

Christian Nitschke

A Framework for Real-time 3D
Reconstruction by Space Carving using
Graphics Hardware

Diploma Thesis

YOUR KNOWLEDGE HAS VALUE



- We will publish your bachelor's and master's thesis, essays and papers
- Your own eBook and book - sold worldwide in all relevant shops
- Earn money with each sale

Upload your text at www.GRIN.com
and publish for free



A Framework for Real-time 3D Reconstruction by Space Carving using Graphics Hardware

by

Christian Nitschke

Submitted to the Department of Media System Science

in partial fulfillment of the requirements for the degree of
Diplom-Mediensystemwissenschaftler

at the

BAUHAUS-UNIVERSITÄT WEIMAR

Date of Submission	12/06/2006
Date of Defence	12/15/2006

To my Parents

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Christian Nitschke, Weimar 12/06/2006

ABSTRACT

Reconstruction of real-world scenes from a set of multiple images is a topic in Computer Vision and 3D Computer Graphics with many interesting applications. There exists a powerful algorithm for shape reconstruction from arbitrary viewpoints, called Space Carving. However, it is computationally expensive and hence can not be used with applications in the field of 3D video or CSCW as well as interactive 3D model creation. Attempts have been made to achieve real-time framerates using PC cluster systems. While these provide enough performance they are also expensive and less flexible. Approaches that use GPU hardware acceleration on single workstations achieve interactive framerates for novel-view synthesis, but do not provide an explicit volumetric representation of the whole scene. The proposed approach shows the efforts in developing a GPU hardware-accelerated framework for obtaining the volumetric photo hull of a dynamic 3D scene as seen from multiple calibrated cameras. High performance is achieved by employing a shape from silhouette technique in advance to obtain a tight initial volume for Space Carving. Also several speed-up techniques are presented to increase efficiency. Since the entire processing is done on a single PC the framework can be applied to mobile setups, enabling a wide range of further applications. The approach is explained using programmable vertex and fragment processors with current hardware and compared to highly optimized CPU implementations. It is shown that the new approach can outperform the latter by more than one magnitude.

ACKNOWLEDGMENTS

At first, I want to thank my supervisor, Oliver Bimber for providing me with the opportunity to spend a one-year research program at Takemura Lab in Osaka University, Japan. This work would not exist in the current form without his great support. I also want to thank my second supervisor, Kiyoshi Kiyokawa for coming to Germany just to attend my defense, for all the good discussions and comments when I got stuck in the daily details, for the wonderful time in Japan and the efforts in extending my stay in the last minute. It is a good place to thank both for reading the thesis in one week. I know myself, a lot of pages came together. I would also like to thank Atsushi Nakazawa for directly supervising my work with interesting discussions and background knowledge.

My parents and my family always supported me and gave me a backing on my way. I want to thank you for all the support and that you never said "no" to my ideas.

"The stone unhewn and cold becomes a living mould,
The more the marble wastes the more the statue grows."
Michelangelo Buonarroti

CONTENTS

1	Introduction	1
1.1	Application	1
1.2	Classification	2
1.3	Performance	3
1.4	Contribution	4
1.5	Overview	5
2	Related Work	7
2.1	Shape from Silhouette	7
2.1.1	Image Segmentation	8
2.1.2	Foundations	8
2.1.3	Performance of View-Independent Reconstruction	9
2.1.3.1	CPU	9
2.1.3.2	GPU Acceleration	9
2.1.4	Performance of View-Dependent Reconstruction	10
2.1.4.1	CPU	10
2.1.4.2	GPU Acceleration	10
2.1.5	Conclusion	11
2.2	Shape from Photo-Consistency	12
2.2.1	Foundations	13
2.2.2	Performance of View-Independent Reconstruction	14
2.2.2.1	CPU	14
2.2.2.2	GPU Acceleration	14
2.2.3	Performance of View-Dependent Reconstruction	15
2.2.3.1	CPU	15
2.2.3.2	GPU Acceleration	16
2.2.4	Conclusion	17
3	Fundamentals	18
3.1	Camera Geometry	18
3.1.1	Pinhole Camera Model	18
3.1.2	Camera Parameters	19
3.1.2.1	Intrinsic Parameters	19
3.1.2.2	Extrinsic Parameters	20
3.1.2.3	Radial Lens Distortion	21
3.1.3	Camera Calibration	21
3.2	Light and Color	23
3.2.1	Light in Space	23
3.2.1.1	Radiance	23

Contents

3.2.2	Light at a Surface	24
3.2.2.1	Irradiance	24
3.2.2.2	Radiosity	24
3.2.2.3	Lambertian and Specular Surfaces	25
3.2.3	Occlusion and Shadows	25
3.2.4	Light at a Camera	26
3.2.5	Color	26
3.2.6	Color Representation	27
3.2.6.1	Linear Color Spaces	27
3.2.6.2	Non-linear Color Spaces	28
3.2.6.3	Color Metric	29
3.2.7	CCD Camera Color Imaging	30
3.3	3D Reconstruction from Multiple Views	30
3.3.1	Visual Hull Reconstruction by Shape from Silhouette	30
3.3.1.1	Shape from Silhouette	30
3.3.1.2	Discussion	31
3.3.1.3	The Visual Hull	32
3.3.1.4	Silhouette-Equivalency	33
3.3.1.5	Number of Viewpoints	34
3.3.1.6	Conclusion	34
3.3.2	Photo Hull Reconstruction by Shape from Photo-Consistency	35
3.3.2.1	Shape from Photo-Consistency	36
3.3.2.2	Discussion	36
3.3.2.3	Photo-Consistency	37
3.3.2.4	The Photo Hull	38
3.3.2.5	The Space Carving Algorithm	39
3.3.2.6	Voxel Visibility	40
3.3.2.7	Conclusion	41
4	Basic Algorithm	42
4.1	Data	42
4.1.1	Camera Parameters	42
4.1.2	Image Data	43
4.2	Reconstruction	44
4.2.1	3D Data Representation	44
4.2.2	Volumetric Bounding Box	45
4.2.3	Maximal Volume Intersection	46
4.2.4	Visual Hull Approximation	46
4.2.5	Photo-Consistent Surface	46
4.2.5.1	Active Source Camera Test	47
4.2.5.2	Photo Consistency Test	48

Contents

5	Advanced Algorithm	49
5.1	Overview	49
5.1.1	Deployment	49
5.1.2	Process Flow	49
5.2	Texture Processing	50
5.2.1	Lookup Table for Projection Coordinates	51
5.2.2	Mapping Image Data into Textures	52
5.2.3	Texture Upload and Processing Performance	52
5.2.4	GPU Image Processing	54
5.3	Destination Cameras	54
5.3.1	Discussion	55
5.3.1.1	Ray Casting vs. Multi Plane Sweeping	56
5.3.1.2	Virtual vs. Natural Views	57
5.3.2	Interleaved Depth Sampling	58
5.3.3	Active Destination Cameras	60
5.3.3.1	Source Camera Viewing Ray	60
5.3.3.2	Intersection of Volume and Source Camera Viewing Ray	61
5.3.3.3	Activity Decision	62
5.4	Reconstruction	62
5.4.1	Vertex Data	63
5.4.2	Vertex Shader Visual Hull Approximation	64
5.4.2.1	Decreasing the Sampling Error for Interleaved Sampling	64
5.4.2.2	Early Ray Carving	64
5.4.3	Fragment Shader Photo-Consistent Surface	65
5.4.3.1	Filling Holes	65
5.4.3.2	Modified Active Source Camera Decision	66
5.4.4	Fragment Shader Color Blending	67
5.4.5	Fragment Shader Render to Texture	68
5.5	Postprocessing	70
5.5.1	Extracting Texture Data	70
5.5.2	Filling Interior Volume Data	71
5.5.2.1	Ambiguities	71
5.5.2.2	Performance	72
6	Experiments	73
6.1	System Setup	73
6.2	Implementation	73
6.3	Datasets	74
6.4	Performance	75
6.4.1	Abstract Data Performance Experiments	76
6.4.1.1	CPU-GPU Texture Upload	76

Contents

6.4.1.2	Interleaved Sampling	76
6.4.1.3	Early Ray Carving	76
6.4.1.4	Fragment Shader CIELab-RGB Conversion	77
6.4.1.5	Porting all Load to the Fragment Processor	77
6.4.1.6	GPU-CPU Texture Read-back	77
6.4.1.7	FBO Texture Size	78
6.4.1.8	Impact of CPU-GPU Texture Upload on overall Performance	78
6.4.1.9	Number of Source Cameras	78
6.4.1.10	Number of Destination Cameras	79
6.4.2	Concrete Data Performance Experiments	79
6.4.2.1	Algorithmic Features	80
6.4.2.2	Destination Cameras	80
6.4.2.3	Volumetric Resolution	81
6.4.2.4	Volumetric Bounding Box	81
6.4.2.5	<i>PCS</i> Increments	81
6.4.3	Conclusion	82
6.4.3.1	Algorithmic Features	82
6.4.3.2	Parameters	82
6.4.3.3	GPU/CPU Comparison	82
6.5	Quality	83
6.5.1	Concrete Data Quality Experiments	83
6.5.1.1	Volumetric Resolution	83
6.5.1.2	Volumetric Bounding Box	83
6.5.1.3	<i>PCS</i> Increments	84
6.5.2	Visual Experiments	84
6.5.2.1	Image Segmentation	84
6.5.2.2	Interleaved Sampling and <i>MVI</i>	84
6.5.2.3	Camera Viewing Cone Intersection	84
6.5.2.4	Reconstruction of <i>VHA</i> and <i>PCS</i>	85
6.5.2.5	Volumetric Resolution	85
6.5.2.6	<i>PCS</i> Increments	85
6.5.2.7	Geometrical Score for Active Source Camera Computation	86
6.5.2.8	Range of Color Distances for Active Source Camera Computation	86
6.5.2.9	Labeling of Interior Space	86
6.5.3	Conclusion	86
6.5.3.1	Image Segmentation	87
6.5.3.2	<i>BB</i> , <i>MVI</i> and Viewing Cone Intersection	87
6.5.3.3	<i>VHA</i> and <i>PCS</i>	87
6.5.3.4	<i>PCS</i> Parameters	87
6.5.3.5	Labeling of Interior Space	88

Contents

7 Discussion and Enhancements	89
7.1 Summary	89
7.2 Limitations	90
7.3 Future Work	90
7.3.1 Online System	90
7.3.2 Performance	91
7.3.3 Quality	91
7.4 Annotation	92
A Abstract Data Performance Experiments	93
B Concrete Data Performance Experiments	98
C Concrete Data Quality Experiments	104
D Visual Experiments	108
References	127

LIST OF FIGURES

1.1	A Taxonomy of 3D Shape Acquisition Techniques	2
2.1	Image-based Ray Sampling of the Visual Hull (IBVH)	11
2.2	Multi Plane-Sweep Implementation of Space Carving	15
3.1	Perspective Projection in Pinhole Camera	19
3.2	Relation of Coordinate Systems for Image Formation	20
3.3	Point Correspondences for Linear Camera Calibration	22
3.4	Light at a Camera	26
3.5	Volume Intersection	31
3.6	Limitations of Volume Intersection	32
3.7	Silhouette-Equivalency	34
3.8	Photo-Consistency	36
3.9	Visibility and Non-Photo-consistency	39
4.1	Nested Volumes	45
5.1	Hardware Deployment of Reconstruction Framework	50
5.2	Process Flow in Reconstruction Framework	51
5.3	Destination Camera Setup	56
5.4	Interleaved Sampling in 2D	59
5.5	Active Destination Camera Test	62
5.6	Early Ray Carving	64
5.7	Ray-based Sampling	65
5.8	Geometrical Score for Active Source Camera Decision	66
5.9	FBO Rendertexture	69
5.10	Labeling Interior Volume	71
6.1	Multiple Viewpoint Video Capturing Setup	74
6.2	Concrete Hardware Deployment of Reconstruction Framework	75
1.1	Performance CPU-GPU Texture Upload	94
1.2	Performance Interleaved Sampling	94
1.3	Performance Early Ray Carving	94
1.4	Performance Raw CIELab-RGB Conversion	95
1.5	Performance Vertex/Fragment Shader Balancing	95
1.6	Performance GPU-CPU Texture Read-back	95
1.7	FBO Texture Size Performance Impact 1	96
1.8	FBO Texture Size Performance Impact 2	96
1.9	Impact of CPU-GPU Texture Upload on System Performance	97
1.10	Impact of Source Cameras on Rendering Performance	97
1.11	Impact of Destination Cameras on Rendering Performance	97
2.1	Performance of Algorithmic Features (Dancer1)	99
2.2	Performance of Algorithmic Features (Dancer2)	99
2.3	Performance of Algorithmic Features (Paper Houses)	99

List of Figures

2.4	Destination Camera Rendering Performance (Dancer1)	100
2.5	Destination Camera Rendering Performance (Dancer2)	100
2.6	Destination Camera Rendering Performance (Paper Houses)	100
2.7	Performance Impact of Volumetric Resolution (Dancer1)	101
2.8	Performance Impact of Volumetric Resolution (Dancer2)	101
2.9	Performance Impact of Volumetric Resolution (Paper Houses)	101
2.10	Performance Impact of Volumetric Bounding Box (Dancer1)	102
2.11	Performance Impact of Volumetric Bounding Box (Dancer2)	102
2.12	Performance Impact of Volumetric Bounding Box (Paper Houses)	102
2.13	Performance Impact of <i>PCS</i> Increments (Dancer1)	103
2.14	Performance Impact of <i>PCS</i> Increments (Dancer2)	103
2.15	Performance Impact of <i>PCS</i> Increments (Paper Houses)	103
3.1	Quality Impact of Volumetric Resolution (Dancer1)	105
3.2	Quality Impact of Volumetric Resolution (Dancer2)	105
3.3	Quality Impact of Volumetric Resolution (Paper Houses)	105
3.4	Quality Impact of Volumetric Bounding Box (Dancer1)	106
3.5	Quality Impact of Volumetric Bounding Box (Dancer2)	106
3.6	Quality Impact of Volumetric Bounding Box (Paper Houses)	106
3.7	Quality Impact of <i>PCS</i> Increments (Dancer1)	107
3.8	Quality Impact of <i>PCS</i> Increments (Dancer2)	107
3.9	Quality Impact of <i>PCS</i> Increments (Paper Houses)	107
4.1	Image Segmentation Experiment (Dancer2)	109
4.2	Image Segmentation Experiment (Dancer1)	110
4.3	Conventional/Interleaved Sampling Visualization (Dancer1) 1	111
4.4	Conventional/Interleaved Sampling Visualization (Dancer1) 2	112
4.5	Camera Viewing Cone Intersection Visualization (Dancer1) 1	113
4.6	Camera Viewing Cone Intersection Visualization (Dancer1) 2	114
4.7	<i>PCS</i> Reconstruction Experiment (Dancer1) 1	115
4.8	<i>PCS</i> Reconstruction Experiment (Dancer1) 2	116
4.9	<i>PCS</i> Reconstruction Experiment (Dancer1) 3	117
4.10	<i>PCS</i> Reconstruction Experiment (Dancer1) 4	118
4.11	<i>PCS</i> Reconstruction Experiment (Dancer2) 1	119
4.12	<i>PCS</i> Reconstruction Experiment (Dancer2) 2	120
4.13	<i>PCS</i> Increments Experiment (Paper Houses) 1	121
4.14	<i>PCS</i> Increments Experiment (Paper Houses) 2	122
4.15	<i>PCS</i> Increments Experiment (Dancer1)	123
4.16	Impact of Geometrical Score sc_k^g Experiment (Dancer1)	124
4.17	Impact of Range $r(k)$ on Color Similarity Score sc_k^c Experiment (Dancer1)	125
4.18	Interior Volume Labeling Experiment (Dancer1)	126