the user to acquire high-quality OMs, and fast obtain the camera parameters of images in OMs. The camera parameters can be used in many applications. In this work, we will use the parameters to perform background removal in Chapter 3, 3D reconstruction and novel view generation in Chapter 4.

Fig. 3 shows the processing flowchart of the proposed calibration method. To calibrate the motorized object rig, we first use the camera mounted on the AutoQTVR to capture some feature points, whose 3D positions are known beforehand. The 2D and 3D positions of the feature points are used to estimate the intrinsic and extrinsic camera parameters. In our experiments, the calibration object, called the physical control cube (PCC) [24], is shown in Fig. 4. With the estimated extrinsic camera parameters, we can reconstruct the kinematic model of the rig. Then, we apply a simple and practical model, completely and parameter continuous (CPC) model [1][69], to formulate the relation among the three axes. Finally, we provide a visual tool showing the axes for users to adjust the motorized object rig. If the intersections of the rays are not close enough, the user can adjust the motorized object rig according to the estimated result, and then the axes will be estimated again. The whole process will be repeated until the intersections of the rays are close enough. After calibration, reliable extrinsic parameters of the camera will be available with the kinematic model.



**Fig. 3. Processing Flowchart of Calibration**

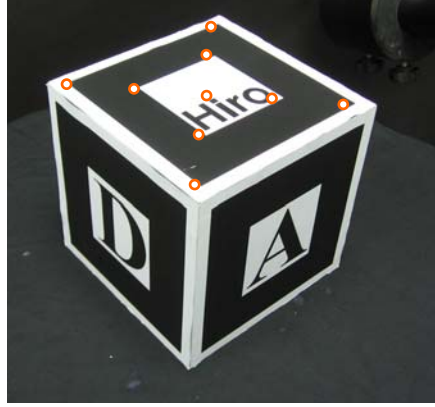
## 2.1. Estimation of Camera Parameters

We adopt the method proposed by Zhang [66] to estimate the intrinsic camera parameters. The method performs camera calibration with at least two images of a known planar pattern captured at different orientations.

On the other hand, we adopt the method presented in [9] and [24] to estimate the extrinsic camera parameters, by first using the method proposed by Kato et al. [27] to obtain a set of



initial extrinsic parameters, and then applying Iterative Closest Point (ICP) principle [3] to refine them.



**Fig. 4. The calibration object, called physical control cube (PCC), and the extracted feature points used to obtain intrinsic camera parameters.**

## 2.2. Completely and Parameter Continuous (CPC) Kinematic Model

A CPC model stands for the completely and parameter continuous kinematic model [69]. A complete model means the model provides enough parameters to express any variation of the actual robot structure, and parameter continuity implies no model singularity by adopting a singularity-free line representation [46].

This model was motivated by the special needs of robot calibration. It is assumed that the robot links are rigid. A CPC kinematic model for a revolution/prismatic joint can be represented as follows (we refer the reader to [69] for detail descriptions):

$${}^i T_{i+1} = \mathbf{Q}_i V_i \quad (1)$$

where  ${}^i T_{i+1}$  denotes the transformation matrix between any two consecutive joint frames, i.e., the  $(i+1)$ -th reference frame to the  $i$ -th reference frame.  $\mathbf{Q}_i$  is the motion matrix defined as follows:

$$\mathbf{Q}_i = \begin{cases} \mathbf{Rot}_z(q_i) & ; \text{for revolute joint} \\ \mathbf{Trans}([0 \ 0 \ q_i]') & ; \text{for prismatic joint} \end{cases} \quad (2)$$

$$q_i = \text{sign} \times q_i' \quad ; \text{sign} \in \{+1, -1\}$$

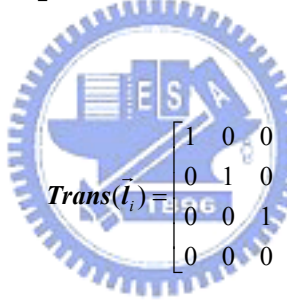
$q_i'$  denotes joint value, which means the rotation angle for a revolution joint, or the amount of displacement for a prismatic joint, and  $V_i$  denotes the constant shape matrix. The shape matrix is a general transformation matrix given by

$$V_i = \begin{bmatrix} \mathbf{r} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = \mathbf{R}_i \mathbf{Rot}_z(\beta_i) \mathbf{Trans}(\vec{l}_i) \quad (3)$$

where

$$\mathbf{R}_i = \begin{bmatrix} 1 - \frac{b_{i,x}^2}{1 + b_{i,z}} & \frac{-b_{i,x}b_{i,y}}{1 + b_{i,z}} & b_{i,x} & 0 \\ \frac{-b_{i,x}b_{i,y}}{1 + b_{i,z}} & 1 - \frac{b_{i,y}^2}{1 + b_{i,z}} & b_{i,y} & 0 \\ -b_{i,x} & -b_{i,y} & b_{i,z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

and



$$\mathbf{Trans}(\vec{l}_i) = \begin{bmatrix} 1 & 0 & 0 & l_{i,x} \\ 0 & 1 & 0 & l_{i,y} \\ 0 & 0 & 1 & l_{i,z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

The rotation matrix  $\mathbf{R}_i$  is used to describe the relative orientation of the two consecutive joint axes, details can be found in Appendix,  $\mathbf{Rot}_z(\beta_i)$  is used to align the  $x$ - and the  $y$ -axes. Notice that the CPC convention requires that any two consecutive joint axes have a nonnegative inner product, i.e.,  $b_{i,z} \geq 0$ . In general, this requirement can be achieved by changing the sign of one of the joint values of consecutive joints. This is because changing the sign of the joint value is equivalent to reversing the joint axis for both revolution and prismatic joints [53].

With the CPC kinematic model [69], the kinematic parameter identification problem can be decomposed into many kinematic parameter calibration sub-problems for each prismatic or revolute joint. Suppose we have a robot with  $n$  joints. The transformation matrix from world reference frame,  $w$ , to end-effector reference frame,  $n$ , can be expressed as follows:

$${}^n T_w = {}^n T_{n-1} \dots {}^0 T_w = \mathbf{Q}_0 V_0 \dots \mathbf{Q}_n V_n \quad (6)$$

### 2.3. Kinematic Calibration Using the CPC Model

In this section, we will introduce how to apply the CPC model to estimate the transformation matrices among the coordinate systems defined on the motorized object rig. As shown in Fig. 5, we define three axes of three different reference frames on the rig. Let  $\bar{z}_c$ ,  $\bar{z}_t$  and  $\bar{z}_p$  denote the z-axes of the camera coordinate system (CCS), the tilt-axis coordinate system (TCS), and the pan-axis coordinate system (PCS), respectively.

For convenience, let the camera be the “end-effector” of the motorized object rig. Thus, we can obtain the corresponding robot pose with the method described in Section 2.1. In general, the orientations of the x- and the y-axes of the coordinate systems need not to be specified in formulating the kinematics of the motorized object rig. Therefore, the redundant parameter  $\beta_i$  in (3) can be set to zero, and the transformation matrix from object coordinate system (OCS) to camera coordinate system (CCS) can be simplified as follows:

$${}^c T_o = {}^c T_t \times {}^t T_p \times {}^p T_o = \mathbf{Q}_0 \times V_0 \times \mathbf{Q}_1 \times V_1 \times \mathbf{Q}_2 \times V_2 \quad (7)$$

where  ${}^b T_a$  denotes the transformation matrix from coordinate system  $a$  to coordinate system  $b$ .

Since the motorized object rig is composed of two revolution joints, the motion matrix  $\mathbf{Q}_0$  is a constant matrix which can be set to identity, whereas  $\mathbf{Q}_1$  and  $\mathbf{Q}_2$  are the rotation matrices about the  $\bar{z}_t$ - and the  $\bar{z}_p$ -axes, respectively. The equations of  $\mathbf{Q}_0$ ,  $\mathbf{Q}_1$  and  $\mathbf{Q}_2$  are given by

$$\begin{aligned} \mathbf{Q}_0 &= \mathbf{I}_{4 \times 4} \\ \mathbf{Q}_1 &= \mathbf{Rot}_z(\theta_t), \text{ where } \theta_t = \text{sign}_t \times q_t' \\ \mathbf{Q}_2 &= \mathbf{Rot}_z(\phi_p), \text{ where } \phi_p = \text{sign}_p \times q_p' \end{aligned} \quad (8)$$

where  $\text{sign}_t$  and  $\text{sign}_p$  are either +1 or -1, and  $q_t'$  and  $q_p'$  are the rotation angle about the tilt and the pan axes, respectively. Substituting (8) into (7), we have

$$\begin{aligned}
{}^cT_o &= V_0 \times \text{Rot}_z(\theta_t) \times V_1 \times \text{Rot}_z(\phi_p) \times V_2 \\
&= R_0 \times \text{Trans}(l_0) \times \text{Rot}_z(\theta_t) \times R_1 \times \text{Trans}(l_1) \times \text{Rot}_z(\phi_p) \times R_2 \times \text{Trans}(l_2) \\
&= \begin{bmatrix} [{}^c r_o(\theta_t, \phi_p)]_{3 \times 3} & [{}^c \vec{t}_o(\theta_t, \phi_p)]_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}
\end{aligned} \tag{9}$$

where  ${}^c r_o$  and  ${}^c \vec{t}_o$  are the rotation matrix and the translation vector of the transformation matrix  ${}^c T_o$ . From (9), we have

$${}^c r_o(\theta_t, \phi_p) = r_0 \times r_z(\theta_t) \times r_1 \times r_z(\phi_p) \times r_2 \tag{10}$$

and

$${}^c \vec{t}_o(\theta_t, \phi_p) = r_0 \times \vec{l}_0 + r_0 \times r_z(\theta_t) \times r_1 \times \vec{l}_1 + r_0 \times r_z(\theta_t) \times r_1 \times r_z(\phi_p) \times r_2 \times \vec{l}_2 \tag{11}$$

In the following subsections, we will show how to solve the parameters,  $r_0$ ,  $\vec{l}_0$ ,  $r_1$ ,  $\vec{l}_1$ ,  $r_2$ ,  $\vec{l}_2$  in (10) and (11).

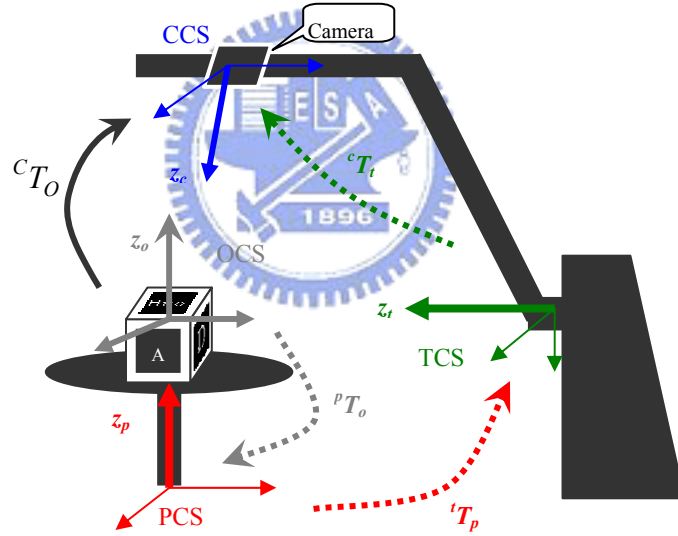


Fig. 5. The schematic of motorized object rig.

### 2.3.1. Rotation Parts

In order to simplify the calibration process, we calibrate one axis at a time. Therefore, when calibrating the tilt-axis, the pan-axis is held still, i.e.,  $\phi_p$  can be regarded as a constant, and thus  $r_1 \times r_z(\phi_p) \times r_2$  becomes a constant term denoted by  $x$ . By substituting  $x$  into (10), we

have

$${}^c \mathbf{r}_o(\theta_i, \phi_p) = \mathbf{r}_0 \times \mathbf{r}_z(\theta_j) \times \mathbf{x} \quad (12)$$

Equation (12) can be rewritten in the following form

$$\mathbf{x} = \mathbf{r}_z(-\theta_j) \mathbf{r}_0^{-1} {}^c \mathbf{r}_o(\theta_i, \phi_p) \quad (13)$$

By maneuver the tilt axis to two different joint values,  $\theta_i$  and  $\theta_j$ , from (12) and (13), we have

$${}^c \mathbf{r}_o(\theta_i, \phi_p) \times {}^c \mathbf{r}_o(\theta_j, \phi_p)^{-1} \times \mathbf{r}_0 = \mathbf{r}_0 \times \mathbf{r}_z(\theta_i - \theta_j) \quad (14)$$

Multiplying  $[0 \ 0 \ 1]^t$  on both sides of (14), we have

$$\begin{aligned} & {}^c \mathbf{r}_o(\theta_i, \phi_p) \times ({}^c \mathbf{r}_o(\theta_j, \phi_p))^{-1} \times \bar{\mathbf{b}}_0 = \bar{\mathbf{b}}_0 \\ \Rightarrow & \left[ {}^c \mathbf{r}_o(\theta_i, \phi_p) \times ({}^c \mathbf{r}_o(\theta_j, \phi_p))^{-1} - \mathbf{I}_{3 \times 3} \right]_{3 \times 3} \times \bar{\mathbf{b}}_0 = \boldsymbol{\varepsilon} \approx \vec{0} \\ \Rightarrow & \mathbf{a} \bar{\mathbf{b}}_0 = \boldsymbol{\varepsilon} \end{aligned} \quad (15)$$

where  $\boldsymbol{\varepsilon}$  denotes the error vector induced by the observation noise, and  $\bar{\mathbf{b}}_0$  can be estimated by minimizing  $\|\boldsymbol{\varepsilon}\|^2$ . It is well known that  $\bar{\mathbf{b}}_0$  is the unit eigenvector of  $\mathbf{a}'\mathbf{a}$  corresponding to the smallest eigenvalue  $\lambda$ . Note that the direction of  $\bar{\mathbf{b}}_0$  has to be determined such that its z-component is positive. By substituting the estimated  $\bar{\mathbf{b}}_0$  into (4), we have the orientation matrix  $R_\theta$ .

The stability of the solution to  $\bar{\mathbf{b}}_0$  can be realized with the following derivation. By substituting (12) to the definition of  $\mathbf{a}$  we have

$$\begin{aligned} \mathbf{a} &= \mathbf{r}_a - \mathbf{I}_{3 \times 3} \\ \mathbf{r}_a &= \mathbf{r}_0 \mathbf{r}_z(\theta_i - \theta_j) \mathbf{r}_0^{-1} \end{aligned} \quad (16)$$

From (16), it is obvious that  $\bar{\mathbf{b}}_0$  is the rotation axis of  $\mathbf{r}_a$ . However, if the difference of the rotation angles  $(\theta_i - \theta_j)$  is close to zero, estimating the rotation axis of  $\mathbf{r}_a$  becomes ill-posed and then the solution to  $\bar{\mathbf{b}}_0$  may not be stable. To avoid this singular configuration, one must make  $(\theta_i - \theta_j)$  as large as possible. This gives a useful guidance to selecting the joint angles for kinematics calibration.

Once  $\mathbf{r}_0$  is available, (14) can be rewritten as follows

$$\begin{aligned} {}^c \mathbf{r}_o(\theta_i, \phi_p) \times ({}^c \mathbf{r}_o(\theta_j, \phi_p))^{-1} \times \mathbf{r}_0 &= \mathbf{r}_0 \times \mathbf{r}_z(\theta_i - \theta_j) \\ &= \mathbf{r}_0 \times \mathbf{r}_z(\text{sign}_t \times \Delta q_i') \end{aligned} \quad (17)$$

The sign parameter  $\text{sign}_t$  can be determined by minimizing the following function

$$\text{sign}_t = \arg \left\{ \min_{\text{sign}_t = +1, -1} \sum_{j=1}^M \left\| \mathbf{r}_0 \times \mathbf{r}_z(\text{sign}_t \times \Delta q_i') - {}^c \mathbf{r}_o(\theta_i, \phi_p) \times ({}^c \mathbf{r}_o(\theta_j, \phi_p))^{-1} \times \mathbf{r}_0 \right\|^2 \right\} \quad (18)$$

Our next step is to solve the rotation matrix  $\mathbf{r}_1$  of  ${}^t T_p$  also using (10). Now that  $\mathbf{r}_0$  is calibrated, the tilt axis can be moved when calibrating  $\mathbf{r}_1$ . For convenience, let us define

$${}^c \tilde{\mathbf{r}}_o(\theta_i, \phi_p) = (\mathbf{r}_0 \times \mathbf{r}_z(\theta_i))^{-1} \times {}^c \mathbf{r}_o(\theta_i, \phi_p) \quad (19)$$

By maneuvering the pan axis to two joint angles, say  $\phi_i$  and  $\phi_j$ , from (10) and (19), we have

$$\begin{aligned} {}^c \tilde{\mathbf{r}}_o(\theta_i, \phi_i) &= \mathbf{r}_1 \times \mathbf{r}_z(\phi_i) \times \mathbf{r}_2 \\ {}^c \tilde{\mathbf{r}}_o(\theta_j, \phi_j) &= \mathbf{r}_1 \times \mathbf{r}_z(\phi_j) \times \mathbf{r}_2 \end{aligned} \quad (20)$$

Equation (20) can be rewritten as follows

$$\begin{aligned} \mathbf{r}_2 &= \mathbf{r}_z(-\phi_i) \times (\mathbf{r}_1)^{-1} \times {}^c \tilde{\mathbf{r}}_o(\theta_i, \phi_i) \\ &= \mathbf{r}_z(-\phi_j) \times (\mathbf{r}_1)^{-1} \times {}^c \tilde{\mathbf{r}}_o(\theta_j, \phi_j) \end{aligned} \quad (21)$$

where yields

$${}^c \tilde{\mathbf{r}}_o(\theta_i, \phi_i) \times ({}^c \tilde{\mathbf{r}}_o(\theta_j, \phi_j))^{-1} \times \mathbf{r}_1 = \mathbf{r}_1 \times \mathbf{r}_z(\phi_i - \phi_j) \quad (22)$$

Multiplying  $[0 \ 0 \ 1]^t$  on both sides of (22), we have

$$\begin{aligned} {}^c \tilde{\mathbf{r}}_o(\theta_i, \phi_i) \times ({}^c \tilde{\mathbf{r}}_o(\theta_j, \phi_j))^{-1} \times \bar{\mathbf{b}}_1 &= \bar{\mathbf{b}}_1, \\ \Rightarrow \left[ {}^c \tilde{\mathbf{r}}_o(\theta_i, \phi_i) \times ({}^c \tilde{\mathbf{r}}_o(\theta_j, \phi_j))^{-1} - \mathbf{I}_{3 \times 3} \right]_{3 \times 3} \times \bar{\mathbf{b}}_1 &= \boldsymbol{\varepsilon} \approx \vec{0}, \\ \Rightarrow \mathbf{a} \bar{\mathbf{b}}_1 &= \boldsymbol{\varepsilon} \end{aligned} \quad (23)$$

Again, by solving an eigenvalue problem, we obtain  $\bar{\mathbf{b}}_1$  which leads to the rotation matrix  $\mathbf{r}_1$ .

The sign parameter  $\text{sign}_p$  for  $\phi_p$ , and also be determined by minimizing an objective function similar to (18).

The final orientation parameter  $\mathbf{r}_2$  can be computed with the following objective function derived from (10).

$$\min_{\mathbf{r}_2} \sum_{i,j} \left\| {}^c \mathbf{r}_o(\theta_i, \phi_j) - \mathbf{r}_0 \times \mathbf{r}_z(\theta_i) \times \mathbf{r}_1 \times \mathbf{r}_z(\phi_j) \times \mathbf{r}_2 \right\|_F^2 \quad \text{subject to} \quad \mathbf{r}_2^t \mathbf{r}_2 = \mathbf{I}_{3 \times 3} \quad \text{and} \quad \det \mathbf{r}_2 = 1 \quad (24)$$

This constrained optimization problem can be solved with a method similar to the one proposed in [3].

### 2.3.2. Translation Parts

By substituting the estimated rotation matrices into (11), we have the following linear equations for the translation parameters:

$${}^c\vec{t}_o = \mathbf{M}_{3 \times 9} [l_{0,x} \quad l_{0,y} \quad 0 \quad l_{1,x} \quad l_{1,y} \quad 0 \quad l_{2,x} \quad l_{2,y} \quad l_{2,z}]^t \quad (25)$$

where  $\mathbf{M}_{3 \times 9} = [\mathbf{r}_0 \quad \mathbf{r}_0 \times \mathbf{r}_z(\theta_1) \times \mathbf{r}_1 \quad \mathbf{r}_0 \times \mathbf{r}_z(\theta_1) \times \mathbf{r}_1 \times \mathbf{r}_z(\phi_1) \times \mathbf{r}_2]$ .

By moving the pan and the tilt joints to different positions, we have an over-determined system of the translation parameters which can be solved using the least square method.

### 2.3.3. Axes Adjustment



After solving the kinematic parameters of the motorized object rig, we can compute its forward kinematic model as follows:

$$\begin{aligned} {}^c\mathbf{T}_o(\theta_t, \phi_p) &= \mathbf{V}_0 \times \mathbf{Q}_1 \times \mathbf{V}_1 \times \mathbf{Q}_2 \times \mathbf{V}_2 \\ &= \mathbf{R}_0 \times \text{Trans}(\vec{l}_0) \times \text{Rot}_z(\theta_t) \times \mathbf{R}_1 \times \text{Trans}(\vec{l}_1) \times \text{Rot}_z(\phi_p) \times \mathbf{R}_2 \times \text{Trans}(\vec{l}_2) \end{aligned} \quad (26)$$

Given the tilt angle,  $\theta_t$ , and the pan angle,  $\phi_p$ , we can use (26) to determine the pose of the camera. Also, the forward kinematic model can be used to find the representations of  $\vec{z}_c$ ,  $\vec{z}_t$  and  $\vec{z}_p$  axes, i.e., the orientation and position of these three axes. First, the transformation matrix from the reference frame of the tilt axis to the CCS can be determined as  ${}^c\mathbf{T}_t = \mathbf{V}_0$ . Thus, the unit direction vector of the tilt axis  $\vec{z}_t$ , denoted by  $\vec{O}_t$ , can be derived as follows

$$\vec{O}_t = {}^c\mathbf{T}_t \times [0 \quad 0 \quad 1 \quad 0]^t = \mathbf{V}_0 \times [0 \quad 0 \quad 1 \quad 0]^t \quad (27)$$

The position of the tilt axis, denoted by  $\vec{P}_t$ , is given by (28)

$$\vec{P}_t = {}^c\mathbf{T}_t \times [0 \quad 0 \quad 0 \quad 1]^t = \mathbf{V}_0 \times [0 \quad 0 \quad 0 \quad 1]^t \quad (28)$$

Similarly, the orientation and position of the pan axis  $\vec{z}_p$ , denoted by  $\vec{O}_p$  and  $\vec{P}_p$ , can be

found to be

$$\vec{O}_p = {}^cT_t \times {}^tT_p [0 \ 0 \ 1 \ 0]^T = V_0 \times \mathbf{Rot}_z \times V_1 \times [0 \ 0 \ 1 \ 0]^T \quad (29)$$

and

$$\vec{P}_p = {}^cT_t \times {}^tT_p [0 \ 0 \ 0 \ 1]^T = V_0 \times \mathbf{Rot}_z \times V_1 \times [0 \ 0 \ 0 \ 1]^T, \quad (30)$$

respectively.

By using equations (27)-(30), the positions and orientations of the three axes of  $\vec{z}_c$ ,  $\vec{z}_t$  and  $\vec{z}_p$  can be evaluated and then can be illustrated as shown in Fig. 6(a). The positions of these three axes can be adjusted to minimize the distance among them. According to our experiences, when the maximum distance among these three axes is smaller than a threshold value of 15 mm, the effect of the miss-alignment of these three axes is negligible.

## 2.4. Experimental Results of Calibration

Our method is implemented on the PC platform with CPU P4-3.0GHz and 1GB RAM and the motorized object rig is AutoQTVR. Fig. 6 shows the result before aligning the three axes of the rig where the estimates of the three axes are shown in Fig. 6(a), and the acquired OM of a toy shark is shown in Fig. 6 (b). The estimation and adjustment process is repeated five times to align the three axes of the rig and the result is shown in Fig. 7. From the frontal view of Fig. 7(d), we show that the tilt axis can be effectively adjusted to be perpendicular to the pan axis and optical axis of camera with our method. Moreover, from the top view of Fig. 7 (d), the intersections of the three axes are close enough. Some images of the OM of the toy shark are shown in Fig. 7(a). After the visual hull of the shark is constructed, shown in Fig. 7(c), the centralization process can be performed, and the resulted OM is shown in Fig. 7(b).

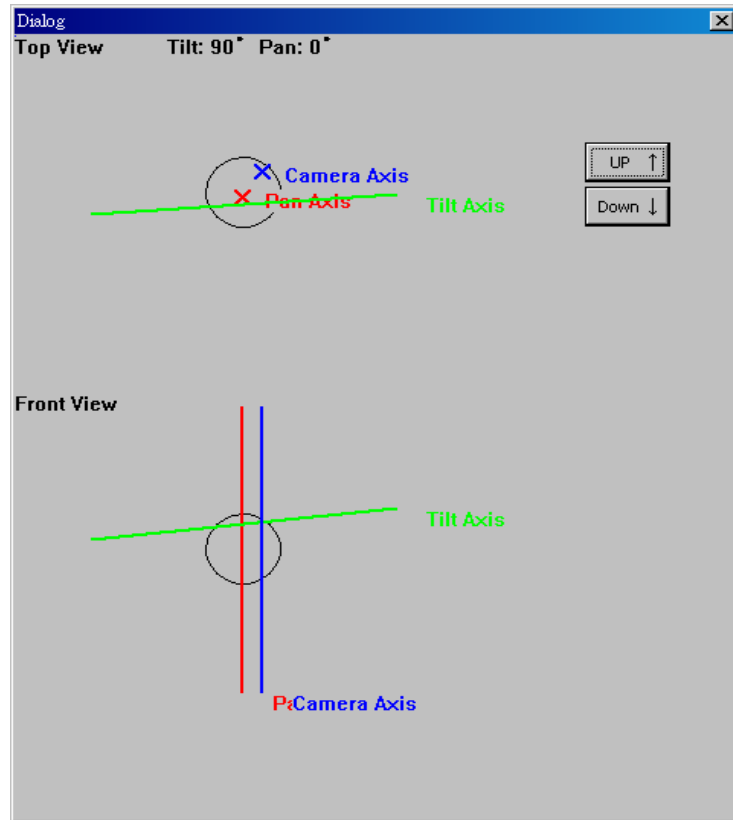
The process time (includes capturing time and computation time) of the calibration process relies on the amounts of the photographs are used. To reduce the process time we have to use small amounts of the photographs. Therefore, we generate some synthetic data to investigate



how many photographs we need and what the relations between the amounts of photographs and the accuracy of the estimated parameters are. We use 3D Studio Max to render the PCC object with known camera parameters. Three sets of synthetic data with different numbers of images (48, 24 and 12 images, respectively) are generated. The 48-image set is obtained with four different tilt angles ( $\theta_t = 90^\circ, 60^\circ, 30^\circ, \text{and } 0^\circ$ ) and twelve different pan angles ( $\phi_p$  is from  $0^\circ$  to  $330^\circ$  with an angle interval of 30 degree). The 24-image set is taken with three different tilt angles ( $90^\circ, 60^\circ, 30^\circ$ ) and eight different pan angles, and the 12-image set are taken with three different tilt angles ( $90^\circ, 60^\circ, 30^\circ$ ) and four different pan angles. In the experiments, our method is applied to the three data sets to estimate the camera parameters and the estimated parameters are compared with the ground truth. To quantify the error of the estimated camera parameters, some 3D points are randomly selected to calculate their 2D positions using the ground truth and the estimated camera parameters, and then the Euclidean distance between the ground-truth position and the estimated position is calculated. The results are shown in Table 1. The process time includes shooting process and camera parameter estimation. The error is mean Euclidean distance. From our experiments, we found that only 12 images are enough to obtain a set of highly accurate parameters. That is, we only need to take 12 pictures at each adjustment-calibration process, and the processing time needed, including capturing and processing, is about 7 minutes.

**Table 1. Processing time and accuracy of the calibration processing**

The Number of Images	Processing Time	Euclidean Distance
48	About 25 min	1.62
24	About 8 min	1.63
12	About 7 min	1.63



(a)



(b)

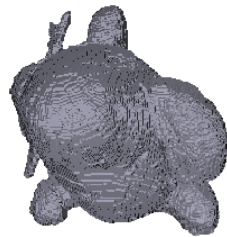
**Fig. 6. The OM of the toy shark before calibration. (a) shows the estimated relation among 3 axes, and (b) shows the OM of the toy shark. The cross markers indicate the center of images.**



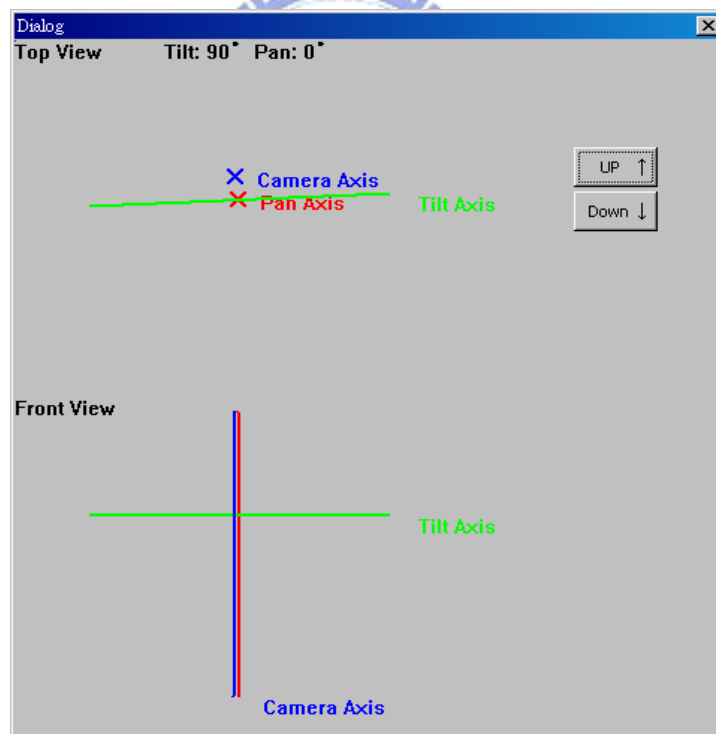
(a)



(b)



(c)



(d)

**Fig. 7. The result of the toy shark experient. (a) shows some images of the OM of the toy shark after calibration, while (b) shows that after centralization. (c) shows the Visual Hull of the shark, and (d) shows the estimated axes after calibration.**

# Chapter 3

## Background Removal

In order to reduce the user intervention, the basic idea to develop the OM background removal system is follows. First, an automatic segmentation will be applied to obtain initial segmentation results. If some results are not satisfied, the user can correct one of them though the provided user interface. After modification, the corrected result can be automatically propagated to the other images, and used to refine the segmentation results.

In this work, we treat the segmentation problem as a labeling problem. We assign every pixel a label for a given OM. These labels are  $F$  (Foreground),  $B$  (Background), and  $U$  (Uncertain), and the image used to record the labels is called *trimap*. OM notations to which we will refer are: is defined as follows. Let  $I_{\theta,\phi}$  denote the image taken at pan angle  $\theta$  and tilt angle  $\phi$ . An equi-tilt set  $O_\phi$  is defined as a subset of the images in an OM captured at the same tilt angle  $\phi$ , i.e.,

$$O_\phi = \{I_{\theta,\phi} \mid 0 \leq \theta \leq 2\pi\} \quad (31)$$

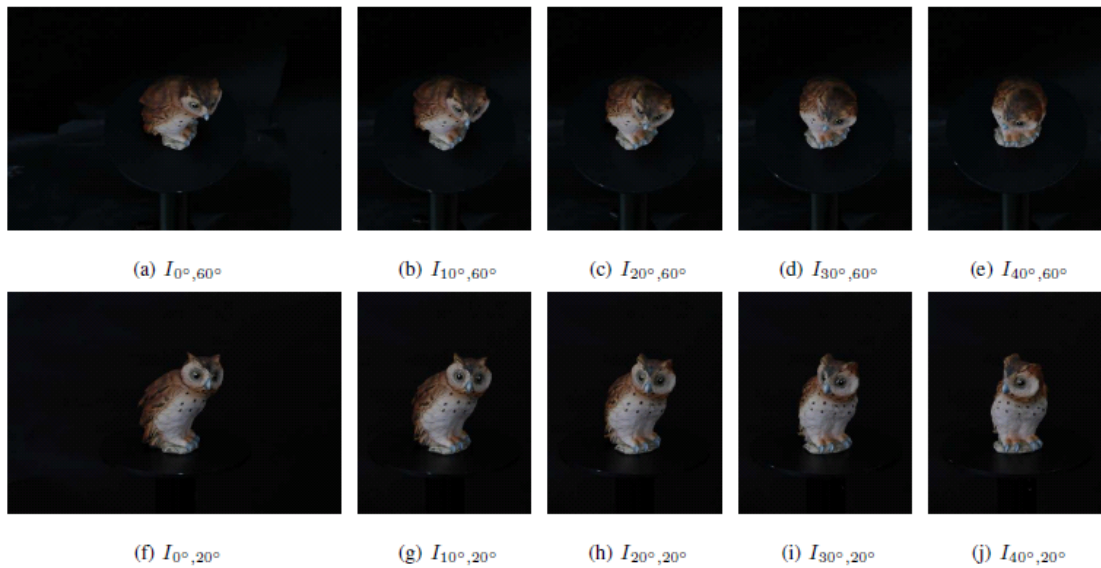
Finally, an OM  $O$  is defined as

$$\begin{aligned} O &= \{O_\phi \mid -\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}\} \\ &= \{I_{\theta,\phi} \mid 0 \leq \theta \leq 2\pi, -\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}\} \end{aligned} \quad (32)$$

Fig. 8 shows a portion of the two equi-tilt sets that are contained in the OM of the pottery owl.

Based on the idea, the flowchart of our system is shown in Fig. 9. It includes three main stages: initial labeling, label updating, and alpha estimation. For initial labeling, we extract reliable foreground and background pixels based on some OM characteristics. The details are described in section 3.1. For label updating,  $U$  pixels are updated using spatial and temporal coherence based on the extracted foreground and background. After label updating,

intermediate segmentation may contain some misclassified pixels. To correctly classify these pixels, user modification can be done at this point through the provided user interface. After modification, the label updating stage is again used to obtain more accurate results. After user intervention, most pixels are classified as foreground or background except the pixels that may be composites of the foreground and background. For alpha estimation, the method proposed by Chuang *et al.* [13] can be applied to calculate the alpha value for each  $U$  pixel. Using the alpha value, we can product a smooth contour blending when we integrate OM into a new background.

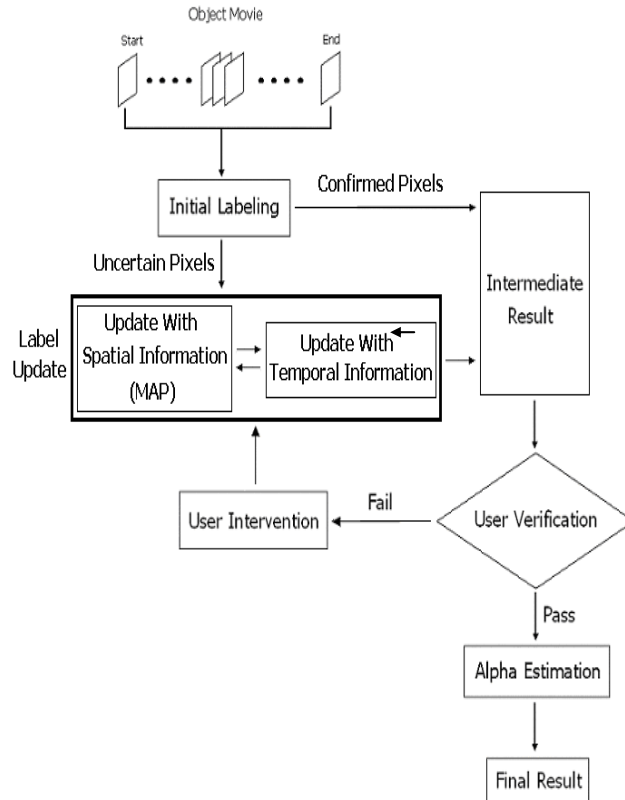


**Fig. 8. Part of the two different equi-tilt sets before applying the OM segmentation method. Except for leftmost two images in the figure, the remainder of the images in this thesis are cropped in order to show more examples.**

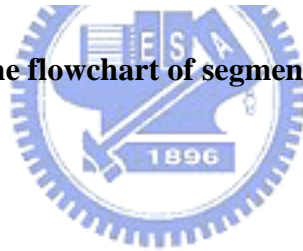
In this thesis, two approaches are proposed to propagate the corrected information. The first method utilizes motion vectors to propagate the corrected information to other frames that some segmentation errors occur. The details are described in section 3.2. The method works well for most cases, but requires more user intervention for some cases due to error motions.

The situation could be even worse for the first method. To compute the motion field, the motion estimator usually assumes that the sampling rate of the video camera is high enough to minimize the frame-to-frame motion. However, to keep the data size and cost reasonable, the

sampling rate of the OM is generally low. A popular alternative approach is to interpolate the dense motion field from a set of image correspondences. Because the difference between the images is caused fully by the changes in the 3D viewpoints, the perspective distortion makes the correspondence problem extremely difficult. In our experience, generating enough correspondences is still a problem, even with some popular tools, e.g., such as the *KLT* feature tracker [52] or the *SIFT* features [34]. Additionally, to filter out the potentially false correspondences, the class of the transformation, e.g., translational, affine, or a more complex -transformation, need to be considered so that the images can be aligned as accurately as possible, and a robust estimation can be performed. The translational motion is often the prominent transformation in many of the video source used to demonstrate the information propagation scheme. However, the nature of the transformation existed in the OM cannot be easily modeled without 3D object information. In practice, without some user intervention or knowledge of the 3D information, a usable motion field between any possible pair of the neighboring images in the OM is quite hard to compute. Therefore, the second approach is proposed for efficiently propagate the corrected information by learning shape priors. The details are described in section 3.3.



**Fig. 9. The flowchart of segmentation method.**



### 3.1. Initial Labeling

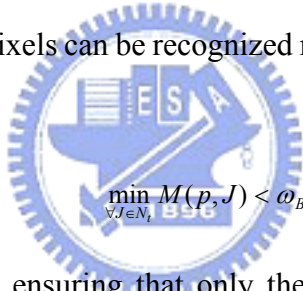
From our observation, OM has three basic characteristics which can help the method generate the trimap:

1. When an equi-tilt set of the OM is captured, a large proportion of the background scene is static.
2. Only one interesting object is presented in every image of the OM.
3. The foreground and background color distributions are distinct in most cases.

The trimap labeling method comprises *B*-labeling and *F*-labeling. Each equi-tilt set of the OM is processed individually by the trimap labeling method. Given an equi-tilt set  $O_\phi$ , the

trimap of each image in  $O_\phi$  is initialized to  $U$ . During the  $B$ -labeling, pixels are examined to be labeled as  $B$  based on the color difference. During the  $F$ -labeling, all pixels that are still labeled as  $U$  are examined to be labeled  $F$  based on the background model.

- 1.)  $B$ -labeling: By the first characteristic, if the color of a pixel varies barely throughout the equi-tilt set  $O_\phi$ , then the pixel should be the background and labeled  $B$ . Since an equi-tilt set  $O_\phi$  can be treated as a short video sequence, a pixel  $B$  is labeled by examining its color difference compared with the corresponding pixels in both directions of the video sequence. Let  $p = [u \ v]^T$  denote a pixel of a video frame  $I_t$ , i.e., an image of the equi-tilt set  $O_\phi$ . Let  $I_t(p)$  be the color of pixel  $p$  in the frame  $I_t$ . Let  $N_t$  be the set of neighboring frames of  $I_t$ . To relieve the camera noises and consider the color changes caused by the lighting, a measure based on the block color difference with respect to the mean is used such that the background pixels can be recognized reliably. Each pixel  $p$  in  $I_t$  is labeled  $B$  if

$$\min_{J \in N_t} M(p, J) < \omega_B \quad (33)$$


Here,  $\omega_B$  is the threshold ensuring that only the pixel with a small color variation is labeled  $B$ . The measure  $M(p, J)$  is defined as follows

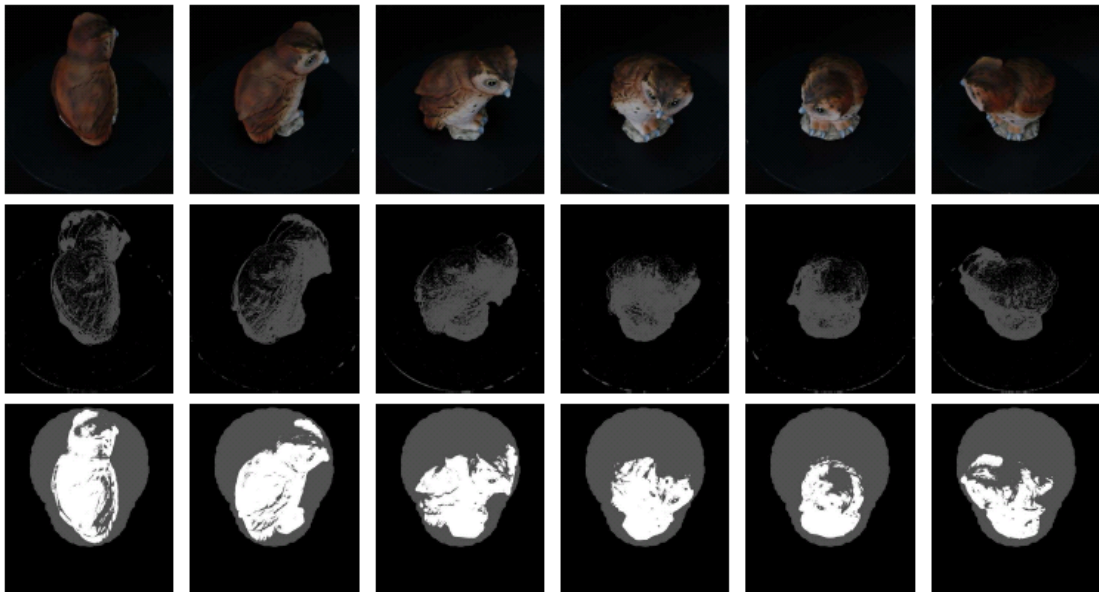
$$M(p, J) = \frac{1}{|W|} \sum_{x \in W} |(I_t(x) - \bar{I}_t(p)) - (J(x) - \bar{J}(p))|_2 \quad (34)$$

where  $W$  is a small window centered at the pixel  $p$ ,  $|W|$  denotes the number of pixels in the window,  $\bar{I}_t(p)$  is the mean color of the window  $W$  on the image  $I_t$ ,  $\bar{J}(p)$  is the mean color of the window  $W$  on the image  $J$ , and  $|\cdot|_2$  denotes the  $L_2$ -norm. For each pixel on a frame  $I_t$ , the measure is examined in both directions of the video sequence, i.e., backward and forward. Figure 4 shows a portion of the equi-tilt set after applying the above procedure, where  $N_t = \{I_{t-1}, I_{t+1}\}$ .

Most of the  $B$  pixels are exactly within the background as shown in Fig. 10, but there are exceptions, such as the pixels of a uniform colored patch of the object. The concept of



label consistency is then introduced. If the pixels at the same image position do not have the same label throughout the whole sequence, then they are re-labeled as  $U$ . Finally, by the second characteristic of the OM, mathematical morphology is applied to filter out the remained noises such that only one  $U$  region exists, surrounded by the  $B$  region. Notably, all the images in  $O_\phi$  until now had the same trimap consisting only the  $B$  and  $U$  labels. Fig. 11 shows an example of such a global background mask.



**Fig. 10** The top row shows a portion of the input image sequence taken from an equi-tilt set of the pottery owl OM. For all the images in the middle and bottom rows, the black pixels correspond to the classified background regions. The foreground regions are colored white, and the unknown regions are colored gray. The middle row shows the corresponding result during the  $B$ -labeling for each image. Notably, to filter out the incorrectly classified pixels and obtain the global background mask used during  $F$ -labeling, label consistency and mathematical morphology are used as shown in Fig. 11. Finally, the bottom rows shows the generated trimap for each image that is used to activate the graph cut image segmentation.

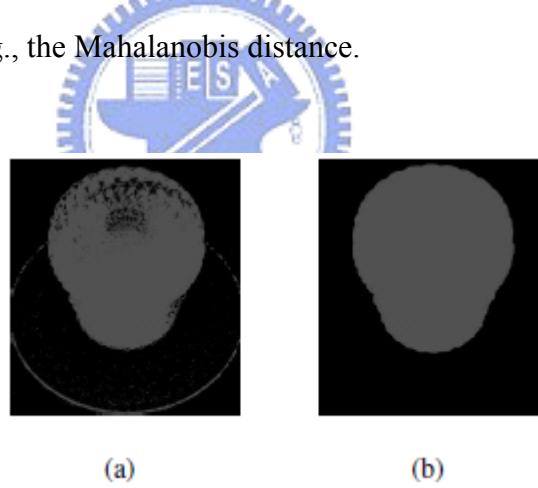
- 2.) *F-labeling*: By the third characteristic of the OM, each pixel whose color differs widely from the background model can be labeled  $F$ . To learn the background model of a given image, the  $B$  pixels that are reasonably close to the boundary between the  $B$  and  $U$  regions

are collected and clustered by using K-means. Let  $\mu_{\theta,\phi}^i$  denote the mean color of the  $i^{\text{th}}$  cluster for image  $I_{\theta,\phi}$ . Each pixel  $p$  with the label  $U$  in the image  $I_{\theta,\phi}$  is examined and labeled  $F$  if

$$\min_{\forall i} |I_{\theta,\phi}(p) - \mu_{\theta,\phi}^i|_2 < \omega_F \quad (35)$$

where  $\omega_F$  is a strict threshold to ensure that only the pixels that differ widely from the background model are labeled  $F$ .

Fig. 10 shows the result of the trimap labeling. The trimap of each image is used to activate the graph cut image segmentation. Notably, in this OM segmentation problem, the variation between the colors drawn from the background and foreground is strong. Thus, for a given pixel, to determine the similarity of its color to the foreground or background model in the graph cut image segmentation, the distance measure should consider the statistical variation, e.g., the Mahalanobis distance.



**Fig. 11. (a) The result including the label consistency concept is included; (b) The global background mask obtained by applying the mathematical morphology on (a).**

### 3.2. Label Updating with Motion Vectors

The label updating stage consists of spatial (intra-frame) updating and temporal (inter-frame) updating. The spatial followed by the temporal updating process will be iterated until it is stable. That is, label updating will repeat until there is no label change.

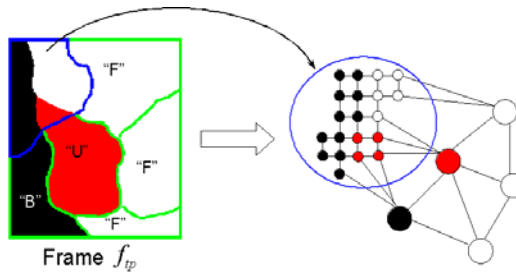
### 3.2.1. Spatial updating

We construct a graph for every frame of an OM and apply the *Maximum a Posteriori* (MAP) method to achieve an optimal labeling set of the graph. The vertex on which we perform MAP labeling represents either a watershed region in which all pixels have the same label or a single pixel. An example of graph construction is shown in Fig. 12. We first apply watershed segmentation [60] on every frame of an OM. If all pixels of a watershed region have the same label, the watershed region is represented by a vertex. Otherwise the watershed region is split into pixels, each of which is represented by a vertex in the graph. There is an edge between two nodes only if they neighbor each other in the image.

We then define a label field  $L = \{l_v | l_v \in [1..N], v \in \text{Vertex}\}$  on the graph. Given measurement  $M = \{\theta_v | v \in \text{Vertex}\}$ , we estimate labeling field  $L$  by maximizing the a posteriori probability. Using the Bayes rule, the a posteriori probability density function can be expressed as

$$P(L|M) \propto P(M|L) \cdot P(L) \quad (36)$$

To maximize the posteriori probability  $P(L|M)$  is to maximize the observation term  $P(M|L)$  and the prior term  $P(L)$ .



**Fig. 12. 2D spatial graph construction example.**

We model the observation probability as a Gaussian distribution. Because a 3D object may contain many colors, for a  $U$  vertex, we find a fixed number of its neighboring  $F$  vertices and perform color quantization [43] on this set of vertices. After color quantization, the set of vertices are separated into several clusters.

Because a vertex of a graph can represent either a single pixel or a watershed region, we weight the contribution of the  $F$  vertices by its area and distance from the  $U$  vertex. That is, when evaluating a vertex's observation term, we weight the collected neighboring vertices by their number of pixels and by the distance between the  $U$  vertex and the collected neighboring vertices. The evaluation equation of the observation term is:

$$P(I_i | \{l_i = k\} \cup \bar{L}_i) = \begin{cases} \frac{1}{K_F} \exp\left[-\|I_i - \mu_i^c("F_c")\|_{\Sigma_{F_c}^{-1}}^2\right] & ; k = "F_c", c = 1, 2, \dots, C \\ \frac{1}{K_B} \exp\left[-\|I_i - \mu_i("B")\|_{\Sigma_B^{-1}}^2\right] & ; k = "B" \\ \frac{1}{K_u} & ; k = "U" \end{cases} \quad (37)$$

$$\mu_i("k") = \frac{\sum_{j \in N_i, l_j = "k"} \omega_{ij} I_j}{\sum_{j \in N_i, l_j = "k"} \omega_{ij}}$$

$$\omega_{ij} = Area_j * e^{-distance_{ij}}, \quad \bar{L}_i = L - \{l_i\},$$

where  $C$  is the number of clusters of foreground after color quantization,  $1/K_F$ ,  $1/K_B$ , and  $1/K_U$  are normalizing constants,  $I_i$  is the color vector of vertex  $I$ , and  $distance_{ij}$  is the mean distance between two regions corresponding to vertex  $i$  and  $j$ .

We model the prior term as a Gibbs distribution [30]. A Gibbs distribution takes the form.

$$P(L) = \frac{1}{Z} \exp(-E(L)/T) \quad (38)$$

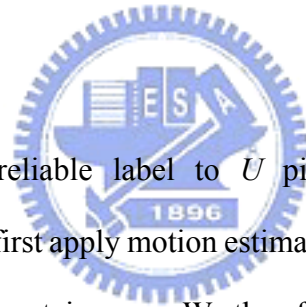
where  $Z$  is a normalizing factor which is a constant for all the configurations, so there is no need to compute the value of  $Z$ ;  $T$  is a constant called the temperature which is normally assigned to be 1;  $E(L)$  is the *priori energy* given in (49). For a  $U$  vertex  $i$ , we weight its 2-site clique potentials by the area of its neighboring vertices when evaluating its prior term.

$$E(L) = \sum_{i \in V} \left[ \frac{\sum_{(i,j) \in C_2} \omega_j V_2(l_i, l_j)}{\sum_{(i,j) \in C_2} \omega_j} \right], \quad \omega_j = Area_j \quad (39)$$

where  $C_2$  is the collection of 2-site cliques, and  $V_2(l_i, l_j)$  is the clique potential function of

clique  $(i,j)$  and is dependent on the configuration  $(l_i,l_j)$ . Let  $A_{l_i,l_j}$  be the constant value to be assigned to  $V_2(l_i,l_j)$ . Here we consider  $A_{l_i,l_j}$  to be  $A_{FF} < A_{BB} < A_{UU} < A_{UF} = A_{UB} = A_{BU} = A_{FU} < < A_{BF} = A_{FB}$ . By giving  $A_{FF}$  a smaller value than the others, the cliques with configuration  $(F,F)$  will have a higher occurrence probability. That is, the foreground will expand more rapidly toward the border than the background, because most pixels should belong to the foreground after previous processes.  $A_{UU}$  have a lower energy value than  $A_{UF}$ ,  $A_{UB}$ ,  $A_{BU}$ , and  $A_{FU}$ , because the probability of having  $(U,U)$  is higher than those of having the other four configurations.  $(B,F)$  and  $(F,B)$  are the least preferred configurations, because it represents sharp boundary. In most cases, the boundary pixels are composites of foreground and background, so these pixels should remain undecided until the alpha estimation stage.

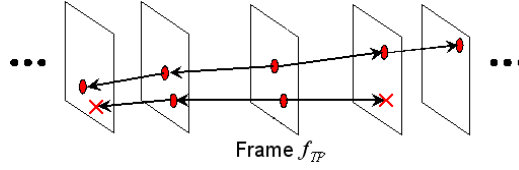
### 3.2.2. Temporal Updating



This process assigns a reliable label to  $U$  pixels based on temporal information (inter-image information). We first apply motion estimation to every  $U$  pixel by block matching on neighboring frames within a certain range. We then filter out unreliable motion vectors since the motion estimation for those vectors may be erroneous. For every  $U$  pixel, we find its corresponding pixel in each frame, and call this series of pixels a *worm*. A worm of an  $U$  pixel  $i$  consists of pixels that can be reached from  $i$  through estimated motion vectors. As shown in Fig. 13 the worm  $w_i$  is a series of pixels stemming from  $i$ , and the worm  $w_j$  stems from  $j$ . The two ends of a worm will terminate at pixels that don't have reliable motion vectors. As shown in Fig. 13, the length of the worm  $w_j$  is two.

After the worm of  $U$  pixel  $i$  is constructed, we will assign a label to  $i$  based on the label information of its worm. To assign a label to  $i$ , there are three conditions. One, if all labels of the worm are  $U$ , the label of the pixel  $i$  remains  $U$ . Two, if a worm contains both F and B labels (which is a contradiction), then the label of the pixel  $i$  remains  $U$ . Three, if worm labels are

either  $U$  and  $F$  or  $U$  and  $B$ , then we will change  $U$  pixel  $i$ 's label to either  $F$  or  $B$ .



**Fig. 13. The worms of the uncertain pixels  $i_1$  and  $i_2$**

### 3.3. Label Updating with Shape Priors

The proposed approach aims to let every single image segmentation, rather than only those in neighboring viewing directions, benefit from the segmentation results of the images captured in all possible viewing directions. Besides the problem of computing a reliable motion field, one more shortcoming of the information propagation scheme is that the information can only be propagated from neighboring images, because of the error accumulation problem which is hard to avoid when computing the motion field. The proposed approach overcomes this limitation with the help of the reconstructed 3D model, and some preliminary results have been shown in [29]. A quality motion field can also be computed between any pair of the neighboring images after the 3D object is reconstructed.

Fig. 14 illustrates the process flowchart of the proposed approach. Given an OM with the intrinsic and extrinsic parameters of the camera calibrated for all views [4], the proposed method starts with the automatic initial segmentation, which aims to provide some tentative segmentation results based entirely on the color and contrast information. To take the shape prior into account, the user is required to select a subset of acceptably segmented images. The 3D shape is then generated from these selected images. The reconstructed 3D model can be used to infer the shape of the object in any given 3D configuration of a view. For each image of

the OM, a quality segmentation result can be computed by incorporating the inference of shape of the object into the segmentation algorithm, along with the original color and contrast information. The main advantage of the approach is that each time the user gives some intervention to a part of the OM, the influence can be propagated to the whole OM segmentation problem. Thus, if the user is still not satisfied with the OM segmentation result, then interactive background removal tools can be utilized to refine some problematic images. This procedure can be repeated in order to refine the OM segmentation result further.

Notably, to apply our method, camera parameters are indeed required for 3D reconstruction. The reconstructed 3D model may be inaccurate due to calibration error, which may then introduce errors when the shape priors are extracted from the inaccurate 3D model. However, the final 2D image segmentation results are not very sensitive to small errors in the shape priors, as long as the errors are within a few pixels. In our experiments, we used the method described in chapter 2 to estimate the camera parameters, and the calibration errors are less than 3 pixels in general. Since the shape prior of the object is expressed by using a volumetric representation in this approach, a reliable 3D reconstruction method is desired. The volumetric graph cuts proposed by Vogiatzis et al. [30] are adopted in this case.

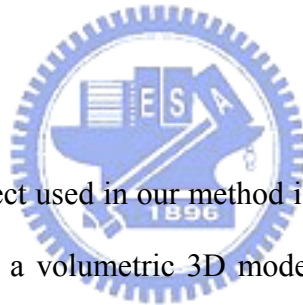
### **3.3.1. Spatial updating**

The background removal tool proposed by Boykov and Jolly [11] on which our OM segmentation method is built, is described here. Graph cut image segmentation requires the user to interactively mark some pixels as being inside the foreground objects, and others as a part of the background scene. The two disjoint sets of marked pixels serve as the foreground and background hard constraints, respectively. All the other pixels are considered to be unknown, and then they can be classified into the foreground or background by Markov random field (MRF) optimization. Each candidate segmentation is associated with an energy that considers

the following properties. For each foreground pixel of the candidate segmentation, a penalty is given to reflect on whether its color fits into the foreground model. The model can be learned from the foreground pixels marked by the user.

A penalty is similarly given to each background pixel based on the similarity of its color to the background model. Next, the algorithm penalizes every pair of the adjacent pixels where one is inside the foreground and the other is outside according to how likely a boundary is probable to appear between the adjacent pixels. A small penalty is generally given for the adjacent pixels that have a large difference in their colors. The algorithm determines the optimal segmentation by finding the global minimum among all segmentations that meet the specified hard constraints.

### 3.3.2. Temporal Updating

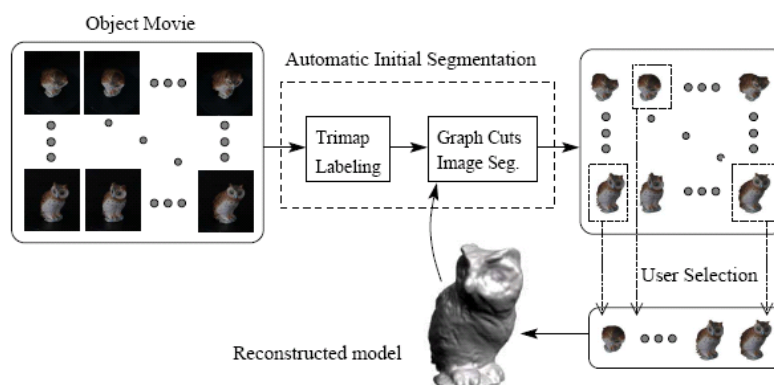


The shape prior of the object used in our method is expressed by a volumetric 3D model. The problem of reconstructing a volumetric 3D model from multiple calibrated images has been widely investigated in the last decade. Besides the camera calibration, these algorithms also require the silhouettes of the object in all the images. However, obtaining these silhouettes is exactly what we want to solve. The proposed method avoids this contradiction based on the observation that a subset of the OM is sufficient for the 3D reconstruction. Sufficient number of images that have satisfactory segmentations after the automatic initial segmentation. The user is then required to select a subset of acceptably segmented images to accomplish the 3D reconstruction. Vogiatzis et al. recently proposed a graph cut-based method, called volumetric graph cuts [62], to solve the reconstruction problem. This work adopts Vogiatzis *et al.*'s algorithm to learn the shape prior.

Besides the color and contrast information, a good inference of the shape available for any possible view of the object can provide the favorable information on solving the OM



segmentation problem. Because the camera is calibrated for all images in the OM, a good shape prior of the object can be obtained to rectify the segmentation errors in some problematic views by projecting the reconstructed 3D model. Fig. 15 illustrates the idea of the segmentation refinement. For each image with the discontented segmentation result, the projection of the reconstructed 3D model under the same viewpoint is integrated to serve as the foreground hard constraints, together with the previously generated trimap. The graph cut image segmentation is then applied again to obtain the satisfied segmentation result. Significantly, the photo-consistent reconstruction is mandatory to obtain a good shape. The visual hull can only represent an approximate geometry of the object, and tends to be fatter than the real object, regardless of whether the object is convex or concave. This characteristic of the visual hull could be more obvious when the number of images available to be used is limited. Consequently, the projection of the visual hull might introduce unreliable foreground hard constraints in the segmentation refinement. Fig. 15 also illustrates the problem when photo-consistent reconstruction is not used. Here, directly using the projection of the base surface on  $C_3$  and  $C_5$  imposes incorrect foreground hard constraints, and lead to failed segmentation results.



**Fig. 14. The process flowchart of the new system.**

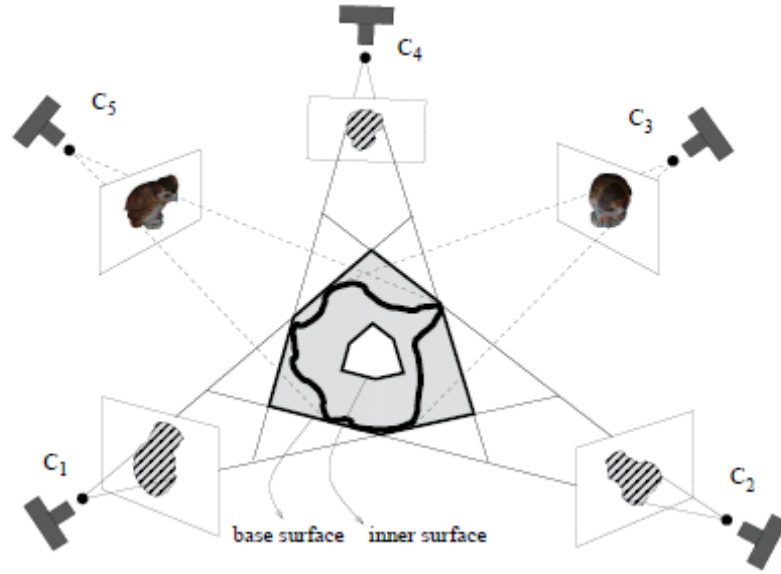


Fig. 15.  $C_1$ ,  $C_2$ , and  $C_4$  denote the views adopted to build the visual hull. Notably, the true surface of the object is assumed to be between the base and inner surfaces. Although the segmentation results of  $C_3$  and  $C_5$  are poor, they can be improved by incorporating the projection of the reconstructed model into the graph cut image segmentation algorithm.



### 3.4. Experimental Results of Background Removal

Each OM consists of 360 images from 10 equi-tilt sets  $O_{\frac{0\pi}{18}}, O_{\frac{1\pi}{18}}, \dots, O_{\frac{9\pi}{18}}$ . Each equi-tilt set had 36 images captured equally from pan angle  $0^\circ$  to pan angle  $2\pi$  with the image size 3000x2000 pixels. In all the experiments, because the lens distortion occurred in an area far from the center of the image, and the object was mostly located in the center of the image, each image is cropped to about 1000x1000 pixels before evaluating our OM segmentation method. The experiments were performed on a 2.4GHz Pentium 4 desktop with 1 GB memory. The remainder of the experiments section is arranged as follows. First, the results of the automatic initial segmentation are shown. The reconstructed 3D models, from which the shape prior can be extracted, are then shown. Following this, we demonstrate how to rectify the segmentation errors existing in some problematic images using the obtained shape prior.

### 3.4.1. Initial Segmentation Results

To reduce the response time to the user, the automatic initial segmentation can be carried out on the downsized OM. After obtaining the initial segmentation results, the set of segmented images chosen by the user was then resized to the original image size to generate the base surface used in the 3D reconstruction.

To help the user select the segmented images that are successful, the segmented images were sorted according to their energies after the graph cut image segmentation, i.e., a low energy means a potentially good segmentation. When finding the optimal surface within the base surface, besides the selected images, all the other images in the OM can also be considered when computing the photo-consistency scores. Additionally, the automatic initial segmentation does not need to be applied on all the equi-tilt sets of the OM. Experimental results shows that about 3 or 4 equi-tilt sets captured in the relatively small tilt angles can yield enough satisfactory segmentation results for the 3D reconstruction job.

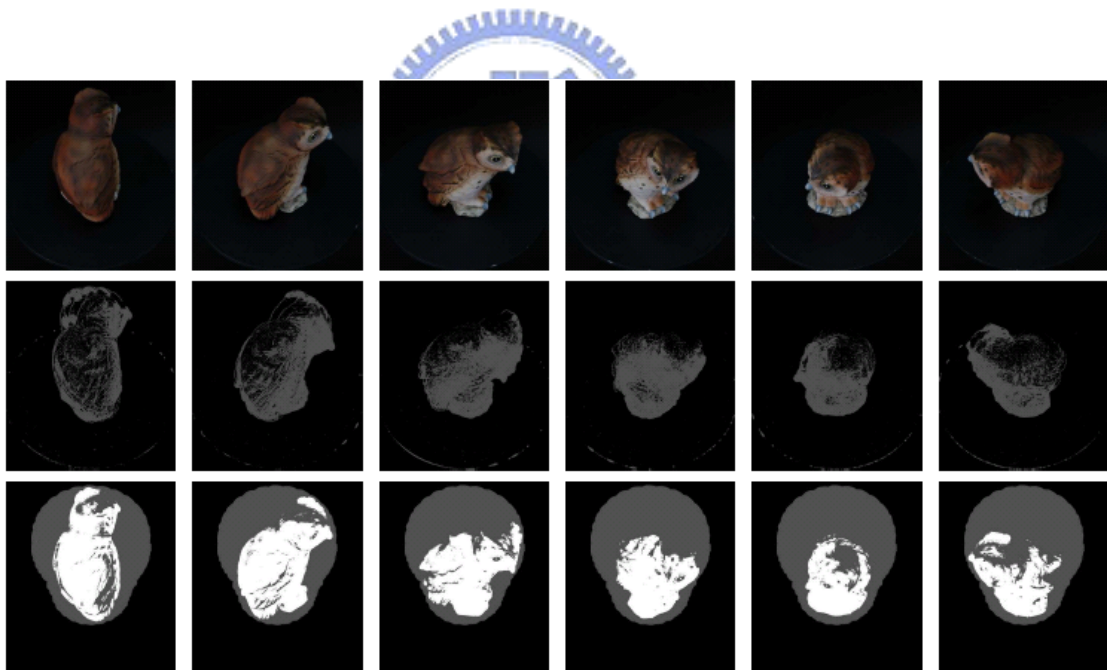
First, the automatic initial segmentation was applied to the pottery owl OM. Fig. 17 shows the results of the automatic initial segmentation for the pottery owl with respect to the image sequence as shown in Fig 16. Because of the low contrast boundaries of the pottery owl, the black screen and the shadows caused by the lighting, automatic foreground extraction of the whole OM could be a demanding challenge when applying methods based on color and contrast information alone. However, since different geometries, textures, and lightings are presented in different viewing directions of the pottery owl, the foreground can be automatically separated from the background in some images. For the other problematic images, the segmentation errors can be rectified in the next run by incorporating the learned shape prior into the segmentation process. To learn the shape prior, 36 segmented images were selected for the 3D reconstruction of the pottery owl. Fig. 18 shows the results of the automatic initial segmentation for a portion of the equi-tilt set in the toy house OM. Since the tower of the house had mixed

together with the black screen in some viewing directions in the photo studio arranged for capturing the OM, the tower was difficult to separate from the background without the shape knowledge learnt from the other successfully segmented views. To rectify the segmentation errors, 48 segmented images were selected for the 3D reconstruction of the toy house.

### 3.4.2. Rectification of Segmentation Errors

This section describes the refinement of the segmentation results with the learned shape prior. Fig. 19 shows the rectification of the segmentation errors for each problematic image in Fig. 17 and Fig. 18, which are denoted by the red circles. Since the projection of the reconstructed model can provide a good inference of the shape for the object in each calibrated view, a robust segmentation result can be achieved even when the boundary of the object goes through the low-contrast and shadowed regions where the foreground and background color distributions can not be effectively separated. On each trimap that includes the projection of the reconstructed model, the learned shape prior provides much information about the segmentation problem that the original foreground hard constraints do not reveal. Fig 20 indicates that the background removal of the pottery cat OM increases the benefit of using shape priors. Because the foreground and background color distributions are entirely mixed with each other in some difficult regions, the images are quite difficult to segment by using only the color and contrast information. Moreover, for such a troublesome OM, segmentation errors generally appear in several consecutive images at the same time. Consequently, propagating successful segmentation results by using the motion field becomes quite unstable due to the error accumulation problem when estimating the motion field. For such a difficult object, the automatic initial segmentation might not provide enough successful segmentation results for the 3D reconstruction. Here, an equi-tilt set was manually segmented by using the interactive background removal tool. Both the automatic and manual segmentation results were used to accomplish the 3D reconstruction job. The problematic segmentation results was then refined

be refined using shape priors obtained from the reconstructed 3D model. To measure the segmentation improvement, the proposed method was applied to the synthetic data composed of the rendering result of the 3D model and random background noises, as depicted in Fig. 21. Since the silhouette is known in the synthetic data set, the error between the segmentation result produced by our method and the silhouette can be calculated. The Hausdroff distance was adopted to measure the segmentation errors. In our experiment, four levels of background noises were composed to the synthetic data, and 10 and 20 ground truth images were randomly selected to learn the shape prior. The results of Fig. 22 indicate that shape information is indeed critical to alleviate eliminate segmentation errors, and ensures that the segmentation method is robust to background noises. Fig. 23 shows the comparison between ground truth and the segmentation results produced by the proposed method with shape prior 2.



**Fig 16** The top row shows a portion of the input image sequence taken from an equi-tilt set of the pottery owl OM. For all the images in the middle and bottom rows, the black pixels correspond to the classified background regions. The foreground regions are colored white, and the unknown regions are colored gray. The middle row shows the corresponding result during the B-labeling for each image. Finally, the bottom rows show the generated trimap for each image that is used to activate the graph cut image segmentation.



Fig. 17. The results of the automatic initial segmentation corresponding to the image sequence shown in Fig 16. The three images on the left show the segmentation results that should be selected for the 3D reconstruction, while the others shows results that should be excluded and refined in the next run. The red circles denote the noticeable segmentation errors in each image.

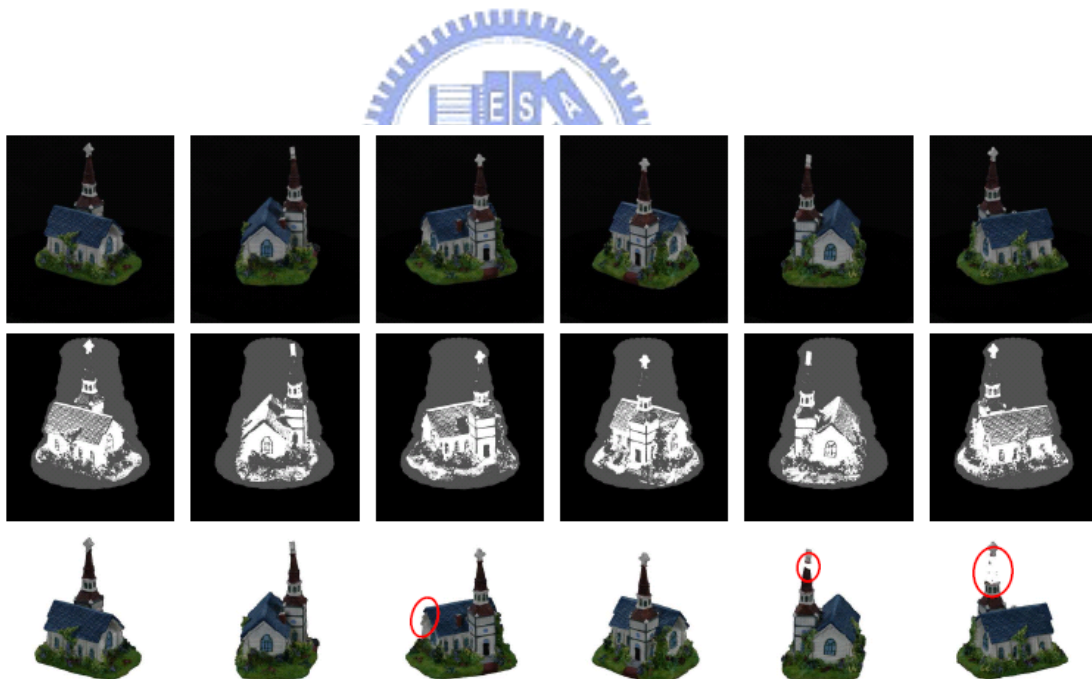


Fig. 18. The top row shows a portion of an equi-tilt set for the toy house OM. The middle row shows the trimap labeling result for each image. Finally, the bottom row shows the results of the automatic initial segmentation. The red circles indicate the noticeable segmentation errors in each image, to be rectified in the next run.

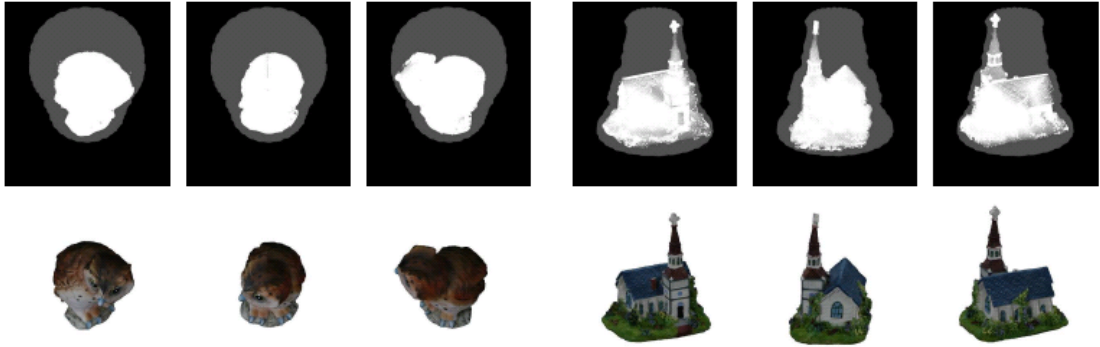


Fig. 19. The rectification of the segmentation errors for the pottery owl in Fig. 17 and the toy house in Fig. 18. Top row shows the refined trimaps. The segmentation results are shown bottom row for each image.

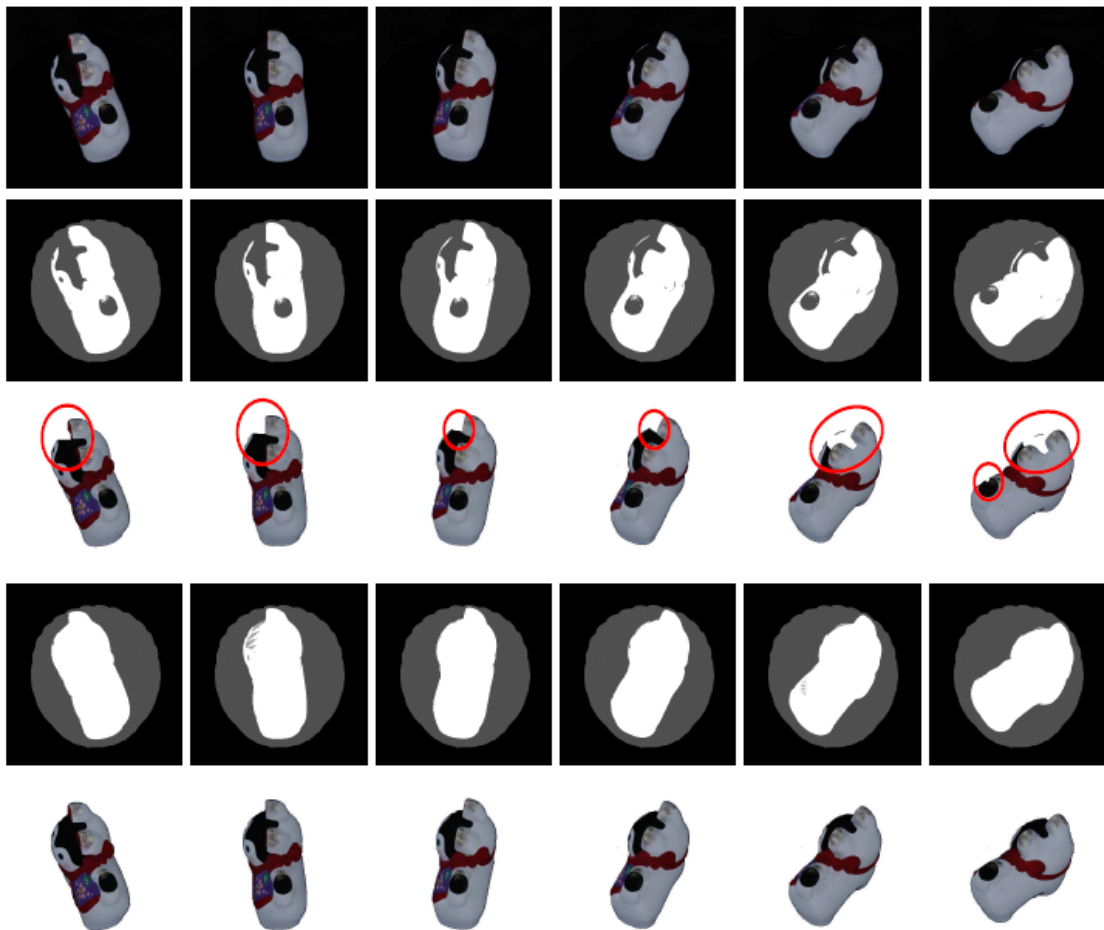
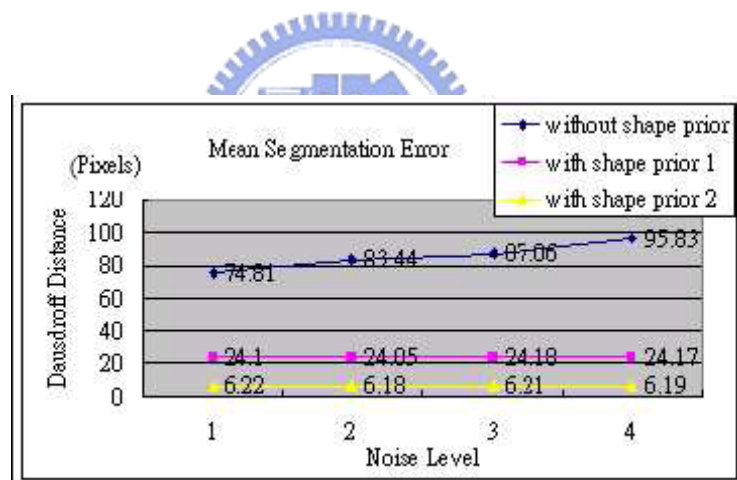


Fig 20. The first row shows six consecutive images in an equi-tilt set of the pottery cat OM. The second row shows the result of trimap labeling. The third row shows the result of the automatic initial segmentation. In the fourth row, the projection of the reconstructed 3D model provides the information on regions that is quite difficult to obtain by the methods based on color and contrast alone. The last row shows the refinement of the segmentation result by using shape priors.



**Fig. 21.** The Armadillo that is the 3D model adopted to generate the synthetic data.



**Fig. 22** Mean segmentation errors on the synthetic data. The image size is 800 x 600. In the experiments, the 3D shape was reconstructed by randomly selecting ground truth images. The shape prior 1 was learnt by using 10 images, and the shape prior 2 was learnt by 20 images.





**Fig. 23.** The first row shows six consecutive images in an equi-tilt set of the Armadillo OM. The second row shows the result of trimap labeling. The third row shows the result of the automatic initial segmentation. In the fourth row, the projection of the reconstructed 3D model provides the information on regions that is quite difficult to obtain by the methods based on color and contrast alone. The fifth row shows the refinement of the segmentation result by using shape priors. The last row shows the comparison between the segmentation results produced by the proposed method and the ground truth. The red solid lines denote the contours of the ground truth, and the green dot lines denote the segmentation results produced by the proposed method

# Chapter 4

## Object Movie-Based 3D Reconstruction

To the best of our knowledge, graph cuts based methods for 3D reconstruction can produce better results than other passive methods. In the section 4.1, we will briefly describe the volumetric graph cuts algorithm, and the problems of the algorithm. To solve the problems, we propose a two-phase approach to recover 3D object surfaces with silhouette preserved and high photo-consistency properties from multi-view images. In the first phase, a silhouette-preserved volumetric graph cuts algorithm is proposed to obtain a silhouette-preserved 3D surface. In the second phase, the 3D surface will be refined using gradient descent optimization. The positions of the vertices on the surface will be adjusted along the normal directions to make sure the surface has high photo-consistency such that more detail features of 3D surface can be recovered.

### 4.1. Volumetric Graph Cuts

The volumetric 3D reconstruction problem can be expressed as a labeling problem, which involves deciding whether a given voxel within the volume is inside or outside the surface of the object. The idea of the volumetric graph cuts is as follows. The true surface is assumed to be between a given base surface  $S_{base}$  and a parallel inner surface  $S_{in}$ . The base surface is an approximation of the true surface, encloses the true surface. In practice, the base surface can be obtained from the visual hull [28]. Each candidate surface under this assumption is then scored mainly according to whether the points on the surface are photo-consistent. The algorithm finds the optimal surface by solving the minimum cut of a corresponding weighted graph. Fig. 24

shows the idea of volumetric graph cuts algorithm.

Specifically, for each voxel  $x \in R^3$ , let  $\rho(x)$  be the photo-consistency score of  $x$ , where a lower value represents a better photo-consistency. For a candidate surface  $S$ , let  $V(S)$  be the volume between  $S$  and the base surface. Each candidate surface is associated with the energy function consisting of the integral of the photo-consistency score  $\rho(x)$  on the surface and the size of the volume  $V(S)$ . The true surface  $S$  is determined by finding the global minimum of the energy function  $E(S)$  among all candidate surfaces  $S$ ,

$$S^* = \arg \min_S E(S) \quad (40)$$

where

$$E(S) = \iint_S \rho(x) dA + \lambda \iiint_{V(S)} dV \quad (41)$$

In (41), the first integral tends toward a photo-consistent surface, while the second, called the *ballooning term*, prefers a fatter reconstructed model. The reason for preferring a fatter model is that finding the global minimum can result in a trend to remove the protrusive parts of the object. The goal of the ballooning term is to counterbalance the protrusion flattening problem. Vogiatzis *et al* [62] describes the detailed formulation and graph construction.

As is well known, solving the two terminals min-cut problem is equivalent to finding the maximum a posteriori (MAP) estimation of a MRF with two labels. The graph cut energy minimization, such as that used in the volumetric graph cuts, is widely adopted in many computer vision applications. Similar to most of the energy functions that can be minimized by the graph cut, (41) also includes the data and boundary properties.

Let  $V$  be the set of voxels within the base surface. Let  $N$  be a neighborhood system defined for  $V$ , which containing the set of all pairs of neighboring voxels. Let  $L = \{l_i \mid \forall x_i \in V\}$  be a family of random variables defined on the set  $V$ , in which each variable takes a label  $l_i$  from

$\{1,0\}$ . Given a candidate surface  $S$ , a corresponding random field  $L$  is uniquely defined such that for any voxel  $p$  in  $V$

$$l_i = \begin{cases} 1 & \text{;if voxel } x_i \text{ is within the surface } S \\ 0 & \text{;otherwise} \end{cases} \quad (42)$$

In the discrete case, it can be easily proven that the energy function  $E(S)$  in (41) associated with a candidate surface  $S$  can be rewritten as  $E(L)$  which corresponds to the joint of data and boundary properties of a random field  $L$

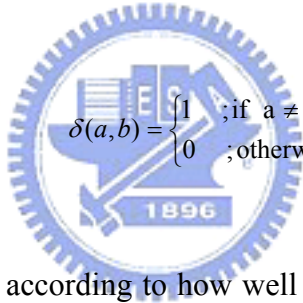
$$E(L) = \sum_{x_i \in V} D(x_i) + \sum_{(x_i, x_j) \in N} B(x_i, x_j) \quad (43)$$

where

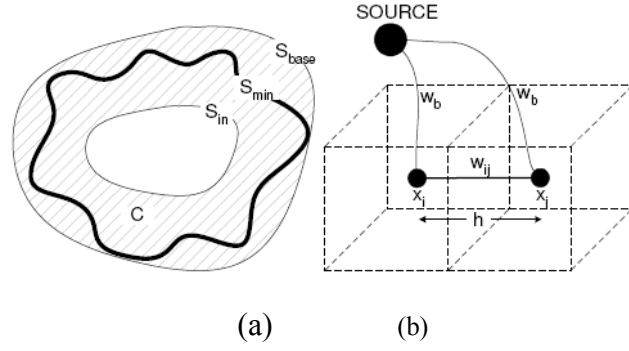
$$D(x_i) = \lambda \cdot \delta(l_i, 1) \quad (44)$$

$$B(x_i, x_j) = \frac{\rho(x_i) + \rho(x_j)}{2} \cdot \delta(l_i, l_j)$$

and

$$\delta(a, b) = \begin{cases} 1 & \text{;if } a \neq b \\ 0 & \text{;otherwise} \end{cases} \quad (45)$$


Here,  $D(x_i)$  is the penalty according to how well the voxel  $x_i$  fits into the given label  $l_i$ , while  $B$  can maintain the smoothness prior such that the physical property in the neighborhood of the space offers some coherence and does not change abruptly [30]. In the implementation, The edge weight, as shown in Fig. 24(b), between two neighbor voxels  $x_i$  and  $x_j$  is defined as  $w_{i,j} = \frac{4}{3} \pi h^2 \cdot \frac{(\rho(i) + \rho(j))}{2}$ , where  $h$  is the voxel size. And every voxel is connected to SOURCE, the terminal node indicates inside object, with the weight  $w_b = \lambda h^3$ . With the graph  $G$  constructed this way, the graph cut algorithm is then applied to find  $S_{min}$ .

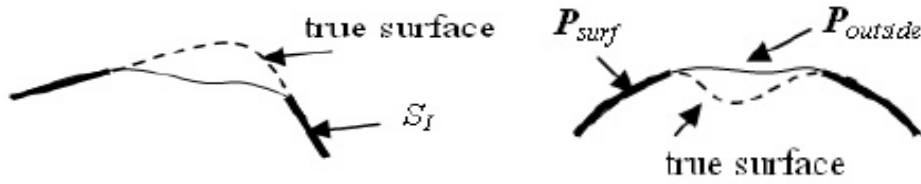


**Fig. 24. Illustration of volumetric graph cuts algorithm.** (a) Graph cuts algorithm is used to find the  $S_{min}$  surface between  $S_{base}$  and  $S_{in}$  in volumetric graph cuts. (b)  $x_i$  and  $x_j$  are the neighbor voxels. The edge weight between these two voxels is represented as  $w_{ij}$  and the edge weight between voxels and source node is represented as  $w_b$ .  $h$  means the length between two voxels.

#### 4.1.1. Problem I: Not Preserving Concavity-Convex Features

Since the graph cut algorithm usually prefers shorter cuts, concavity-convex features may be lost. This problem was described in [57] in detail. As shown in Fig. 25, the dotted line is the true surface of object, and the solid line is the surface decided by volumetric graph cuts. Although the voxels on the true surface has high photo-consistency, the total energy is not minimized because the distance of this path is longer.

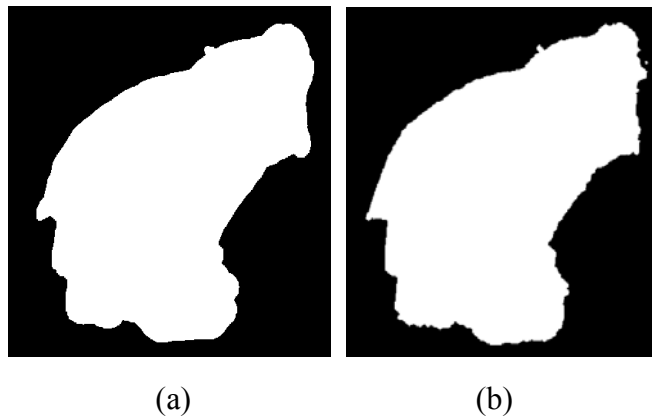
To counterbalance this problem, a simple constant penalty  $\lambda$  in (44) is chosen to penalize all voxels that are not inside the surface. One problem with the volumetric graph cuts is that the parameter  $\lambda$  has to be chosen through trial and error in order to obtain a satisfactory result. Furthermore, the ballooning term could lead to a tug-of-war between the original protrusion flattening problem and the following concavity filling problem, where the concavities presented in the object are filled. For some objects, a befitting ballooning term still can not be found out to obtain a correctly reconstructed object even after an exhaustive search of the parameter  $\lambda$ . The phenomenon is also demonstrated in one of our experiments.



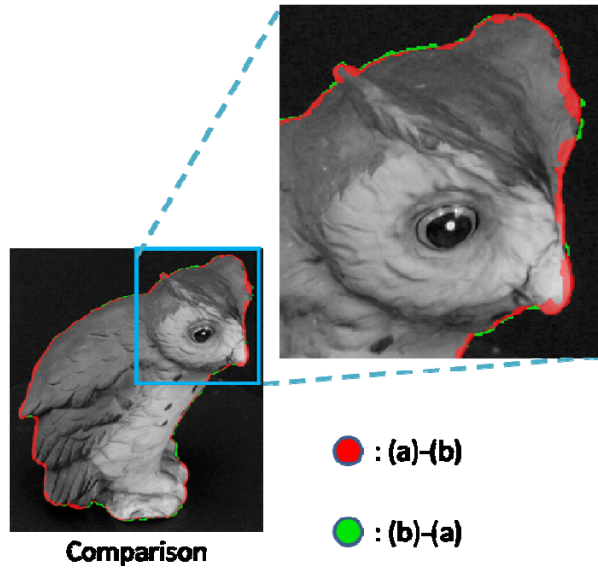
**Fig. 25** Two cases that cause errors may occur in volumetric graph cuts. Because of the shorter cut property of volumetric graph cuts, Concavity-Convex feature will be flattened in volumetric graph cuts.

#### 4.1.2. Problem II: Not Preserving Silhouettes

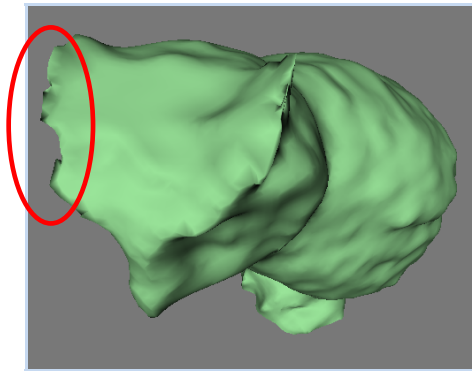
Because the silhouette information is not considered in [62], the inaccuracy can be observed on the silhouette of volumetric graph cuts result. Fig. 28 shows the reconstructed 3D model of potty owl using volumetric graph cuts algorithm. The ear of the reconstructed 3D is incomplete so that its projected silhouette may not match with the input silhouette. Fig. 26 shows an input silhouette image and the projected silhouette image, and the comparison is shown in Fig. 27. The green and red pixels indicate the differences between the input silhouette and the projected silhouette.



**Fig. 26.** Silhouette images. (a) is the input silhouette image for 3D reconstruction. (b) is the silhouette image generated from the reconstruction model using volumetric graph cuts algorithm.



**Fig. 27.** The comparison between silhouette images shown in Fig. 26. The unmatched regions are colored in red and green.

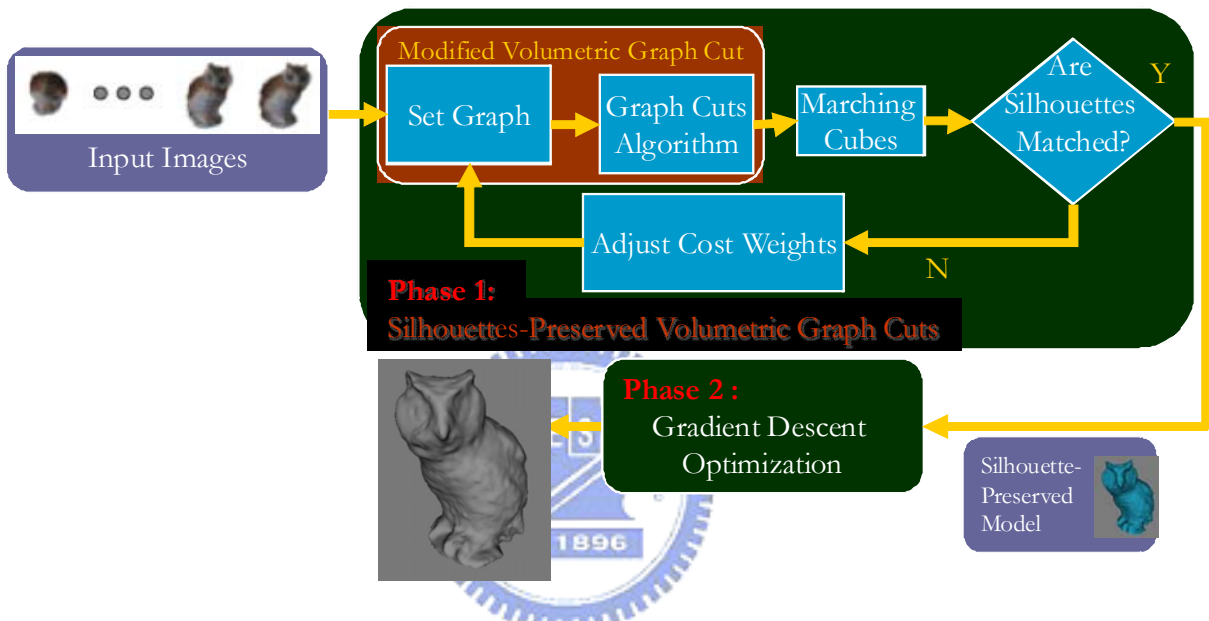


**Fig. 28.** The broken ears is caused by not considering the silhouette information in volumetric graph cuts.

## 4.2. Our Approach

In this section, we first propose a modified volumetric graph cuts algorithm by introducing discrete medial axis constraints into traditional volumetric graph cuts algorithm and develop a two-phase method, as shown in Fig. 29, to solve the problems that silhouettes and concavity-convex features are not preserved. In the first phase, a surface that its silhouette strongly matches the input images is constructed by the modified volumetric graph cuts in an iterative way. This algorithm starts from volumetric graph cuts algorithm and improves the reconstruction result by adjusting the optimization cost according to the output of volumetric

graph cuts in an iterative way. These iterative steps would be performed until the silhouettes of the obtained 3D surface match the input silhouette images. To generate silhouette images, the matching cube algorithm [33], a famous triangulation method, is applied to obtain the triangle-based 3D model, and the triangle model is rendered. At this moment, we have got a global solution surface. In order to make the surface fit the local solution, it is refined by gradient descent method in the second phase.



**Fig. 29.** The flowchart of our approach. This approach contains two phases. In the first phase, a silhouette-preserved model is generated by a silhouette-preserved volumetric graph cuts algorithm. Then, the result of phase 1 is refined by gradient descent in phase 2.

#### 4.2.1. Discrete Medial Axis Constraint

To counterbalance the problem I described in Section 4.1.1, a simple constant penalty  $\lambda$  in (44) is chosen to penalize all voxels that are not inside the surface. But to achieve better performance, the definition of  $D(p)$  should consider the likelihood that the voxel  $p$  is inside or outside the surface with respect to the available observations. Unfortunately, until now, it is still not clear on how to compute a good estimate of the likelihood based on the available observations. Here, we present a new definition of  $D(p)$  based on the medial axis of the object,



which has been proven to work well as shown in the experiments.

The medial axis of the 3D object is defined as the centers of all maximal spheres in the object that touch the shell of the object at two or more points. In practice, the medial axis is represented by a set of discrete voxels interior to the 3D object, called discrete medial axis (DMA). The DMA of a volumetric model can be obtained by analyzing the 3D distance field, which is computed by the distance transformation method. A good overview of these methods has been provided by Cuisenaire [14]. The local maxima in the 3D distance field are examined to serve as the DMA. Because undesired branches might exist, which is considered to be meaningless, only the large enough connected components of the voxels in the DMA are retained. Compared to the original volumetric graph cuts, we first compute the DMA of the base surface, which is assumed to be an adequate approximation of the DMA of the true surface. The DMA itself is imposed as the hard constraint of the object such that the voxels in the DMA are enforced to be inside the object, while the voxels in the neighborhood of the DMA act as the soft constraint that are very probable to be inside the object.

Specifically, let  $V_A$  be the set of voxels in the DMA. Let  $d_x$  be the minimum distance from the voxel  $x$  to its nearest voxel in  $V_A$ . Computing the minimum distances for all voxels can be accelerated by using the distance transformation method to obtain an approximate solution. For each voxel within the base surface, the possibility of being inside the true surface is considered to be inversely proportional to the minimum distance. Thus, we define the new data property  $D_A(x)$ , into which the DMA constraint has been embedded

$$D_A(x_i) = \begin{cases} \infty \cdot \delta(f_i, I) & ; \forall x_i \in V_A \\ \lambda \cdot \exp\left(\frac{-d_{x_i}^2}{2\sigma^2}\right) \cdot \delta(f_i, I) & ; \text{otherwise} \end{cases} \quad (46)$$

Here, (48) guarantees that the voxels in  $V_A$  are always labeled as being inside the surface. Additionally, (49) encourages the voxels in the neighborhood of the DMA to be labeled as being inside the surface. Notably, the parameter  $\lambda$  adjusts the strength of the soft constraint, while  $\sigma^2$  controls the influenced range. The energy function with the new data property  $D_A(x_i)$  is

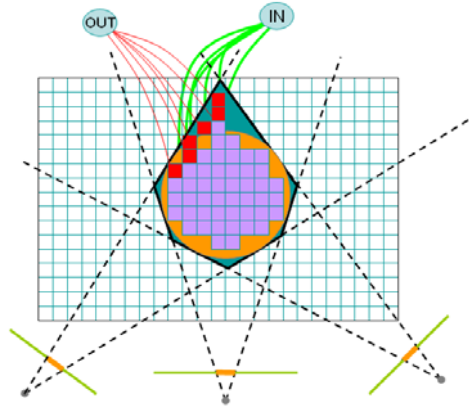
globally minimized by using the graph cut technique similar to [62].

#### 4.2.2. Silhouettes-Preserved Volumetric Graph Cuts

This algorithm is based on the modified volumetric graph cuts. However it uses the output of volumetric graph cuts as a feedback to adjust the edge weight between voxels and SOURCE. These steps run in an iterative way until the silhouettes completely match the observed pictures. In the first step, we run the modified volumetric graph cuts and construct the mesh and to generate the silhouette maps in every view. Then in step 2, check if silhouette matches the input images. If a voxel is not projected in the silhouette maps, we will increase the edge weight between this voxel and SOURCE node and perform volumetric graph cuts again. These steps run in an iterative way until the silhouette of volumetric graph cuts result matches all the input images.

Fig. 30 shows the idea of phase 1. The orange circle represents the object to be reconstructed. The purple grids represent the voxels labeled as inner of object after volumetric graph cuts. And the silhouette does not match the image captured by the left camera unless one of the red grids is added. So we increase the edge weight between those grids and SOURCE node (in object node) and run volumetric graph cuts again to get a silhouette-preserved model.

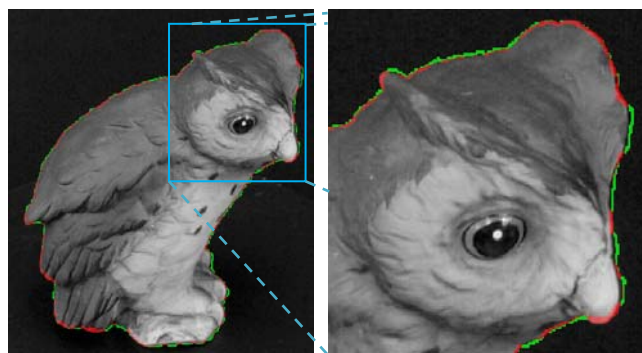
A silhouette of output of phase 1 is shown in Fig. 32, and compared with the input silhouette is shown in Fig. 32. The silhouette of reconstructed 3D model almost matches the input silhouette except a few quantization errors caused by the marching cube. The improvement of phase 1 can also be observed by a 3D mesh shown in Fig. 33, e.g., the broken ear is fixed.



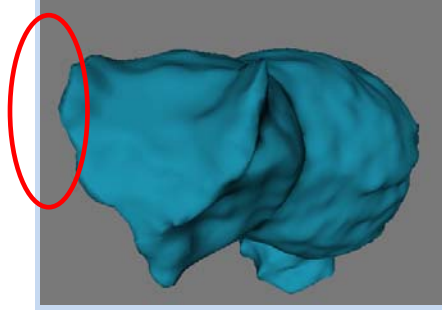
**Fig. 30. Silhouette-preserved volumetric graph cuts algorithm. Orange circle: the 3D object to be reconstructed. Purple grid: The voxels labeled “IN Object” after volumetric graph cuts. Red grid: The voxels have to increase edge weights to match the silhouettes.**



**Fig. 31. A silhouette image projected from the reconstructed 3D model using the silhouette-preserved volumetric graph cuts.**



**Fig. 32 Comparison between silhouette images shown in Fig. 31 and Fig. 26 (a).**



**Fig. 33. The reconstructed 3D model after phase 1. Notably, The broken ears are fixed.**

### 4.2.3. Gradient Descent Using Photo Consistency Constraint

Before starting the phase 2 method, we define the problem that in phase 2 we try to solve first. The phase 2 method takes the following as input :

- a set of n images  $I = \{I_i | i = 1..n\}$  ;
- a set of projection matrices  $P = \{P_i | i = 1..n\}$  ;
- an Initial shape  $S_0$  ;

Then, our purpose is to find a 3D surface  $S_{max}$  that maximizes the energy function  $E(S)$ , where  $E$  is defined as (48).

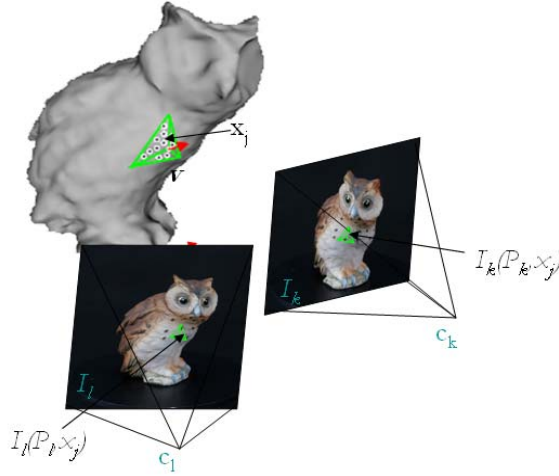
$$E(S) = \int_S g(x) dS \quad (48)$$

Where the  $g$  is the photo-consistency function, and  $x$  is the point on the surface. In our implement, the equation (2) is used to approach (1).

$$E'(S) = \sum_{v_i \in S} Z(v_i) \quad (49)$$

$$Z(v) = \frac{1}{N(l, k)} \sum_{c_l \in C(v)} \sum_{c_k \in C(v) \& c_k \neq c_l} \frac{\sum_{x_j \in X(v)} (I_l(P_l, x_j) - \bar{I}(P_l))(I_k(P_k, x_j) - \bar{I}(P_k))}{\sqrt{\sum_{x_j \in X(v)} (I_l(P_l, x_j) - \bar{I}(P_l))^2} \sqrt{\sum_{x_j \in X(v)} (I_k(P_k, x_j) - \bar{I}(P_k))^2}} \quad (50)$$

where  $v$  is the vertex of 3D mesh,  $C(v)$  is the Camera set can observe vertex  $v$ ,  $I(P, x)$  is the color that 3D point  $x$  projected by matrix  $P$  on image  $I$ ,  $N(l, k)$  is the num of pair of  $l$  and  $k$ , and  $X(v)$  is the 3D point set that lies on the triangle which contain vertex  $v$ .



**Fig. 34** The meaning of symbols in (50)

The gradient descent is used to adjust the vertices of the 3D mesh along their normal directions with the following update function (51). The vertices of the 3D mesh are updated by turns until all vertices converge on their local maxima.

$$v_t = v_{t-1} + \kappa(Z(v^+) - Z(v^-)) \vec{n} \quad (51)$$

$$v^+ = v + \sigma \vec{n}, \quad v^- = v - \sigma \vec{n} \quad (52)$$

where  $\kappa$  and  $\sigma$  are tuning parameters.

The gradient descent algorithm has the property that refined surface may converge at the local maximum. However, it would still do well in our work because we can get a good initial surface from phase one. After refined by phase two, the surface should be in a state with high photo consistency.

### 4.3. Experimental Results of 3D Reconstruction

The first experiment involved the toy house, which was also adopted to demonstrate the advantage of using the DMA constraint. The toy house was chosen deliberately because it

represents a difficult 3D reconstruction problem, due to noticeable protrusions and concavities in the object. Fig. 35 demonstrates the difficulty of reconstructing the toy house, indicated by the tug-of-war between the protrusion flattening problem and the concavity filling problem. Without the DMA constraint, even if the concavities all around the house are going to be filled, the ballooning term still cannot correctly deal with the tower even after it has been exhaustively searched. Fig. 36 illustrates the benefit of the DMA constraint for improving the volumetric graph cuts algorithm. The visualization of the photo-consistency scores is also provided. Fig. 37 shows the successfully reconstructed model of the toy house by imposing the DMA constraint to alleviate this difficulty. Notably, the algorithm can properly reconstruct both the protrusive parts, i.e. the tower and chimney of the toy house, and the concavities all around the house.

The second experiment adopted the pottery owl. Fig. 38 shows the reconstructed model. Although the ears of the pottery owl are thin and sharp, they were correctly reconstructed with the DMA constraint. Additionally, the concavities around the eyes and feet were handled properly.

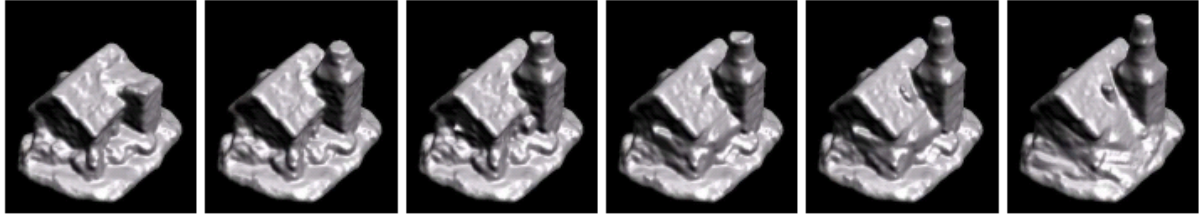
We test our two-phase approach with the potty owl model and two synthesis models. Fig. 39 shows the result of real owl model. The result of phase one algorithm is shown in Fig. 39(a), and the result of phase two method is shown in Fig. 39(b). We can easily find the details of Fig. 39(b) are stronger than those of Fig. 39(a).

In order to make sure the refinement in phase two is correct, two synthesis models, the bunny and the buddha, are tested. Fig. 40 shows the result of phase 1 and phase 2 and ground truth.

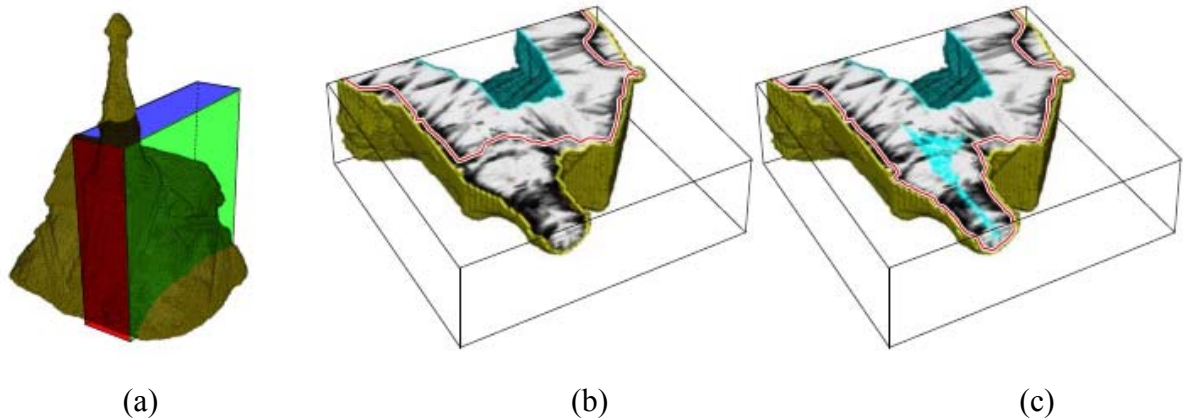
In a complicated model case, as shown in Fig. 41, the improvement of our work is totally shown. We compare our work with traditional volumetric graph cuts in this case. Because of the influence of self occlusion, the head of buddha is cut off, as shown in Fig. 41(a). However, our phase one method can fix this error, as shown in Fig. 41(b). Then, the enhancement of details by

phase two can be observed from the buddha face.

At the end of the experiment, we extract the texture of 3D reconstruct results from input images. And the textured models are rendered to original view to compare with the original image, as shown in Fig. 42.



**Fig. 35.** The reconstructed model of the toy house by using the volumetric graph cuts algorithm without imposing the DMA constraint. The ballooning term is increased gradually from left to right. The figure indicate that reconstructing the toy house is a difficult task without the DMA constraint.



**Fig. 36.** Visualization and comparison of the 3D reconstruction algorithm. Both (b) and (c) are taken from a cross-section of the visual hull for the toy house, which is shown in (a). The golden voxels correspond to the base surface in all three images. The cyan voxels denote the inner surface, which is parallel to the base surface. Additionally, the voxels in VA are also colored cyan in (c). The photo-consistency scores between the base and inner surfaces are shown, where the darker region indicates a better photo-consistency. Additionally, the line within the base and inner surfaces represents the reconstructed surface of the object. In (b), without the DMA constraint, although the reconstructed surface passes through the worse photo-consistency regions, the integral of the energy on the entire surface is lower. Consequently, the protrusive part (i.e., the tower of the house) is flattened incorrectly. The image in (c) shows the correctly reconstructed surface for the same portion of the object with the DMA constraint.

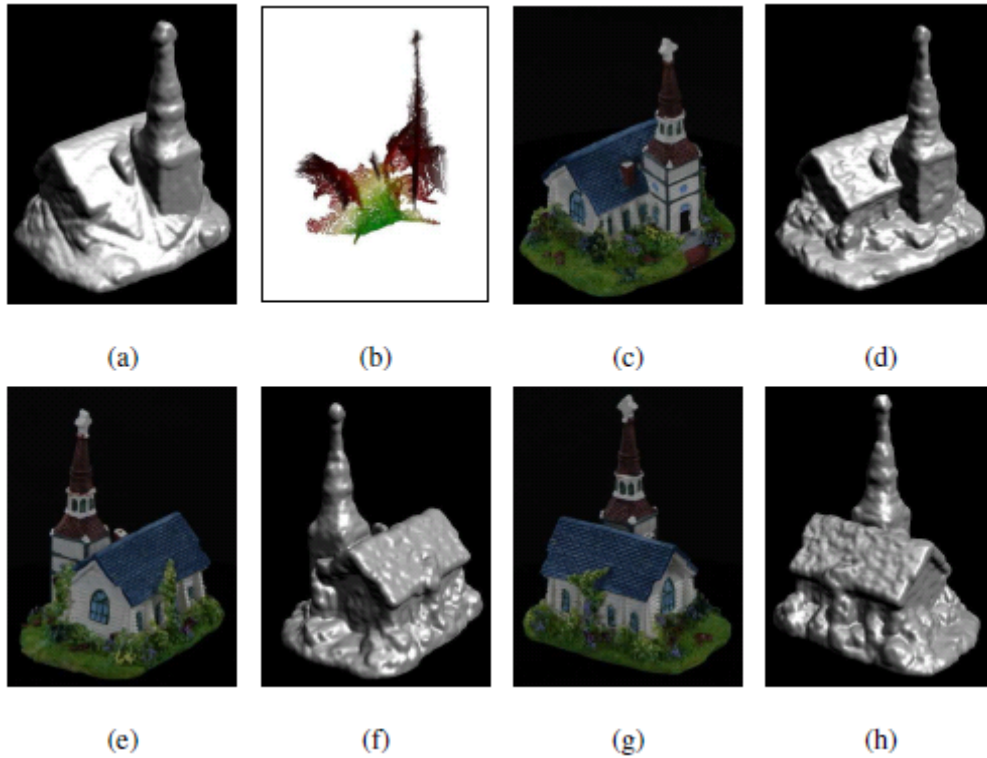


Fig. 37. Image (a) shows the visual hull generated from the available silhouettes of the toy house to act as the base surface in the algorithm; (b) the DMA of the visual hull that is considered to be an approximate DMA of the toy house. Images (c)-(h) show the reconstructed model from three different viewpoints of the toy house, together with the images captured at similar viewpoints.

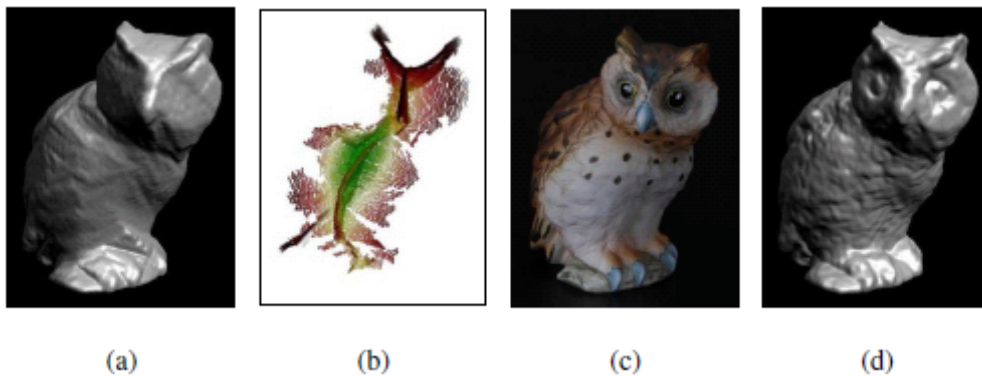
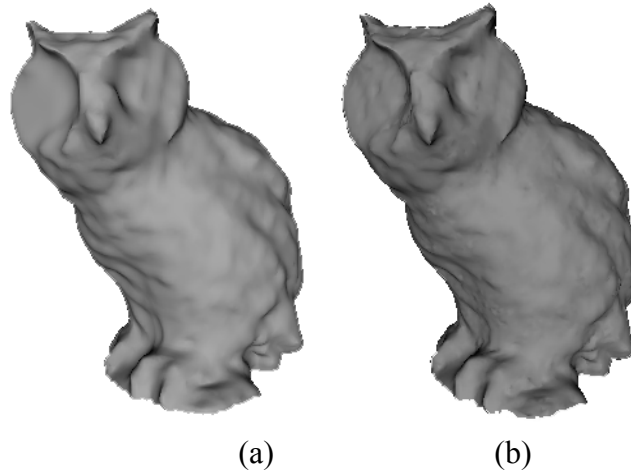
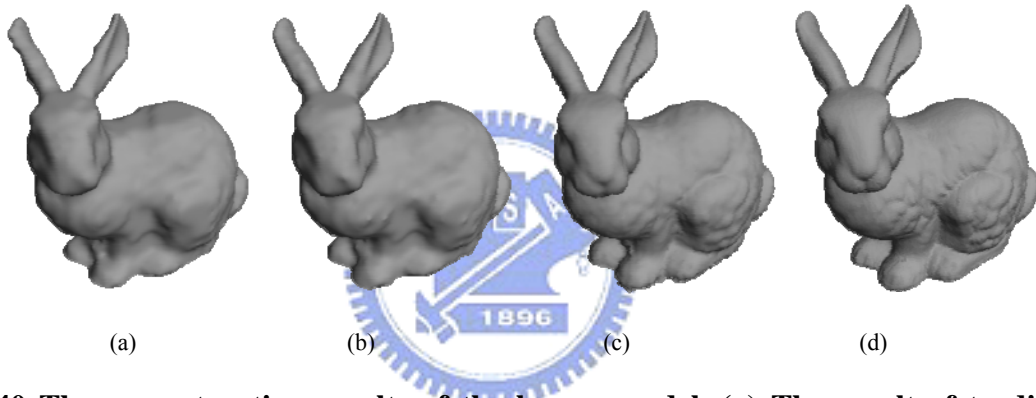


Fig. 38. (a) The visual hull of the pottery owl. (b) The DMA of the visual hull. (c) An example image of the pottery owl MVI. (d) The reconstructed model of the pottery owl by using our method.

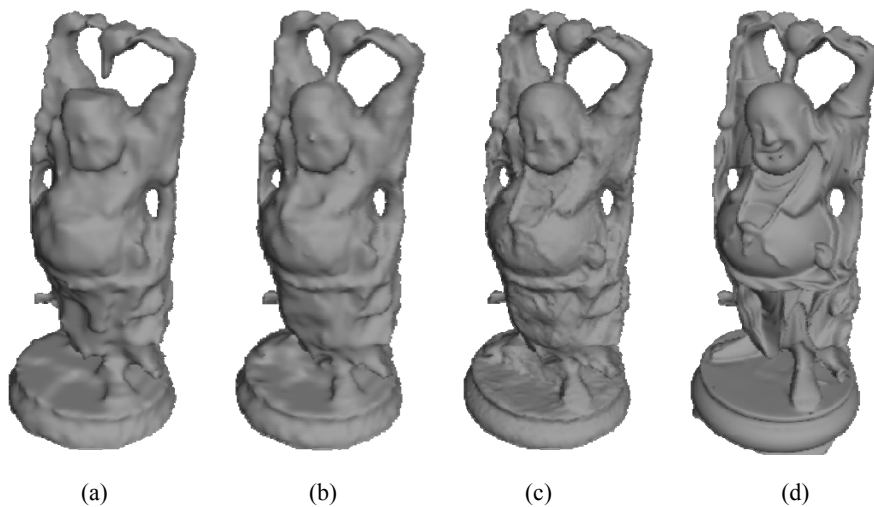




**Fig. 39** The reconstructed owl models. (a) The result of phase one. (b) The result of phase two.



**Fig. 40** The reconstruction results of the bunny model. (a) The result of traditional volumetric graph cuts algorithm (b) The result of phase one (c) The result of phase two (b) The ground truth.



**Fig. 41** The reconstructed buddha models. (a) The result of traditional volumetric graph cuts. (b) The result of phase 1. (c) the result of phase 2 (d) The ground truth.



**Fig. 42 Comparison between our result and original image. Left is the original image and the right is the result reconstructed by our method.**



# Chapter 5

## Augmented Stereo Panoramas

In [25], Hung et al. proposed a method to integrate object movies into a panorama in a visually 3D-consistent way. With the proposed method, the user can easily author and browse the augmented panorama. Lo et al. [32] have successfully applied the technique to construct a kiosk for visual museum. In this thesis, we extend our previous work on augmented panorama to augmented stereo panorama. We develop an interactive system which allows the user to integrate stereo OMs into a stereo panorama, and interactively browse the augmented stereo panorama.



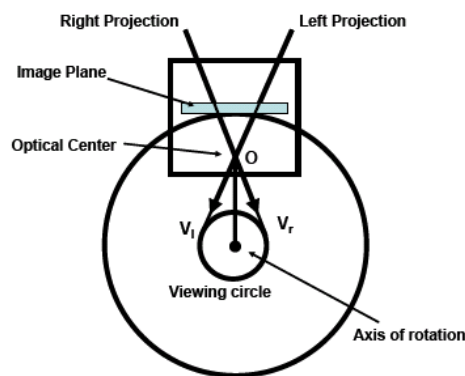
### 5.1. Generation of Stereo Panoramas

To generate stereo panoramas, Huang and Hung [23] proposed a method to automatically generate a stereo panorama with two cameras. One of the cameras is rotating on the axis and the other is off-center rotating. This method generates two sets of panorama, one for the left view and the other for the right view.

The method, named Parallel Ray Interpolation for Stereo Mosaicing (PRISM) proposed in [68], is to stitch mosaics seamlessly for aerial images. The authors generated stereo panorama from an aerial camera. The aerial camera, which undergoes a dominant translational motion, is mounted on an aerial plane. To calibrate the aerial camera, they estimate the extrinsic parameters of the camera by an aerial instrumentation system, such as GPS, INS and laser profiler. After estimating camera parameters, they rectified the captured images to eliminate rotational components.

Shum and He proposed concentric mosaic [54] to capture rays in the environment. Those rays are all tangent to several specific circles and form several cylindrical images with different radius. The concentric mosaic can render scenes at any view point toward any viewing direction inside the circle. Shum and Szeliski [54] further use the concentric mosaic to generate stereo. Because the depth of any vertical strip of captured rays is not identical, they apply depth correction for captured rays.

In this thesis, we adopt the method proposed in [45], because their method is easy to implement. Their method generates stereo panoramas by stitching vertical strips of a series of images captured by a video camera. These image strips can approximate the desired circular projection on a cylindrical image surface. As shown in Fig. 43, the camera with an optical center  $O$  and an image plane is rotated about the rotation axis behind the camera. Strips at the left side of the image are seen from viewpoint  $V_r$ , and strips at right side of the image are seen from viewpoint  $V_l$ . The left strips are extracted for the right panorama and the right strips are for the left panorama. Therefore, the left panoramic image can be constructed from strips located at the right side of images and the right panoramic image can be constructed from strips located at the left side of images.

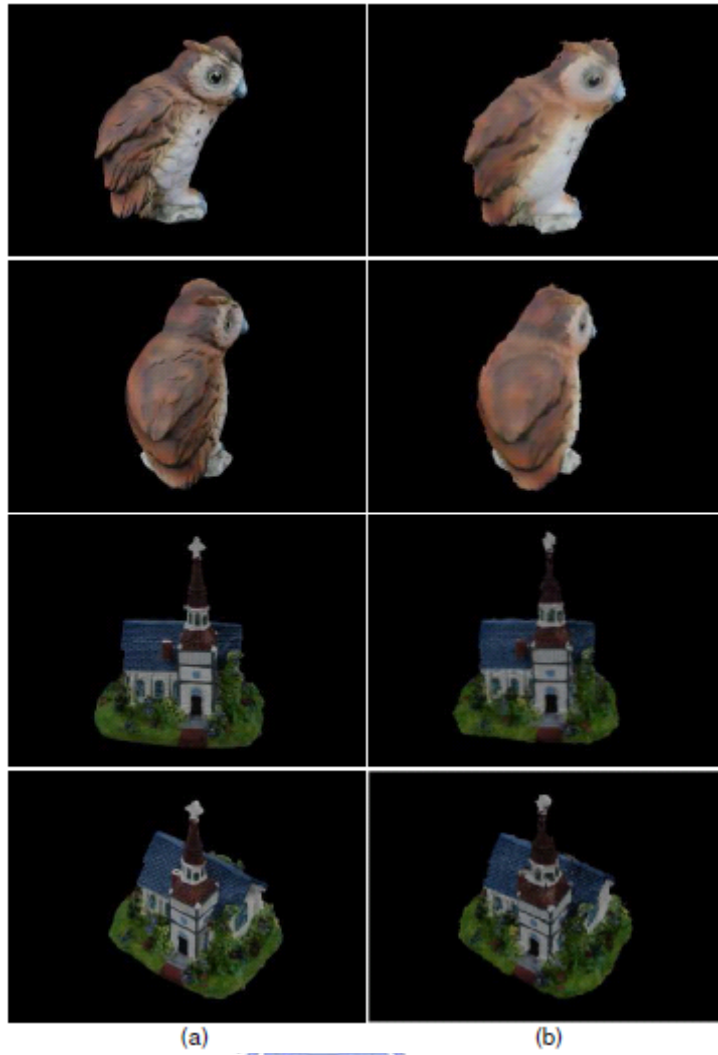


**Fig. 43. A diagram shows the idea to create a stereo panorama using a video camera.**

## 5.2. Generation of Stereo Object Movies

In this section, we will describe how to generate stereo OMs from acquired monocular OMs. For convenience, we assume that the acquired OM is for left view, named left-OM. For each image  $I_{\theta,\phi}^L$  in an left-OM, our goal is generate the image  $I_{\theta,\phi}^R$  for right view with the help of 3D model.

Once the viewing baseline is determined, the camera parameters of right view can be calculated. This distance between two viewpoints is usually about 5cm to 7cm, which is the average interval of general human eyes, while the real distance depends on where the object is placed in virtual environment. We first use the image  $I_{\theta,\phi}^L$  as the texture of the reconstructed 3D model in Chapter 4, and render the 3D model with the calculated camera parameters. Next, we render the 3D model on the right view again by using image  $I_{\theta,\phi+1}^L$  to be the texture of the 3D model. The image  $I_{\theta,\phi+1}^L$  is the right-side neighboring image of  $I_{\theta,\phi}^L$ , and then composite the two rendered images. This approach is very simple, and can be accelerated by industry-standard graphics hardware. Hence, the final binocular OM consists of a set of these images that are generated in this way for each view in the original monocular OM, as demonstrated in Fig. 44.



**Fig. 44 (a) The original OM images. (b) Our rendering results of binocular views.**

### **5.3. Augmenting Stereo Panoramas with Stereo OMs**

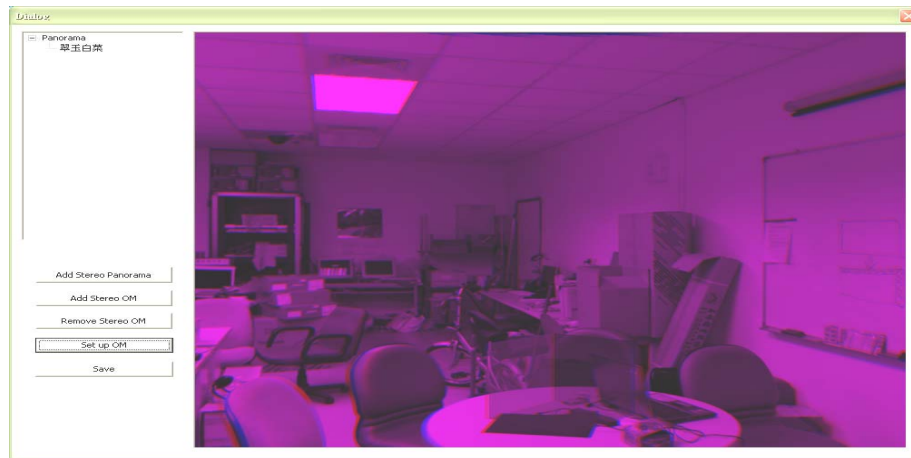
To integrate stereo OMs into a stereo panorama, we have to know where the objects will be inserted in. As mentioned in [25], to achieve the task for a monocular panorama, the user is only required to specify four vertices of a cuboid to define a 3D reference frame, named shadow reference frame (SRF), in a 2D dewarped view. This reference frame defines where the shadow of the object is supposed to be projected onto. Once the user has specified a SRF in the dewarped panoramic view, the geo-metric transformation between this SRF and the panorama reference frame (PRF) can be computed using this information [8]. By referring to the SRF, the

user can insert stereo OMs into the stereo panorama in a visually 3D-consistent way. Each stereo OM is associated with a reference frame, named object reference frame (ORF), so the user can manipulate the stereo OM according to the orientation and location where the user desires. In this work, we extend the method to augment a stereo panorama with stereo OMs. Here, two approaches are developed for the users to quickly and accurately specify shadow reference frames in a stereo panorama. One is a 2D approach, which the user can determine the SRFs in dewarped views of the stereo panorama. The other is a 3D approach, which allows the user to specify the SRFs in 3D space with stereoscopic display devices. Fig. 45 shows an example that users integrate stereo OMs into a panorama in 3D mode. The 3D approach is intuitive while the 2D approach does not require the stereoscopic devices.

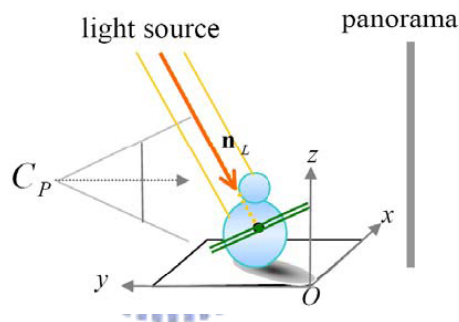
When rendering, the left panorama with left-OMs and right panorama with right-OMs are processed separately but in the same way. When rendering, we sequentially render the background layer, the shadow layer and the object layer. The background layer is composed by the de-warped view of the panorama. After the viewing direction of viewer is specified, we can dewarp the view according to the specified viewing direction and render it.

An OM with no 3D geometric model is impossible to generate a realistic shadow. To cope with this, we assume the shadow to be generated is produced by a set of parallel light sources. The lighting directions of the parallel sources can either be estimated from photographs containing the global illumination or manually specified by the user. We then can generate shadow of an OM by putting the correct shadow map at the correct position with respect to a user-specified SRF. As shown in Fig. 46, we generate a viewing image by composing the image of the OM correspond to the viewing direction  $n_L$  and its shadow, on the x-z plane of the SRF, produced by shadow map.

To render object layer, we first compute the viewing direction, from the center of PRF ( $C_p$ ) to the center of the ORF, and select and render the image of the OM according to the viewing direction.



**Fig. 45. The UI allows users to integrate stereo OMs into a stereo panorama in 3D mode.**



**Fig. 46. Illustration of casting shadow for an object movie.**

## 5.4. Experimental Results of Augmented Stereo Panoramas

Fig. 47 shows the stitched results of a stereo panorama from photos taken in our laboratory. Fig. 48 shows the result of integrating a stereo OM into the stereo panorama. The shadow is properly rendered under the inserted object and the perceived depth of the OM is consistent with its nearby scene objects. Fig. 49 shows the consecutive views of rotating the stereo OM in the stereo panorama.



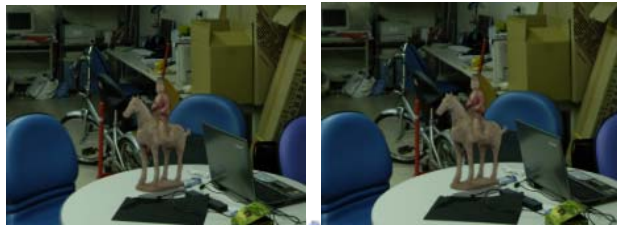


(a)



(b)

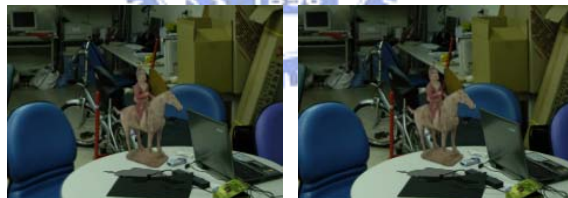
**Fig. 47: Stitching result of a stereo panorama.**



(a)

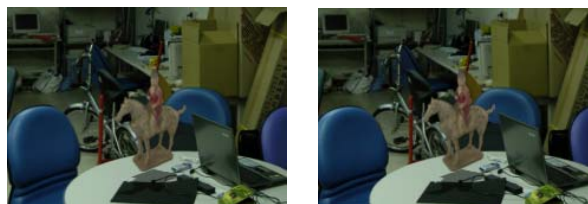
(b)

**Fig. 48: Result of the augmented panorama with a stereo OM. (a) shows the rendered left view, and (b) shows the right view.**



(a)

(b)



(c)

(d)

**Fig. 49 Rotating the OM in the augmented stereo panorama. (a) and (c) are the left views. (b) and (d) are the right views.**

# Chapter 6

## Conclusion and Future Work

This work proposes methods of acquiring high quality OMs including object rig calibration, and background removal. Furthermore, to allow additional applications, this work also develops a 3D reconstruction method to obtain the 3D information of the object, and a new method called augmented stereo panoramas to construct interactive 3D virtual worlds.

To calibrate a motorized object rig, this work first applies the CPC kinematic model to formulate the 3D configuration of the device, and then proposes a method of estimating the parameters of the CPC model of the device. Furthermore, a visual tool is provided to guide users to adjust the controllable axes of the rig according to the estimated results. The proposed method has two major advantages. First, only a small number of images of the calibration object is required. Second, the camera parameters of any views can be obtained with the estimated parameters after calibration.

Since fully automatic segmentation method remains an open problem, this work develops interactive segmentation methods for minimizing the user intervention. First, the initial segmentation results are automatically obtained based on observed characteristics of OM. If some segmentation results do not satisfy user expectations, then the user can modify misclassified pixels in only a few images, and propagate the corrected result to all frames through spatial and temporal coherence. In contrast with other segmentation methods, the proposed method incorporates the shape prior to the image segmentation process. The shape prior introduced into each image of the OM is extracted from the 3D model reconstructed using the volumetric graph cuts algorithm. Experimental results demonstrate the shape

information is indeed critical to eliminating segmentation errors, and ensures that the segmentation method is robust to background noises. Moreover, the proposed OM segmentation process requires only a small amount of user intervention, namely selecting a subset of acceptable segmentations of the OM following the initial segmentation process.

Some graph-cut-based methods have recently been proposed to reconstruct 3D models from multi-view images, and can yield acceptable results. However, such methods have two problems namely that concavity-convex features and silhouettes are not preserved. This work proposes a two-phase approach is proposed to solve these problems. In the first phase, a modified volumetric graph cuts algorithm is applied to obtain a silhouette-preserved 3D surface. This algorithm starts from volumetric graph cuts algorithm and enhances the reconstruction result by iteratively adjusting the optimization cost based on the output of volumetric graph cuts. These iterative steps are performed until the silhouettes of the obtained 3D surface match the input silhouette images. In the second phase, the 3D surface is refined using gradient descent optimization. The positions of the vertices of the 3D model are adjusted along the normal directions to ensure that he surface has high photo-consistency.

Panoramas and OMs are conventionally adopted image-based techniques for modeling and rendering 3D scenes and objects. This work presents a method that allows users to generate an augmented stereo panorama by interactively integrating stereo OMs into a stereo panorama. A user can directly browse the stereo object movies of interest by navigating in the augmented stereo panorama with a stereoscopic display. The augmented stereo panoramas enhance the user's interactive experience by elevating better depth perception.

A major limitation of the proposed method for 3D reconstruction is that it cannot effectively handle specular objects, because the zero-mean normalized cross correlation (ZNCC), which is adopted to measure the photo-consistency score, is not robust to specular reflection. Future work will be to apply to the OM some diffuse-specular separation techniques before 3D reconstruction. Relighting also can be performed with the reconstructed 3D models

after separating the reflection components. Another plan is to further reduce the user intervention by analyzing the energy of the minimum cut after the initial segmentation, and then automatically identifying a subset of acceptable segmented images.



# References

- [1] Adobe Inc. [Online]. Available: <http://www.adobe.com/products/photoshop/>
- [2] M. Agrawal and L. S. Davis, "Complete Camera Calibration Using Spheres : A Dual-Space Approach," *International Conference on Computer Vision*, vol. 2, 2003.
- [3] K. S. Arun, T. S. Huang and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 698-700, 1987.
- [4] T. Beier and S. Neely, "Feature-Based Image Metamorphosis," *Computer Graphics*, vol. 26, no. 2, pp. 35-42, 1992.
- [5] J.-Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the Algorithm," Intel Corporation, Microprocessor Research Labs, OpenCV Documents, 1999.
- [6] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images," *International Conference on Computer Vision*, 2001, pp. 105–112.
- [7] C. Chang, G. Bishop, and A. Lastra, "LDI tree: A hierarchical representation for image-based rendering," *Computer Graphics*, pp. 291–298, August 1999.
- [8] C.-S. Chen, C. K. Yu, Y.-P. Hung, "New Calibration-free Approach for Augmented Reality Based on Parameterized Cuboid Structure," *International Conference on Computer Vision*, (1999) 30-37
- [9] C.-S. Chen and W.-Y. Chang, "On Pose Recovery for Generalized Visual Sensors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 848-861, July 2004.
- [10] C.-W. Chen, L.-W. Chan, Y.-P. Tsai, and Y.-P. Hung, "Augmented stereo panoramas," *Asian Conference on Computer Vision*, vol. 1, pp. 41–49, 2006.
- [11] S. Chen and L. Williams, "View interpolation for image synthesis," *Computer Graphics*, pp. 279–288, August 1993.
- [12] S. E. Chen, "QuickTimeVR – an image-based approach to virtual environment navigation," *Computer Graphics*, pp. 29-38, August 1995.
- [13] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski, "A bayesian approach to digital matting," *Conference on Computer Vision and Pattern Recognition*, pp. 264-271, December 2001.

- [14] O. Cuisenaire, "Distance transformations: Fast algorithms and applications to medical image processing," Ph.D. dissertation, Universit' e Catholique de Louvain, Belgium, 1999.
- [15] X. Déecoret, F. Durand, F. X. Sillion, and J. Dorsey, "Billboard clouds for extreme model simplification," *Computer Graphics*, vol. 22, no. 3, pp. 689-696, 2003.
- [16] J. Denavit and R. S. Hartenberg, "A Kinematic notation for lower-pair mechanisms based on matrices," *Journal of Applied Mechanics* , vol. 22, no. 2, pp. 215-221, June 1955.
- [17] D. Freedman and T. Zhang, "Interactive graph cut based segmentation with shape priors," *Conference on Computer Vision and Pattern Recognition*, pp. 755-762, 2005.
- [18] D. B. Goldman, B. Curless, A.n Hertzmann, and S. M. Seitz, "Shape and Spatially-Varying BRDFs from Photometric Stereo," *International Conference on Computer Vision*, 2005.
- [19] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," *Computer Graphics*, pp. 43-54, 1996.
- [20] C. Gu and Lee, M. C., "Semi-automatic segmentation and tracking of semantic video objects," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 572-584, 1998.
- [21] R. I. Hartley, E. Hayman, L. de Agapito and I. D. Reid. "Camera calibration and the search for infinity," *International Conference on Computer Vision*, pp. 510-517, September 1999.
- [22] A. Hornung, and L. Kobbelt, "Hierarchical Volumetric Multi-view Stereo Reconstruction of Manifold Surfaces based on Dual Graph Embedding," *Conference on Computer Vision and Pattern Recognition*, 2006.
- [23] H. C. Huang, and Y. P.: Hung, "Panoramic Stereo Imaging System with Automatic Disparity Warping and Seaming," *Graphical Models and Image Processing*, vol. 60, no. 3, pp.196-208, 1998.
- [24] C. R. Huang, C. S. Chen and P. C. Chung, "Tangible Photo-Realistic Virtual Museum," *IEEE Computer Graphics and Applications*, vol. 25, no.1, pp. 15-17, 2005.
- [25] Y. P. Hung, C. S. Chen, Y. P. Tsai, and S. W. Lin, "Augmenting Panoramas with Object Movies by Generating Novel Views with Disparity-Based View Morphing," *Journal of Visualization and Computer Animation*, Special Issue on Hallucinating the Real World from Real Images, vol. 13, pp. 237-247, 2002
- [26] Y. P. Hung and Y. P. Tsai, "Trail-dependent intelligent scissors based on multi-scale image segmentation," *Asian Conference on Computer Vision*, pp. 539-544, 2002.
- [27] H. Kato and M. Billinghurst, "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferenciing," *2nd IEEE and ACM International Workshop on Augmented Reality, IWAR*, 1999.
- [28] A. Laurentini, "The visual hull concept for silhouette based image understanding." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16.2 (1994), 150-162.

- [29] M. Levoy and P. Hanrahan, "Light Field Rendering," *Computer Graphics*, pp.31-42, 1996.
- [30] S. Z. Li, *Markov Random Field Modeling in Computer Vision*, Springer-Verlag, Tokyo, 1995.
- [31] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," *Computer Graphics*, pp. 303–308, 2004.
- [32] W.-Y. Lo, Y.-P. Tsai, C.-W. Chen, and Y.-P. Hung, "Stereoscopic Kiosk for Virtual Museum," *International Computer Symposium*, 2004.
- [33] W. E. Lorensen and H. E. Cline, "Marching Cubes: a high resolution 3D surface construction algorithm," *Computer Graphics*, pp.163-169, 1987.
- [34] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [35] D.G. Lowe, "Fitting parameterized three-dimensional models to images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 5, pp. 441-450, 1991.
- [36] H. Luo and A. Eleftheriadis, "An interactive authoring system for video object segmentation and annotation," *Signal Processing: Image Communication*, vol. 17, p559-572, 2001.
- [37] S. P. Mallick, T. E. Zickler, D. J. Kriegman, and P. N. Belhumeur, "Beyond lambert: reconstructing specular surfaces using color," *Conference on Computer Vision and Pattern Recognition*, 2005.
- [38] B. Marcotegui, P. Correia, F. Marques, R. Mech, R. Rosa, M. Wollborn, F. Zanoguera "A Video Object Generator Tool Allowing Friendly User Interaction" *International Conference of Image Processing*, 1999.
- [39] S. J. Maybank and O. D. Faugeras, "A Theory of Self-Calibration of a Moving Camera," *International Journal of Computer Vision*, *International Journal of Computer Vision*, vol. 8. no. 2, pp. 123-152, Aug. 1992.
- [40] E. N. Mortensen and W. A. Barrett, "Intelligent scissors for image composition," *Computer Graphics*, pp.191–198, 1995.
- [41] E. N. Mortensen and W. A. Barrett, "Toboggan-based intelligent scissors with a four-parameter edge model," *Conference on Computer Vision and Pattern Recognition*, pp. 2452–2458, 1999.
- [42] S. Nayar, S. Nene, and H. Murase, "Subspace methods for robot vision," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 750-758, 1996.
- [43] M.T. Orchard, C.A. Bouman, "Color Quantization of Image," *IEEE Transactions on Signal Processing*, vol. 39, no. 12, pp. 2677-2690, 1991.
- [44] I. Patras, E. Hendriks, and I. Lagendijk, "Video Segmentation by MAP Labeling of Watershed Segments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.

23, no. 3, pp. 326-332, 2001.

[45] S. Peleg, and M. Ben-Ezra, "Stereo Panorama with a Single Camera," *Computer Vision and Pattern Recognition*, pp.395-401, 1999

[46] K. S. Roberts, "A New Representation for a Line," *Computer Vision and Pattern Recognition*, pp. 635-640, June 1988.

[47] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: interactive foreground extraction using iterated graph cuts," *Computer Graphics*, pp. 309–314, 2004.

[48] D. Ruprecht and H. Muller, "Image Warping with Scattered Data Interpolation", *IEEE Computer Graphics and Applications*, vol. 3, pp. 37-43. 1995.

[49] S. M. Seitz and C. M. Dyer, "Photorealistic scene reconstruction by voxel coloring," *Computer Vision and Pattern Recognition*, 1997.

[50] S. M. Seitz and C. M. Dyer, "View morphing," *Computer Graphics*, pp. 21–30, August 1996.

[51] J., Shade S. Gortler, L.-W. He, and R. Szeliski, "Layered depth images," *Computer Graphics*, pp. 231–242, July 1998.

[52] J. Shi and C. Tomasi, "Good features to track," *Computer Vision and Pattern Recognition*, 1994, pp.593–600.

[53] S. W. Shih, *Kinematic and Camera Calibration of Reconfigurable Binocular Vision Systems*, Ph.D. thesis, National Taiwan University, 1995.

[54] H. Y. Shum, and L. W. He, "Rendering with Concentric Mosaics," *Computer Graphics*, (1999) 299-306.

[55] H. Y. Shum and R. Szeliski, "Stereo Reconstruction from Multiperspective Panoramas," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 45-62, Jan 2004.

[56] H.-Y. Shum, S. B. Kang, S.-C. Chan, "Survey of Image-Based Representations and Compression Techniques," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 11, pp. 1020-1037, 2003.

[57] S. Tran, and L. Davis, "3D surface reconstruction using graph cuts with surface constraints," *European Conference on Computer Vision*, 2006

[58] R. Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 323-344, Aug. 1987.

[59] G. Turk and M. Levoy, "Zippered polygon meshes from range images," *Computer Graphics*, pp. 311-318, 1994

[60] L. Vincent and P. Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583-598, 1991.



- [61] G. Vogiatzis, C. Hernández, and R. Cipolla, “Reconstruction in the round using photometric normals,” CVPR 2006.
- [62] G. Vogiatzis, P. H. S. Torr, and R. Cipolla, “Multi-view stereo via volumetric graph-cuts,” *Computer Vision and Pattern Recognition*, 2005
- [63] D. Wang, “Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 539-546, 1998.
- [64] J. Wang and M. F. Cohen, “An iterative optimization approach for unified image segmentation and matting.” *International Conference on Computer Vision*, pp. 936-943, 2005.
- [65] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, “Surface light fields for 3D photography,” *Computer Graphics*, pp.287–296, 2000.
- [66] Z. Zhang, “A Flexible New Technique for Camera Calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, Nov 2000.
- [67] H. Zhong, L. Wenyin, S. Li, “Interactive Tracker – A Semiautomatic System for Video Object Segmentation,” *International Conference on Multimedia and Expo*, pp. 645-648, 2001.
- [68] Z. Zhu, E. M. Riseman, and A. R. Hanson, “Parallel-Perspective Stereo Mosaics,” *International Conference on Computer Vision*, pp. 345-352, 2001.
- [69] H. Zhuang, Z. S. Roth and F. Hamano, “A Complete and Parametrically Continuous Kinematic Model for Robot Manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 4, pp.451-463, Aug 1992.

# Appendix

## Singularity-Free Line Representation

Let a line, called **B-Line**, be in 3D space, and a plane, called **B-plane**, be perpendicular to the line and passes through the origin of reference coordinate system  $\{\mathbf{O}, \bar{x}, \bar{y}, \bar{z}\}$ , as shown in Figure A. Let  $\bar{\mathbf{b}} = (b_x, b_y, b_z)$  be the unit vector along **B-Line** and lie in the upper half-space of  $\{\mathbf{O}, \bar{x}, \bar{y}, \bar{z}\}$  coordinate system, where  $b_x$  and  $b_y$  are the x and y components of the  $\bar{\mathbf{b}}$  vector defined on the  $\{\mathbf{O}, \bar{x}, \bar{y}, \bar{z}\}$  coordinate system. Note that  $b_z$ , the z component of the direction unit vector  $\bar{\mathbf{b}}$  can be obtained by equation A.1, if  $b_x$  and  $b_y$  are known,

$$b_z = (1 - b_x^2 - b_y^2)^{1/2} \quad (\text{A.1})$$

Therefore the  $b_x$  and  $b_y$  is enough to represent the orientation of **B-Line**.

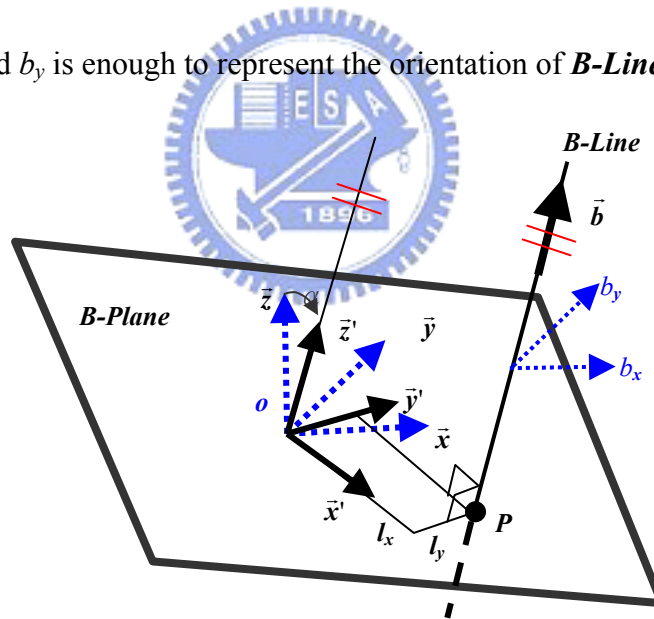


Fig. A. Representation of a line B in 3D space.

Next, we define another coordinate system  $\{\mathbf{O}, \bar{x}', \bar{y}', \bar{z}'\}$ , where  $\bar{x}'$  and  $\bar{y}'$  are 2-D Cartesian coordinate system defined on the **B-plane**, passing through the same origin  $\mathbf{O}$  of  $\{\mathbf{O}, \bar{x}, \bar{y}, \bar{z}\}$  coordinate system. Let the unit vector  $\bar{\mathbf{k}}$  denote the common normal of  $\bar{z}$  and  $\bar{\mathbf{b}}$ . Thus, we have

$$\begin{aligned}\bar{\mathbf{k}} &= \frac{\bar{\mathbf{z}} \times \bar{\mathbf{b}}}{\|\bar{\mathbf{z}} \times \bar{\mathbf{b}}\|}, \\ \bar{\mathbf{z}} \times \bar{\mathbf{b}} &= [0 \quad 0 \quad 1]^t \times [b_x \quad b_y \quad b_x]^t = -b_y \bar{\mathbf{i}} + b_x \bar{\mathbf{j}}, \\ \therefore \bar{\mathbf{k}} &= \left[ \begin{array}{ccc} \frac{-b_y}{\sqrt{b_x^2 + b_y^2}} & \frac{b_x}{\sqrt{b_x^2 + b_y^2}} & 0 \end{array} \right]^t\end{aligned}\tag{A.2}$$

where  $\|\cdot\|$  denotes the Euclidean norm. Thus the  $\{\mathbf{O}, \bar{\mathbf{x}}', \bar{\mathbf{y}}', \bar{\mathbf{z}}'\}$  coordinate system is obtained by rotating an angle  $\alpha$  about the axis  $\bar{\mathbf{k}}$ , where

$$\alpha = \cos^{-1}(\bar{\mathbf{z}} \cdot \bar{\mathbf{b}}) = \cos^{-1}(b_z)\tag{A.3}$$

Let  $\mathbf{R}$  be the rotation matrix  $\mathbf{Rot}(\bar{\mathbf{k}}, \alpha)$ , and it can be calculated by equation (A.4).

$$\begin{aligned}\mathbf{R} &= \mathbf{Rot}(\bar{\mathbf{k}}, \alpha) \\ &= \begin{bmatrix} k_x^2(1 - \cos \alpha) + \cos \alpha & k_x k_y(1 - \cos \alpha) - k_z \sin \alpha & k_x k_z(1 - \cos \alpha) + k_y \sin \alpha & 0 \\ k_x k_y(1 - \cos \alpha) + k_z \sin \alpha & k_y^2(1 - \cos \alpha) + \cos \alpha & k_y k_z(1 - \cos \alpha) - k_x \sin \alpha & 0 \\ k_x k_z(1 - \cos \alpha) - k_y \sin \alpha & k_y k_z(1 - \cos \alpha) - k_x \sin \alpha & k_z^2(1 - \cos \alpha) + \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 - \frac{b_x^2}{1 + b_z} & \frac{-b_x b_y}{1 + b_z} & b_x & 0 \\ \frac{-b_x b_y}{1 + b_z} & 1 - \frac{b_y^2}{1 + b_z} & b_y & 0 \\ -b_x & -b_y & b_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\end{aligned}\tag{A.4}$$

