

**TOWARDS ROBUST VOXEL-COLORING: HANDLING CAMERA
CALIBRATION ERRORS AND PARTIAL EMPTINESS OF
SURFACE VOXELS**

Zeeshan Anwar

Departement of Electrical Engineering
McGill University, Montréal

August 2005

A Thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of
Master of Engineering

© ZEESHAN ANWAR, 2005



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-24935-2

Our file Notre référence

ISBN: 978-0-494-24935-2

NOTICE:

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

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

AVIS:

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

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

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

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

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

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


Canada

ABSTRACT

In this thesis, we present two new methods to overcome the effects of both camera calibration errors and partial emptiness of surface voxels in voxel-coloring. Voxel-coloring is a relatively new volumetric method for 3D scene reconstruction from multiple calibrated views. The quality of reconstruction is affected by the presence of errors in the estimated calibration parameters. Furthermore, a voxel forming a scene surface may be partially empty as there is no prior knowledge about the scene surface. Both of these sources of error result in outlier pixels in voxel projections in the input images. These outlier pixels affect the photo-consistency test of the voxel and tend to result in over-carving of the reconstructed 3D scene. The existing methods to handle these errors are either insufficient or too complex. We propose a method to handle the effect of calibration errors and call it Adaptive Gaussian Averaging. It makes use of the information of the error probability distribution of projected pixel coordinates due to camera parameter errors. We propose another method to reduce the effects of partial emptiness of surface voxels and we call it Area Weighting. In this method we use the pixel count in voxel projections to weight the projections in voxel's color statistics calculations. Our proposed methods are simple and can be incorporated into the existing voxel-coloring algorithms easily. We also conduct experiments on our own calibrated data sets to verify the effectiveness of the proposed methods. The experimental results show that both of our proposed methods have the ability to improve the results of existing algorithms. We also compare the results of our proposed methods with the results of an existing method that handle these errors too, the r-Consistency. We find that our proposed methods have the ability to adapt to the level of errors present in the system, and perform better than r-Consistency when the effect of these errors is higher on voxel-coloring.

RÉSUMÉ

Cette thèse présente deux nouvelles méthodes de reconstruction tridimensionnelle par coloration de voxels. Elles ont été développées dans le but de contourner les effets des erreurs de calibration de la caméra, ainsi que des erreurs causées par les voxels de surface qui peuvent être partiellement vides. L'approche par coloration de voxels est relativement nouvelle pour la reconstruction de scènes à partir de vues multiples et calibrées. La qualité de la reconstruction est affectée par la présence d'erreurs dans l'estimation des paramètres de calibration. De plus, un voxel délimitant la surface de la scène peut être partiellement vide puisqu'il n'existe aucune connaissance *a priori* sur la surface. Ces deux sources d'erreurs entraînent la présence de pixels aberrants lors de la projection des voxels vers les images d'entrée. Ces pixels ont un impact sur le test de photocoherence du voxel, et peuvent provoquer un surtaillage de la scène tridimensionnelle. Les procédés existants d'inclusion de ces erreurs sont en général insuffisants ou trop complexes. La méthode que nous proposons, le moyennage gaussien adaptatif, tient compte des erreurs de calibration. Elle utilise l'information disponible sur la distribution des erreurs dans les coordonnées projetées, qui sont elles-mêmes provoquées par des erreurs de paramètres de la caméra. Nous proposons également une autre méthode pour la réduction des effets de vide partiel de voxels de surface, la pondération de la superficie. Cette approche utilise le nombre de pixels obtenus lors des projections d'un voxel afin de pondérer ces projections lors du calcul des statistiques de couleur du voxel. Les deux techniques proposées sont simples et peuvent être incorporées à des algorithmes de coloration de voxels déjà existants. Des expérimentations ont été effectuées sur des données calibrées afin de vérifier l'efficacité des deux procédés. Les résultats expérimentaux démontrent que les deux méthodes permettent d'améliorer les résultats d'algorithmes existants. Nous comparons de plus nos résultats avec ceux d'une autre technique d'intégration des erreurs, la r-cohérence. Il est démontré que les méthodes proposées peuvent s'adapter à la quantité d'erreurs dans le système, et surpassent la r-cohérence dans les cas où la coloration de voxel se trouve grandement affectée par ces erreurs.

ACKNOWLEDGEMENTS

First of all, I am grateful to my Lord, the most Exalted and the Almighty, for giving me strength out of His mercy to finish the work of this thesis. I am thankful to my sweet and kind supervisor, Frank Ferrie, for his help and guidance in my studies and in the work for this thesis. I pay my gratitude to the Government of Canada whose scholarship made it possible for me to come and study in Canada. I am thankful to Ms. Diane Cyr, the program officer of Canadian Commonwealth Scholarship and Fellowship Program, for her generous and prompt help regarding the scholarship issues. I also acknowledge the help of Yasha, Alexandre and Andrew in acquiring the data sets and their calibration. I am thankful to Matthew Loper for using the framework of his voxel-coloring software, that helped to reduce the total programming time required for the experimental setup. I would like to thank Bruce Culbertson of HP labs and Rahul Bhotika of GE Global Research labs for a number of helpful discussions and for the data sets that I used in my initial experiments.

I am really grateful to all of my nice colleagues of Artificial Perception Lab for their help and encouragement, and for adding fun to my life during the work for this thesis. In particular, I would like to thank Prasun, Isabelle, Matt, Rupert, Frank and Cathy for helping me in correcting my thesis chapters. I also acknowledge the helpful discussions with Matt and the useful insights offered by Prasun to improve my work. And finally, it would be unfair not to thank Prakash for making my life joyful in the stressful moments of work by giving me both company and ride to Indian restaurants and movie theaters.

List of Acronyms

| | |
|----------------|--|
| VoM | Variance of Means |
| AWVoM | Area Weighted Variance of Means |
| AGA | Adaptive Gaussian Averaging |
| MVPS | Maximum Voxels Per Side |
| STD | Standard Deviation |
| GVC | Generalized Voxel Coloring |
| GVC-IB | Generalized Voxel Coloring with Item Buffers |
| GVC-LDI | Generalized Voxel Coloring with Layered Depth Images |
| Pr | Probability |

TABLE OF CONTENTS

| | |
|---|------|
| ABSTRACT | ii |
| RÉSUMÉ | iii |
| ACKNOWLEDGEMENTS | iv |
| List of Acronyms | v |
| LIST OF FIGURES | ix |
| LIST OF TABLES | xiii |
| CHAPTER 1. Introduction | 1 |
| 1. An Introduction to Voxel-Coloring | 2 |
| 1.1. Photo-Consistency | 3 |
| 1.2. The Visibility Principle | 3 |
| 1.3. A Generic Voxel Coloring Algorithm | 3 |
| 1.4. Photo Hull | 4 |
| 1.5. Advantages of Voxel-Coloring | 5 |
| 1.6. Limitations of Voxel-Coloring | 5 |
| 2. Sources of Error in Voxel-Coloring | 6 |
| 3. Advancements in Voxel Coloring Algorithms | 6 |
| 3.1. Voxel Visibility Calculation | 7 |
| 3.2. Photo-Consistency Tests | 8 |
| 3.3. Exploring New Dimensions in Voxel-Coloring | 10 |
| 4. Handling Camera Calibration Errors | 10 |
| 5. Handling Partial Emptiness of Surface Voxels | 11 |
| 6. Contributions | 12 |
| 7. Thesis Outline | 13 |

| | |
|---|----|
| CHAPTER 2. Handling Camera Calibration Errors in Voxel-Coloring | 14 |
| 1. The Effect of Camera Calibration Errors on Voxel-Coloring | 14 |
| 2. Related Work | 15 |
| 2.1. Approximate N-View Stereo | 16 |
| 2.2. Limitations of r-Consistency | 18 |
| 3. Adaptive Gaussian Averaging | 18 |
| 3.1. Probabilistic Modelling of the Effect of Calibration Errors in Voxel-coloring | 18 |
| 3.2. Local Probability Distribution of Pixel Color Intensity in the Presence of Sensor Noise | 20 |
| 3.3. The Distribution of the Projected Pixel Coordinates Error Due to Calibration Errors | 20 |
| 3.4. Uncertainty in a Voxel Projection Color due to Gaussian Calibration Error | 21 |
| 3.5. Multiple Pixel Voxel Projections | 24 |
| CHAPTER 3. Handling Partial Emptiness of Surface Voxels | 26 |
| 1. Partial Emptiness of Surface Voxels: The Problem and its Effect | 26 |
| 2. Related Work | 29 |
| 2.1. Roxels: Responsibility Weighted 3D Volume Reconstruction | 29 |
| 2.2. Stereo Matching with Transparency and Matting | 29 |
| 2.3. r-Consistency | 30 |
| 2.4. Other Related Techniques | 31 |
| 3. Area Weighting to Handle Partial Emptiness of Surface Voxels | 32 |
| 3.1. Area Weighted Variance of Means Photo-Consistency Test | 33 |
| 3.2. Advantages of Area Weighting | 34 |
| 3.3. Limitations of Area Weighting | 34 |
| CHAPTER 4. Experiments and Results | 36 |
| 1. Experimental Setup | 36 |
| 1.1. Data Sets | 36 |
| 1.2. Camera Calibration | 38 |
| 1.3. Voxel-Coloring Algorithm | 40 |
| 1.4. Quantitative Verification of the Test Results | 41 |
| 2. Results for the use of Area Weighting | 42 |
| 2.1. Low Voxel Resolution: MVPS 100 | 46 |
| 2.2. High Voxel Resolution: MVPS 180 | 49 |
| 3. Results for Handling Camera Calibration Errors | 49 |

TABLE OF CONTENTS

| | |
|--|----|
| 3.1. The Effect of Calibration Errors on Voxel-Coloring | 51 |
| 3.2. Improvement in the Voxel-Coloring Results with Adaptive Gaussian Averaging . | 53 |
| 3.3. Experimental Comparison of Adaptive Gaussian Averaging with r-Consistency . . | 65 |
| 3.4. Significance of the Error in the Reconstruction Results | 66 |
| CHAPTER 5. Conclusion | 70 |
| APPENDIX A. Estimation of Pixel Coordinates Error Variance | 74 |
| APPENDIX B. Tables for Quantitative Error Analysis | 77 |
| 1. Quantitative Results for the Use of Area Weighting | 77 |
| 2. Quantitative Error Analysis for Handling Calibration Errors | 79 |
| REFERENCES | 83 |

LIST OF FIGURES

| | |
|---|----|
| 1.1 Photo-Consistency Principle: Left Image shows that an empty voxel projects to different colors in different images. Right image shows that the voxel projections on the scene surface have same color in all the visible images. | 3 |
| 1.2 Visibility Principle: When the initial Volume V is carved to the new volume V' , the number of cameras that can see the point p increases to four. This figure is taken from [18]. . . . | 4 |
| 1.3 Photo Hull: Minimally constrained 3D reconstruction | 5 |
| 2.1 Effect of calibration errors on the projections of a voxel is to shift the projected pixels from their true locations. The shift in the projected pixel coordinates may be different in the different images. | 15 |
| 2.2 The effect of r-Shuffle on the reprojected image is in the form of blurring, taken from [16] | 16 |
| 2.3 The comparison of r-Consistency and photo-consistency, taken from [16] | 17 |
| 2.4 The concept of Adaptive Gaussian Averaging. The projection window size is increased to calculate the Gaussian averaging according to the effect of calibration errors in that part of the image. | 22 |
| 3.1 The effect of partial emptiness of a surface voxel. Almost half of the center voxel V_2 is empty as the scene surface is passing through it. It is clear that the projection of the center voxel V_2 in images I_1 and I_2 contains the outlier pixels representing the scene surface of the neighboring voxels V_1 and V_3 | 27 |
| 3.2 The effect of partial emptiness of an edge voxel. Note that most of the pixels in the voxel projection in I_1 represent the surface contained by the voxel V_3 instead of V_2 | 28 |
| 3.3 The use of Area Weighting to handle the effects of partially occluded voxels. | 35 |
| 3.4 An example when Area Weighting is unable to overcome the effect of outliers as all the projections of a voxel have similar areas. | 35 |

| | |
|---|----|
| 4.1 A single image of each 3D test objects | 37 |
| 4.2 Twelve different views of Piper in a plane at a uniform interval of 30° | 38 |
| 4.3 Comparison of Photo Hull and Visual Hull. | 39 |
| 4.4 Pixels inside the object silhouette were used for the calculation of reprojection error. Colored pixels outside the silhouette of the object were considered as false positive pixels. | 43 |
| 4.5 Reconstruction results that show the effectiveness of Area Weighting for the Piper data set for two different voxel resolutions, i.e. Maximum Voxels Per Side (MVPS). The corresponding analytical comparison is shown in Figure 4.6. | 44 |
| 4.6 Quantitative comparison of the results shown in Figure 4.5 for the Piper data set. The corresponding numbers are given in Table B.1 in Appendix B. | 45 |
| 4.7 The effect of changing the variance threshold on reprojection error and false positive pixel count. These are the results of a VoM test for the Piper data set with MVPS of 180. The exact numbers are given in Table B.1 in Appendix B. | 45 |
| 4.8 Reconstruction results showing the effectiveness of Area Weighting for the Chipmunk data set. | 46 |
| 4.9 Quantitative comparison of results shown in Figure 4.8 for the Chipmunk data set. The exact numbers are given in Table B.2 in Appendix B. | 47 |
| 4.10 Quantitative comparison of results shown in Figure 4.11 for the Puppy data set. The exact numbers are given in Table B.3 in Appendix B. | 47 |
| 4.11 Reconstruction results showing the effectiveness of Area Weighting for the Puppy data set. | 48 |
| 4.12 This figure shows the effect of leaving the single visibility voxels uncolored. It is clear that this method helps to make the holes in the reconstruction results visually more recognizable. The associated quantitative results are given in Table 4.2. | 50 |
| 4.13 Reconstruction results of the AWVoM based voxel-coloring algorithm for different levels of calibration error for two different voxel resolutions for the Piper data set. | 52 |
| 4.14 Quantitative analysis of the effect of increasing calibration error on the reconstruction results shown in Figure 4.13 for the Piper data set. The exact numbers are given in Table B.4 in Appendix B. | 53 |
| 4.15 Piper data set, MVPS = 100. The effect of using AGA with AWVoM to overcome calibration error. | 55 |
| 4.16 Piper data set, MVPS = 180. The effect of using Adaptive Gaussian Averaging with AWVoM to overcome calibration error. | 56 |

| | | |
|------|---|----|
| 4.17 | Quantitative comparison of results shown in Figure 4.15 for the Piper data set for MVPS of 100. The exact numbers are given in Table B.5 in Appendix B. | 57 |
| 4.18 | Quantitative comparison of results shown in Figure 4.16 for the Piper data set for MVPS of 180. The exact numbers are given in Table B.6 in Appendix B. | 57 |
| 4.19 | The effect of using AGA with AWVoM to overcome the calibration errors for the Piper data set. The holes in the reconstruction were NOT been made explicit for these results. | 58 |
| 4.20 | Quantitative comparison of the results shown in Figure 4.19 for the Piper data set. The exact numbers are given in Table B.7 in Appendix B. | 58 |
| 4.21 | Reconstruction results to show the effect of using Adaptive Gaussian Averaging for the Chipmunk data set with average standard deviation of error in pixel coordinates = 0.2 pixels. | 59 |
| 4.22 | Reconstruction results to show the effect of using Adaptive Gaussian Averaging for the Chipmunk data set with average standard deviation of error in pixel coordinates = 2 pixels. | 60 |
| 4.23 | Quantitative comparison of the results shown in Figure 4.21 for the Chipmunk data set with average standard deviation of error in pixel coordinates = 0.2 pixels. The exact numbers are given in Table B.8 in Appendix B. | 61 |
| 4.24 | Quantitative comparison of the results shown in Figure 4.22 for the Chipmunk data set with average standard deviation of error in pixel coordinates = 2 pixels. The exact numbers are given in Table B.9 in Appendix B. | 61 |
| 4.25 | Reconstruction results to show the effect of using Adaptive Gaussian Averaging for the Puppy data set with average standard deviation of error in pixel coordinates = 0.2 pixels. | 62 |
| 4.26 | Reconstruction results to show the effect of using Adaptive Gaussian Averaging for the Puppy data set with average standard deviation of error in pixel coordinates = 2 pixels. | 63 |
| 4.27 | Quantitative comparison of the results shown in Figure 4.25 for the Puppy data set with average standard deviation of error in pixel coordinates = 0.2 pixels. The exact numbers are given in Table B.10 in Appendix B. | 64 |
| 4.28 | Quantitative comparison of the results shown in Figure 4.26 for the Puppy data set with average standard deviation of error in pixel coordinates = 2 pixels. The exact numbers are given in Table B.11 in Appendix B. | 64 |
| 4.29 | Piper data set, MVPS = 100. Reconstruction results comparison of r-Consistency and AWVoM with AGA. Holes are being explicitly shown in these reconstructions. | 67 |

| | |
|---|----|
| 4.30Piper data set, MVPS = 180. Reconstruction results comparison of r-Consistency and AWVoM with AGA. Holes are being explicitly shown in these reconstructions. | 68 |
| 4.31Piper data set, MVPS = 100. Quantitative comparison of the results of r-Consistency with AWVoM using AGA shown in Figure 4.29. The exact numbers are given in Table B.12 in Appendix B. | 69 |
| 4.32Piper data set, MVPS = 180. Quantitative comparison of the results of r-Consistency with AWVoM using AGA shown shown in Figure 4.30. The exact numbers are given in Table B.13 in Appendix B. | 69 |

LIST OF TABLES

| | |
|---|----|
| 4.1 Dimensions of the initial bounding box for each test object | 37 |
| 4.2 The effect of leaving the single visibility voxels uncolored on the reprojection error and the false positive pixel count. | 50 |
| B.1 Quantitative error analysis of Area Weighting for the Piper data set | 77 |
| B.2 Quantitative error analysis of Area Weighting for the Chipmunk data set | 78 |
| B.3 Quantitative error analysis of Area Weighting for the Puppy data set | 78 |
| B.4 The effect of increasing calibration error on reconstruction quality of AWVoM based voxel-coloring algorithm | 79 |
| B.5 Quantitative assessment of using Adaptive Gaussian Averaging to overcome calibration errors for MVPS = 100 for the Piper data set | 79 |
| B.6 Quantitative assessment of using Adaptive Gaussian Averaging to overcome calibration errors for MVPS = 180 for the Piper data set | 80 |
| B.7 Result of using Adaptive Gaussian Averaging without explicitly showing holes for the Piper data set. Average pixel coordinates error STD = 2pixels | 80 |
| B.8 Quantitative assessment of using Adaptive Gaussian Averaging to overcome calibration errors for the Chipmunk data set. Average pixel coordinates error STD = 0.2 pixels | 80 |
| B.9 Quantitative assessment of using Adaptive Gaussian Averaging to overcome calibration errors for the Chipmunk data set. Average pixel coordinates error STD = 2 pixels | 81 |
| B.10 Quantitative assessment of using Adaptive Gaussian Averaging to overcome calibration errors for the Puppy data set. Average pixel coordinates error STD = 0.2 pixels | 81 |
| B.11 Quantitative assessment of using Adaptive Gaussian Averaging to overcome calibration errors for the Puppy data set. Average pixel coordinates error STD = 2 pixels | 81 |

| | | |
|------|---|----|
| B.12 | Quantitative comparison of reconstruction results of r-Consistency with Adaptive Gaussian | |
| | Averaging for the Piper data set. MVPS = 100 | 82 |
| B.13 | Quantitative comparison of reconstruction results of r-Consistency with Adaptive Gaussian | |
| | Averaging for the Piper data set. MVPS = 180 | 82 |

CHAPTER 1

Introduction

3D shape recovery of an object from its images is a fundamental problem in computer vision. The reconstructed shape has abundant uses ranging from the basic applications in robotic navigation and object detection to more advanced applications such as building virtual reality environments. A lot of work has already been done in this area, and different methods have been proposed to get shape from images. These methods are in general called “Shape from X”, where X signifies the underlying principle being used for shape recovery. One of these methods is stereo vision that uses more than one view of the same object taken from different positions. The computational stereo vision, inspired by human vision, was presented by Marr and Poggio in 1979 in [21] for two cameras. Stereo vision has been and is still a hot topic of research in the computer vision community. It is the most commonly used method for shape recovery in practical applications because of its simplicity and reliability. The quality of reconstruction of a stereo vision algorithm can be improved by increasing the number of input views. Due to the recent advancement in the sensor technology in terms of imaging quality and low price, the use of more than just two cameras in stereo vision is becoming popular. The shape reconstruction using only two cameras is sometimes referred as $2\frac{1}{2}D$ as we can only recover the part of the scene visible from both cameras. Therefore, another advantage of using multiple cameras is to get true 3D shape recovery for all regions of the scene. The complete 3D reconstructed models of the scene using multiple cameras are especially useful in object modelling for virtual reality environments [13] and 3D photography.

Traditional stereo reconstruction methods work in image-space. The first step in these methods is to build a disparity map using image correspondence. The next step is to use the camera information to get the shape information from this disparity map. Occlusions and image correspondence are the two main issues faced by traditional stereo. One recent advancement in stereo vision has been to use object-space methods to avoid these problems. These methods are sometimes referred as volumetric stereo vision. These methods work especially well if an approximate

guess of the volume containing the object is already available. Detailed discussions of volumetric methods for 3D scene reconstruction from multiple views can be found in [28, 29, 9]. The most recent of these are voxel-based volumetric reconstruction methods. These methods start from an initial approximate volumetric guess of the 3D scene. The volume is represented as a collection of discrete volume elements called *voxels*. Voxel Coloring [26] and Space Carving [18] are the two most well-known algorithms for voxel-based volumetric reconstruction. Although *Voxel Coloring* was the algorithm proposed by Seitz [27], voxel-based methods for photo-consistent reconstruction of 3D scenes are also known as *voxel-coloring* in general. The reason for this name is that these methods make use of only the color information present in the images. In this thesis, the term *voxel-coloring* will be used as synonym for these methods.

The reconstruction quality of voxel-coloring suffers from a number of error sources, two of which are the camera calibration errors and the partial emptiness of surface voxels. In this thesis, we propose methods to overcome the effect of each these two error sources in voxel-coloring. We propose a method called Adaptive Gaussian Averaging that makes use of the available calibration error distribution information to overcome calibration errors effects. We also propose a simple method called Area Weighting to reduce the effects of partial emptiness of surface voxels on the reconstruction. Our proposed methods are simple and can be used with the existing voxel-coloring algorithms to improve the their reconstruction quality in the presence of these errors.

This chapter is a brief introduction to voxel-coloring which will clarify the concepts used in the rest of the thesis. The last part of this chapter is an overview of the contributions made through this thesis work.

1. An Introduction to Voxel-Coloring

A 3D reconstruction of a scene from its images is called Photorealistic, Photo-Consistent or Color-Consistent if the reprojection of the reconstructed model in the input views can exactly reproduce the input images. Voxel-coloring is based on the concept of photo-consistency and visibility. In voxel-coloring, each surface voxel is checked for the photo-consistency of it's projections in the input images where it is visible. If a voxel is a part of the 3D scene surface, it will project to the pixels of same color in all the visible images. On the other hand, a voxel that is either in the empty space or inside the scene surface, but not on the scene surface, will project to pixels of different colors in the visible images. Therefore, the projections of an empty voxel will not be photo-consistent and it will be removed from the voxelized volume. The concept of photo-consistency becomes clear from Figure 1.1. Once all the non-photo-consistent voxels have been removed, the resulting volume will be a photo-consistent reconstruction of the original 3D scene. Complete calibration information

of the input views is a precondition to voxel-coloring because it is required to accurately calculate visibility and photo-consistency. The main concepts in voxel-coloring are described below and are based on the theory of voxel-coloring presented in [26] and [18].

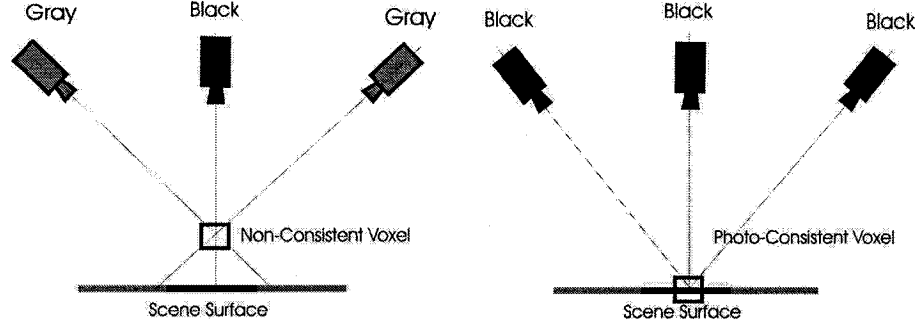


FIGURE 1.1. Photo-Consistency Principle: Left Image shows that an empty voxel projects to different colors in different images. Right image shows that the voxel projections on the scene surface have same color in all the visible images.

1.1. Photo-Consistency.

A point p on a 3D shape surface S is called photo-consistent with a set of input images I_1, \dots, I_k , if it is visible in at least one of the images, and its projections in all visible images have same color. A 3D shape is called photo-consistent with a set of input images if all the points on its surface are photo-consistent with the input images. A photo-consistency test is required for this purpose. If a 3D point or voxel p is found to be photo-consistent in k images by a photo-consistency test C_k , it is called monotonic if p is consistent in all subsets of these k input images, with the size of each subset being $k - 1$.

1.2. The Visibility Principle.

Let p be a point on a volume V 's surface, $Surf(V)$, and $VIS_V(P)$ be the collection of images where p is visible. If $V' \subset V$ is a volume that also has p on its surface, then $VIS_{V'}(P) \supseteq VIS_V(P)$ [18]. The phenomenon of an increase in the number of visible cameras with a reduction in size of the 3D volume during voxel-coloring is shown in Figure 1.2.

1.3. A Generic Voxel Coloring Algorithm.

Step 1: Initialize with an initial volume V that is guaranteed to contain the scene surface

Step 2: Repeat the following two steps for each surface voxel $v \in Surf(V)$ until a non-photo-consistent voxel is found:

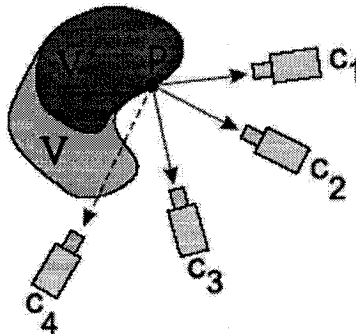


FIGURE 1.2. Visibility Principle: When the initial Volume V is carved to the new volume V' , the number of cameras that can see the point p increases to four. This figure is taken from [18].

- i Project v on all the input images where it is not occluded by $V-v$. Let v be visible in j images and Col_1, \dots, Col_j be the color of the pixels in v 's projections in these images
- ii Determine the consistency of v using a suitable photo-consistency function $C_j(Col_1, \dots, Col_j)$

Step 3: If all the voxels are found photo-consistent, set $V^* = V$ and terminate with V^* as the photo-consistent surface. Otherwise remove v from V , i.e. $V = V - v$, and repeat Step 2.

To get a photo-consistent surface at the end of this algorithm, it is important that the photo-consistency test is monotonic. As the non-consistent voxels are removed or carved from the initial volume V , the number of cameras that can see a voxel increases in general as shown in Figure 1.2. The monotonicity requirement of the test will make sure that if a voxel is found to be non-consistent and is removed at one stage of the algorithm, it cannot be deemed photo-consistent at a later stage with a change in its visibility.

1.4. Photo Hull.

The reconstructed 3D surface that is a result of voxel-coloring is the superset of all the photo-consistent surfaces that can reproduce the input images by reprojection. This is the tightest bound on the recovered 3D surface from multiple images in the absence of any prior information about the scene with arbitrary location of the cameras, under arbitrary lighting conditions, and using only the color statistics. This minimally constrained 3D reconstruction of the scene is called the Photo Hull [18]. Figure 1.3 explains the concept. It is clear that the reconstructed surface has a different shape and is fatter than the original surface, which is in the form of a square. This

deformation is mainly due to the textureless regions and the absence of any prior constraint on the scene geometry.

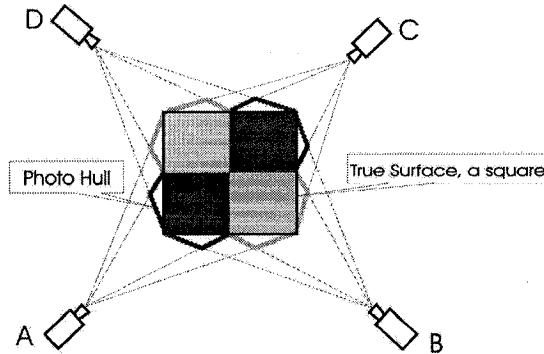


FIGURE 1.3. Photo Hull: Minimally constrained 3D reconstruction

1.5. Advantages of Voxel-Coloring.

Voxel-coloring has the following advantages over other stereo algorithms for the reconstruction from multiple images:

- The Photo Hull, that is the result of voxel-coloring, provides an upper-bound on the quality of 3D reconstruction from N input images of the scene in the absence of any prior assumptions on both the scene shape, lighting and camera distribution.
- Per pixel correspondence, which is still a difficult problem to solve in traditional stereo vision, is not required in voxel-coloring.
- The occluding voxels are handled before the occluded voxels in voxel-coloring. This helps to avoid the problem of occlusions faced by other wide-baseline stereo algorithms.
- The 3D resolution of the reconstructed model can be easily adjusted by changing the number of voxels being used to represent the initial volume containing the scene.

1.6. Limitations of Voxel-Coloring.

Main limitations of voxel-coloring methods are as follows:

- Voxel-coloring algorithms require larger memory for data storage and are slow in terms of processing speed. This problem gets worse at high voxel resolutions.
- Most of the voxel-coloring algorithms are sequential, i.e. the voxels that are closer to the cameras are checked first for their photo-consistency. Therefore, an error made in deciding the photo-consistency of a voxel may affect the visibility of subsequent voxels [19].

2. Sources of Error in Voxel-Coloring

- (i) **Camera calibration errors:** An underlying assumption in voxel-coloring is that accurate camera calibration information is available for all the views. However, the estimated camera parameters inherently have some errors in them. Camera calibration errors shift a voxel projection from its original position in the input images. As a result, even consistent voxels can get carved resulting in holes in the reconstructed 3D model [16, 5, 19].
- (ii) **Non-Lambertian and transparent surfaces:** Another assumption often made in voxel-coloring is that the 3D scenes are completely opaque and have Lambertian surface characteristics. However, these assumptions are not fulfilled completely for all parts of the 3D scene. The effect of this error is to have local artifacts in the reconstructed model [19].
- (iii) **Multiple pixel projections:** In an ideal voxel-coloring algorithm, it is assumed that each 3D voxel represents a point in 3D space and will project to only a single pixel in each image. However, depending on the resolution of the voxel grid being used to model the scene volume, a voxel projection may contain multiple pixels. This effect adds uncertainty to the photo-consistency tests making it more difficult, and hence introduces errors in the reconstructed 3D model [26, 16, 19, 5].
- (iv) **Partially empty surface voxels:** Due to the non-zero size of the voxels, a voxel that represents the scene surface may be only partially filled. This problem is more common for boundary voxels at surface discontinuities. The effect of this error is the inclusion of outlier pixels in voxel projections. As with the camera calibration errors, the outliers may lead to an error in the photo-consistency test and a consistent voxel might be declared non-consistent [33, 8].
- (v) **Sensor noise:** Sensor noise may also effect the color of pixels in voxel projections. In most of the voxel-coloring literature, sensor noise is modelled as zero mean additive Gaussian noise [26, 18].

3. Advancements in Voxel Coloring Algorithms

The basic theory of voxel-coloring was first proposed by Seitz and Dyer in [27] in 1997 and one year later was expanded by Kutulakos and Seitz in [17]. Since then, a good amount of work has been done in this area to improve the quality of reconstruction and to increase its use in other situations that were not proposed in the original work, e.g. non-Lambertian surface and moving scenes. A review of these advances is presented here as an exposition of the state of the art.

3.1. Voxel Visibility Calculation.

Most voxel-coloring methods follow the generic algorithm presented above, i.e. first calculating the visibility of a voxel in the input images and then using the visibility information to check its photo-consistency. The visibility of a voxel at a particular stage in the voxel-coloring algorithm depends on the 3D shape formed by the rest of the voxels. Whenever a voxel is removed for being non-photo-consistent, it affects the visibility of the remaining voxels. Visibility calculation is the most computationally expensive part of a voxel-coloring algorithm. Therefore, to reduce the computational cost, Seitz in [27] proposes an Ordinal Visibility Constraint on the position of the cameras. According to this constraint, no scene point can be a part of the convex hull of the camera centers, i.e. either the scene surface lies completely outside of the convex hull of camera centers or completely inside of it. This condition results in a single sweep voxel-carving algorithm that is nearly real-time. In this method a plane is passed through the voxel volume. Due to the Ordinal Visibility Constraint, only the cameras on one side of the plane can see the voxels on the plane. This reduction in the number of cameras that can see a particular voxel makes the visibility calculation simpler and hence less time consuming. This constraint was removed by Kutulakos in [18] to extend the use of voxel-coloring for arbitrary camera placements. Kutulakos in [18] proposes a multi-sweep algorithm named Space Carving. In this algorithm, instead of a single plane, six planes are passed through the voxel volume along orthogonal axes (-X, X, -Y, Y, -Z, Z). Culbertson in [7] proposes a new algorithm, Generalized Voxel Coloring (GVC), to calculate the visibility without using the sweeping planes. He makes use of the volumetric rendering feature of OpenGL [1] for this purpose. In the first variant of this algorithm, GVC-IB, the ID of a voxel is stored for each pixel of all the input images that can see this voxel. The generated images with pixel colors equal to the ID of the visible voxel are called Item Buffers (IBs). Once a non-photo-consistent voxel is removed, this method requires the recalculation of the visibility of all the remaining voxels. In the second algorithm, GVC-LDI, all the surface voxels in the line of sight of a pixel are stored, in the order of their distance from the pixel, for each pixel of all the input images. This method of storing visibility information is very similar to Layered Depth Image (LDI). Therefore, this algorithm is called GVC-LDI. The advantage of GVC-LDI over GVC-IB is that when a non-photo-consistent voxel is removed, it is easier to identify the voxels whose visibility has been changed and we need to update the visibility of only those voxels. Therefore, GVC-LDI poses less computational cost for visibility updates. However, this is achieved at the expense of larger memory required to store LDIs. A different approach has been proposed by de Bonet in [8] for visibility calculations. In this method, all the voxels are initially assumed to be partially opaque. The visibility and the opacity estimates are optimized in successive iterations.

3.2. Photo-Consistency Tests.

It is considered difficult voxel-coloring to have a good photo-consistency test that can be used for all types of scenes. Additional problems are posed by the error sources in the space carving process and the finite (non-zero) voxel size. Therefore, the focus of most of the work in voxel-coloring has been to design reliable, robust and generic photo-consistency tests. The VoM photo-consistency test is the simplest and is the most common in used. This test assumes that the effect of all the error sources can be modelled as additive Gaussian noise in pixel color. In this test, the mean color of pixels in a voxel projection in each visible image is calculated first. Then the variance σ_{col}^2 of these mean pixel colors is compared with a global variance threshold τ . If the variance is smaller than this threshold, the voxel is declared as photo-Consistent, i.e.

$$(3.1) \quad \begin{cases} \sigma_{col}^2 \leq \tau, & \text{Consistent;} \\ \sigma_{col}^2 > \tau, & \text{Non-Consistent.} \end{cases}$$

The Variance of Means test is not monotonic. A similar but monotonic test can be devised using chi-square statistics. If we assume that the color distribution of a voxel projection in each visible image is Gaussian, the distribution of the variance of means will be a chi-square distribution with $K - 1$ degrees of freedom, where K is the total number of visible images. This test was proposed by Kutulakos and Seitz in [26, 18].

$$(3.2) \quad \lambda_v = \frac{(K - 1)\sigma_{col}^2}{\sigma_o^2},$$

where σ_o^2 is the allowed noise variance and λ_v is the chi-square variable with $K - 1$ degrees of freedom. If the probability of λ_v , $Pr(\lambda_v)$, is less than a threshold value, the voxel is declared non-consistent. If the noise variance represents sensor noise only, it can be estimated by taking images of a single uniformly colored surface. However, when other error sources are present as well, the threshold noise variance value is adjusted empirically.

The above mentioned tests use a global variance threshold for all parts of the scene surface and thus do not produce optimum reconstruction model. Error sources have a greater effect on scene regions with a higher degree of texture and depth variation. Therefore, an optimum noise variance threshold should be adaptive, i.e. it will have low threshold value for low texture regions and high threshold value for highly textured regions. One approach is to use *f-statistics*. The ratio of the variance of the means of pixel colors of voxel projections and the average of the intra-projection pixel color variance can be modelled as an *f-distribution* [5],

$$(3.3) \quad f_K = \frac{\sigma_{col}^2}{\sigma_n^2},$$

where

$$\begin{aligned}
 \sigma_n^2 &= \frac{1}{K} \sum_{k=1}^K \left[\left(\frac{1}{J_k - 1} \right) \sum_{j=1}^{J_k} (x_{jk} - \bar{x}_k)^2 \right], \\
 (3.4) \quad &= \frac{1}{K} \sum_{k=1}^K \sigma_k^2, \\
 &= \overline{\sigma_k^2},
 \end{aligned}$$

and

K = Total number of images that can see the voxel,

J_k = Number of pixels in the projection of the voxel in image k ,

\bar{x}_j = Mean color of pixels in the projection of the voxel in k th image,

σ_{col}^2 = Variance of mean projection colors,

σ_n^2 = Average of intra-projection color variance,

f_K = f -distribution variable with K and $K(J_k - 1)$ degrees of freedom.

Broadhurst in [5] has compared the results of the f -statistics photo-consistency test and the Variance of Means test. For lower voxel resolutions, the f -statistics measure behaved better as a photo-consistency test due to its adaptive nature. However, for higher voxel resolutions, the Variance of Means test outperformed the f -statistics. The reason for this better performance is that at higher voxel resolutions, a voxel projection contains fewer pixels and the intra-projection color variance becomes unstable due to under sampling.

Slabaugh et al. in [29] have also developed an adaptive photo-consistency test called the Adaptive Standard Deviation Test (ASDT). It is similar to the Variance of Means test except that there are two variance thresholds, τ_1 and τ_2 ,

$$(3.5) \quad \sigma_{col}^2 \leq \tau_1 + \tau_2 \sigma_n^2, \text{ Consistent,}$$

where σ_n^2 is the average of the variance of pixel colors in each projection as calculated in Equation (3.4). τ_1 is a global variance threshold, and τ_2 depends on local variance statistics. The authors have shown that their photo-consistency test gives better results as compared to the single threshold Variance of Means test. However, the price for this improvement is having to deal with two adjustable thresholds. Another method proposed by the same authors in [29] uses histogram matching as a photo-consistency check. The advantage of this method is that the threshold parameter does not change for different scenes. The results of this test are comparable to that of ASDT.

The limitations of the histogram matching test are extra computational cost and larger memory requirements.

Probabilistic photo-consistency tests form another class of voxel photo-consistency [4, 2]. The advantages of these methods are better handling of the errors in the voxel-coloring process and that they do not require adjustable variance threshold parameter [4]. Therefore, these tests are global in nature and perform better in the presence of error sources. The limitation of these methods is a high computational cost. Another photo-consistency test has been proposed by Yang and Pollefeys in [35] that makes use of a smoothness constraint to handle textureless regions. John Isidoro in [12] has proposed per-pixel matching as a photo-consistency test that is more robust in handling the outlier pixels in a voxel projection due to different error sources.

3.3. Exploring New Dimensions in Voxel-Coloring.

Carceroni and Kutulakos in [6] and Vedula and Baker in [34] have tried to recover 3D motion and shape from multi-view sequence of images using voxel-coloring. Carceroni has also extended voxel-coloring to non-Lambertian surfaces. Vedula uses a 6D space for motion estimation using voxel-coloring. However, a big problem with voxel-coloring for motion estimation is that it is still far from real-time. Slabaugh et al. propose a warped voxelized space to handle very large scenes in [30]. Fewer and larger voxels are used to represent the scene regions that are farther from the camera and have lower resolution, while a large number of smaller voxels are used to represent the higher resolution closer scene regions. This strategy helps to reduce the amount of voxels required to represent very large 3D scenes and hence improves reconstruction resolution for lower computational cost. Prock and Dyer in [24] and Montenegro in [22] have used a coarse-to-fine approach to reduce the computational cost of voxel-coloring. Saito and Kanade in [25] and Kimura in [15] have extended voxel coloring to projective space.

Martin Granger-Piché in [10], and Cha Zhang and Tsuhan Chen in [37] control the imaging process to improve the quality of reconstruction from voxel-coloring, thus opening a direction for active voxel-coloring. Yezzi and Slabaugh have used active contours to add the smoothness constraints in voxel-coloring in [36]. Sylvain Paris and Francois Sillion in [23] have used additional shape constraints to make the reconstruction more robust for a small-baseline stereo system.

4. Handling Camera Calibration Errors

As described earlier, camera calibration errors add outlier pixels in a voxel projections and may cause erroneous carving of a consistent voxel. Broadhurst in [5], Kutulakos in [16] and Kwon in [19] have discussed the effects of camera calibration errors in voxel-coloring. However, not enough work has been done to handle the effects of these errors. The probabilistic voxel-coloring methods in [4, 2]

try to handle error sources in voxel-coloring. But they do not deal with the calibration errors in a systematically. The only work to explicitly handle the camera calibration errors in voxel-coloring is presented by Kutulakos in [16]. Kutulakos has introduced a new photo-consistency measure called r-Consistency. r-Consistency is more robust against outliers as compared to the originally proposed photo-consistency tests. In r-Consistency, instead of using the color distribution of all the pixels in a voxel projection, only one pixel from each voxel projection is used in photo-consistency test. Therefore, the result of r-Consistency is a blurred 3D model of the scene that, when reprojected, produces images similar to the input images up to an r-Shuffle transform. The concept of r-Consistency and r-Shuffle transform will be explained in more detail in Chapter 2. The limitations of r-Consistency include excessive blurring and fattening of the reconstructed 3D model, and not making any use of the available calibration error information.

Camera calibration errors show some structure that depends on the particular method of calibration and the feature points used for calibration. If we could model the effect of calibration errors as a probability distribution, this information can be used to handle their effects in a more systematic way. In this thesis, we use zero mean additive Gaussian noise as an approximate model for the errors in estimated calibration parameters and their effects on the coordinates of the projected pixels. We use error information in the camera parameters to estimate the error in the coordinates of pixels in the voxel projections. The error estimate in the projected pixel coordinates is used to increase the size of a voxel projection to reduce the effect of outlier pixels on the photo-consistency test. We name our proposed method Adaptive Gaussian Averaging. The derivation of our proposed method will be presented in detail in Chapter 2. The price for using Adaptive Gaussian Averaging is the Gaussian blurring of the reconstructed 3D model. However, the level of blurring will be different for different regions of the reconstructed scene depending on the effects of calibration errors for that particular region. Our method has the ability to adapt to the level of calibration errors present in the system. The effectiveness of our method will be shown experimentally in Chapter 4.

5. Handling Partial Emptiness of Surface Voxels

Some work can be found in the literature on voxel-coloring that describes how to handle the effects of partial emptiness of surface voxels. De Bonet in [8] has developed an iterative optimization scheme to handle the effects of transparent scenes and partially empty voxels. However, no proof for the convergence of this algorithm has been provided, and the imaging model is not correct for opaque scenes [2]. Szeliski in [33] has proposed an optimization of the 3D model using the reprojection error to overcome the effect of partially empty voxels. However, it is a complex framework and does not take into account the effect of non-boundary, partially empty surface voxels. The r-Consistency

test proposed by Kutulakos in [16] is also effective in reducing the effects of outlier pixels due to finite voxel size. Some other techniques to handle this problem can be found in [12, 14, 22]. A more detailed review of these techniques is presented in Chapter 3.

A much simpler method is proposed in this thesis to reduce the effects of partial emptiness of surface voxels. The percentage of outlier pixels in a voxel projection due to its partial emptiness is inversely proportional to the area of the projection, i.e. the number of pixels in it. In other words, a voxel projection that contains more pixels has less chances of having outlier pixels and is a more reliable representation of the scene surface contained in the voxel. Therefore, if a weight is assigned to the color statistics of each voxel projection in proportion to the number of pixels in that projection for photo-consistency test, the effect of this error on the quality of reconstruction can be reduced. The justification of this area weighting scheme and its application to the Variance of Means photo-consistency test is explained in detail in Chapter 3. Our proposed method is simpler and straightforward, and can be used with most of the existing photo-consistency tests to overcome the problem at hand.

An area weighting scheme, similar to our proposed method, has been used by Matthew Loper in his software for voxel-coloring called *Archimedes* [20]. However, we could not find in literature any explanation of the use of this method in voxel-coloring.

6. Contributions

The main contribution of this thesis is to propose methods that can be used with the existing voxel-coloring algorithms to make them more robust against calibration errors and partial emptiness of surface voxels.

- (i) **Handling Camera Calibration Errors:** We propose a method, Adaptive Gaussian Averaging, to reduce the effects of the uncertainty introduced by camera calibration errors in voxel-coloring. This method makes use of the calibration error distribution information in a simple way to handle the calibration error effects systematically.
- (ii) **Analysis of the Effects of Partial Emptiness of Surface Voxels:** An analysis of the problems arising from partial emptiness of surface voxels is presented.
- (iii) **Reducing the Effect of Partial Emptiness of Surface Voxels:** Based on the analysis of the problem, a simple and effective method is proposed to reduce the effects of partial emptiness of surface voxels on the reconstructed scene.
- (iv) **Experimental Verification:** The effects of the above-mentioned error sources and the effectiveness of our proposed methods have been verified experimentally using our own calibration data sets.

7. Thesis Outline

In Chapter 2 a discussion of the effects of camera calibration errors on voxel-coloring is presented. A review of the pertinent literature is also presented in this chapter. Our proposed solution to handle the calibration errors is presented in the last part of this chapter. Chapter 3 comprises an analysis of the effects of partial emptiness of surface voxels on voxel-coloring, a review of the work that has already been done to handle this problem, and a presentation of our proposed method to make voxel-coloring more robust against this source of error. Chapter 4 has two parts. The first part gives an overview of our experimental setup. The second part of this chapter is a discussion of the experimental results to verify our proposed methods. Finally, chapter 5 concludes the thesis with a summary of our work and the directions for future research.

CHAPTER 2

Handling Camera Calibration Errors in Voxel-Coloring

In this chapter the effects of camera calibration errors on voxel-coloring and its solutions are discussed. First, there is a discussion of the effects of this error on voxel-coloring. After this, a review of the already proposed solutions is presented. It is shown that the existing solutions do not overcome the effects of calibration errors in a systematic way. This is followed by the discussion and derivation of our proposed solution to handle calibration errors in voxel-coloring.

1. The Effect of Camera Calibration Errors on Voxel-Coloring

A complete camera calibration for all the input views is a precondition for voxel-coloring. Due to the errors in the calibration process, the estimated calibration parameters might not be accurate. In voxel-coloring, the errors in the estimated camera parameters cause a voxel projection to shift from its true position in the input image. The result of this shift is to add outlier pixels in the projection that may affect the color statistics of the projection. This fact is shown in Figure 2.1. The shape of a voxel projection in an image is shown to be a square only for simplicity. The camera calibration errors depend on the locations of the feature points being used in the calibration process and the accuracy of their image correspondences. In a multiple view system, different views may be affected by different degrees of calibration error. And for a particular view, the calibration errors may have different effect on the voxel projections in different parts of the input image. As a result, the amount of error in the pixel coordinates of the projections of a voxel in different input images may be different. Therefore, different projections of a voxel may contain different amounts of outlier pixels. The effect of the outliers is to reduce the color-consistency of a voxel projections. As a result of these outlier pixels, a consistent voxel might be declared as non-consistent by a photo-consistency test if the test does not take into account the effect of these outliers. Erroneous removal of consistent voxels will result into holes in the reconstructed 3D model. It is well-known that, due

to the sequential nature of voxel-coloring algorithms, a wrongly carved voxel is more dangerous for the quality of reconstruction than a wrongly left-over voxel [7].

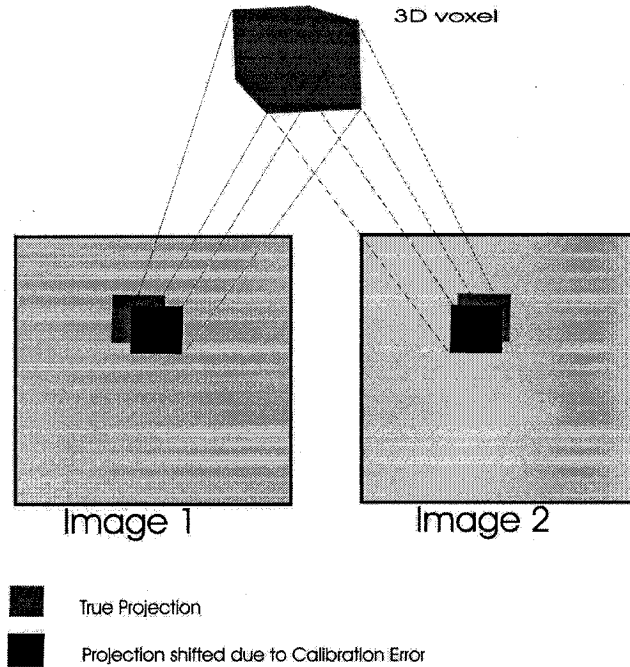


FIGURE 2.1. Effect of calibration errors on the projections of a voxel is to shift the projected pixels from their true locations. The shift in the projected pixel coordinates may be different in the different images.

2. Related Work

It is important to note that it is more difficult to handle the calibration errors in voxel-coloring than in traditional image-space stereo methods. A voxel is usually modelled as a cube, therefore, the problem is that the size and shape of a voxel projection in different images can be different from each other due to the difference in the visible shape of the voxel in different input images. Therefore, two projections of a voxel cannot be compared on per-pixel basis, a commonly used technique in traditional area-correlation based stereo correspondence. As a result, regardless of the variations in the spatial ordering of the pixels in voxel projections, as long as the color distributions of projections of a voxel are same, it will be declared consistent. It also makes it difficult to extend the use of existing methods of handling calibration errors in traditional stereo to voxel-coloring.

The amount of work that has already been done to handle camera calibration errors in voxel-coloring is limited. The effects of camera calibration errors on voxel-coloring have been discussed by Broadhurst in [5] and by Musik Kwon in [19]. Broadhurst has shown that by increasing the level

of calibration errors, the mean RGB reprojection error increases monotonically. However, he does not propose a solution to overcome this problem. Musik Kwon has shown that in the absence of the calibration errors, the distribution of the variance of mean projection colors of consistent voxels has a high peak for a low variance value. An increase in the calibration error results in further spreading of this variance distribution, adding uncertainty in the photo-consistency test. Only Kutulakos in [16] has directly addressed the problem of camera calibration errors in voxel-coloring. A review of his proposed method is presented below.

2.1. Approximate N-View Stereo.

In [16], Kutulakos has introduced the idea of r-Consistency to overcome the effect of camera calibration errors on the reconstruction results of voxel-coloring. The following definitions will be helpful to understand the concept of r-Consistency.

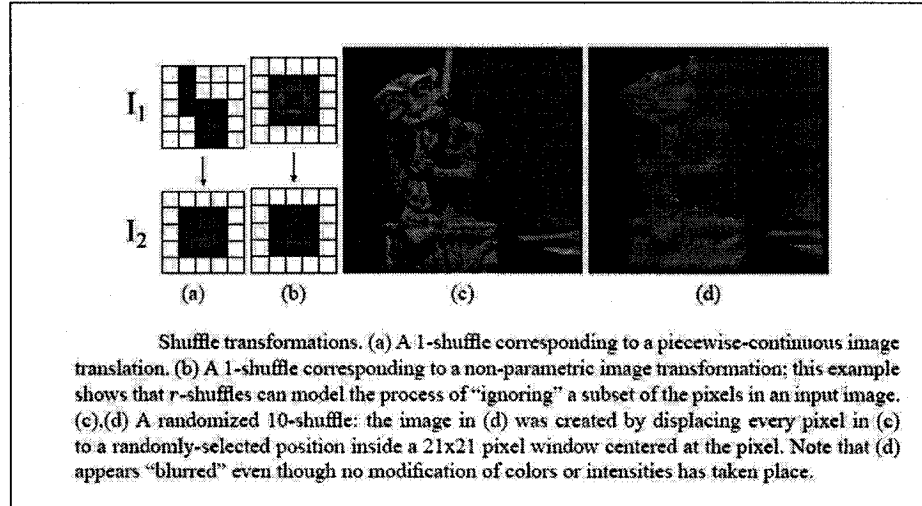
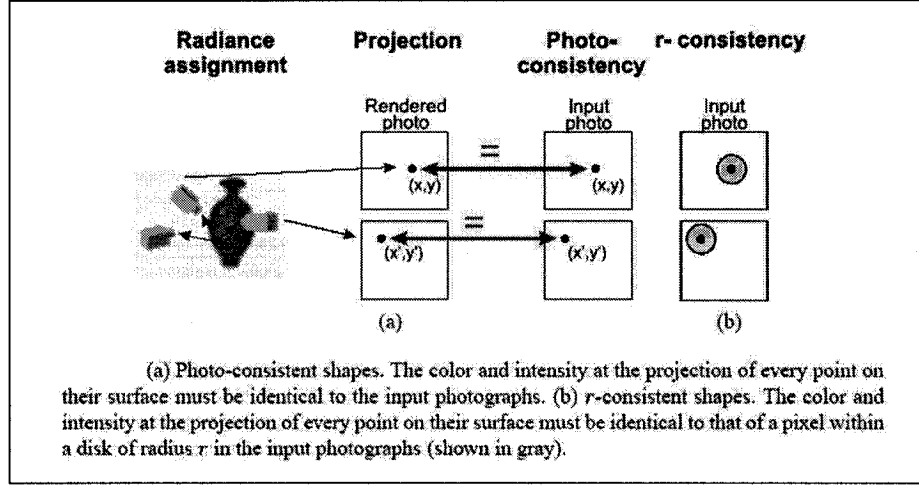


FIGURE 2.2. The effect of r-Shuffle on the reprojected image is in the form of blurring, taken from [16]

r- Shuffle: "A 2D transformation $\tau : I_1 \rightarrow I_2$ is called an r-Shuffle if for every point in image I_2 we can find a point of identical color within a disk of radius r in image I_1 . The constant $r \geq 0$ is called the dispersion radius of τ " [16]. The effect of r-Shuffle on an image is the blurring of that image as shown in Figure 2.2.

r-Consistency: r-Consistency is a refined and more flexible version of the initial photo-consistency test proposed by Kutulakos in [18]. In a normal photo-consistency test, the mean color of all the pixels in each voxel projection is calculated, and then the mean projection colors are matched to see if the voxel is consistent. In r-Consistency, only one pixel taken from each voxel projection is used in its photo-consistency test. If the variance of the best single pixel match is

FIGURE 2.3. The comparison of r -Consistency and photo-consistency, taken from [16]

less than the threshold, the voxel is declared to be photo-consistent. In r -Consistency, each voxel projection has a circular shape with a radius of r pixels. This is shown in Figure 2.3. The result of r -Consistency is an r -Consistent 3D shape. When an r -Consistent 3D shape is projected back to the input images, the reprojected images differ from the original images by an r -Shuffle transform.

r -Consistent 3D Shape: “A volume V is r -Consistent if for every input image, I_i , there exists an r -Shuffle $\tau_i : I_i \rightarrow I'_i$ that makes V photo-consistent with all N input images I'_1, \dots, I'_N ” [18].

The r -Consistent 3D shape is a superset of the original 3D scene. If a shape α is r_1 -Consistent and a shape β is r_2 -consistent, and $r_1 \subseteq r_2$, then α is contained in β . Kutulakos refers to this as the Nesting Theorem. Based on the Nesting Theorem, a 3D shape can be successively refined using a coarse-to-fine reconstruction strategy starting from a larger value of r .

2.1.1. Handling Camera Calibration Error with r -Consistency.

As r -Consistency uses only a single pixel from each voxel projection for its photo-consistency check, it helps to overcome the effects of outlier pixels resulting from calibration errors and other sources of errors. An assumption in r -Consistency is that the camera calibration error should be limited so that it does not affect the visibility of a 3D point in the input images. The Calibration Reconstructibility Theorem describes this assumption as a precondition to get an r -Consistent 3D reconstruction in the presence of calibration errors.

Calibration Reconstructibility Theorem: “An r -Consistent subset of a volume V exists for some $r \geq 0$ if the following condition holds for all $i = 1, \dots, N$ images :

$$Vis_V(c_i) = Vis_V(\hat{c}_i)$$

$Vis_V(c_i)$ is the set of points on volume V that are visible from the viewpoint c_i . In this definition c_1, \dots, c_N are the viewpoints from which the input photographs were taken, and $\hat{c}_1, \dots, \hat{c}_N$ are the incorrect estimates for these viewpoints due to camera calibration errors" [16].

2.2. Limitations of r-Consistency.

r-Consistency is not an optimum solution for the problems caused by the camera calibration errors in voxel-coloring because of the following limitations:

- r-Consistency is a much weaker consistency test compared to the original photo-consistency test since only one pixel from each projection of the voxel is used. Therefore the lower the voxel resolution or the larger the value of dispersion radius r , the fatter and blurred is the reconstructed 3D model. In cases where no background segmentation information is available, e.g. silhouettes, the r-Consistency check will result in a higher number of false matches.
- No method for selecting an appropriate value dispersion radius r for r-Consistency is provided.
- r-Consistency is not a systematic treatment of the calibration errors problem as it does not make any use of the calibration error distribution information. Kutulakos admits that this method is only a first step towards handling calibration errors in voxel-coloring.

3. Adaptive Gaussian Averaging

This section gives the details of our proposed method for systematic handling of the calibration errors in voxel-coloring, which we call Adaptive Gaussian Averaging (AGA).

3.1. Probabilistic Modelling of the Effect of Calibration Errors in Voxel-coloring.

If we know the probability distribution of errors in estimated camera parameters, the uncertainty caused by these errors in voxel-coloring can be modelled probabilistically. For now, it is assumed that a voxel is a 3D point that projects only to one pixel in each visible image. This assumption will be relaxed later for cube shaped voxels that may contain more than a single pixel in their individual projections. Let us assume that X_{2D} represents the projected pixel coordinates in an image of a 3D point X_{3D} . X_{3D} and X_{2D} are related to each other through the camera projection matrix P ,

$$(3.1) \quad X_{2D} = P \cdot X_{3D}.$$

As our primary aim is to see the effect of camera calibration errors on voxel-coloring, we will neglect other error sources to keep the derivation simple. For now, it will be assumed that only two types of error affect voxel-coloring:

- (i) The effect of sensor noise and non-Lambertian surfaces on pixel color. Due to these errors, the color I of a particular pixel at coordinates x, y in the i th image will be a random variable. Let this random variable be represented by \mathbf{I}_{xy}^i and its probability distribution be represented as $f(I_{xy}^i | (x, y))$. As the effect of these errors could be different for each pixel of an image, it is a conditional probability distribution.
- (ii) Error in the coordinates x, y of the pixel in a voxel's projection due to the errors in estimated camera parameters. Therefore, the coordinates x and y of a projected pixel are random variables and can be combined in a random vector $\mathbf{X}_{2D} = [x \ y]'$ having a bivariate probability distribution $f(X_{2D}) = f(x, y)$.

Let $PROJ^i(X_{3D})$ be the event that the projection of a voxel v located at $X_{3D} = [X \ Y \ Z]'$ has the 2D coordinates $X_{2D} = (x, y)$ in image i and the color of the pixel at this location is I_{xy}^i . Let the probability of occurrence of this event be $Pr(PROJ^i(X_{3D}))$. Because of the above two error sources, $Pr(PROJ^i(X_{3D}))$ depends on the following two probabilities:

- (i) Probability that $X_{2D} = (x, y)$ are the true coordinates of the pixel in the projection of the voxel v . Let this probability be represented as $Pr(x, y)$ and it depends on the probability distribution of the errors in the pixel coordinates due to calibration errors.
- (ii) Probability that the color of the pixel at coordinates (x, y) is I_{xy}^i . Let this probability be represented as $Pr(I_{xy}^i | (x, y))$ and it depends on the probability distribution of the sensor noise and non-Lambertian surface effects.

The effect of the above two probabilities can be combined using Bayes rule to calculate $Pr(PROJ^i(X_{3D}))$ as follows:

$$(3.2) \quad Pr(PROJ^i(X_{3D})) = Pr(I_{xy}^i, x, y) = Pr(I_{xy}^i | x, y) \cdot Pr(x, y)$$

Equation 3.2 gives the joint probability distribution of error in the position of a voxel projection, and the color of the pixel in the projection. In voxel-coloring, only the color of the pixels in a voxel projection is used to check its photo-consistency. Therefore, it is important to transfer the uncertainty in the projection coordinates to the uncertainty in the color of the projection. This can be achieved by calculating the marginal probability distribution of projection color in the presence of both sensor noise and calibration errors as follows:

$$(3.3) \quad Pr(I_v^i) = \int_x \int_y Pr(I_{xy}^i, x, y) dy dx = \int_x \int_y Pr(I_{xy}^i | x, y) \cdot Pr(x, y) dy dx,$$

where $Pr(I_v^i)$ is the probability that the color of the pixel in a voxel v 's projection in image i is I in the presence of both sensor noise and camera calibration errors. The accuracy of this probabilistic modelling depends on the accuracy of the probabilistic model used for the error sources.

3.2. Local Probability Distribution of Pixel Color Intensity in the Presence of Sensor Noise.

In most of the voxel-coloring literature, sensor noise and non-Lambertian surface effects are modelled as zero mean additive Gaussian noise [18, 2]. Each color channel of a pixel at a particular location x, y in an image i can be assumed to be corrupted by this zero mean additive Gaussian noise with variance σ_{nxyj}^2 , where j is the color channel i.e. R, G or B. Furthermore, it is assumed that this noise is IID (independent and identically distributed) for a particular color channel j and for all the pixels in an image, thus $\sigma_{nxyj}^2 = \sigma_{nj}^2$. Let I_{xyj}^i be the true intensity value of the color channel j for a pixel at x, y in image i and \mathbf{I}_{xyj}^i is its noisy version that is a random variable. Boldfaced letters are being used to represent random variables. Let \mathbf{n}_j be the zero mean Gaussian noise in the j th color channel of image i with a variance σ_{nj}^2 . Then for each color channel,

$$(3.4) \quad \begin{aligned} \mathbf{I}_{xyR}^i &= I_{xyR}^i + \mathbf{n}_R, \\ \mathbf{I}_{xyG}^i &= I_{xyG}^i + \mathbf{n}_G, \\ \mathbf{I}_{xyB}^i &= I_{xyB}^i + \mathbf{n}_B. \end{aligned}$$

The pdf (probability density function) of each component of a pixel color is, therefore, the normal distribution i.e.

$$(3.5) \quad \begin{aligned} f(\mathbf{I}_{xyR}^i) &= \mathcal{N}(I_{xyR}^i, \sigma_{nR}^2), \\ f(\mathbf{I}_{xyG}^i) &= \mathcal{N}(I_{xyG}^i, \sigma_{nG}^2), \\ f(\mathbf{I}_{xyB}^i) &= \mathcal{N}(I_{xyB}^i, \sigma_{nB}^2). \end{aligned}$$

3.3. The Distribution of the Projected Pixel Coordinates Error Due to Calibration Errors.

The calibration errors in the camera parameters can be compactly represented as a covariance matrix up to second order of approximation [11]. A further assumption is made that the error in the camera parameters has zero mean Gaussian distribution. The error in the pixel coordinates of a voxel projection due to camera parameters error can be modelled as a zero mean bivariate Gaussian

distribution. With the assumption that the projected pixel coordinates error linearly depends on the camera parameters error, the covariance matrix of the camera parameters error can be used to calculate the covariance matrix of the error in the projected pixel coordinates. This covariance matrix depends on both the 3D coordinates of the voxel and 2D coordinates of its projection in an image.

Let C^i be the covariance matrix of the camera parameters error for camera i , and $C_{x,y}^i$ be the covariance matrix of the error in pixel coordinates x, y of a voxel v 's projection in image i . Let X_{3D} be a vector of voxel v 's 3D coordinates and X_{2D}^i be a vector representing the pixel coordinates of a voxel projection in image i . $C_{x,y}^i$ can be calculated from C^i using some function F that depends on X_{3D} , X_{2D}^i and C^i i.e.

$$(3.6) \quad C_{x,y}^i = F(C^i, X_{3D}, X_{2D}^i)$$

With the assumption that the the projected pixel coordinates error is linearly dependent on the error in the estimated camera parameters, the function in equation (3.6) can be expressed as follows:

$$(3.7) \quad \begin{aligned} C_{x,y}^i &= J_P C^i J_P^T, \\ J_P &= \frac{\partial X_{2D}^i}{\partial P^i}, \end{aligned}$$

where J_P is a Jacobian matrix and P^i is the estimated camera parameter matrix. The estimation of the covariance matrix of the camera parameters error and the covariance matrix of the projected pixel coordinates error have been further explained in appendix A. Once the error in the 2D pixel coordinates is estimated, the bivariate Gaussian distribution of a voxel projection coordinates in image i can be expressed as follows:

$$(3.8) \quad f^i(x, y) = \mathcal{N}(\hat{x}, \hat{y}, C_{x,y}^i)$$

where \hat{x} , \hat{y} represent the mean values of the projection coordinates calculated using Equation (3.1).

3.4. Uncertainty in a Voxel Projection Color due to Gaussian Calibration Error.

With the assumption that a voxel projection coordinates have bivariate Gaussian distribution as given by Equation (3.8), Equation 3.3 is reduced to a summation over a window centered on \hat{x} , \hat{y} :

$$(3.9) \quad Pr(I_v^i) = \sum_x \sum_y Pr(I_{xy}^i | x, y) \cdot Pr^i(x, y),$$

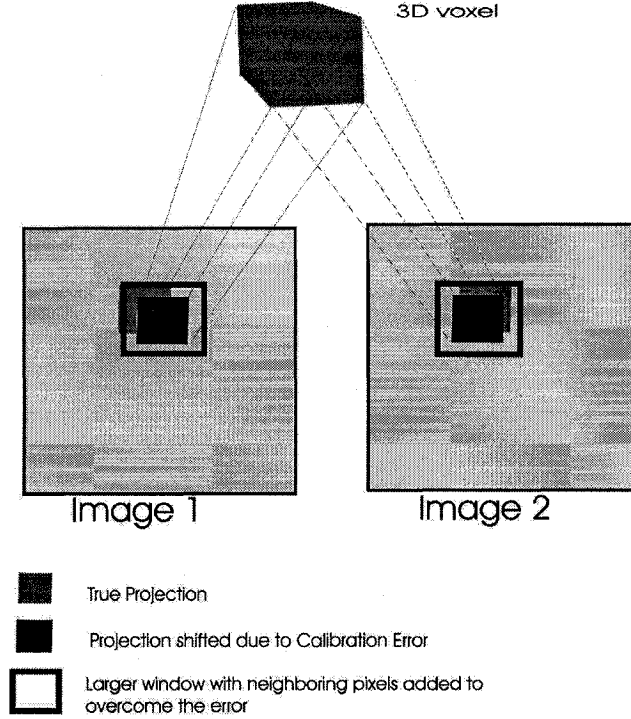


FIGURE 2.4. The concept of Adaptive Gaussian Averaging. The projection window size is increased to calculate the Gaussian averaging according to the effect of calibration errors in that part of the image.

where $Pr(I_{xy}^i|x, y)$ is given by Equation (3.5) and $Pr^i(x, y)$ is given by Equation (3.8). Note that all the derivations in this section are for a single color channel.

The size of the window in Equation (3.9) is limited by $\pm 3\sigma_x$ and $\pm 3\sigma_y$ along both axes, where σ_x and σ_y are the standard deviation of the errors in pixel coordinates along the two axes. For a Gaussian distribution, $\pm 3\sigma$ covers 95% of the samples.

The mean of the above pixel color probability distribution will be:

$$\begin{aligned}
 \text{Mean}(\mathbf{I}_v^i) &= \sum_x \sum_y (I_{xy}^i + \mathbf{n}) \cdot Pr(x, y), \\
 (3.10) \quad &= \sum_x \sum_y I_{xy}^i \cdot Pr(x, y) + \sum_x \sum_y \mathbf{n} \cdot Pr(x, y), \\
 \mu_{col} &= \mu_I + \mu_n,
 \end{aligned}$$

where μ_{col} represents the mean color intensity of the pixel in the voxel projection, μ_I is the Gaussian average of the true color of the pixels in the window and μ_n is the Gaussian average of the additive zero mean Gaussian noise. Although \mathbf{n} is zero mean IID Gaussian noise, μ_n may not be zero due to Gaussian averaging of its samples. The value of μ_n depends mainly on the value of the

noise sample in the pixel at the center of projection window, i.e. the pixel with coordinates \hat{x} , \hat{y} . Because the size of the window used for Gaussian averaging depends on the pixel coordinates error variance i.e. $C_{x,y}^i$, which is a function of projected pixel coordinates (x, y) , we name this technique **Adaptive Gaussian Averaging (AGA)**.

Similarly, the variance of the color distribution of a voxel projection will be:

$$\begin{aligned}
 \text{Variance}(\mathbf{I}_v^i) &= \left(\frac{N}{N-1}\right) \sum_x \sum_y (I_{xy}^i + \mathbf{n} - \mu_{col})^2 \cdot Pr(x, y), \\
 &= \left(\frac{N}{N-1}\right) \sum_x \sum_y (I_{xy}^i + \mathbf{n} - \mu_I - \mu_n)^2 \cdot Pr(x, y), \\
 (3.11) \quad &= \left(\frac{N}{N-1}\right) \sum_x \sum_y (I_{xy}^i - \mu_I)^2 \cdot Pr(x, y) \\
 &\quad + \left(\frac{N}{N-1}\right) \sum_x \sum_y (\mathbf{n} - \mu_n)^2 \cdot Pr(x, y) \\
 &\quad + \left(\frac{N}{N-1}\right) \sum_x \sum_y (I_{xy}^i - \mu_I)(\mathbf{n} - \mu_n) \cdot Pr(x, y),
 \end{aligned}$$

where N is the total number of pixels in the Gaussian averaging window. As the calibration error is independent of sensor noise, the 3rd term in (3.11) will be zero. Thus, Equation (3.11) is reduced as follows:

$$\begin{aligned}
 \text{Variance}(\mathbf{I}_v^i) &= \left(\frac{N}{N-1}\right) \sum_x \sum_y (I_{xy}^i - \mu_I)^2 \cdot Pr(x, y) \\
 (3.12) \quad &\quad + \left(\frac{N}{N-1}\right) \sum_x \sum_y (\mathbf{n} - \mu_n)^2 \cdot Pr(x, y)
 \end{aligned}$$

Equation (3.12) describes how the calibration errors increase the variance of the color distribution of a voxel projection. As both the error sources are independent of each other, ideally the calculated variance should be the sum of the variance of the pixel color due to the calibration error and the variance of the Gaussian noise:

$$(3.13) \quad \text{Variance}(\mathbf{I}_v^i) = \left(\frac{N}{N-1}\right) \sum_x \sum_y (I_{xy}^i - \mu_I)^2 \cdot Pr(x, y) + \sigma_n^2$$

However, due to the Gaussian averaging of the noise samples, the second term in Equation (3.12) is not the true sensor noise variance. Therefore, equation (3.12) may underestimate the effect of sensor noise variance in the calculated variance value. This effect becomes obvious when a small Gaussian averaging window is used in case of small calibration error values. In this case, if we have an estimate of the sensor noise variance, we can use it to improve the estimate of the variance

calculated from (3.12) as follows:

$$(3.14) \quad \text{Variance}(\mathbf{I}_v^1) = \max(\text{Calculated Variance from (3.12), Sensor Noise Variance})$$

Using a Gaussian distribution for the calibration error results into Gaussian blurring of the 3D model. This is the price that has to be paid to avoid the over-carving of 3D reconstructed model.

3.5. Multiple Pixel Voxel Projections.

The above calculations were made under the assumption that a voxel projects to exactly one pixel in each visible input image. However, for lower voxel resolutions, a voxel projection may consist of multiple pixels. The number of pixels in voxel projections as well as their shape could be different in different input images. Therefore, it is hard to know the exact color distribution of the pixels in a voxel projection. For simplicity, the color distribution is approximated as a univariate Gaussian distribution characterized by a mean and a variance. The calculations of the pixel color mean and variance of a voxel projection from Equations (3.10) and (3.12) can be extended to multiple pixel projections by averaging over all the pixels in a voxel projection:

$$(3.15) \quad \begin{aligned} \text{Mean}(\mathbf{I}_v^1) &= \left(\frac{1}{N_R}\right) \sum_R \sum_x \sum_y (I_{xy}^i + \mathbf{n}) \cdot \text{Pr}(x, y), \\ &= \left(\frac{1}{N_R}\right) \sum_R \sum_x \sum_y I_{xy}^i \cdot \text{Pr}(x, y) + \left(\frac{1}{N_R}\right) \sum_R \sum_x \sum_y \mathbf{n} \cdot \text{Pr}(x, y), \\ \mu_{colv} &= \mu_I + \mu_n \approx \mu_I, \end{aligned}$$

where R represents the region of a voxel projection and N_R is the number of pixels in R . Due to the averaging of the color of all the pixels in a projection, the effect of the zero mean Gaussian sensor noise on the mean projection color can be ignored.

Similarly the variance of pixel colors in the voxel projection becomes:

$$\begin{aligned}
 \text{Variance}(\mathbf{I}_v^i) &= \left(\frac{N_R}{N_R - 1}\right) \sum_R \sum_x \sum_y (I_{xy}^i + \mathbf{n} - \mu_{col})^2 \cdot Pr(x, y), \\
 &= \left(\frac{N_R}{N_R - 1}\right) \sum_R \sum_x \sum_y (I_{xy}^i + \mathbf{n} - \mu_I - \mu_n)^2 \cdot Pr(x, y), \\
 &= \left(\frac{N_R}{N_R - 1}\right) \sum_R \sum_x \sum_y (I_{xy}^i - \mu_I)^2 \cdot Pr(x, y) \\
 &\quad + \left(\frac{N_R}{N_R - 1}\right) \sum_R \sum_x \sum_y (\mathbf{n} - \mu_n)^2 \cdot Pr(x, y) \\
 &\quad + \left(\frac{N_R}{N_R - 1}\right) \sum_R \sum_x \sum_y (I_{xy}^i - \mu_I)(\mathbf{n} - \mu_n) \cdot Pr(x, y), \\
 &= \left(\frac{N_R}{N_R - 1}\right) \sum_R \sum_x \sum_y (I_{xy}^i - \mu_I)^2 \cdot Pr(x, y) \\
 &\quad + \left(\frac{N_R}{N_R - 1}\right) \sum_R \sum_x \sum_y (\mathbf{n} - \mu_n)^2 \cdot Pr(x, y).
 \end{aligned}
 \tag{3.16}$$

CHAPTER 3

Handling Partial Emptiness of Surface Voxels

One of the sources of error in voxel-coloring is the partial emptiness of surface voxels. In most of the voxel-coloring literature, this problem has been largely neglected, and voxels are considered as either completely occupying the 3D scene surface or completely a part of empty space. In this thesis, we analyze the effect of partial emptiness of surface voxels on voxel-coloring. Based on our analysis of the problem, we propose a method that makes the voxel-coloring algorithms more robust against the effects of this error source. This chapter begins with an analysis of the effects of this error source on the reconstructed 3D model. It is followed by an overview of the work done by others to handle this problem in voxel-coloring. In the last section of this chapter, our proposed method to reduce the effects of this error source on voxel-coloring is presented.

1. Partial Emptiness of Surface Voxels: The Problem and its Effect

An ideal voxel is a 3D point that projects only to one pixel in each visible image. In most voxel-coloring algorithms, a voxel is modelled as a cube and it may contain multiple pixels in its projection. It has been shown in the literature that the reconstruction results of a voxel-coloring algorithm are improved when a voxel is modelled as a cube, and is precisely projected in the input image as a cube [31, 7]. A voxel-coloring algorithm starts with an approximate guess of the 3D scene volume that is supposed to contain the true scene surface. As there is no prior information available about the shape of the scene surface, and each voxel is in the form of a cube, the faces of the cubic surface voxels might not be exactly on the scene surface. Rather, the scene surface may pass through a voxel leaving it partially empty. When such a voxel is projected in the visible input images, its projections will contain outlier pixels, i.e. the pixels corresponding to the scene surface represented by the voxels other than the voxel being projected. This phenomenon is clear in Figure 3.1 and Figure 3.2 for voxels at a flat scene surface and at a depth discontinuity. To explain the problem more clearly, a two-dimensional view of the 3D voxels is shown in these figures.

3.1 PARTIAL EMPTINESS OF SURFACE VOXELS: THE PROBLEM AND ITS EFFECT

Instead of a perspective projection, an orthographic projection of the 2D voxels has been used for this purpose. From these figures, it is clear that the problem of outlier pixels due to the partial emptiness of surface voxels has more of an effect for the voxels at surface discontinuities. Therefore, in most of the related work to solve this problem, only the emptiness of the edge voxels has been considered. However, as is clear from Figure 3.1, the effect of partial emptiness of voxels at flat scene surfaces cannot be neglected either.

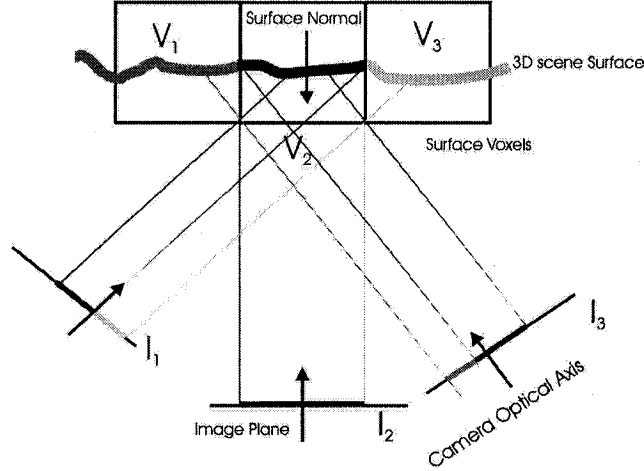


FIGURE 3.1. The effect of partial emptiness of a surface voxel. Almost half of the center voxel V_2 is empty as the scene surface is passing through it. It is clear that the projection of the center voxel V_2 in images I_1 and I_2 contains the outlier pixels representing the scene surface of the neighboring voxels V_1 and V_3 .

A voxel, which is not on the true scene surface but becomes visible due to the erroneous carving of a partially empty surface voxel, may lead to hole-carving in the reconstructed 3D model [7]. Therefore, a surface voxel that is at most 50% empty, should be declared consistent by the photo-consistency test to avoid hole-carving. However, due to the outlier pixels in the projections of such a partially empty voxel, the color distributions of its projections may not agree with each other. This may lead to the erroneous carving of such a voxel. Therefore, to avoid artifacts in the reconstruction, a photo-consistency test should take into account the effect of outlier pixels due to the partial emptiness of surface voxels. Consideration of outlier pixels due to this problem becomes especially important in the following situations:

- (i) When no foreground/background segmentation information is available, which is the case for most practical situations. In this case, it becomes important to use a strict color variance threshold for the photo-consistency test to avoid accidental matches with the background pixels. However, in the presence of outlier pixels, a strict variance threshold may result in the carving of partially empty surface voxels.

3.1 PARTIAL EMPTINESS OF SURFACE VOXELS: THE PROBLEM AND ITS EFFECT

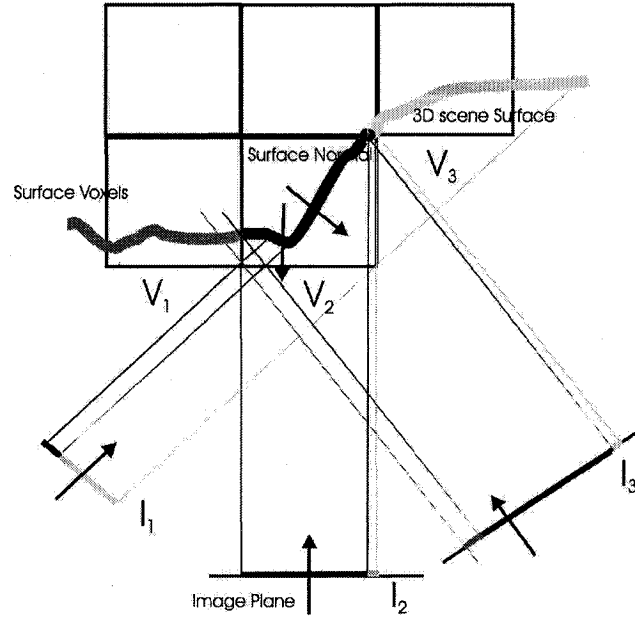


FIGURE 3.2. The effect of partial emptiness of an edge voxel. Note that most of the pixels in the voxel projection in I_1 represent the surface contained by the voxel V_3 instead of V_2 .

- (ii) Increasing the number of input images that can see the same scene surface increases the constraint on the photo-consistency test of a voxel, thereby reducing the chances of accidental matches. However, for a partially empty surface voxel, an increase in the number of input views will also increase the variance of the projection colors due to the outlier pixels. For instance, the pixels in the projections of a partially empty surface voxel in two images taken from distant viewpoints may represent two completely different 3D scene surfaces, which would be the surfaces contained by adjacent voxels. Therefore, the effect of the partial emptiness of surface voxels increases with an increase in number of input viewpoints.
- (iii) For lower voxel resolutions, when a voxel projection contains larger number of pixels, representing the unknown projection color distribution with just a mean and a variance is not enough. Therefore, lower voxel-resolutions pose extra uncertainty for a photo-consistency test. As a result, the effect of outlier pixels due to partial emptiness of surface voxels is higher at lower voxel resolutions.
- (iv) The effect of outlier pixels due to this problem is higher for highly textured images, and for scenes with lots of depth discontinuities.

2. Related Work

The problem of having partially empty surface voxels has been acknowledged in the voxel-coloring literature. A discussion of the related work to overcome the effects of this problem is presented below.

2.1. Roxels: Responsibility Weighted 3D Volume Reconstruction.

The Roxels algorithm was proposed by de Bonet and Viola in 2000 [8]. Their main contribution is to extend voxel-coloring to transparent scenes. In order to handle transparent scenes, the authors assume that no voxel is either completely empty or completely opaque. Therefore, their method is also able to reduce the effects of partially empty surface voxels. Their algorithm is iterative and parallel as compared to traditional voxel-coloring algorithms which are sequential.

In the Roxels algorithm, each 3D voxel $v(x, y, z)$ has an associated color $c(x, y, z)$, an opacity value $\alpha(x, y, z)$ and a responsibility value $r(x, y, z)$. Both $\alpha(x, y, z)$ and $r(x, y, z)$ vary in a range between 0 and 1. At the start of the algorithm, all the voxels are assigned equal responsibility and opacity values and same color. The algorithm works iteratively by comparing the reprojected images with the input views. After each iteration, the reprojection error is used to improve the color and opacity estimates of each voxel. The algorithm iterates until converging at stable values of color, opacity and responsibility for each voxel. The authors have shown the results of their algorithm for both synthetic and real data sets in [8].

2.1.1. Limitations of Roxels Algorithm.

The Roxel's algorithm has the following limitations when dealing with the problem of partially empty surface voxels:

- The authors claim that the algorithm converged on all of their data sets, however, no proof was provided of the convergence of their algorithm. Due to the large number of parameters being optimized, there is no guarantee that the algorithm will converge for arbitrary data sets.
- The Roxels algorithm equates the uncertainty of a voxel's photo-consistency and its opacity. This is not true for the image formation model of opaque scenes [2].
- The authors only used the data sets black background that allowed for easy background subtraction. As mentioned earlier, the problem of partially empty surface voxels is more prominent where foreground/background segmentation is not possible.

2.2. Stereo Matching with Transparency and Matting.

Another related piece of work that overcomes the problem of partial emptiness of surface voxels is presented by Richard Szeliski and Polina Golland in [33]. In this paper, the authors propose

a technique to handle the errors caused by partially transparent boundary pixels. The voxels that projects to these pixels would be partly transparent as well. The proposed technique is quite similar to the voxel coloring method proposed by Seitz and Dyer in 1997 [27]. However, instead of building a generic 3D model of the scene, the reconstructed model is built only for one virtual view only. In this method, after an initial estimate of the color and transparency of the voxels, the estimates are refined to take the mixed pixels into account, i.e. the pixels representing the depth discontinuities and occlusions in an image. The refinement is performed first by reprojecting the 3D voxels in the input images using their estimated color, visibility and opacity, and then optimizing these estimates by reducing the reprojection error between the input and reprojected images. Three different type of cost functions on reprojection error are used in their model refinement process:

- a The weighted error norm of the original image and the reprojected image. The weights can take into account the angle between the optical axes of the assumed virtual camera and input viewpoints.
- b A weak smoothness constraint on colors and opacities.
- c A prior distribution of opacities of the voxels.

The total cost function of the reprojection error is a weighted sum of the above three cost functions.

2.2.1. *Limitations.*

- This algorithm only takes into account the partially transparent boundary pixels. As the voxel resolution is equal to the pixel resolution, the problem of partially empty surface voxels on the planar scene surface does not arise in this case.
- As all the disparity calculations and refinements are performed in the image domain for a single virtual camera view, the reconstructed 3D model is highly view-dependent.
- The authors make use of smoothness constraint and prior distribution of the opacities. Whereas the purpose of our work is to propose a method that could handle this problem for arbitrary scenes with arbitrary camera configurations without making any prior assumptions.

2.3. **r-Consistency.**

An overview of r-Consistency, a photo-consistency measure proposed by Kutulakos in [16], is presented in Chapter 2 while discussing its usefulness to handle camera calibration errors. The r-Consistency test has the ability to handle the effects of outlier pixels in voxel projections caused by different error sources. Therefore, r-Consistency is also useful to handle the effects partial emptiness of surface voxels. However, the limitation of this method is that it does not take into

account the particular nature of the problem. Therefore, it is just a first step towards making the photo-consistency tests robust against the error sources that cause outlier pixels in voxel projections.

2.4. Other Related Techniques.

Another related piece of work is presented by Isidoro and Sclaroff in [12]. The authors introduce a new photo-consistency measure to avoid the problem of outliers in voxel projections to make the voxel-coloring more robust. The authors refer to this photo-consistency test as Oriented Per-Pixel Matching (OPPM). In this method, instead of representing the color statistics of a voxel projection by a mean and a variance, individual pixels from each voxel projection are used in its photo-consistency test. Their photo-consistency measure assures that smaller number of outlier pixels do not affect the consistency of a partially empty voxel. The robustness of OPPM is further increased by weighting the difference of pixel colors of two projections by the cosine of the angle between the optical axes of the corresponding viewpoints. Thus the viewpoints that are close to each other would have higher weights given to the difference of their pixel colors. The authors compare the results of using OPPM as photo-consistency with those of the Variance of Means photo-consistency test. Their method results in fewer holes in the reconstructed 3D model. One limitation of their method is the use of per-pixel matching which is highly computationally expensive. Further, the difference metric used to check the photo-consistency is a nonsystematic approach for handling sensor noise and camera calibration errors that are usually modelled as Gaussian noise.

Kang and Szeliski in [14] have proposed several interesting techniques for outliers handling resulting from occlusions in multi-view stereo. Some of their proposed methods, such as the use of spatially shiftable windows and temporal selection of the input views, can also be used to handle the outliers caused by the partially empty surface voxels in voxel-coloring. The limitation of their proposed techniques is the increased computational cost for multiple view voxel-coloring.

One approach to handle the uncertainty due to the outlier pixels, especially at low voxel resolutions, is to use a coarse-to-fine strategy as proposed by Montenegro et al. in [22]. In this method, if the uncertainty is high in deciding the photo-consistency of a larger voxel, the decision for its photo-consistency is delayed for later stage when the voxel size would be reduced. The delay in the decision to remove a voxel helps to avoid the erroneous carving of a partially empty surface voxel.

Another class of methods that can handle outliers in voxel-coloring are the probabilistic approaches [4, 2]. These probabilistic methods make use of the background statistics and calculate the carving probability as a likelihood ratio. The use of background variance statistics helps to

reduce the uncertainty in photo-consistency of a voxel in the presence of outliers. The probabilistic methods are especially useful when no foreground/background segmentation information, e.g. silhouettes, is available.

3. Area Weighting to Handle Partial Emptiness of Surface Voxels

In this section, we present our proposed method to overcome the effect of outliers in voxel-coloring resulting from partial emptiness of surface voxels. Like most robust computer vision algorithms [32], we have made use of the problem analysis to make voxel-coloring robust against this error source.

The amount of outlier pixels in the projections of a partly empty surface voxel depends on the following factors:

- (i) The percentage of the emptiness of the surface voxel. The higher the emptiness, the higher the amount of outliers.
- (ii) The angle between the unit normal of the 3D surface contained in the voxel and the camera optical axis of the input viewpoint. This effect is clear in Figures 3.1 and 3.2. In Figure 3.1, the optical axis of the central camera is almost aligned with the normal of the surface represented by voxel V_2 . Therefore, the projection of voxel V_2 in image I_2 does not contain outlier pixels. Whereas the projections of voxel V_2 in image I_1 and I_2 contain outliers as the camera optical axes for these images form an angle of approximately 45° with respect to the surface normal. The same concept is true in Figure 3.2 where the edge voxel is partly empty.
- (iii) The angle between the optical axes of the cameras that can see the voxel. For a given emptiness of a surface voxel, the higher the angular spread of the input views, the more outliers in the voxel projections.

For a perspective camera model, the pixel coordinates of a 3D point can be calculated as follows:

$$(3.1) \quad x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z},$$

where (x, y) are the sensor coordinates and (X, Y, Z) are coordinates of the 3D point in the camera reference frame. The number of pixels in the voxel projection in different visible input images are usually different. Using Equation (3.1), and from Figure 3.1 and Figure 3.2, it is clear that the number of pixels in the voxel projection depends on the following factors

- (i) The focal length f of the camera; the projection area is proportional to f^2 .

- (ii) The distance of the image plane from the voxel along the optical axis of the camera, i.e. the depth Z of the 3D point; the projection area is proportional to $\frac{1}{Z^2}$.
- (iii) The angle between the principle axis of the camera and the unit normal of the surface represented by the voxel; the larger the angle, the smaller the area of projection

If the number of pixels in voxel projections is normalized for their difference in focal length and depth for different images, by multiplying the number of pixels with $(\frac{f}{Z})^2$, a voxel projection area will only depend on the 3^{rd} factor in the above list, i.e. the angle between the surface normal and the camera optical axis. Further, it is important to note that the amount of outliers in a voxel projection also depends on the same factor. In other words, the larger the voxel projection area, the smaller the amount of outlier pixels in the projection. Therefore, the pixels in larger area projections of a voxel are better representations of the surface contained by that voxel. Hence, if the normalized number of pixels in a voxel projection are used as weights when calculating the color statistics of the voxel i.e. its mean color and variance, the effect of the outliers due to the emptiness of surface voxels will be reduced. This is our proposed solution to reduce the effects of partial emptiness of surface voxels in voxel-coloring. We call this technique **Area Weighting**.

3.1. Area Weighted Variance of Means Photo-Consistency Test.

Our proposed solution is simple to use with existing photo-consistency tests to make them robust against the outliers problem. In order to show the applicability of our method with the existing techniques, we are using Area Weighting with the Variance of Means photo-consistency test. We call the new robust photo-consistency test Area Weighted Variance of Means (AWVoM). Let μ_i be the mean color (Red, Green or Blue) of the pixels in a voxel v 's projection in the i th image, N be the total number of images that can see a voxel and w_i be the weight given to a projection color statistics in proportion to its normalized area, i.e. the normalized pixel count. The Variance of Means (VoM) photo-consistency test can be modified as AWVoM as follows:

$$(3.2) \quad \text{The average of the mean projection colors of a voxel} = \mu_v = \frac{1}{N} \sum_{i=1}^N w_i \cdot \mu_i,$$

and

$$(3.3) \quad \text{The variance of mean projection colors of a voxel} = Var_v = \frac{N}{N-1} \sum_{i=1}^N w_i \cdot (\mu_i - \mu_v)^2$$

A similar area weighting scheme has been used by Matthew Loper in his software called Archimedes [20], which is an implementation of Generalized Voxel Coloring (GVC) Algorithm [7]. However, the author does not explain the use of this technique. We were unable to find any

description of this technique in voxel-coloring literature as well. Therefore, one contribution of our work in this thesis is to provide the rationale to use Area Weighting and to verify it experimentally.

3.2. Advantages of Area Weighting.

Our proposed technique has following advantages over the existing techniques to handle the outlier pixels:

- Because of its simplicity, Area Weighting can be easily added to most existing photo-consistency tests.
- There is very little computational cost involved in incorporating the Area Weighting, especially in comparison with the methods proposed in [4;7;15].
- Area Weighting can handle the partial emptiness of both the edge voxels and voxels on continuous surface. This is evident from Figures 3.1 and 3.2.
- At low voxel resolutions, a voxel might be partially occluded in some views. As a result the mean color of the pixels in the occluded projection could be significantly different from the mean color of the projections where the voxel is fully visible. This problem will occur even if the voxel is completely opaque, i.e. not empty at all. As shown in Figure 3.3, the area of the partially occluded projection will be smaller compared to the projections where the voxel is completely visible. Therefore, the use of Area Weighting will increase the robustness of voxel-coloring against these errors as well.

3.3. Limitations of Area Weighting.

Area Weighting is an approximate method and is based on the assumption that the amount of inliers in a voxel projection are proportional to the projection area. This method will fail when all projections of a partially empty surface voxel have similar areas. An example of this circumstance is shown in Figure 3.4, where the angles between the normal of the surface contained in a voxel and the optical axes of all three cameras are same. However, as it will be shown in the next chapter, for most of the cases Area Weighting improves the reconstruction results of a voxel-coloring algorithm.

3.3 AREA WEIGHTING TO HANDLE PARTIAL EMPTINESS OF SURFACE VOXELS

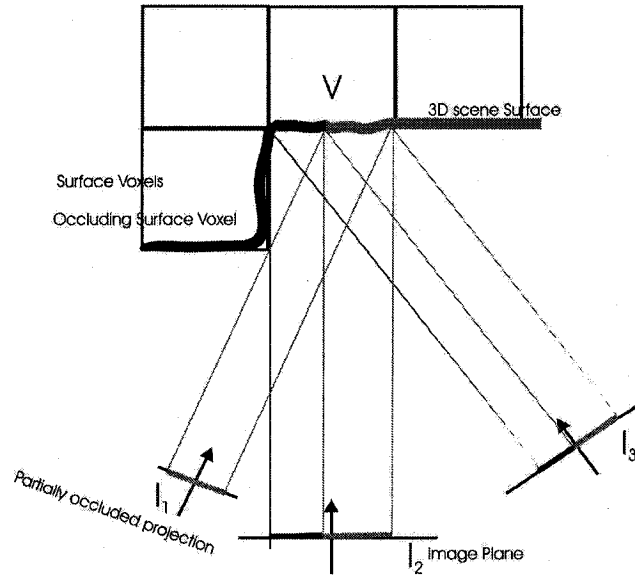


FIGURE 3.3. The use of Area Weighting to handle the effects of partially occluded voxels. The occluded projection of the voxel V in image I_1 captures only half of the surface contained in the voxel. Therefore, the color distribution of the pixels in this projection may be quite different from the color distribution of the pixels in the other two projections. The use of Area Weighting will reduce the role of the voxel projection in I_1 in the photo-consistency test.

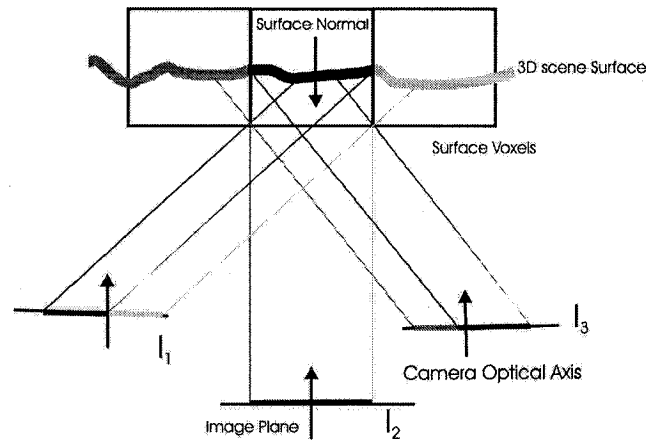


FIGURE 3.4. An example when Area Weighting is unable to overcome the effect of outliers as all the projections of a voxel have similar areas.

CHAPTER 4

Experiments and Results

This chapter presents the experimental verification of our proposed methods to handle the effects of both camera calibration errors and partial emptiness of surface voxels. The first part of the chapter will explain the experimental setup. The results and a discussion of these results will be presented in the second part of this chapter. It will be shown that our proposed methods improve the reconstruction results in the presence of errors.

1. Experimental Setup

This section gives the necessary details of our experimental setup and verification methods.

1.1. Data Sets.

We have used our own calibrated data sets for the experiments. Three different objects, namely a pipe player statue (Piper), a stuffed chipmunk (Chipmunk) and a stuffed dog (Puppy), that are being used as 3D scene models are shown in Figure (4.1). These objects were selected for their shape and texture qualities. Most manmade objects have symmetric surface properties. However, the test objects do not, and thus the results can be extended to arbitrary 3D scenes. Piper has rich surface variation details but its texture variation is relatively smooth. The other two objects have smoother surfaces. However, these two objects have textureless surface patches of uniform color. As was explained in Chapter 1 and was shown in Figure 1.3, a lack of texture in voxel coloring fattens the reconstructed 3D model in the absence of any prior smoothness constraint. The dimensions of the initial bounding box for the test objects are given in Table 4.1. Two different voxel resolutions were used for our experiments; a Maximum Voxels Per Side (MVPS) of 100 and 180 in the initial bounding box. We used the same voxel size for all the voxels. Therefore, the number of voxels along each side of the bounding box may be different and will be proportional to

the length of that side. Table 4.1 gives the size of a voxel for two different voxel resolutions for all three objects.

Twelve different calibrated views of each object, taken at a uniform angular separation of 30° , were used as input to the voxel-coloring algorithms. These input views cover the 3D object from all directions in one plane, as shown in Figure 4.2 for Piper. Therefore, the output of a voxel-coloring algorithm is a complete 3D model of the object. The input images were acquired using a single fixed camera, while the object was rotated on a turntable. The resolution for each image was 640×480 pixels.

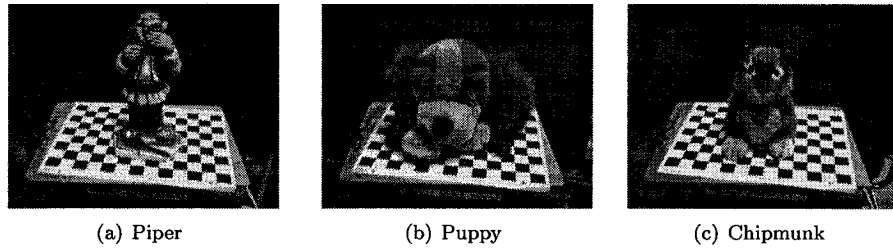


FIGURE 4.1. A single image of each 3D test objects

TABLE 4.1. Dimensions of the initial bounding box for each test object

| Test Object | Initial Bounding Box Dimensions in mm ($X \times Y \times Z$) | Voxel Size in mm | |
|-------------|---|------------------|--------------|
| | | $MVPS = 100$ | $MVPS = 180$ |
| Piper | $120 \times 110 \times 210$ | 2.1 | 1.167 |
| Puppy | $260 \times 190 \times 180$ | 2.6 | 1.44 |
| Chipmunk | $130 \times 150 \times 170$ | 1.7 | 0.94 |

The use of background segmentation information is quite common in voxel-coloring algorithms. The advantage of using this information is that the voxels that project to the background pixels can be easily detected and removed. However, in most practical situations, it is not possible to separate the object from its background perfectly. Furthermore, if a sufficient number of input views are used, the difference between the quality of a Photo Hull obtained from voxel-coloring using silhouettes and a Visual Hull is sometimes not noticeable. This fact is clear from Figure 4.3. A Visual Hull is a 3D reconstructed model of an object using only its silhouettes without making any use of color information. Therefore, all the reconstruction results presented in this chapter were obtained without using any background segmentation information. However, as will be explained later, we will need background segmentation to perform a precise quantitative analysis of the reconstruction results. Therefore, we have used uniform colored background surfaces in our data sets to obtain background segmentation easily.

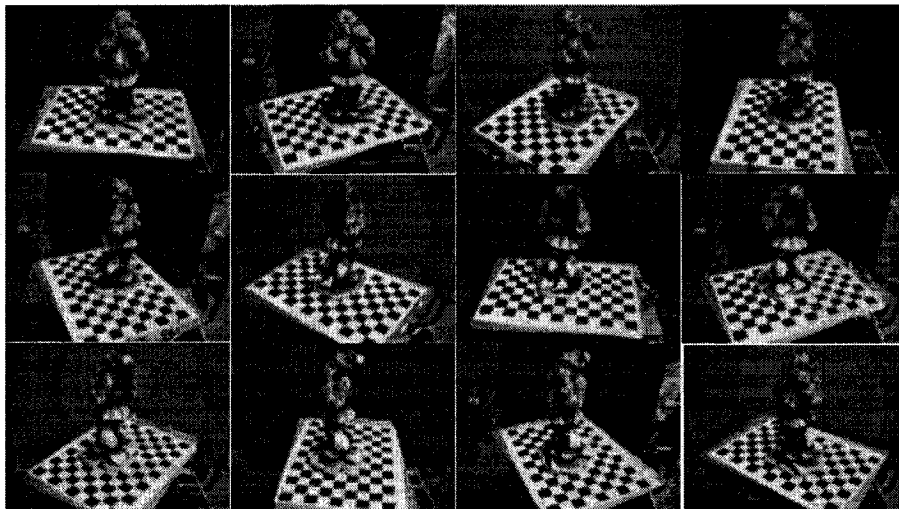


FIGURE 4.2. Twelve different views of Piper in a plane at a uniform interval of 30°

1.2. Camera Calibration.

The intrinsic and extrinsic parameters of the camera were estimated using a Matlab Camera Calibration Toolbox [3]. The toolbox was designed for the calibration of a single camera. However, we were able to easily extend its use for multiple cameras. Technical details of this calibration toolbox can be found in [3]. In this calibration method, a checkerboard is used as a calibration object. At least two images of the checkerboard are required for estimating both intrinsic and extrinsic parameters of a camera. We used twenty extra views of the calibration object at different angles to make sure that the calibration feature points are properly distributed in both the image and the object space. The toolbox provides sufficient information about the distribution of the calibration errors. The reprojection error for all the feature points in all the input views, and the standard deviation of the error in each estimated calibration parameter, were also available from the toolbox.

The reprojection error of the estimated calibration parameters was very small and its standard deviation was around 0.2 pixels. In order to test our proposed method of improvement for different levels of camera calibration error, we controlled the calibration error by adding zero mean Gaussian noise to the coordinates of the image feature points before calibration. Two new camera calibrations with the average reprojection error standard deviations of 1 and 2 pixels were obtained using this method.

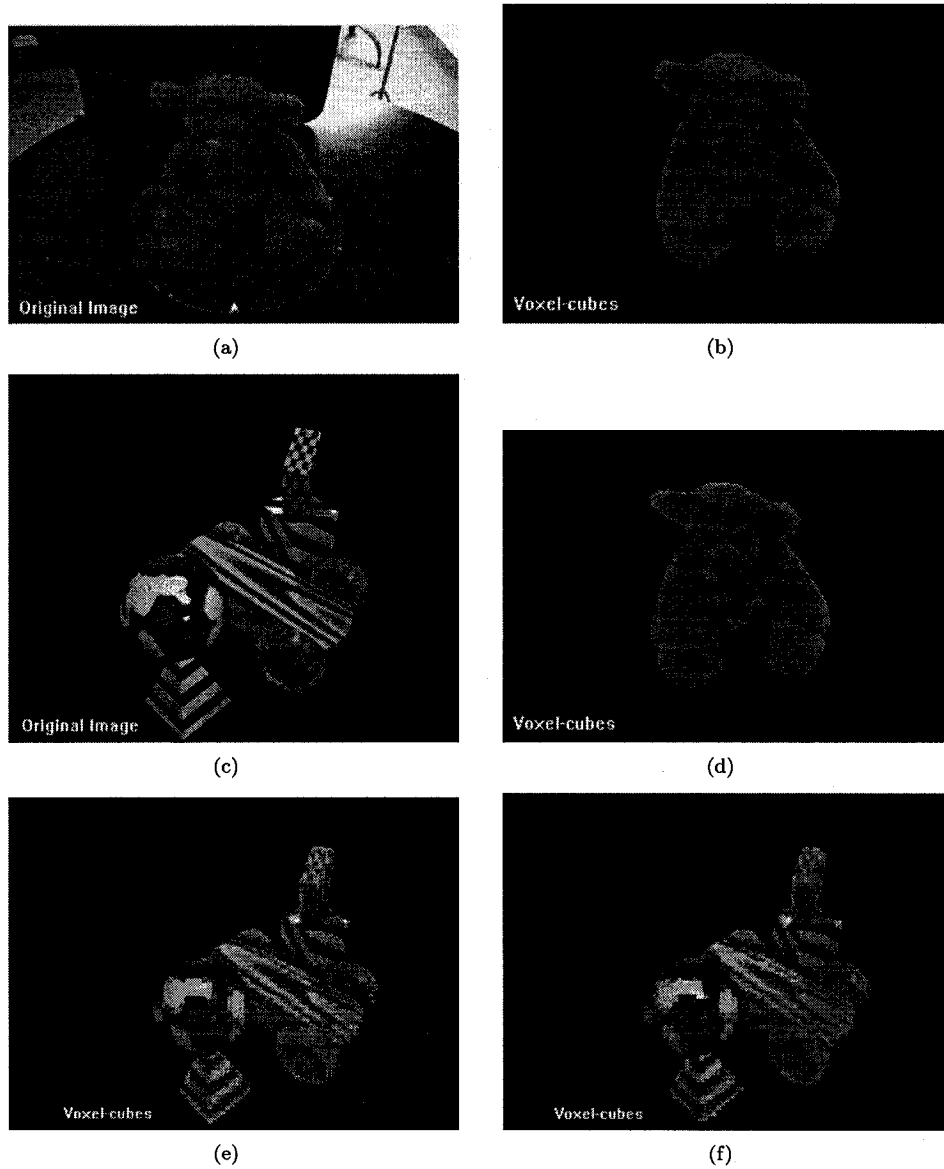


FIGURE 4.3. Comparison of Photo Hull and Visual Hull. Background segmentation information, i.e. silhouettes, was used to generate Photo Hull. (a) is one of the ten input images of a stuffed bear. This data set is courtesy of Matthew Loper [20]. (b) is the reconstructed Visual Hull and (d) is the reconstructed Photo Hull for this data set. (c) is one of the seventeen input images of a synthetic toy car. This data set is courtesy of Bruce Culbertson [7]. (e) is the Visual Hull and (f) is the Photo Hull for this data set. Because of a good number of input views, the difference between the two reconstructions is very small for the toy car. The Visual Hull looks even smoother and better than Photo Hull for this data set.

1.3. Voxel-Coloring Algorithm.

The methods of improvement proposed in this thesis can be used with most of existing voxel-coloring algorithms. For the experiments in this thesis, we used the Generalized Voxel Coloring using Item Buffers (GVC-IB) algorithm [7] for visibility calculations. Photo-consistency tests are based on the Variance of Means (VoM) method which was discussed in Chapter 1. It is one of the simplest photo-consistency tests. For the photo-consistency check of a voxel in this method, the sum of the variance of all three color channels is compared against a threshold value. We used the RGB color space for our experiments.

A voxel is assumed to be a perfect 3D cube and is projected accurately on the input images for visibility calculations. No assumption was made about the shape of a voxel projection in an image and it could vary significantly in different images. Therefore, the calculation of a voxel projection color statistics was a little more laborious but accurate. The size of a voxel is adjusted by changing the resolution of the voxel grid that represents the initial scene volume.

1.3.1. Pseudo-Code for Voxel-Coloring Algorithm.

The pseudo-code for the voxel-coloring algorithm that was implemented and used in our experiments for this chapter is presented below:

Step 1: Initialize a List of Surface Voxels (LSV) of the initial volume that contains the 3D object.

Step 2: Visibility Calculation

For each input view

- (i) Render a new image for the volume represented by the LSV. To render an image, the color of a voxel is assumed to be equal to its ID. Thus, the color of a pixel in the rendered image will determine the ID of the surface voxel visible from that pixel. Therefore, each rendered image contains the voxel visibility information and is called an Item Buffer (IB).

Step 3: Accumulating Voxel Color Statistics

- (i) Initialize a list to store the color statistics of all the surface voxels in the LSV
- (ii) *For each pixel of each input image*
 - *If the color of the same pixel in the corresponding Item Buffer is a valid surface voxel, ID then add this pixel's coordinates and color in the voxels statistics list*

Step 4: Photo-Consistency Check

For each voxel v in the LSV, if the voxel v is visible in more than one input image

- (i) *For each visible input image*

- *If Adaptive Gaussian Averaging flag is ON, then calculate the error variance of 2D coordinates of the pixels in voxel v 's projection and add neighboring pixels in the voxel's projection with calculated Gaussian error probability.*
- *Calculate the mean color of all the pixels in the projection.*
- (ii) *Calculate average of the mean colors of all projections of the voxel. Use projection areas as weights if the Area Weighting flag is ON.*
- (iii) *Calculate the variance of the mean projection colors. Use projection areas as weights if the Area Weighting flag is ON.*
- (iv) *Add the variance of the three color channels to calculate a sum variance*
- (v) *If the sum variance is less than the global variance threshold, then the voxel v is declared consistent and is left in the LSV*
- (vi) *Else remove v from the LSV, set v as non-consistent, and increment the carved voxels count.*
- (vii) *Add the neighboring voxels of the carved voxel v in the LSV if these new voxels are not already present in the LSV or have not previously been declared non-consistent.*

Step 5: Algorithm Termination Check

- (i) *If no voxel was removed during photo-consistency check step, terminate the algorithm. The voxels that are still present in the LSV form the surface of the photo-consistent reconstructed 3D model.*
- (ii) *Else Go back to Step2.*

At the end of the algorithm, the average color of each voxel is used for texture mapping of the rendered 3D model.

1.4. Quantitative Verification of the Test Results.

It is important to have a quantitative measure of error to test and compare the quality of reconstruction of different voxel-coloring algorithms. A voxel-coloring algorithm is supposed to generate a Photo Hull that is a superset of all the possible photo-consistent surfaces. A Photo Hull should be able to reproduce the input image by reprojection. Therefore, an obvious measure to test the quality of reconstruction of a voxel-coloring algorithm is the color difference of the input images and the reprojected images, i.e. the reprojection error in pixel color. A voxel-coloring algorithm is usually supposed to reconstruct only the 3D object present in the image and not the background surfaces. As a result, all the pixels that belong to the background would be black in the reprojected image. Therefore, one method to see the difference between an input and a reprojected image is to take into account only the pixels that are not black in the reprojected image. However, this method is flawed for the following reasons:

- (i) If a part of the 3D object surface is black, it will not be used in the calculation of reprojection error.
- (ii) An over-carving of the 3D model will result in black spots on the reconstructed surface. Therefore, ignoring the black pixels in the reprojected image will lead to overlooking the effect of over-carving.
- (iii) The voxels that do not represent the object surface but have been falsely declared as part of the 3D object will project on the background pixels. These background pixels will have some color and are given the name of *false positive pixels* in this thesis. The false positive pixels are unwanted regardless of their color and reprojection error. Therefore, using all the colored pixels in the reprojected image to calculate the reprojection error will not take into account the effect of false positives properly.

The reprojection error calculation can be made more accurate and the above-mentioned problems can be avoided by using the object/background segmentation information, i.e. silhouettes. With the availability of the object/background segmentation information, the quantitative error measure is composed of the following two parts:

Reprojection Error: The reprojection error of pixel color is calculated only for the pixels belonging to the object surface, i.e. the pixels inside the silhouette of the object. An average of the reprojection error in each color channel over all the input views was used as an error measure in algorithm comparisons.

False Positive Pixel Count: All the pixels in the reprojected images that are outside the silhouettes of the object, and have some color, are considered as false positive pixels. The sum of the false positive pixels in all reprojected images for all input views was used as the second error measure.

Figure 4.4 illustrates the regions used for the calculation of reprojection error and false positive pixel count. While comparing two different voxel-coloring algorithms, an algorithm will be declared better if it results in a lower reprojection error for the same or smaller number of false positive pixels, or fewer false positive pixels for the same or lower reprojection error. It is important to note that object/background segmentation information is not being used in the reconstruction process itself because it reduces the effectiveness of voxel-coloring. This has been discussed earlier in Section 1.1.

2. Results for the use of Area Weighting

In this section, the results of two voxel-coloring algorithms will be compared to evaluate the effect of using our proposed Area Weighting. The basic algorithm uses simple Variance of Means

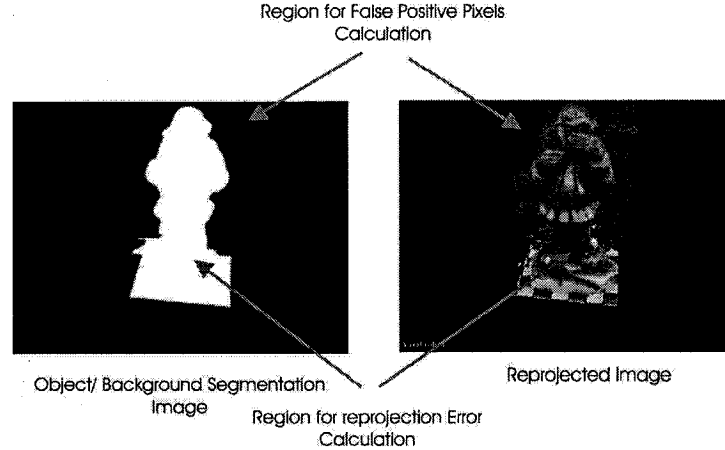


FIGURE 4.4. Pixels inside the object silhouette were used for the calculation of reprojection error. Colored pixels outside the silhouette of the object were considered as false positive pixels.

(VoM) as its photo-consistency test. The enhanced algorithm makes use of Area Weighted Variance of Means (AWVoM) test. The projection area weights reduce the calculated variance of the mean projection colors. Therefore, VoM and AWVoM will result in significantly different amounts of voxel carving for the same variance threshold; the algorithm using AWVoM will carve less. For performance comparison, we need to see which algorithm performs better under similar conditions. In this case an adequate condition for comparison could be the same number of voxels carved from the initial volume. The better algorithm will be the one that carves fewer holes in the reconstructed model, i.e. has low reprojection error for the same false positive pixel count or vice versa.

Another important fact is that for VoM based photo-consistency tests, an increase in the variance threshold reduces the over-carving of the reconstructed 3D model, resulting in lower reprojection error. However, this is achieved at the expense of higher false positive pixel count in the reprojected images. This fact is clear from Figure 4.7, which compares the results of simple VoM tests for two different variance thresholds for the Piper data set.

In Chapter 3, we have discussed the normalization of a voxel projection areas for difference in focal length f and depth Z . However, if all the cameras have the same focal length and the distance of an object from the cameras is also almost similar, which is the case for our test data sets, we have noticed that the normalization does not make much difference in the reconstruction results. Therefore, we did not normalize the projection areas for f and Z differences in our experiments. This helped to reduce the computational cost of the algorithm.

Figures 4.5 to 4.11 compare the reconstruction results of VoM and AWVoM photo-consistency tests. These results were obtained for two different voxel resolutions with Maximum Voxels Per

4.2 RESULTS FOR THE USE OF AREA WEIGHTING

Side (MVPS) of 100 and 180, for all the three data sets. Note that in quantitative comparison of the results, the norm of the average reprojection error in the three color channels was used, i.e.

$$(2.1) \quad \text{Total Reprojection Error} = \sqrt{\epsilon_R^2 + \epsilon_G^2 + \epsilon_B^2},$$

where ϵ_R , ϵ_G and ϵ_B are the average reprojection errors in each color channel.

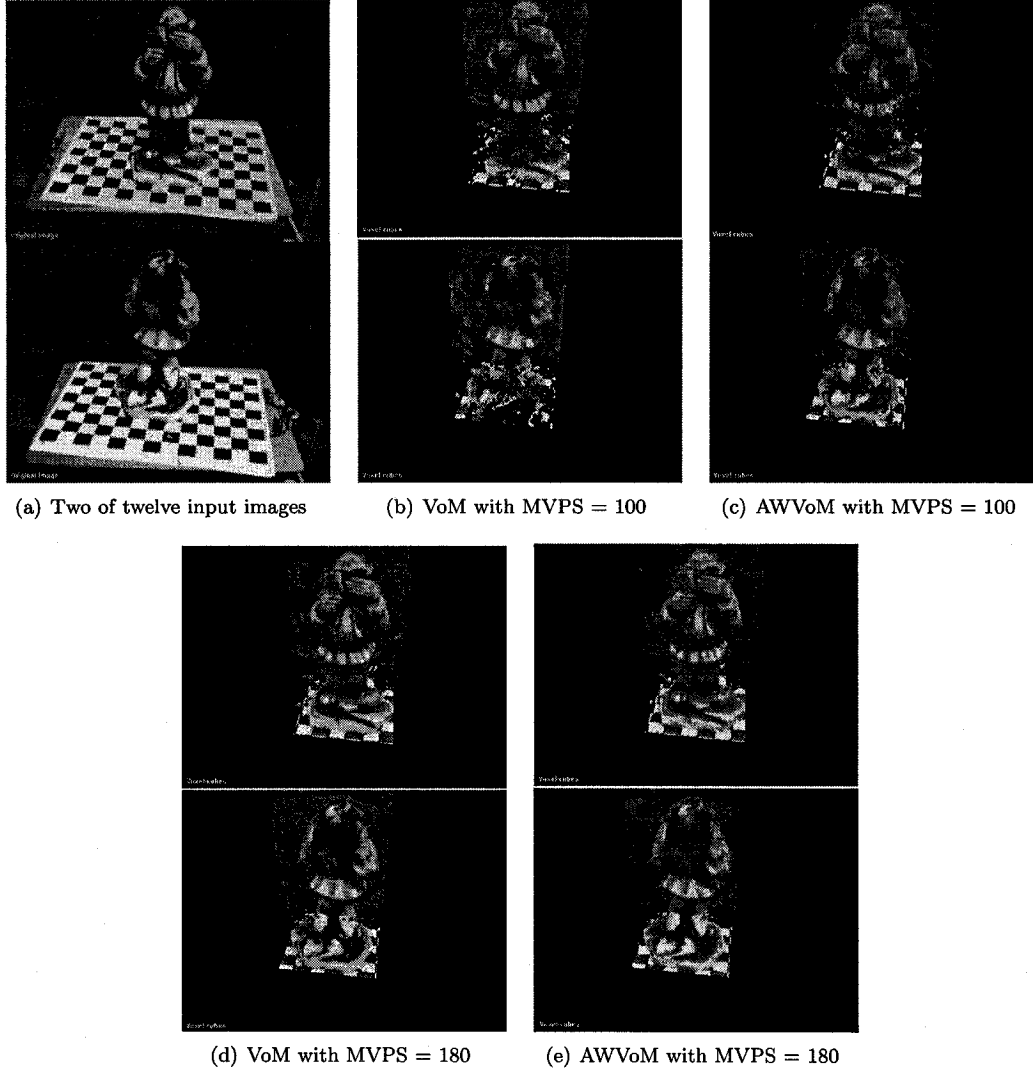


FIGURE 4.5. Reconstruction results that show the effectiveness of Area Weighting for the Piper data set for two different voxel resolutions, i.e. Maximum Voxels Per Side (MVPS). The corresponding analytical comparison is shown in Figure 4.6.

4.2 RESULTS FOR THE USE OF AREA WEIGHTING

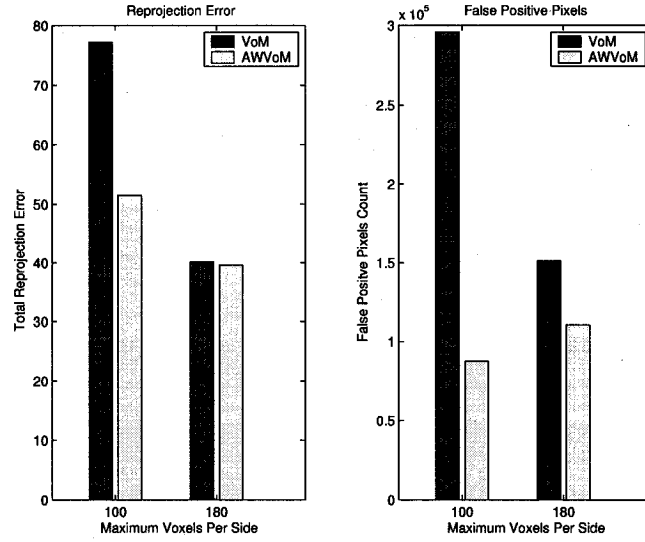


FIGURE 4.6. Quantitative comparison of the results shown in Figure 4.5 for the Piper data set. The corresponding numbers are given in Table B.1 in Appendix B.

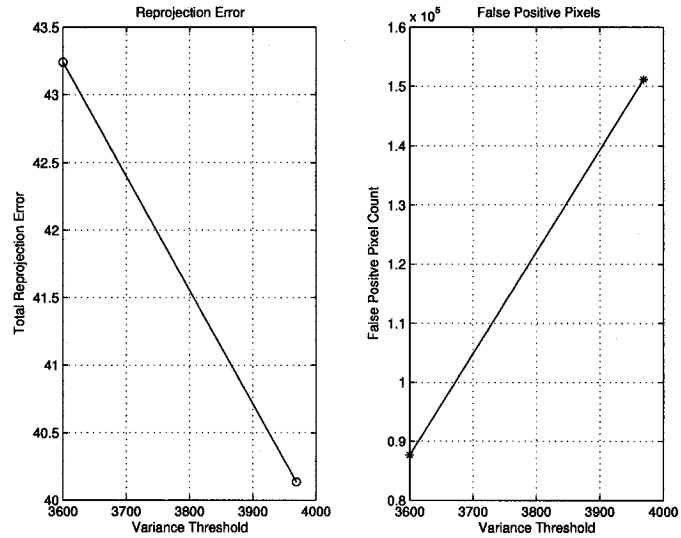


FIGURE 4.7. The effect of changing the variance threshold on reprojection error and false positive pixel count. These are the results of a VoM test for the Piper data set with MVPS of 180. The exact numbers are given in Table B.1 in Appendix B.

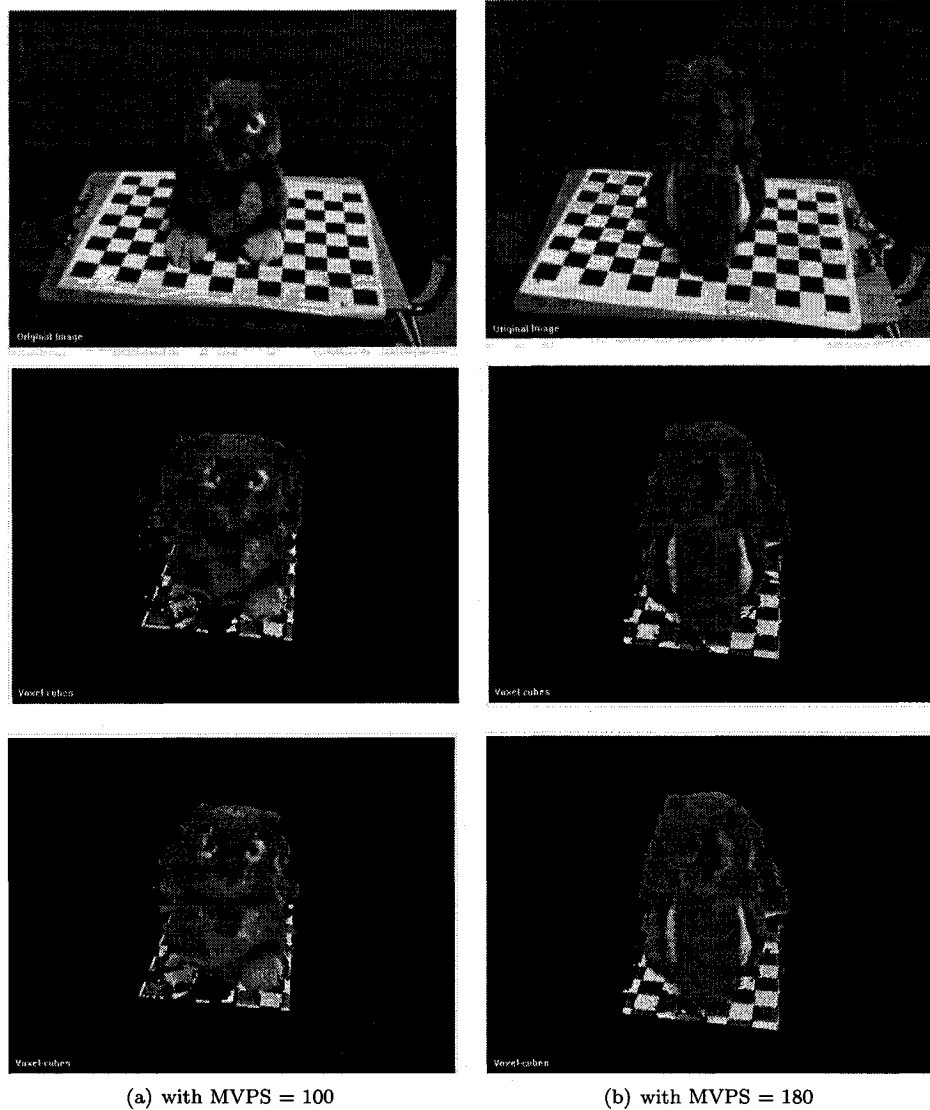


FIGURE 4.8. Reconstruction results showing the effectiveness of Area Weighting for the Chipmunk data set. Each column shows the results for one particular voxel resolution. The first picture in each column is the original input image. The second picture is the reconstruction result of simple Variance of Means (VoM) test. The last picture in each column is the reconstruction result of Area Weighted Variance of Means (AWVoM) test. The corresponding quantitative analysis is shown in Figure 4.9.

2.1. Low Voxel Resolution: MVPS 100.

A maximum of 100 voxels along the longest side of the initial bounding box of the scene volume yields a relatively coarse voxel resolution. It is clear from Tables B.1, B.2 and B.3 in Appendix B that VoM and AWVoM tests remove similar numbers of voxels from the initial volume.

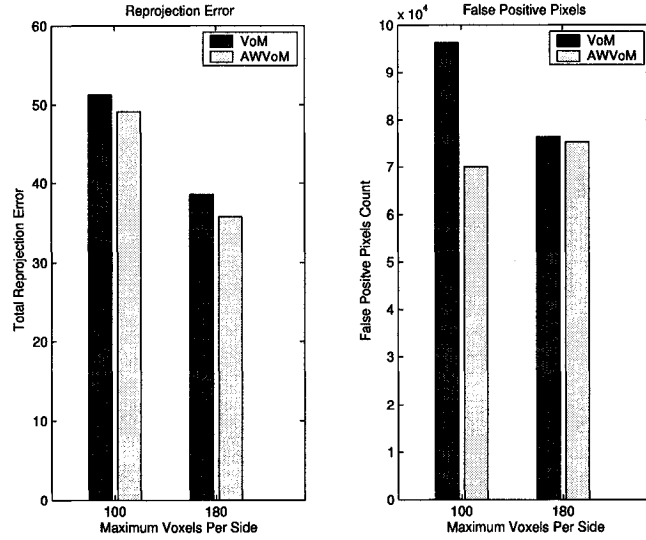


FIGURE 4.9. Quantitative comparison of results shown in Figure 4.8 for the Chipmunk data set. The exact numbers are given in Table B.2 in Appendix B.

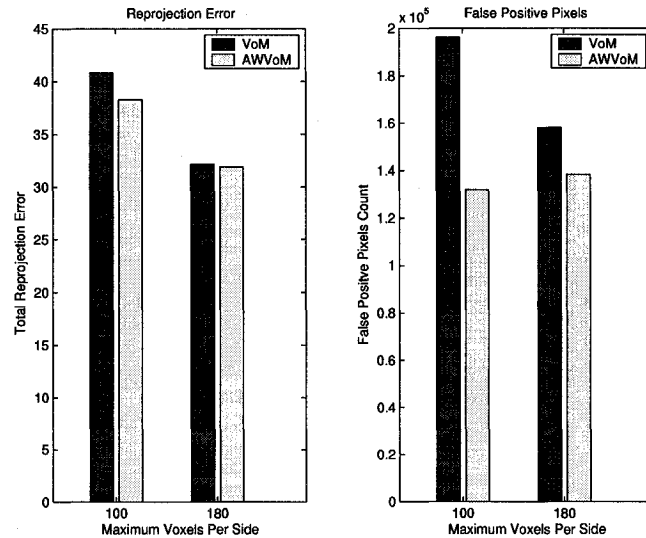


FIGURE 4.10. Quantitative comparison of results shown in Figure 4.11 for the Puppy data set. The exact numbers are given in Table B.3 in Appendix B.

Although the simple VoM test removes a slightly higher number of voxels, it results in more false positive pixels in the reprojected images. At the same time, the simple VoM test results in higher reprojection error due to over-carving of the 3D scene surface. In contrast, AWWoM test results in fewer false positive voxels around the object and also less over-carving of the object surface. The difference is visually clear from Figures 4.5, 4.8 and 4.11 showing the reconstructions. It is

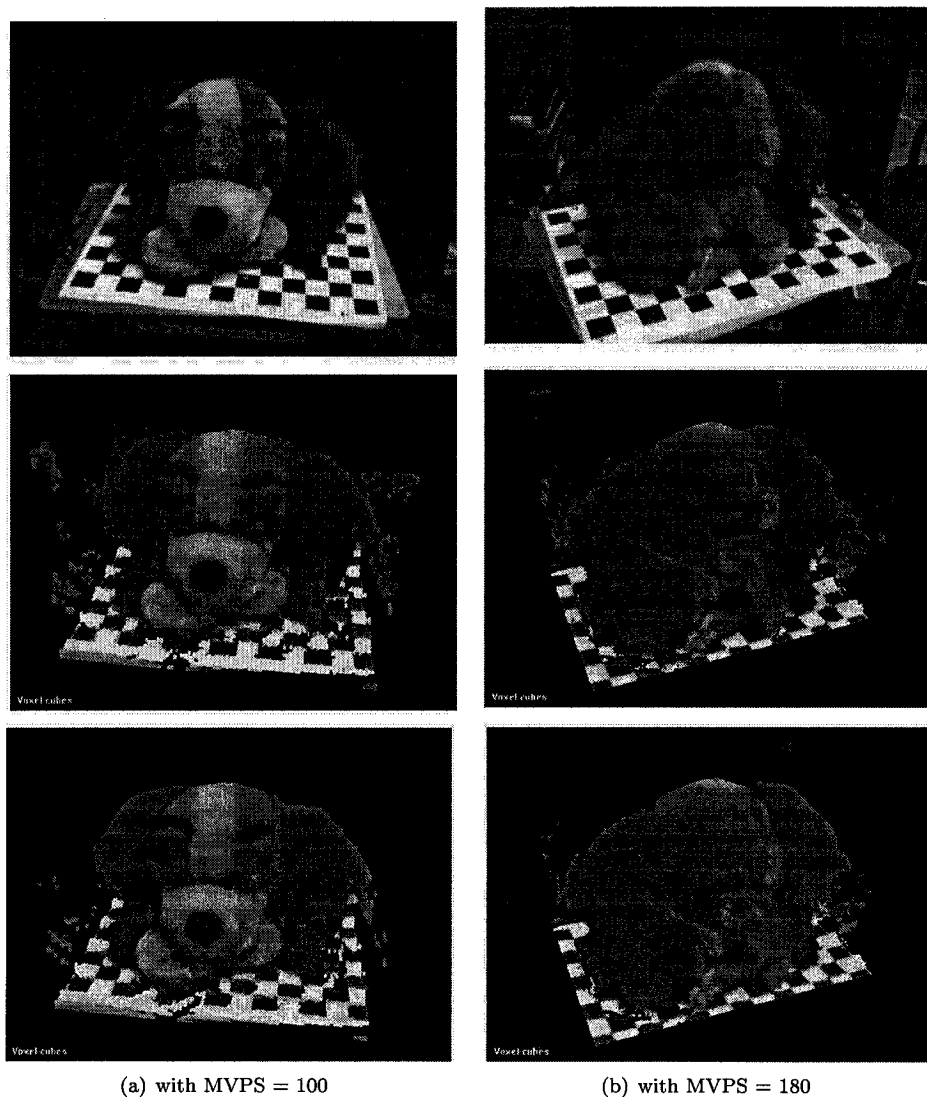


FIGURE 4.11. Reconstruction results showing the effectiveness of Area Weighting for the Puppy data set. Each column shows the results for one particular voxel resolution. The first picture in each column is the original input image. The second picture is the reconstruction result of simple Variance of Means (VoM) test. The last picture in each column is the reconstruction result of Area Weighted Variance of Means (AWVoM) test. The corresponding quantitative analysis is shown in Figure 4.10.

also verified from the quantitative comparison of these results shown in the form of bar graphs in Figures 4.6, 4.9 and 4.10 . These results clearly show that AWVoM is performing much better than VoM test for this voxel resolution.

The difference in the quality of reconstruction is the highest for the Piper data set where AWVoM test results in a three times smaller false positive pixel count while simultaneously reducing

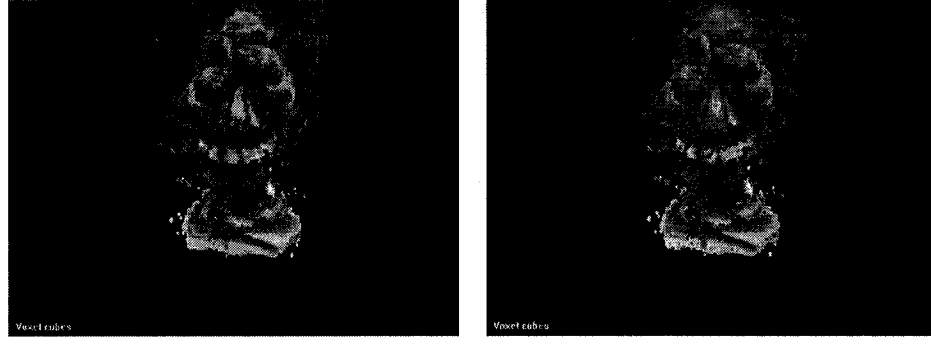
the reprojection error by half. The improvement achieved for the Chipmunk data set is the lowest. From the figures showing the reprojected images of the reconstructed models, it is clear that over-carving has stronger effects on regions with color and surface discontinuities. This is illustrated by the checkerboard floor being affected the most by over-carving. The Piper data set has the highest degree of surface discontinuities. Therefore, Area Weighting is more effective for this data set. Furthermore, for Chipmunk, the voxel size is 1.7mm for this voxel resolution, which is the smallest in all the three data sets. Therefore, the effect of partial emptiness of surface voxels is also less for Chipmunk, and so is the improvement achieved.

2.2. High Voxel Resolution: MVPS 180.

For the higher voxel resolution, the difference in the quality of reconstruction of the two tests is less significant. Nonetheless, AWVoM still performs better than VoM, i.e. AWVoM results in less reprojection error and, at the same time, fewer false positive pixels in the reprojected images. It is also clear from the figures showing the quantitative comparison of the results that the reconstruction quality has improved for both tests with an increase in the voxel resolution. An increase in the voxel resolution has reduced both the reprojection error and the false positive pixels count. This trend is quite predictable. One reason for this improvement is that at higher voxel resolution, the effect of the outlier pixels is reduced due to smaller voxel size. Another reason for the improvement is to have fewer pixels in voxel projections that reduce the ambiguity in the VoM based photo-consistency tests [5]. As at higher voxel resolution, the effect of partial emptiness of surface voxel is reduced, the improvement achieved using our proposed method is also reduced. This confirms that Area Weighting is effective in reducing the effect of this error source, and adapts to the amount of error present.

3. Results for Handling Camera Calibration Errors

In this section, it will be shown experimentally that camera calibration errors do have an impact on the 3D reconstruction quality of a voxel-coloring algorithm. Further, it will be shown that the reconstruction quality of an existing voxel-coloring algorithm can be significantly improved in the presence of calibration errors by combining our proposed method, Adaptive Gaussian Averaging (AGA) with AWVoM. For comparison purposes, a voxel-coloring algorithm using the AWVoM consistency test will be used as a basic algorithm that does not take calibration errors into account. The results of the basic algorithm will then be compared with that of an enhanced algorithm that makes use of our proposed Adaptive Gaussian Averaging. In the last part of this section, the results of this enhanced algorithm will also be compared with those of the r-Consistency test proposed by Kutulakos in [16].



(a) Without leaving single visibility voxels uncolored. (b) Single visibility voxels are left uncolored to make the holes more visually clear.

FIGURE 4.12. This figure shows the effect of leaving the single visibility voxels uncolored. It is clear that this method helps to make the holes in the reconstruction results visually more recognizable. The associated quantitative results are given in Table 4.2.

TABLE 4.2. The effect of leaving the single visibility voxels uncolored on the reprojection error and the false positive pixel count.

| | Reprojection Error (R, G, B) | False Positive pixel Count |
|-------------------|------------------------------|----------------------------|
| No Explicit Holes | 48.3962, 46.6454, 43.7232, | 83359 |
| Explicit Holes | 50.9198, 48.9524, 45.2727 | 82953 |

When the two algorithms are compared for their camera calibration error handling capability, the better algorithm should result in less over-carving of the object surface for the same number of false positive pixels. When a consistent surface voxel is carved by error, the voxel behind this carved surface voxel is exposed and is usually visible in only one image. The erroneous carving of a consistent surface voxel creates a hole on the surface of the reconstructed object. The holes carved in the surface of a 3D object can be shown more explicitly by not coloring the voxels behind them, i.e. the voxels that are visible in a single image only. In this case, any black spot on the continuous surface of a reconstructed 3D model will show that a hole has been carved. A better algorithm will result in fewer black spots, i.e. holes, on the surface of the reconstructed 3D model. However, this method to see the hole carving is only an approximation because of the following factors:

- (i) If the cameras are placed close to each other, a voxel behind a wrongly carved surface voxel might be visible in more than one image. In this case, this method will fail to show the holes in the 3D reconstruction.
- (ii) The surface voxels that lie on the surface discontinuities and on concave surfaces might be visible in a single image only. Leaving these single visibility voxels uncolored will show holes without any over-carving.

Due to the above limitations, we have used this method only for the Piper data set. We also compare the results of different algorithms without making the holes explicit in the reconstructions. Figure 4.12 shows the effect of leaving the single visibility voxels uncolored. It can be seen that the holes in the reconstructed model becomes visually more clear. Table 4.2 shows its effect on the reprojection error and false positive pixel count. The numbers in the table suggest that making the holes explicit bias the quantitative results by increasing the reprojection error and reducing the false positive pixel count. However, the difference is quite nominal.

The results shown in this section have been generated for two different voxel sizes with three different levels of calibration error. Therefore, the algorithms will be compared in two different ways:

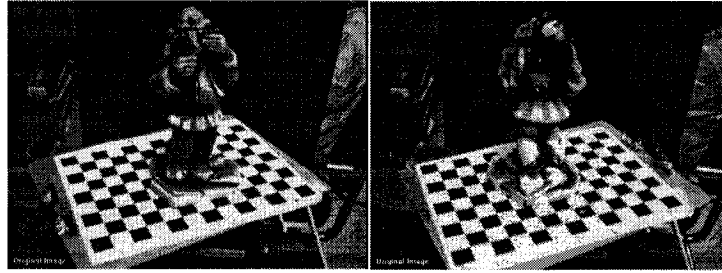
- (i) Comparison for different levels of calibration error for a fixed voxel resolution.
- (ii) Comparison for different voxel sizes for a fixed calibration error.

3.1. The Effect of Calibration Errors on Voxel-Coloring.

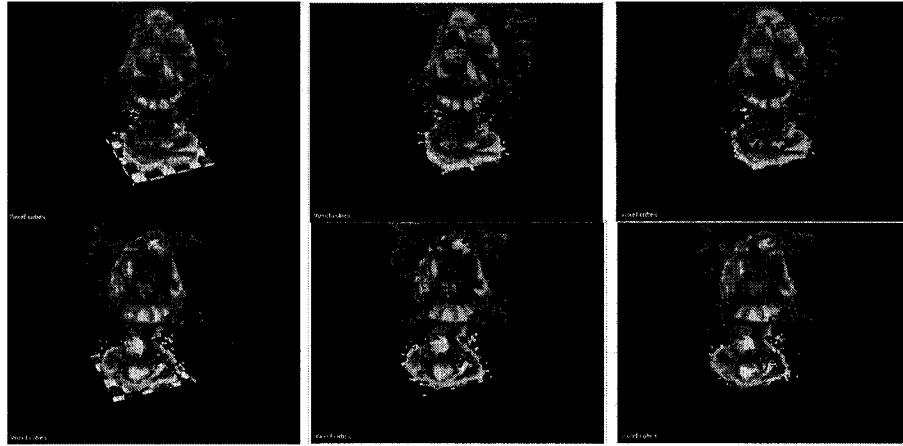
Figure 4.13 shows the effects of increasing camera calibration error on the reconstruction quality of an AWVoM based voxel-coloring algorithm. The corresponding quantitative results are shown in Figure 4.14. It is clear that an increase in calibration error results in higher reprojection error. However, the effect is more pronounced at lower voxel resolutions in terms of holes and over-carving. Figure 4.14 shows that the effect of an increase in the calibration error on the false positive pixel count is not monotonic. At low values of calibration error, the ambiguity resulting from the error increases the false matches and hence the false positive pixels. At higher calibration error, the effect of the error is same on both the consistent and non-consistent voxels, i.e. over-carving of the voxels. Therefore, the false positive pixels are also reduced at high calibration error values.

It is interesting to note that an increase in the calibration error has more effect on the color and surface discontinuities. Therefore, the checkerboard floor is the first to start disappearing with an increase in the calibration error. Another important fact to note is that the effect of calibration error on the reconstruction quality is reduced at higher voxel resolutions. At a low voxel resolution with MVPS of 100, an increase in the calibration error results in over-carving of the floor and the object surface. In contrast, at a higher voxel resolution with MVPS of 180, only the floor disappears. Over-carving of the object surface is hardly noticeable at this voxel resolution. This phenomenon can be explained by the fact that a voxel projection contains fewer pixels at higher voxel resolutions. And the VoM based photo-consistency test faces less ambiguity and is more robust against the error sources at higher voxel resolutions. Another reason is that for a smaller voxel size, the outlier pixels caused by the calibration errors will be closer to the center of a voxel

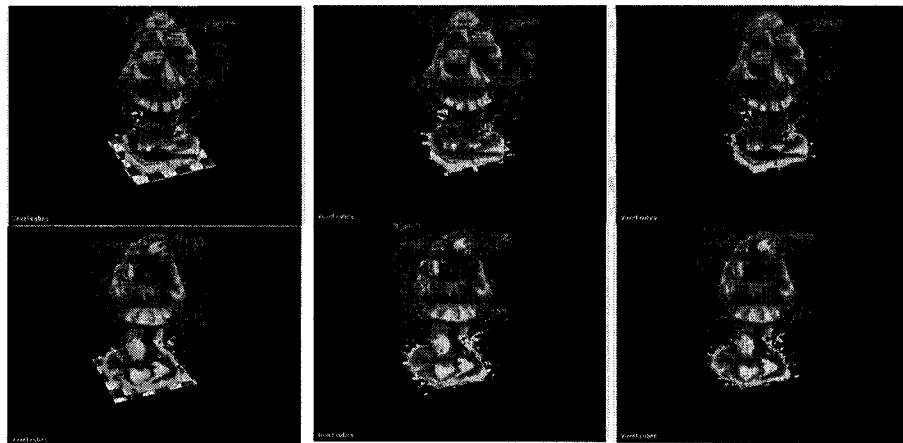
4.3 RESULTS FOR HANDLING CAMERA CALIBRATION ERRORS



(a) Two of the twelve input views



(b) Reconstruction for three different levels of calibration error with MVPS = 100



(c) Reconstruction for three different levels of calibration error with MVPS = 180

FIGURE 4.13. Reconstruction results of the AWWoM based voxel-coloring algorithm for different levels of calibration error for two different voxel resolutions for the Piper data set. All left side reconstructions are for an average pixel coordinates error standard deviation (STD) = 0.2 pixels, reconstructions shown in the center are for STD = 1 pixel, and those shown on the right are for STD = 2 pixels. The corresponding quantitative analysis is presented in Figure 4.14.

4.3 RESULTS FOR HANDLING CAMERA CALIBRATION ERRORS

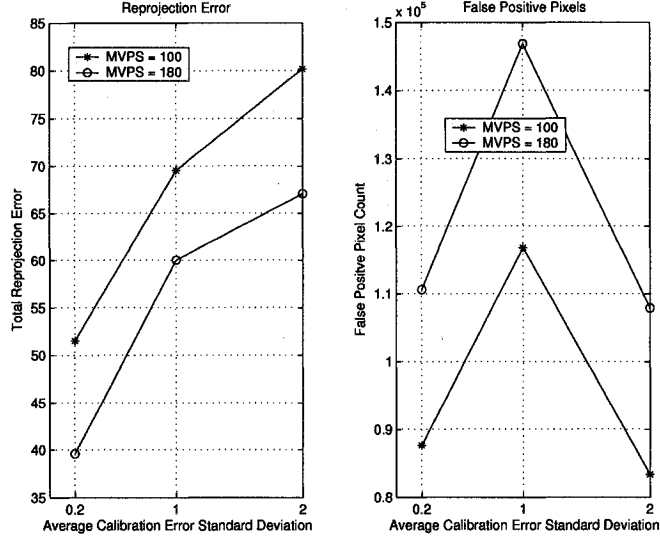


FIGURE 4.14. Quantitative analysis of the effect of increasing calibration error on the reconstruction results shown in Figure 4.13 for the Piper data set. The exact numbers are given in Table B.4 in Appendix B.

projection and hence will be similar in color to the other pixels in the projection. As a result, they have less effect on the projection color statistics.

Figure 4.13 shows the effect of increasing calibration error on the reconstruction results of voxel-coloring for one data set only. More extensive experiments were not conducted as the effect of calibration errors on voxel-coloring has already been discussed in the literature [5, 19].

3.2. Improvement in the Voxel-Coloring Results with Adaptive Gaussian Averaging.

Figures 4.15 and 4.16 show the effect of using Adaptive Gaussian Averaging (AGA) with AWVoM to overcome the calibration error effects for the Piper data set. These reconstruction results were obtained for three different levels of calibration error and for two different voxel resolutions. The holes in the reconstructed 3D models have been made explicit by leaving the single visibility voxels uncolored. The corresponding quantitative comparison of the reconstruction results for the reprojection error and the false positive pixel count is presented in Figures 4.17 and 4.18.

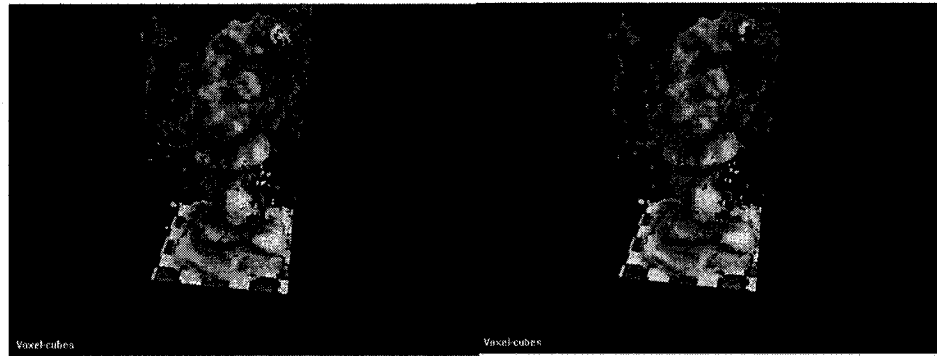
The improvement in the reconstruction quality is visually clear from the figures and is quantitatively verified as well. The use of AGA reduces the number of holes in the reconstructed model. It can be seen that the advantage of using AGA increases with an increase in calibration error. This shows the ability of AGA to adapt to the level of error present in the system. Moreover, the improvement offered by AGA is higher for lower voxel resolutions. This is because at lower voxel

resolutions, the effect of calibration errors on the reconstruction quality is higher in general for VoM based photo-consistency tests. This was discussed in Section 3.1. It is clear from the results that AGA not only reduces the over-carving of the 3D model, and hence the reprojection error, but also reduces the number of falsely selected voxels, i.e. false positive pixel count in reprojected images.

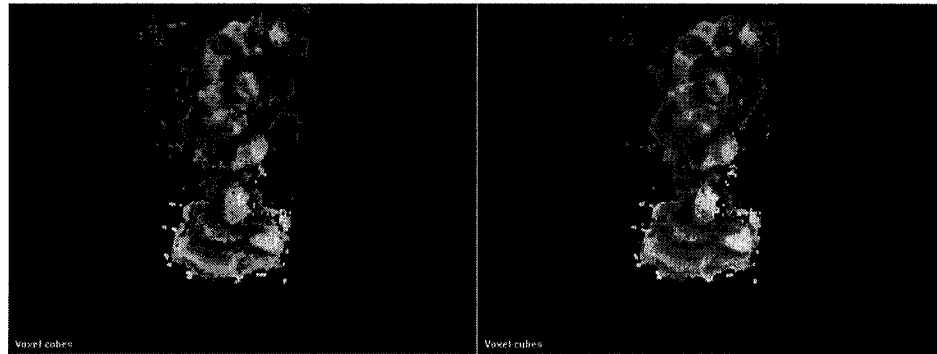
The reconstruction results shown in Figure 4.19 were obtained without making the holes explicit in the reconstructed models for Piper data set. These results have been obtained for only one calibration error value. As these results are consistent with the results with explicit holes for the same calibration error level, it was not required to repeat the experiment without explicit holes for the other two calibration error levels for the Piper data set.

Figures 4.19 to 4.28 compare the results of simple AWVoM based algorithm with enhanced AWVoM based algorithm making use of AGA for the Chipmunk and the Puppy data sets. The holes in the reconstructions were not made explicit for these results. The effect of AGA for these two data sets is consistent with the Piper data set; that is more improvement in the reconstruction quality is obtained for lower voxel resolutions and higher calibration errors. It is interesting to note that for the Chipmunk data set, the simple AWVoM based algorithm actually performs better than AGA for low calibration error when the average standard deviation of the error in pixel coordinates is approximately 0.2 pixels. The voxel size for the Chipmunk data set is the lowest among all three data sets. Therefore, calibration errors have a weaker effect on the reconstruction results for this data set. As the calibration error is very small, the Gaussian averaging is not of much help. Rather, the Gaussian averaging makes the VoM based photo-consistency test more ambiguous by adding extra pixels in the projections of a voxel. Therefore, in this case, the negative effects of Gaussian averaging are higher than the improvement offered by it. This suggests that for very low calibration errors and high voxel resolutions, the use of Adaptive Gaussian Averaging could result in further degradation of the reconstruction quality rather than offering any improvement.

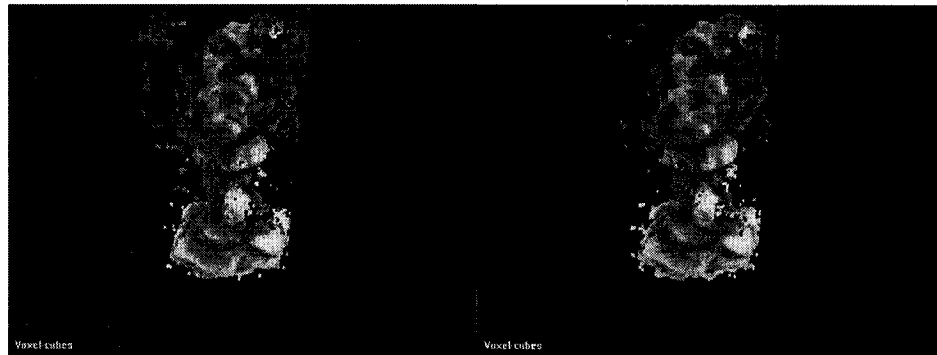
4.3 RESULTS FOR HANDLING CAMERA CALIBRATION ERRORS



(a) Average Pixel Coordinates Error STD = 0.2 pixels



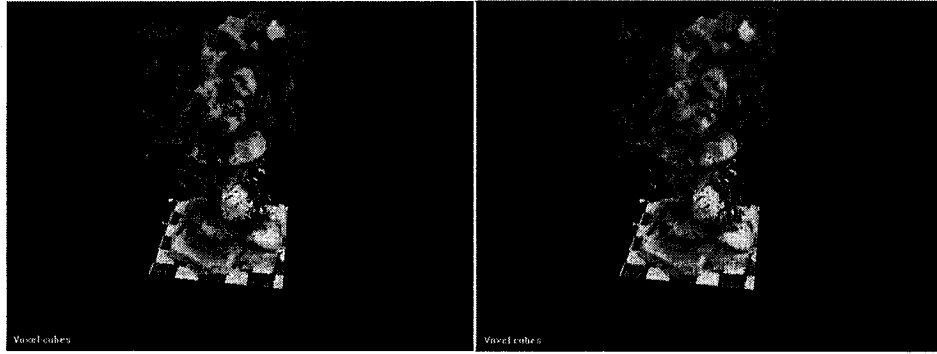
(b) Average Pixel Coordinates Error STD = 1 pixels



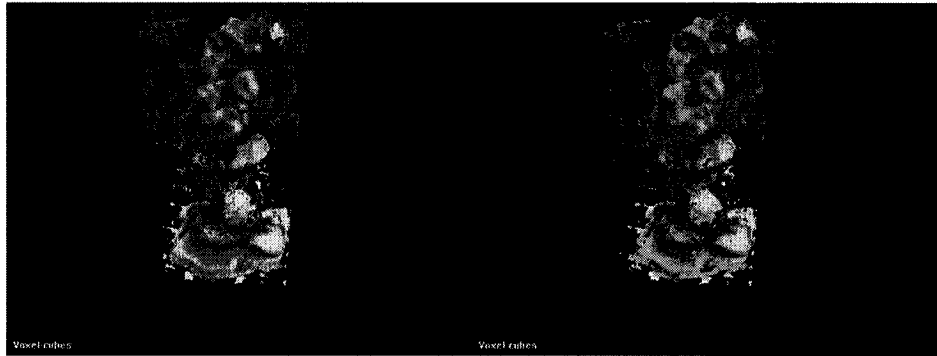
(c) Average Pixel Coordinates Error STD = 2 pixels

FIGURE 4.15. Piper data set, MVPS = 100. The effect of using AGA with AWWoM to overcome calibration error. The holes in the reconstruction results were made explicit by leaving the single visibility voxels uncolored. All left hand side reconstruction results are for a simple AWWoM based algorithm. The reconstruction results after adding AGA in the algorithm are shown in the right hand side pictures. The associated quantitative error comparison is shown in Figure 4.17.

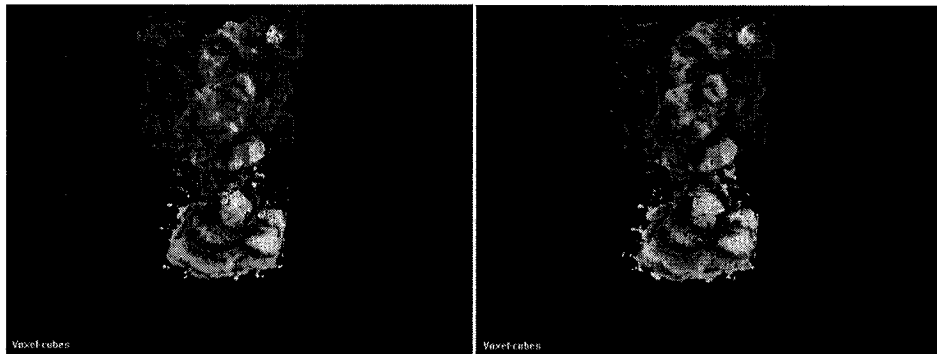
4.3 RESULTS FOR HANDLING CAMERA CALIBRATION ERRORS



(a) Average Pixel Coordinates Error STD = 0.2 pixels



(b) Average Pixel Coordinates Error STD = 1 pixels



(c) Average Pixel Coordinates Error STD = 2 pixels

FIGURE 4.16. Piper data set, MVPS = 180. The effect of using Adaptive Gaussian Averaging with AWVoM to overcome calibration error. The holes in the reconstruction results were made explicit by leaving the single visibility voxels uncolored. All left hand side reconstruction results are for a simple AWVoM based algorithm. The reconstruction results after adding Adaptive Gaussian Averaging in the algorithm are shown in the right hand side pictures. The associated quantitative error comparison is shown in Figure 4.18.

4.3 RESULTS FOR HANDLING CAMERA CALIBRATION ERRORS

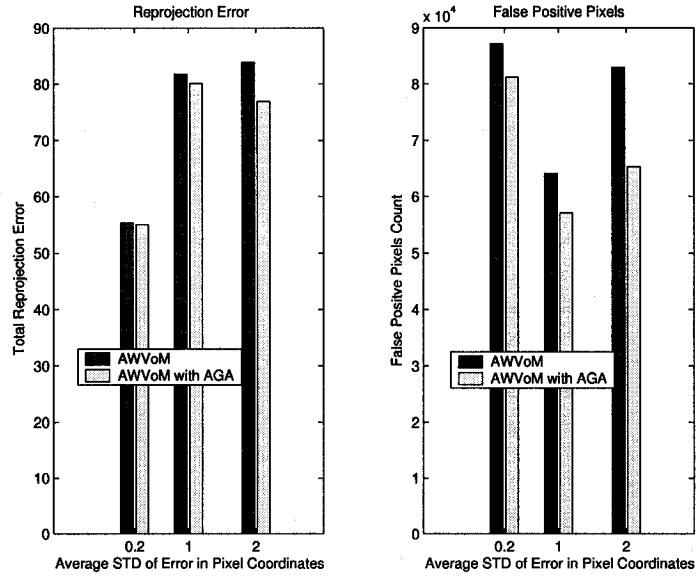


FIGURE 4.17. Quantitative comparison of results shown in Figure 4.15 for the Piper data set for MVPS of 100. The exact numbers are given in Table B.5 in Appendix B.

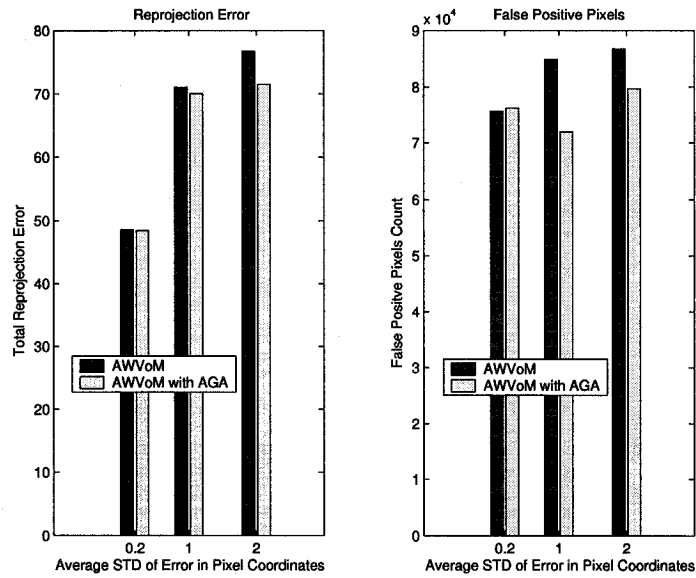
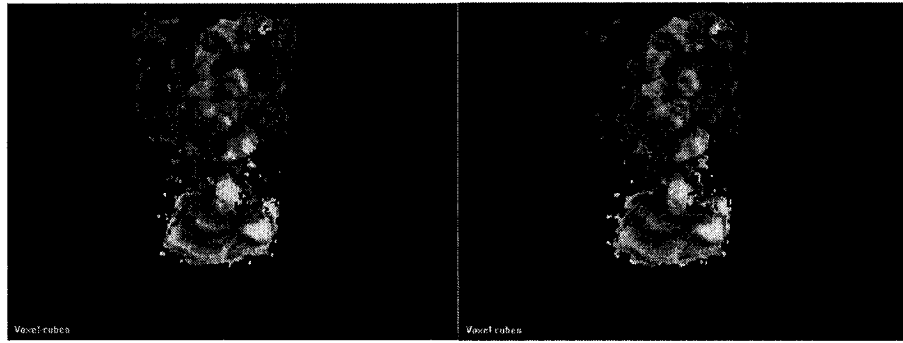
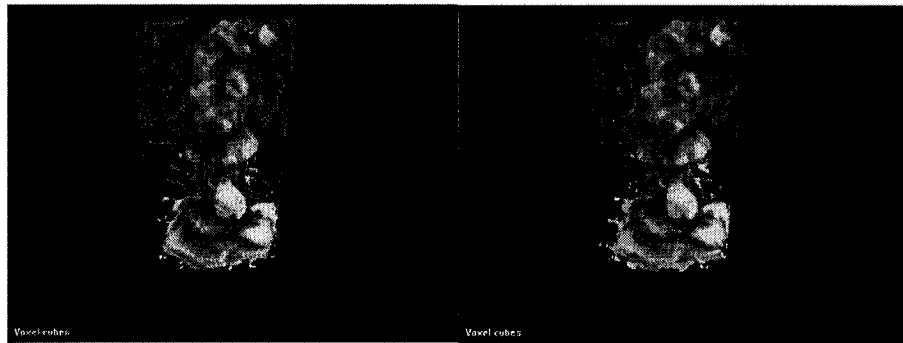


FIGURE 4.18. Quantitative comparison of results shown in Figure 4.16 for the Piper data set for MVPS of 180. The exact numbers are given in Table B.6 in Appendix B.

4.3 RESULTS FOR HANDLING CAMERA CALIBRATION ERRORS



(a) Reconstruction for MVPS = 100



(b) Reconstruction for MVPS = 180

FIGURE 4.19. The effect of using AGA with AWVoM to overcome the calibration errors for the Piper data set. The holes in the reconstruction were NOT been made explicit for these results. The value of average STD of error in the pixel coordinates is 2 pixels. All left hand side reconstruction results are for a simple AWVoM based algorithm. The reconstruction results after adding AGA to the algorithm are shown in the right hand side pictures. The associated quantitative error comparison is shown in Figure 4.20.

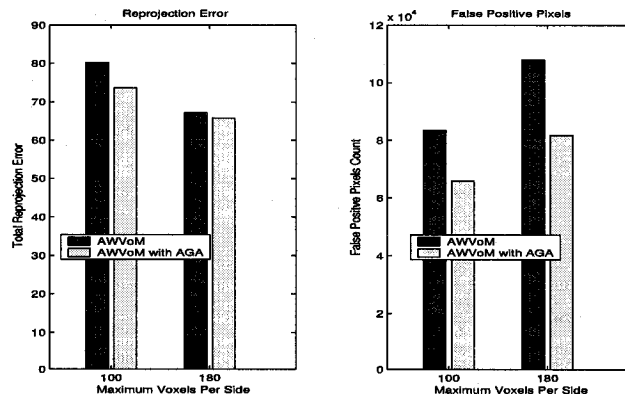


FIGURE 4.20. Quantitative comparison of the results shown in Figure 4.19 for the Piper data set. The exact numbers are given in Table B.7 in Appendix B.

4.3 RESULTS FOR HANDLING CAMERA CALIBRATION ERRORS

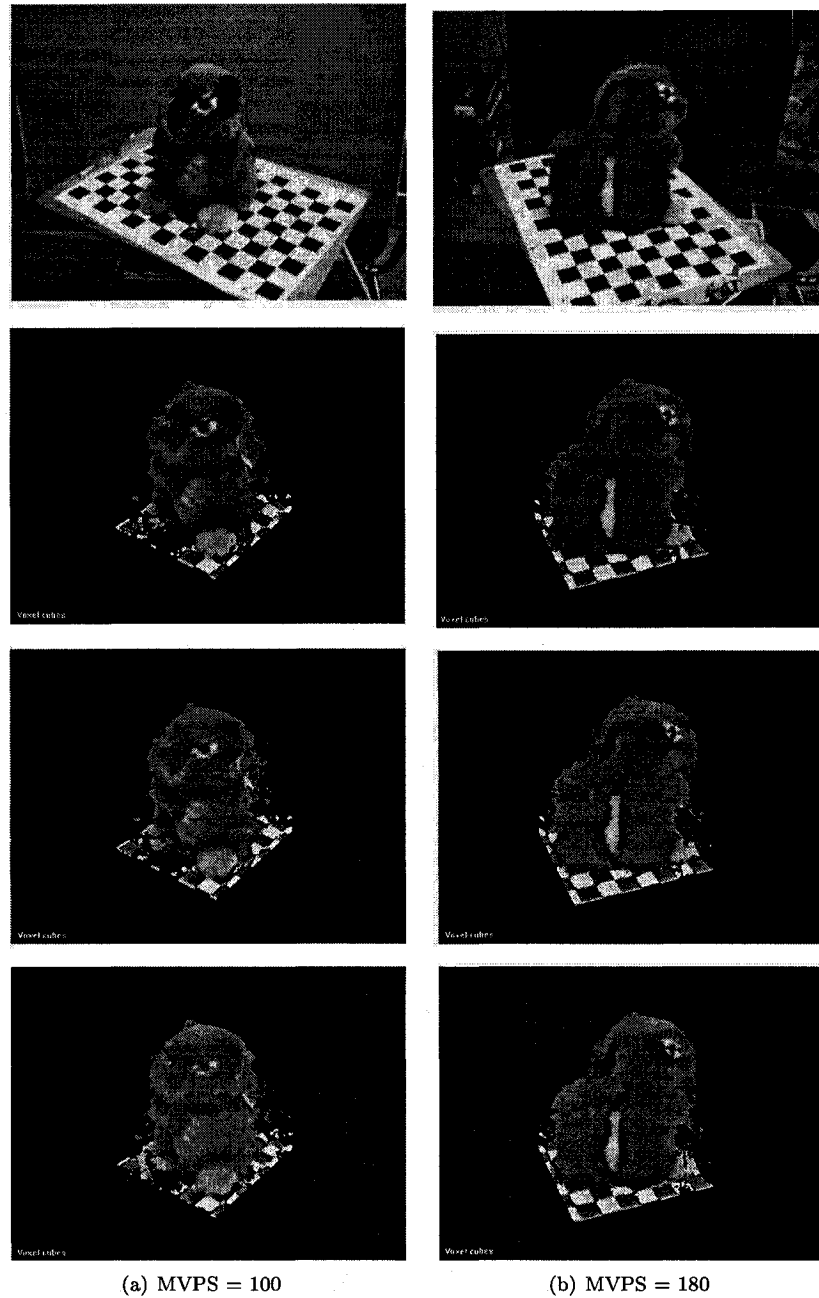


FIGURE 4.21. Reconstruction results to show the effect of using Adaptive Gaussian Averaging for the Chipmunk data set with average standard deviation of error in pixel coordinates = 0.2 pixels. The first picture in each column is the original input image. The second picture is the result of a simple AWVoM test. The third picture is the reconstruction obtained using AGA with AWVoM. The last picture is the reconstruction result of the r-Consistency test. The associated quantitative error comparison is shown in Figure 4.23.

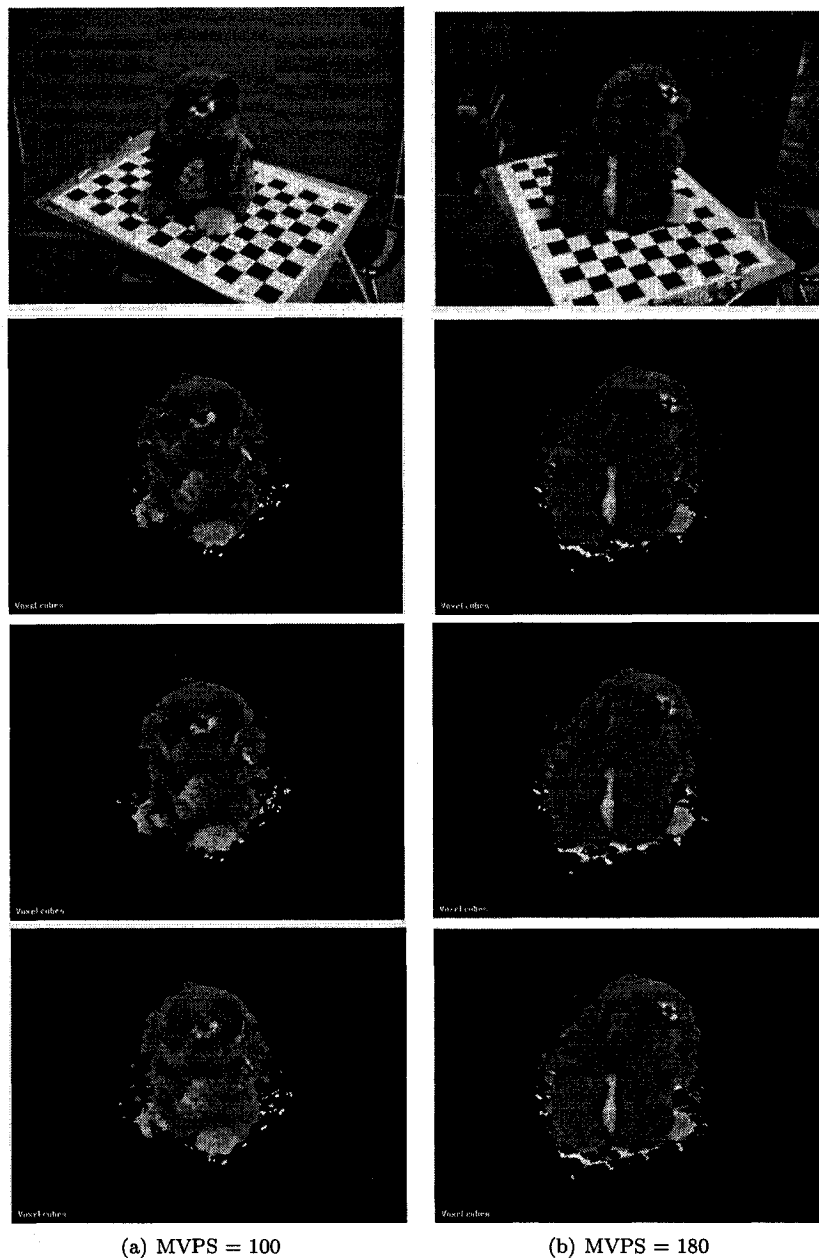


FIGURE 4.22. Reconstruction results to show the effect of using Adaptive Gaussian Averaging for the Chipmunk data set with average standard deviation of error in pixel coordinates = 2 pixels. The first picture in each column is the original input image. The second picture is the result of a simple AWVoM test. The third picture is the reconstruction obtained using AGA with AWVoM. The last picture is the reconstruction result of the r-Consistency test. The associated quantitative error comparison is shown in Figure 4.24.

4.3 RESULTS FOR HANDLING CAMERA CALIBRATION ERRORS

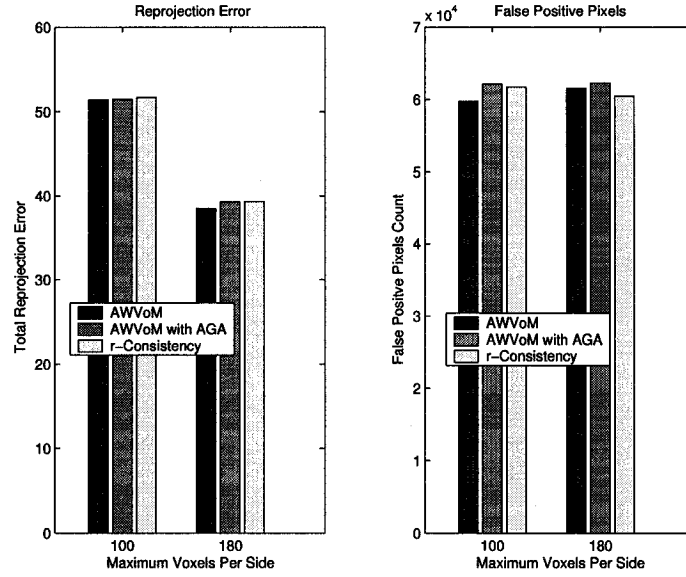


FIGURE 4.23. Quantitative comparison of the results shown in Figure 4.21 for the Chipmunk data set with average standard deviation of error in pixel coordinates = 0.2 pixels. The exact numbers are given in Table B.8 in Appendix B.

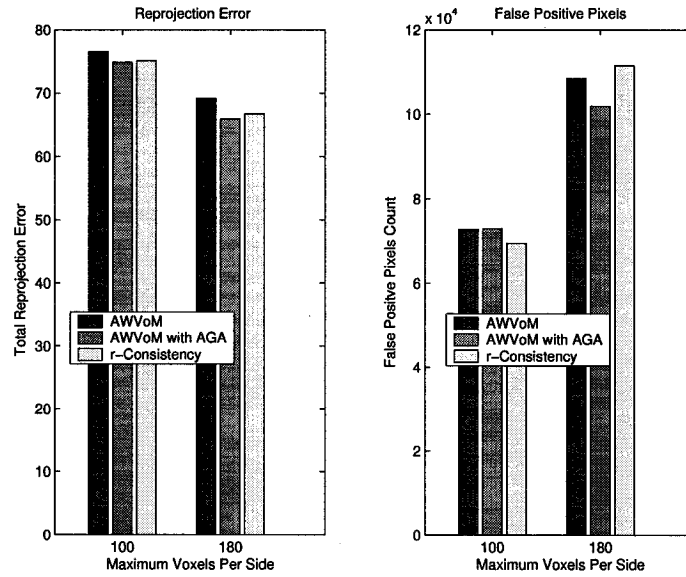


FIGURE 4.24. Quantitative comparison of the results shown in Figure 4.22 for the Chipmunk data set with average standard deviation of error in pixel coordinates = 2 pixels. The exact numbers are given in Table B.9 in Appendix B.

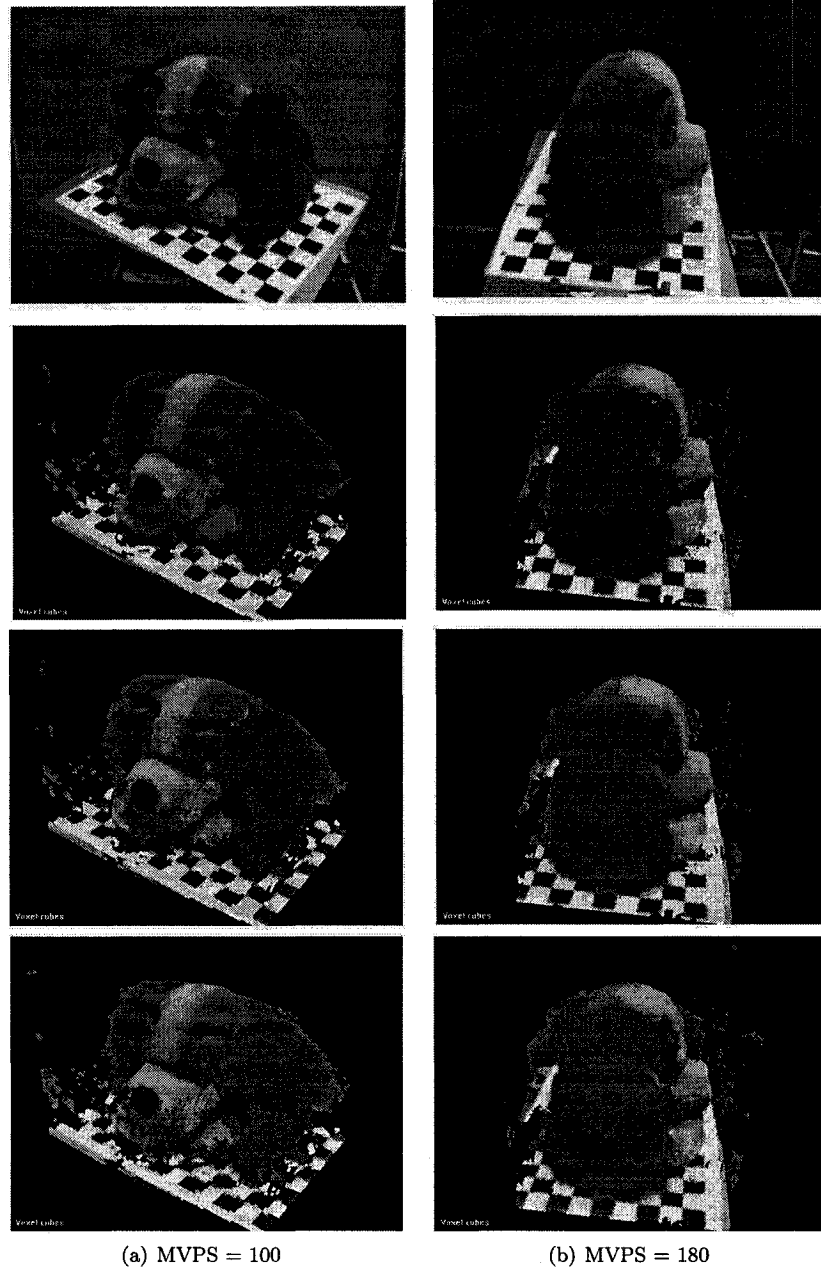


FIGURE 4.25. Reconstruction results to show the effect of using Adaptive Gaussian Averaging for the Puppy data set with average standard deviation of error in pixel coordinates = 0.2 pixels. The first picture in each column is the original input image. The second picture is the result of a simple AWVoM test. The third picture is the reconstruction obtained using AGA with AWVoM. The last picture is the reconstruction result of the r-Consistency test. The associated quantitative error comparison is shown in Figure 4.27.

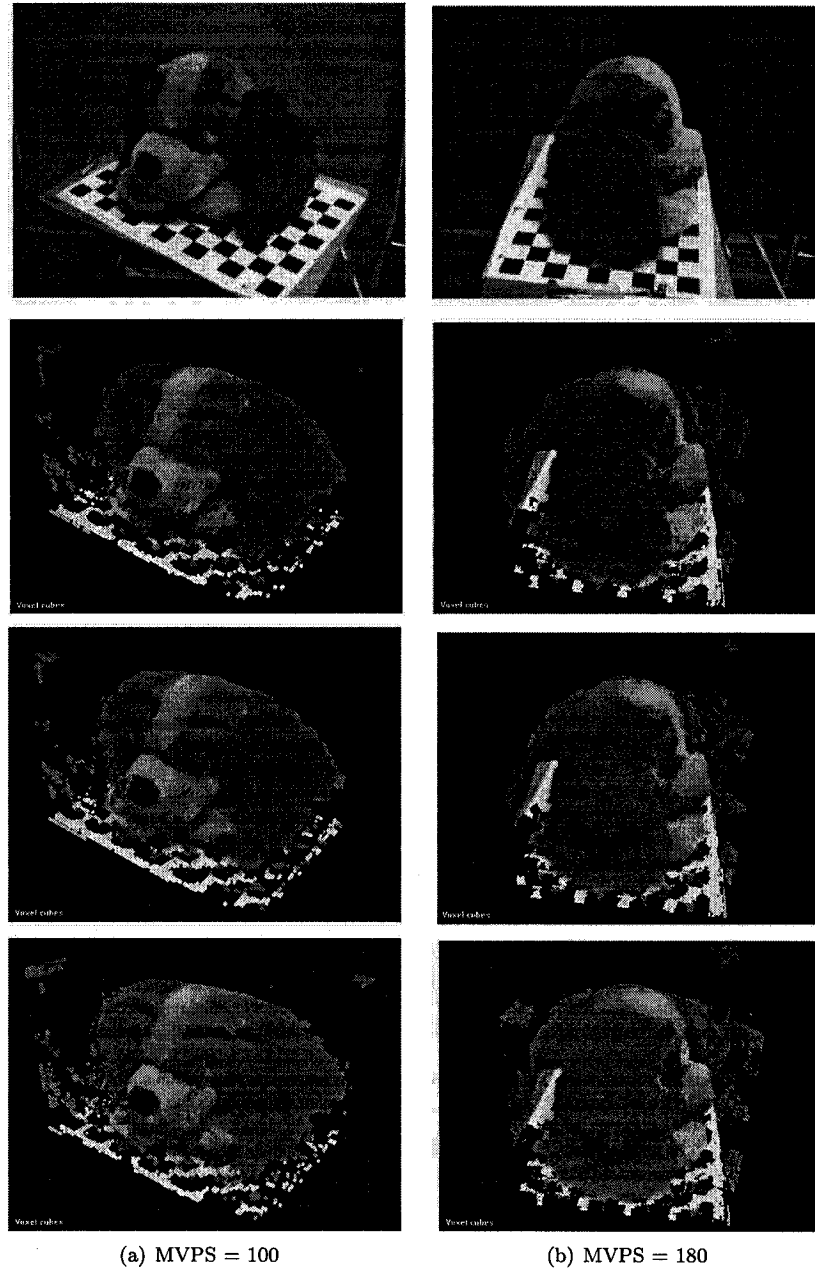


FIGURE 4.26. Reconstruction results to show the effect of using Adaptive Gaussian Averaging for the Puppy data set with average standard deviation of error in pixel coordinates = 2 pixels. The first picture in each column is the original input image. The second picture is the result of a simple AWVoM test. The third picture is the reconstruction obtained using AGA with AWVoM. The last picture is the reconstruction result of the r-Consistency test. The associated quantitative error comparison is shown in Figure 4.28.

4.3 RESULTS FOR HANDLING CAMERA CALIBRATION ERRORS

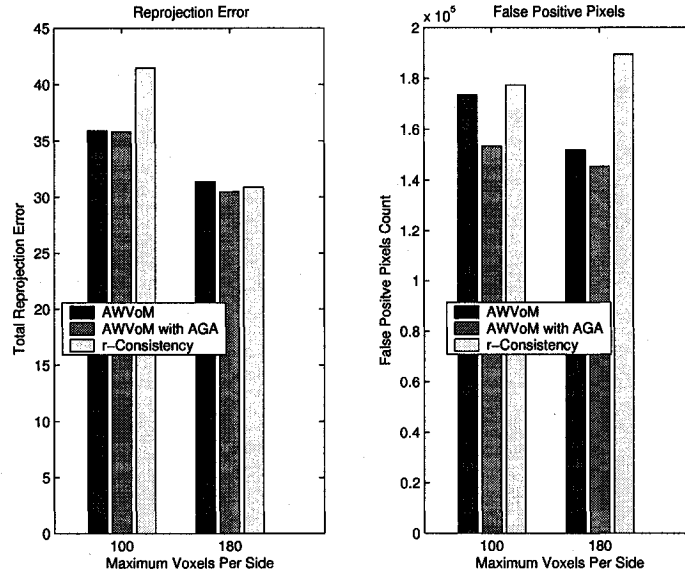


FIGURE 4.27. Quantitative comparison of the results shown in Figure 4.25 for the Puppy data set with average standard deviation of error in pixel coordinates = 0.2 pixels. The exact numbers are given in Table B.10 in Appendix B.

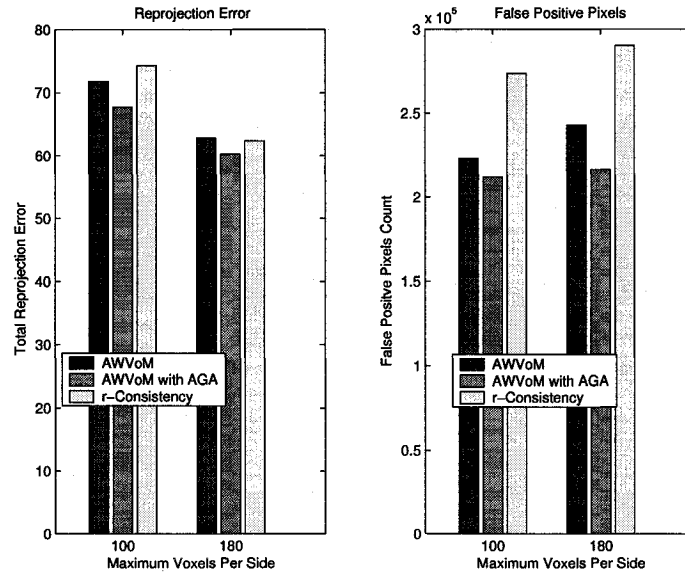


FIGURE 4.28. Quantitative comparison of the results shown in Figure 4.26 for the Puppy data set with average standard deviation of error in pixel coordinates = 2 pixels. The exact numbers are given in Table B.11 in Appendix B.

3.3. Experimental Comparison of Adaptive Gaussian Averaging with r-Consistency.

r-Consistency is a photo-consistency measure proposed by Kutulakos in [16] to handle camera calibration errors as well as the effects of having finite size voxels in voxel-coloring. A detailed review of this paper was presented in Chapter 2. In this section, the results of the r-Consistency test will be compared with the results of our proposed methods, i.e. the Area Weighted Variance of Means (AWVoM) based Photo-Consistency test that uses Adaptive Gaussian Averaging. As no rule has been described in [16] to select the value of the dispersion radius r , its value is controlled by the voxel resolution, i.e. the voxel size. Moreover, instead of using circular voxel projection windows, we are using the actual voxel projections. This simplification reduces the computational complexity of the algorithm.

Figures 4.29 and 4.30 show the reconstruction results of r-Consistency and our proposed methods for three different values of calibration error and for two different voxel resolutions for the Piper data set. The holes in the reconstructed 3D model have been made explicit by not coloring the single visibility voxels. Quantitative comparison of these results is presented in Figures 4.31 and 4.32. Figures 4.21 to 4.28 compare the results of r-Consistency with AGA for the Chipmunk and Puppy data sets for two different calibration error values and for two different voxel resolutions. Note that single visibility voxels in the reconstructions for these two data sets have not been left uncolored. Therefore, the holes are not explicit in these reconstructions.

For lower voxel resolution with MVPS of 100, the results of our proposed methods are better than those of r-Consistency for all levels of calibration error for both the Piper and Puppy data sets. The difference in the reconstruction quality increases with the calibration error, thereby confirming the capability of our proposed methods to adapt to higher calibration errors. As a result, for lower voxel resolution and for higher calibration errors, our proposed methods perform consistently better than r-Consistency for all the data sets. At lower voxel resolution, the only case when the r-Consistency performs better than our proposed methods is for the Chipmunk data set with very small calibration error.

At a higher voxel resolution with MVPS of 180, r-Consistency performs better than our proposed methods for lower calibration errors. But for a higher calibration error, the performance of both methods is comparable. This again confirms that the effectiveness of our proposed methods increase with the level of errors present in the system.

The fact that the r-Consistency test performs better than our proposed methods in certain cases can be explained. As was described earlier in this chapter in Section 3.1, both the camera calibration errors and the partial emptiness of surface voxels have less effect on the quality of reconstruction for higher voxel resolutions, i.e. smaller voxel sizes. Furthermore, it was discussed

in Section 3.2 that the negative effect of Gaussian averaging on the Variance of Means based photo-consistency test is also higher for smaller voxel sizes. Therefore, the improvement offered by our proposed methods is reduced at higher voxel resolutions with low calibration error.

One interesting fact is that for the Chipmunk data set, both AGA and r-Consistency result in a further degradation of the reconstruction quality rather than any improvement when the average standard deviation of error in pixel coordinates is around 0.2 pixels. In this case, the negative effects of these methods overpower the improvement offered by them. Another interesting observation is that for the Puppy data set, simple AWWoM performs better than the r-Consistency. In this case, the r-Consistency test is unable to handle the effects of larger voxel size.

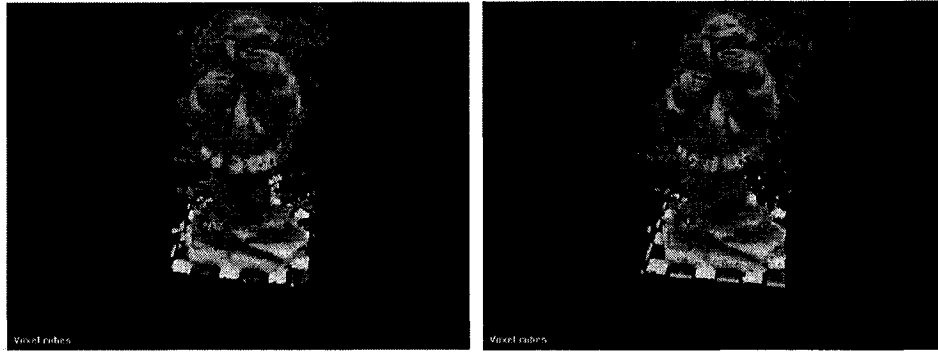
These results clearly show that our proposed methods work better than r-Consistency when the effects of error sources are higher, i.e. low voxel resolutions and high calibration errors.

3.4. Significance of the Error in the Reconstruction Results.

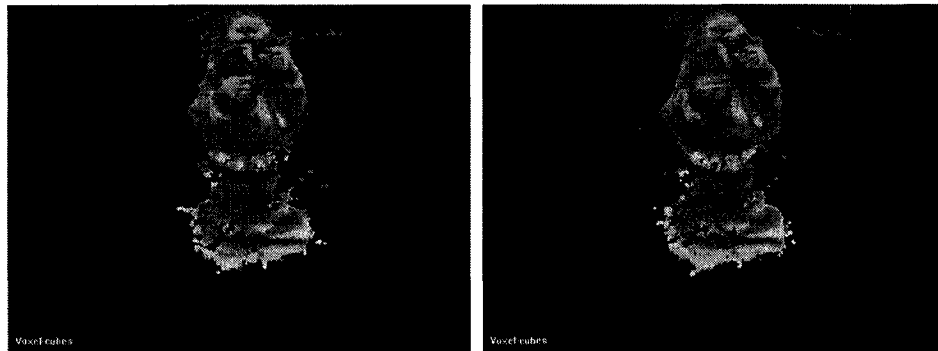
We used the reprojection error in the pixel colors and the false positive pixel counts in the reprojected images as an error measure in this thesis. However, reprojection error gives an estimate of the quality of reconstructed Photo Hull of the 3D object. As a Photo Hull is a superset of all possible photo-consistent reconstructions of a 3D object, reprojection error is limited in determining how closely the 3D reconstruction resembles the original 3D object. Moreover, the reprojection error may be effected by the shape and texture of the 3D object. It was seen that for the Piper data set, the reprojection error was the highest as its surface was rich in depth discontinuities. Similarly, the reprojection error for a 3D object with little surface texture will be lower in general. The reprojection error for the Puppy data set was the lowest due to numerous same color surface regions. Therefore, two reconstruction results with similar reprojection error may actually be different in their quality of reconstruction. Hence, it is important to take these factors into account while evaluating the reconstruction quality of a voxel-coloring algorithm based on the reprojection error only.

Another observation is that small surface artifacts in the reconstructed model are not visible in all the input views. An example is the small holes carved in the 3D model when the voxel behind the carved voxel remains visible in at least one image. Due to the texture mapping of the voxel behind the carved voxel, these holes are sometimes not visible on the reconstructed surface. Therefore, these small surface artifacts will have little effect on the reprojection error. The reprojection error is highly affected by over-carving that introduces black spots on the reconstruction model. As a result, the difference in the reprojection error of two reconstructions usually shows the difference of their over-carving. It is clear from the reconstruction results presented in this chapter that a total average reprojection error difference of 1-2 units can be safely neglected.

4.3 RESULTS FOR HANDLING CAMERA CALIBRATION ERRORS



(a) Average pixel coordinates error STD = 0.2 pixels



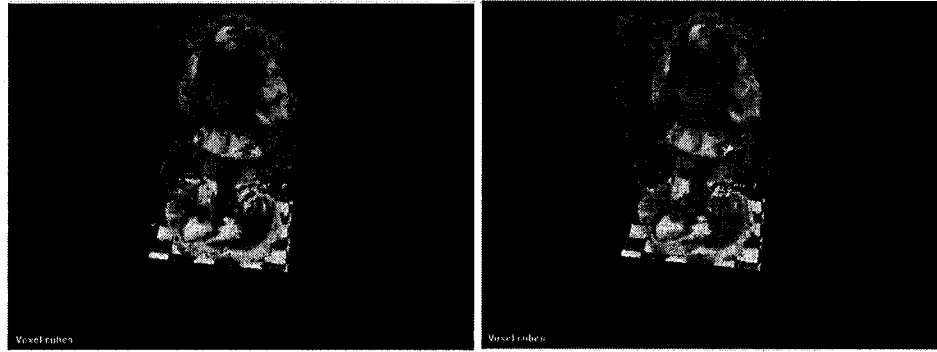
(b) Average pixel coordinates error STD = 1 pixels



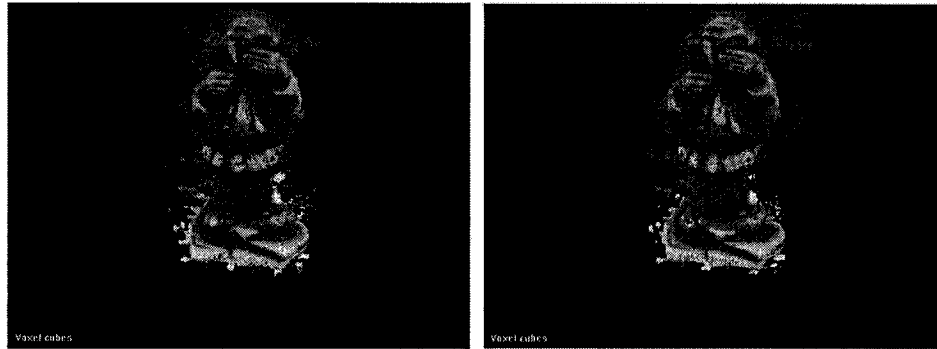
(c) Average pixel coordinates error STD = 2 pixels

FIGURE 4.29. Piper data set, MVPS = 100. Reconstruction results comparison of r-Consistency and AWVoM with AGA. Holes are being explicitly shown in these reconstructions. The left hand side reconstruction pictures are for r-Consistency and the right hand side reconstruction pictures are for AWVoM with AGA. The associated quantitative comparison of the results is presented in Figure 4.31.

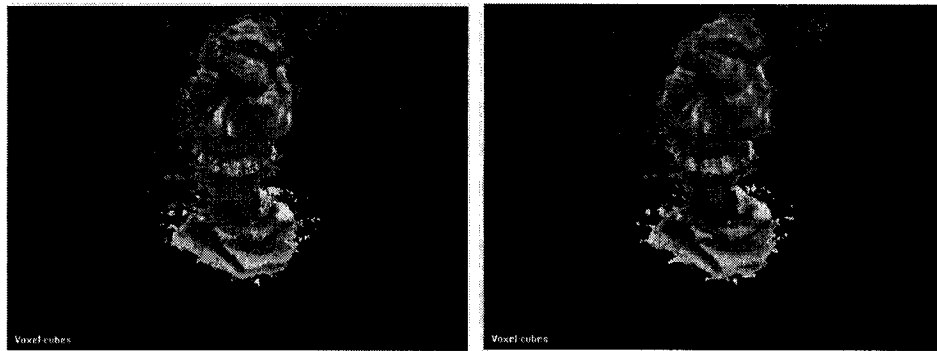
4.3 RESULTS FOR HANDLING CAMERA CALIBRATION ERRORS



(a) Average pixel coordinates error STD = 0.2 pixels



(b) Average pixel coordinates error STD = 1 pixels



(c) Average pixel coordinates error STD = 2 pixels

FIGURE 4.30. Piper data set, MVPS = 180. Reconstruction results comparison of r-Consistency and AWVoM with AGA. Holes are being explicitly shown in these reconstructions. The left hand side reconstruction pictures are for r-Consistency and the right hand side reconstruction pictures are for AWVoM with AGA. The associated quantitative comparison of the results is presented in Figure 4.32.

4.3 RESULTS FOR HANDLING CAMERA CALIBRATION ERRORS

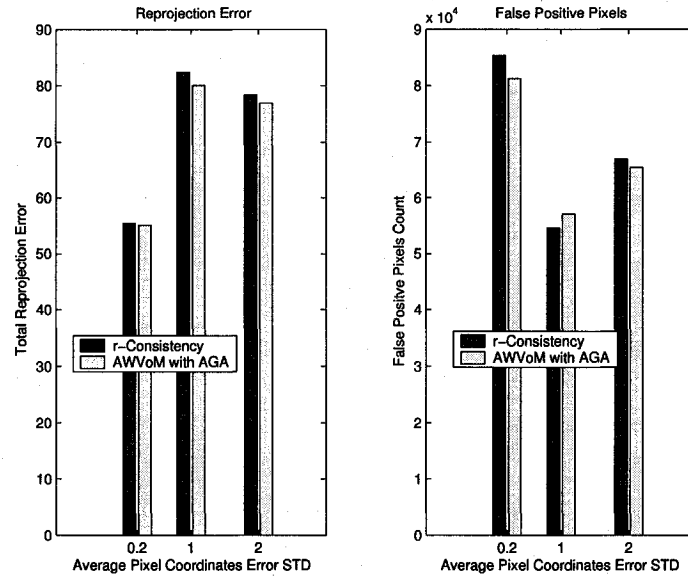


FIGURE 4.31. Piper data set, MVPS = 100. Quantitative comparison of the results of r-Consistency with AWVoM using AGA shown in Figure 4.29. The exact numbers are given in Table B.12 in Appendix B.

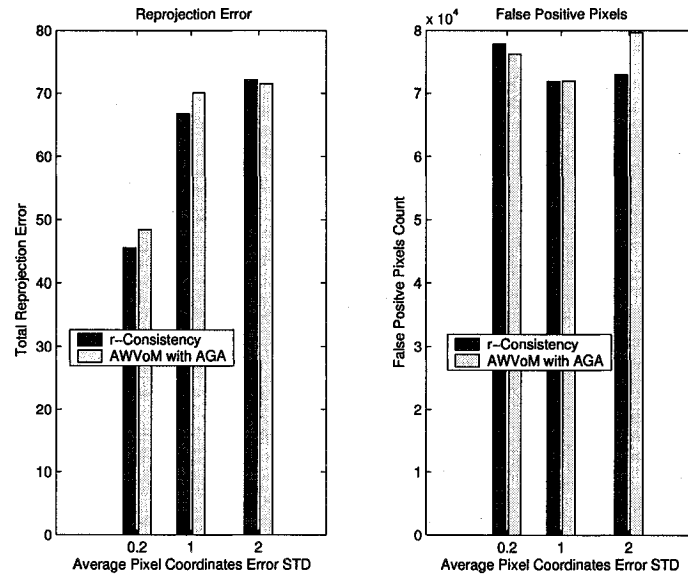


FIGURE 4.32. Piper data set, MVPS = 180. Quantitative comparison of the results of r-Consistency with AWVoM using AGA shown shown in Figure 4.30. The exact numbers are given in Table B.13 in Appendix B.

CHAPTER 5

Conclusion

The work in this thesis investigated the usefulness of voxel-coloring algorithms for 3D scene reconstruction from multiple views. We found that the state-of-the-art in this area can be improved by addressing the problems caused by some error sources in the process that effect the quality of 3D reconstruction. Our focus has been to investigate the effects of two of these sources of error; partial emptiness of surface voxels due to finite size of a voxel and the calibration errors in estimated camera parameters. Based on our analysis of these error sources, we proposed two methods to reduce the effect of these error sources on voxel-coloring. We also experimentally verified the effectiveness of our proposed techniques.

As has been shown with the experiments presented in Chapter 4, calibration errors do have an impact on the quality of reconstruction, and an increase in the calibration error results in noticeable degradation of the reconstruction quality. In order to overcome the effect of this error, it is important to understand its properties. Similar to many approaches in the computer vision literature, we modelled the errors in estimated camera parameters as zero mean Gaussian noise, and the resulting errors in the projected pixel coordinates as a zero mean bivariate Gaussian distribution. Using the assumption that the error in the projected pixel coordinates is linearly dependent on the error in the camera parameters, we were able to calculate the distribution parameters of the errors in projected pixel coordinates, i.e. the covariance matrix. Once we have the distribution of the effect of calibration error in the pixel coordinates, the mean of the projected pixel's color distribution is described by a Gaussian average of a window centered at that pixel. The size of this averaging window depends on the amount of error in pixel coordinates. Therefore, we called our proposed method Adaptive Gaussian Averaging (AGA).

Adaptive Gaussian Averaging can be used with most of existing voxel-coloring algorithms and photo-consistency tests. In order to gauge the effectiveness of our proposed method, we tested it with a Variance of Means photo-consistency test using GVC-IB [7] as the voxel-coloring algorithm.

We found that incorporating our proposed method resulted in a significant improvement in the reconstruction results. The difference in the results was more prominent for lower voxel resolutions and for higher calibration errors. The results of our proposed method were also compared with those of r-Consistency, a method proposed by Kutulakos in [16] to handle both the effect of finite voxel size and calibration errors. Our proposed solution had better results for low voxel resolutions and for higher calibration errors. One advantage of our proposed method is that it makes precise use of the calibration error distribution information and can be explained well analytically.

During our experiments, it was interesting to note that calibration errors have less effect for higher voxel resolutions, at least for the Variance of Means photo-consistency test. We noticed that if the calibration error is sufficiently small, i.e. a standard deviation of less than half a pixel, and the voxel resolution is fairly high, the degradation in the quality of reconstruction is not noticeable and can be safely ignored especially if the speed of the algorithm is an issue.

One of the effects of a finite sized voxel is that a surface voxel can be partially empty. In this thesis, we have analyzed the effects of this error on the quality of reconstruction of a voxel-coloring algorithm. We could not find this analysis in the literature. An interesting result is that the amount of outlier pixels present in a voxel projection in an image has a direct relation to the number of pixels in the projection. Therefore, using the areas of projection of a voxel as weights in calculating its color statistics for a photo-consistency check can be helpful in reducing the effects of this error. This was indeed the case and was verified by our experimental results. As the effect of this error is less for higher voxel resolutions, the improvement achieved in the reconstruction quality was less as well. The results confirm that our proposed solution is able to reduce the effects of this error on voxel-coloring.

Voxel-coloring has emerged as a method for the reconstruction of 3D objects from multiple views. The strength of this method is the simplicity of the concept, which is quite helpful in case of multiple input views. The technique has an inherent advantage in occlusion handling and in easy control of the reconstructed 3D model resolution. However, due to the assumption that the shape of a 3D voxel is a cube, the shape and size of a voxel projection in different images can be different. As a result, we cannot compare two voxel projections on a per pixel basis. This problem is more pronounced at low voxel resolutions. Therefore, as long as two voxel projections have similar color distributions, even if their pixels have different spatial distributions, the voxel will be considered photo-consistent. This adds further uncertainty in the photo-consistency check and hence results in a “fatter” reconstruction. Another limitation is the 3D space quantization. Due to these limitations, the Photo Hull, which is the result of a voxel-coloring algorithm, is only a good approximation of the actual 3D shape. Therefore, one possible use of voxel-coloring could

be in a hybrid algorithm that uses the low-voxel resolution results of a voxel-coloring algorithm to reduce the disparity space for pixel correspondence in traditional stereo. Voxel-coloring is especially helpful when the 3D scene to be reconstructed is contained by a closed bounding box, i.e. the initial volume guess is very close to the original 3D surface. In the absence of such a good guess, there will be extra iterations of the algorithm that will take more time, and will also result in higher error propagation due to the sequential nature of the algorithm. Nonetheless, a Photo Hull is the minimally constrained reconstruction and thus it is helpful in understanding the upper bound on the quality of reconstruction that can be achieved by any algorithm for given imaging conditions.

We have presented two simple methods to handle two error sources in voxel-coloring. The effect of using Adaptive Gaussian Averaging on the reconstruction is to cause the blurring of the 3D surface. The r-Consistency has a similar effect on the reconstruction quality as it causes r-Shuffle of the reprojected images. One interesting extension of our work could be to use extra shape constraints to reduce the uncertainty in the reconstructed 3D surface. One such shape constraint can be obtained by controlling the imaging process, i.e. active voxel-coloring. If we know the uncertainty map of the 3D shape obtained from a voxel-coloring algorithm, the imaging process can be controlled to improve the shape in the high uncertainty regions of the reconstructed model.

We have seen that r-Consistency proposed by Kutulakos in [16] has the ability to overcome the effect of outlier pixels in voxel projections due to different error sources. We found that r-Consistency performed better than our proposed methods for higher voxel resolutions and for low calibration errors. However, it does not make any use of the available information about the calibration error distribution. Therefore, one extension of our work could be to use the calibration error distribution information to adaptively select the value of r in r-Consistency.

Another interesting direction for future research is to use the spatial characteristics of the distribution of pixels in a voxel projection during the photo-consistency test. This would add an extra constraint and hence would reduce the fattening of the Photo Hull.

The quality of a voxel-coloring algorithm relies highly on the photo-consistency test. Most of the photo-consistency tests found in the literature make use of one or more threshold parameters that are selected empirically and depend on surface characteristics of a particular data set. Although a few photo-consistency tests have been proposed that do not require a user defined threshold parameter [29, 4], a better and reliable photo-consistency test that does not require any threshold parameter and can be used for different types of scenes is still an open question for research.

To conclude, voxel-coloring is a simple method for 3D reconstruction from multiple views. It offers the solution of some of the problems faced by the traditional stereo. There are still many areas

in voxel-coloring that require further research to improve its practicality. The most recommended use of this method will be as a first stage of a traditional stereo algorithm, especially when dealing with a large number of input views.

APPENDIX A

Estimation of Pixel Coordinates Error Variance

The method of calculating the covariance matrix of the error in camera parameters and 2D projected pixel coordinates points is presented in this appendix. This method is described in detail in [11]. The calibration error in the camera parameters depends on the calibration method used and also on the feature points used in the process. Therefore, when a 3D point is projected in an image using the estimated camera parameters, we will have different values of the error for different parts of the image and also for different values of the 3D point depth. If we assume that the error in the 2D feature points used in the calibration is distributed as zero mean Gaussian, the error in the estimated camera parameters can be approximated to have a Gaussian distribution as well. The effect of the the feature points error has an almost linear effect on the estimated camera parameter if the error in the feature points is sufficiently small, i.e. a few pixels. Using this linearity assumption, the covariance matrix of the camera parameters error for a particular camera k can be estimated up to a first order of approximation as follows:

$$(0.1) \quad C_P = (J^T C_{X_{2D}}^{-1} J)^+ = \left(\sum_i J_i^T C_{X_{2Di}}^{-1} J_i \right)^+,$$

where the Jacobian, J_i , is the derivative of the 2D projected point coordinates with respect to the estimated camera parameters, i.e.

$$(0.2) \quad J_i = \frac{\partial X_{2Di}}{\partial P},$$

where P is the camera parameter matrix, $()^+$ denotes the pseudo-inverse operation and C_P is the covariance matrix of the estimated camera parameters error. $C_{X_{2D}}$ is the covariance matrix of the error in all the calibration feature points and $C_{X_{2Di}}$ is the covariance matrix of the i th 2D feature point, X_{2Di} . This method is valid when the error in the feature points is Gaussian. Another

assumption is that the error is only in the 2D feature point coordinates and not in the locations of the corresponding 3D points.

For maximum likelihood estimation, when the camera projection matrix parameters are estimated directly during camera calibration, J_i can be calculated as follows:

$$J_i = \frac{\partial X_{2Di}}{\partial P} = \frac{1}{w_i} \begin{bmatrix} X_{3Di}^T & \mathbf{0}^T & -\tilde{x}_i X_{3Di}^T \\ \mathbf{0}^T & X_{3Di}^T & -\tilde{y}_i X_{3Di}^T \end{bmatrix},$$

and

$$X_{3Di} = [X_i \ Y_i \ Z_i \ 1]^T = \text{3D point},$$

$$X_{2Di} = [\tilde{x}_i \ \tilde{y}_i]^T = \left[\frac{x_i}{w_i} \ \frac{y_i}{w_i} \right]^T = \text{2D point},$$

$$[x_i \ y_i \ w_i]^T = P \cdot [X_i \ Y_i \ Z_i \ 1]^T.$$

Once the covariance matrix of the camera parameter error is available, the uncertainty in the projected pixel coordinates can be calculated as follows:

$$(0.3) \quad C_{x,y} = J_P C_P J_P^T,$$

where $C_{x,y}$ is the covariance matrix of the bivariate Gaussian distribution of the error in the coordinates of the projected pixel at x, y . J_P is given by equation (0.2) and C_P is calculated as in equation (0.1). Again the linearity assumption was used in the error transfer; that is the error in the pixel coordinates changes linearly with a change in the error of the camera parameters. Due to the linearity assumption, it is important that the feature points used in the calibration should be well distributed in both the image and the 3D object space to have a reliable transfer of error to the projected pixel coordinates.

The Camera Calibration Toolbox [3] that we have used for the calibration of our data sets estimates the following camera parameters along with the associated uncertainties,

$$\text{Two focal lengths} = f_x \text{ and } f_y,$$

$$\text{Principle point coordinates} = C_x \text{ and } C_y,$$

$$\text{Four radial distortion coefficients} = k_1 \text{ to } k_4,$$

$$\text{Three rotation angles} = R_1, R_2 \text{ and } R_3,$$

$$\text{The translation vector} = [T_1 \ T_2 \ T_3]^T,$$

where

$$f_x = f/s_x,$$

$$f_y = f/s_y,$$

s_x, s_y = sizes of pixel along two axes.

The individual variances of these estimated parameters is combined in a 14×14 diagonal matrix which forms the approximate camera parameters covariance matrix C_P . The Jacobian of the projected 2D point location w.r.t. these camera parameters is calculated using a function provide in the toolbox that is called *ProjectPoints*. Once, C_P and J_P are available, the uncertainty in the position of the 2D projected point can be easily calculated using equation (0.3).

APPENDIX B

Tables for Quantitative Error Analysis

This appendix contains the tables for quantitative results of the experiments presented in Chapter 4.

1. Quantitative Results for the Use of Area Weighting

TABLE B.1. Quantitative error analysis of Area Weighting for the Piper data set

| Photo-Consistency Test | Carving Threshold | Voxels Carved | Surface Voxels Count in 3D Model | Reprojection Error (R, G, B) | | False Positive pixel Count |
|------------------------|-------------------|---------------|----------------------------------|------------------------------|----------|----------------------------|
| MVPS = 100 | | | | | | |
| Simple VoM | 69 ² | 253931 | 19433 | 47.6804, 41.0220 | 44.8704, | 296001 |
| AWVoM | 55 ² | 253487 | 18298 | 31.5699, 27.9181 | 29.6677, | 87639 |
| MVPS = 180 | | | | | | |
| Simple VoM | 63 ² | 1410929 | 73839 | 24.7091, 22.0366 | 22.6851, | 151138 |
| AWVoM | 55 ² | 1338811 | 71088 | 24.4508, 21.5885 | 22.4552, | 110636 |
| Simple Simple VoM | 60 ² | 1439730 | 72402 | 26.5015, 23.6044 | 24.7048, | 87655 |

B.1 QUANTITATIVE RESULTS FOR THE USE OF AREA WEIGHTING

TABLE B.2. Quantitative error analysis of Area Weighting for the Chipmunk data set

| Photo-Consistency Test | Carving Threshold | Voxels Carved | Surface Voxels Count in 3D Model | Reprojection Error (R, G, B) | | False Positive pixel Count |
|------------------------|-------------------|---------------|----------------------------------|------------------------------|----------|----------------------------|
| MVPS = 100 | | | | | | |
| Simple VoM | 61 ² | 444689 | 36270 | 31.2679, 27.6371 | 29.7743, | 96198 |
| AWVoM | 50 ² | 442377 | 34075 | 29.9740, 26.4670 | 28.5107, | 70019 |
| MVPS = 180 | | | | | | |
| Simple VoM | 58 ² | 2382937 | 136557 | 23.4935, 20.9308 | 22.2775, | 76345 |
| AWVoM | 50 ² | 2340567 | 140570 | 21.8728, 19.4277 | 20.6201, | 75249 |

TABLE B.3. Quantitative error analysis of Area Weighting for the Puppy data set

| Photo-Consistency Test | Carving Threshold | Voxels Carved | Surface Voxels Count in 3D Model | Reprojection Error (R, G, B) | | False Positive pixel Count |
|------------------------|-------------------|---------------|----------------------------------|------------------------------|----------|----------------------------|
| MVPS = 100 | | | | | | |
| Simple VoM | 60 ² | 315449 | 28635 | 25.0255, 22.5928 | 22.9839, | 196302 |
| AWVoM | 50 ² | 313373 | 28146 | 23.3444, 21.2477 | 21.6445, | 132024 |
| MVPS = 180 | | | | | | |
| Simple VoM | 58 ² | 1766602 | 110033 | 19.8243, 17.6197 | 18.0449, | 158121 |
| AWVoM | 50 ² | 1749374 | 110774 | 19.6206, 17.5206 | 17.9779, | 138293 |

2. Quantitative Error Analysis for Handling Calibration Errors

TABLE B.4. The effect of increasing calibration error on reconstruction quality of AWVoM based voxel-coloring algorithm

| Average Pixel Coordinates Error Standard Deviation | Reprojection Error (R, G, B) | False Positive Pixel Count |
|--|------------------------------|----------------------------|
| MVPS = 100 | | |
| 0.2 pixels | 31.5699, 29.6677, 27.9181 | 87639 |
| 1.0 pixels | 41.9523, 40.1431, 38.3167 | 116831 |
| 2.0 pixels | 48.3962, 46.6454, 43.7232 | 83359 |
| MVPS = 180 | | |
| 0.2 pixels | 24.4508, 22.4552, 21.5885 | 110636 |
| 1.0 pixels | 36.1520, 34.3548, 33.4362 | 146883 |
| 2.0 pixels | 40.2155, 38.6685, 37.2561 | 107899 |

TABLE B.5. Quantitative assessment of using Adaptive Gaussian Averaging to overcome calibration errors for MVPS = 100 for the Piper data set

| Algorithm | Reprojection Error (R, G, B) | False Positive Pixel Count |
|--|------------------------------|----------------------------|
| Average pixel coordinates error STD = 0.2 pixels | | |
| simple AWVoM | 34.0964, 32.0134, 29.5854 | 87112 |
| AWVoM with AGA | 33.8916, 31.8386, 29.4215 | 81187 |
| Average pixel coordinates error STD = 1 pixels | | |
| simple AWVoM | 49.7029, 47.6892, 44.0049 | 64121 |
| AWVoM with AGA | 48.6404, 46.6783, 43.1968 | 57127 |
| Average pixel coordinates error STD = 2 pixels | | |
| simple AWVoM | 50.9198, 48.9524, 45.2727 | 82953 |
| AWVoM with AGA | 46.4655, 44.7614, 41.9205 | 65360 |

B.2 QUANTITATIVE ERROR ANALYSIS FOR HANDLING CALIBRATION ERRORS

TABLE B.6. Quantitative assessment of using Adaptive Gaussian Averaging to overcome calibration errors for MVPS = 180 for the Piper data set

| Algorithm | Reprojection Error (R, G, B) | False Positive Pixel Count |
|--|------------------------------|----------------------------|
| Average pixel coordinates error STD = 0.2 pixels | | |
| simple AWVoM | 30.3909, 28.0352, 25.4957, | 75648 |
| AWVoM with AGA | 30.2686, 27.9217, 25.4384 | 76211 |
| Average pixel coordinates error STD = 1 pixels | | |
| simple AWVoM | 43.2595, 41.2114, 38.4316 | 84832 |
| AWVoM with AGA | 42.6402, 40.6570, 37.8912 | 71952 |
| Average pixel coordinates error STD = 2 pixels | | |
| simple AWVoM | 46.6483, 44.6296, 41.3907 | 86692 |
| AWVoM with AGA | 43.3839, 41.5361, 38.8225 | 79654 |

TABLE B.7. Result of using Adaptive Gaussian Averaging without explicitly showing holes for the Piper data set. Average pixel coordinates error STD = 2pixels

| Algorithm | Reprojection Error (R, G, B) | False Positive Pixel Count |
|----------------|------------------------------|----------------------------|
| MVPS = 100 | | |
| simple AWVoM | 48.3962, 46.6454, 43.7232 | 83359 |
| AWVoM with AGA | 44.1649, 42.6457, 40.4393 | 65757 |
| MVPS = 180 | | |
| AWVoM | 40.2155 , 38.6685, 37.2561 | 107899 |
| AWVoM with AGA | 39.4808, 37.9474, 36.3390 | 81701 |

TABLE B.8. Quantitative assessment of using Adaptive Gaussian Averaging to overcome calibration errors for the Chipmunk data set. Average pixel coordinates error STD = 0.2 pixels

| Algorithm | Reprojection Error (R, G, B) | False Positive Pixel Count |
|-------------------------------|------------------------------|----------------------------|
| MVPS = 100 | | |
| simple AWVoM | 31.4, 29.9246, 27.5499 | 59730 |
| AWVoM with Gaussian Averaging | 31.4254, 29.9639, 27.6404 | 62103 |
| r-Consistency | 31.3337, 30.03, 28.0514 | 61711 |
| MVPS = 180 | | |
| simple AWVoM | 23.4746, 22.2522, 20.8756 | 61538 |
| AWVoM with Gaussian Averaging | 23.9667, 22.7232, 21.306 | 62267 |
| r-Consistency | 23.9383, 22.7474, 21.3829 | 60439 |

B.2 QUANTITATIVE ERROR ANALYSIS FOR HANDLING CALIBRATION ERRORS

TABLE B.9. Quantitative assessment of using Adaptive Gaussian Averaging to overcome calibration errors for the Chipmunk data set. Average pixel coordinates error STD = 2 pixels

| Algorithm | Reprojection Error (R, G, B) | False Positive Pixel Count |
|----------------|------------------------------|----------------------------|
| MVPS = 100 | | |
| simple AWVoM | 45.5936, 44.6193, 42.2434 | 72790 |
| AWVoM with AGA | 44.6233, 43.6048, 41.2967 | 72885 |
| r-Consistency | 44.7688, 43.8193, 41.3909 | 69458 |
| MVPS = 180 | | |
| simple AWVoM | 41.0779, 40.1616, 38.4742 | 108385 |
| AWVoM with AGA | 39.2233, 38.2496, 36.6523 | 101807 |
| r-Consistency | 39.6562, 38.7511, 37.0101 | 111474 |

TABLE B.10. Quantitative assessment of using Adaptive Gaussian Averaging to overcome calibration errors for the Puppy data set. Average pixel coordinates error STD = 0.2 pixels

| Algorithm | Reprojection Error (R, G, B) | False Positive Pixel Count |
|----------------|------------------------------|----------------------------|
| MVPS = 100 | | |
| simple AWVoM | 22.0102, 20.205, 19.9133 | 173621 |
| AWVoM with AGA | 21.9387, 20.1935, 19.8708 | 153457 |
| r-Consistency | 25.3233, 23.4474, 22.9512 | 177418 |
| MVPS = 180 | | |
| simple AWVoM | 19.3913, 17.6841, 17.2716 | 151976 |
| AWVoM with AGA | 18.8619, 17.1637, 16.782 | 145427 |
| r-Consistency | 19.2374, 17.344, 16.8967 | 189408 |

TABLE B.11. Quantitative assessment of using Adaptive Gaussian Averaging to overcome calibration errors for the Puppy data set. Average pixel coordinates error STD = 2 pixels

| Algorithm | Reprojection Error (R, G, B) | False Positive Pixel Count |
|----------------|------------------------------|----------------------------|
| MVPS = 100 | | |
| simple AWVoM | 43.1469, 41.1159, 40.0445 | 222943 |
| AWVoM with AGA | 40.6557, 38.6668, 37.7376 | 211995 |
| r-Consistency | 45.0324, 42.4262, 41.1396 | 273767 |
| MVPS = 180 | | |
| simple AWVoM | 37.843, 35.8912, 34.9728 | 243053 |
| AWVoM with AGA | 36.3683, 34.4183, 33.5131 | 216616 |
| r-Consistency | 38.0047, 35.483, 34.3747 | 290328 |

B.2 QUANTITATIVE ERROR ANALYSIS FOR HANDLING CALIBRATION ERRORS

TABLE B.12. Quantitative comparison of reconstruction results of r-Consistency with Adaptive Gaussian Averaging for the Piper data set. MVPS = 100

| Algorithm | Reprojection Error (R, G, B) | False Positive Pixel Count |
|--|------------------------------|----------------------------|
| Average pixel coordinates error STD = 0.2 pixels | | |
| r-Consistency | 34.0291, 31.9617, 29.8297 | 85327 |
| AWVoM with AGA | 33.8916, 31.8386, 29.4215 | 81187 |
| Average pixel coordinates error STD = 1 pixels | | |
| r-Consistency | 50.0025, 47.9996, 44.4004 | 54603 |
| AWVoM with AGA | 48.6404, 46.6783, 43.1968 | 57127 |
| Average pixel coordinates error STD = 2 pixels | | |
| r-Consistency | 47.2521, 45.5110, 42.6523 | 66930 |
| AWVoM with AGA | 46.4655, 44.7614, 41.9205 | 65360 |

TABLE B.13. Quantitative comparison of reconstruction results of r-Consistency with Adaptive Gaussian Averaging for the Piper data set. MVPS = 180

| Algorithm | Reprojection Error (R, G, B) | False Positive pixel Count |
|--|------------------------------|----------------------------|
| Average pixel coordinates error STD = 0.2 pixels | | |
| r-Consistency | 28.4672, 26.2019, 24.0470 | 77831 |
| AWVoM with AGA | 30.2686, 27.9217, 25.4384 | 76211 |
| Average pixel coordinates error STD = 1 pixels | | |
| r-Consistency | 40.6135, 38.5947, 36.1864 | 71911 |
| AWVoM with AGA | 42.6402, 40.6570, 37.8912 | 71952 |
| Average pixel coordinates error STD = 2 pixels | | |
| r-Consistency | 43.7924, 41.8676, 39.1210 | 72932 |
| AWVoM with AGA | 43.3839, 41.5361, 38.8225 | 79654 |

REFERENCES

- [1] *Opengl: The industry's foundation for high performance graphics*, <http://www.opengl.org/>.
- [2] R. Bhotika, D.J. Fleet, and K.N. Kutulakos, *A probabilistic theory of occupancy and emptiness*, ECCV02, 2002, p. III: 112 ff.
- [3] Jean-Yves Bouguet, *Camera calibration toolbox for matlab*, http://www.vision.caltech.edu/bouguetj/calib_doc/, 2004.
- [4] A. Broadhurst, T.W. Drummond, and R. Cipolla, *A probabilistic framework for space carving*, ICCV01, 2001, pp. I: 388–393.
- [5] Adrian Broadhurst, *A probabilistic framework for space carving*, Ph.D. thesis, Cambridge University, Department of Computer Science, 2001.
- [6] Rodrigo L. Carceroni and Kiriakos N. Kutulakos, *Multi-view scene capture by surfel sampling: From video streams to non-rigid 3d motion, shape and reflectance*, Int. J. Comput. Vision **49** (2002), no. 2-3, 175–214.
- [7] W. Bruce Culbertson, Thomas Malzbender, and Gregory G. Slabaugh, *Generalized voxel coloring*, ICCV '99: Proceedings of the International Workshop on Vision Algorithms (Kerkyra, Greece), Springer-Verlag, 2000, pp. 100–115.
- [8] J.S. de Bonet and P. Viola, *Roxels: Responsibility weighted 3d volume reconstruction*, ICCV99, 1999, pp. 418–425.
- [9] C.R. Dyer, *Volumetric scene reconstruction from multiple views*, FIU01, 2001, pp. 469–489.
- [10] Martin Granger-Piché, Emric Epstein, and Pierre Poulin, *Interactive hierarchical space carving with projector-based calibrations.*, VMV, 2004, pp. 159–166.
- [11] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, 2003.

- [12] John Isidoro and Stan Sclaroff, *Contour generator points for threshold selection and a novel photo-consistency measure for space carving*, Tech. Report BUCS-TR-2003-025, Boston University Department of Computer Science, 2003.
- [13] T. Kanade, P. J. Narayanan, and P. W. Rander, *Virtualized reality: concepts and early results*, VSR '95: Proceedings of the IEEE Workshop on Representation of Visual Scenes (Washington, DC, USA), IEEE Computer Society, 1995, p. 69.
- [14] S.B. Kang, R. Szeliski, and J. Chai, *Handling occlusions in dense multi-view stereo*, CVPR01, 2001, pp. I:103–110.
- [15] M. Kimura, H. Saito, and T. Kanade, *3d voxel construction based on epipolar geometry*, ICIP99, 1999, pp. III:135–139.
- [16] Kiriakos N. Kutulakos, *Approximate n-view stereo*, ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part I (London, UK), Springer-Verlag, 2000, pp. 67–83.
- [17] Kiriakos N. Kutulakos and Steven M. Seitz, *A theory of shape by space carving*, Tech. report, University of Rochester, 1998.
- [18] ———, *A theory of shape by space carving*, Int. J. Comput. Vision **38** (2000), no. 3, 199–218.
- [19] Musik Kwon, Kyoung Mu Lee, and Sang Uk Lee, *A statistical error analysis for voxel coloring*, International Conference on Image Processing (2003), 425.
- [20] Matthew Loper, *Archimedes: Shape reconstruction from pictures*, http://loper.org/matt/Archimedes/Archimedes_docs/html/index.html/, 2002.
- [21] D. Marr and T.A. Poggio, *A computational theory of human stereo vision*, RoyalP **B-204** (1979), 301–328.
- [22] A.A. Montenegro, M. Gattass, P.C.P. Carvalho, and L.C.P.R. Velho, *Adaptive space carving*, 3DPVT04, 2004, pp. 199–206.
- [23] Sylvain Paris and François Sillion, *Robust acquisition of 3d informations from short image sequences*, Graph. Models **65** (2003), no. 4, 222–238.
- [24] Andrew C. Prock and Charles R. Dyer, *Towards real-time voxel coloring*, Proc. Image Understanding Workshop, 1998, pp. 315–321.
- [25] H. Saito and T. Kanade, *Shape reconstruction in projective voxel grid space from large number of images*, CVPR99, 1999, pp. II: 49–54.
- [26] S.M. Seitz and C.R. Dyer, *Photorealistic scene reconstruction by voxel coloring*, IJCV **35** (1999), no. 2, 151–173.

- [27] Steven M. Seitz and Charles R. Dyer, *Photorealistic scene reconstruction by voxel coloring*, CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97) (Washington, DC, USA), IEEE Computer Society, 1997, p. 1067.
- [28] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer, *A survey of methods for volumetric scene reconstruction from photographs*, International Workshop on Volume Graphics, 2001, pp. 81–100.
- [29] G.G. Slabaugh, W.B. Culbertson, T. Malzbender, M.R. Stevens, and R.W. Schafer, *Methods for volumetric reconstruction of visual scenes*, IJCV **57** (2004), no. 3, 179–199.
- [30] Gregory G. Slabaugh, Thomas Malzbender, and W. Bruce Culbertson, *Volumetric warping for voxel coloring on an infinite domain*, SMILE '00: Revised Papers from Second European Workshop on 3D Structure from Multiple Images of Large-Scale Environments (London, UK), Springer-Verlag, 2001, pp. 109–123.
- [31] E. Steinbach and B. Girod, *3-d reconstruction of real-world objects using extended voxels*, ICIP00, 2000, pp. Vol I: 569–572.
- [32] Charles V. Stewart, *Robust parameter estimation in computer vision*, SIAM Rev. **41** (1999), no. 3, 513–537.
- [33] R. Szeliski and P. Golland, *Stereo matching with transparency and matting*, IJCV **32** (1999), no. 1, 45–61.
- [34] S. Vedula, S. Baker, S.M. Seitz, and T. Kanade, *Shape and motion carving in 6d*, CVPR00, 2000, pp. II: 592–598.
- [35] R. Yang, M. Pollefeys, and G. Welch, *Dealing with textureless regions and specular highlights: A progressive space carving scheme using a novel photo-consistency measure*, ICCV03, 2003, pp. 576–584.
- [36] Anthony Yezzi, Greg Slabaugh, Adrian Broadhurst, Roberto Cipolla, and Ron Schafer, *A surface evolution approach o probabilistic space carving*, 3dpvt, 2002, p. 618.
- [37] Cha Zhang and Tsuhan Chen, *A system for active image-based rendering*, ICME, july 2003.