View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Repository@USM

3D MORPHING AND SHAPE TRANSFORMATION USING SLICES

SHAMIMA YASMIN

UNIVERSITI SAINS MALAYSIA

2009

3D MORPHING AND SHAPE TRANSFORMATION USING SLICES

by

SHAMIMA YASMIN

Thesis submitted in fulfilment of the requirements
for the degree of
Doctor of Philosophy

December 2009

ACKNOWLEDGEMENTS

I would like to extend my heartfelt thanks to all who have contributed to my research work as well as provided me intellectual and moral support throughout my doctoral stint at USM... but

- First of all, I would like to extend my sincere gratitude and thanks to Associate Professor Dr. Abdullah Zawawi Talib for accepting me as a graduate student, and for his advice, supervision and guidance.

- Next, I would like to extend my gratitude and heartfelt thanks to the Ministry of Higher Education, Malaysia for sponsoring my graduate work.

- Next, I would like to thank my colleagues at USM especially to Mr. Asef Al-Khateeb, Mr. Reza Zare, Mr. Syed Amin Hosseini Seno, Mr. Haider Ali Mshali and Mr. Amir Akhvan Masoumi for their unflagging support and moral encouragement during my research tenure at USM.

- Special thanks to Ms. Lim Lian Tze for her LaTeX thesis template.

- Special thanks to my parents without whose blessings, goodwill, love, advice and moral encouragement it would have been impossible for me to complete my work. I dedicate this dissertation to them.

TABLE OF CONTENTS

Acknowledgements ii

Table of Contents iii

List of Tables vii

List of Figures viii

List of Abbreviations xix

Abstrak xx

Abstract xxii

CHAPTER 1 – INTRODUCTION

1.1 Overview and Motivation 1

1.2 Problem Statements 4

1.3 Objectives 5

1.4 Research Scope and Limitations 6

1.5 Methodology 7

1.6 Applications and Benefits of the Research 9

1.7 Contributions 10

1.8 Thesis Organisation 11

CHAPTER 2 – LITERATURE REVIEW

2.1 Classification 12

2.2 Surface-based Approach 14

2.3 Volume-based Approach 19

2.4 Shape Transformation Involving more than Two Objects 24

2.5 Discussion on Surface-based and Volume-based Approaches 25

2.6 Interpolation in Shape Transformation 28

2.7 Surface Reconstruction in Shape Transformation 29

2.8 Discussion on Interpolation and Surface Reconstruction Methods 32

2.9 Summary 33

CHAPTER 3 – A NEW METHOD FOR 3D MORPHING USING SLICES

3.1 Introduction 34

3.2 Algorithm Overview 35

3.3 The Algorithm in Detail 38

3.3.1 Data Traversal and Slicing of Data 38

3.3.2 Boundary Extraction 42

3.3.3 Boundary Projection 43

3.3.4 Boundary Traversal and Extraction of Point Clouds 44

3.3.5 Interpolation of Point Clouds 46

3.3.6 Orientation and Translation of Interpolated Point Clouds 49

3.3.7 Surface Reconstruction 50

3.4 Implementation and Evaluation 51

3.5 Empirical Evaluation 65

3.6 Discussion and Comparison with Other Algorithms 75

3.7 Summary 78

CHAPTER 4 – A NEW APPROACH FOR SURFACE RECONSTRUCTION USING SLICES

4.1 Introduction 79

4.2 Overview of the Algorithm 80

4.3 Contour Preprocessing 81

4.4 Surface Reconstruction 82

4.4.1 Separation of Disconnected Regions 83

4.4.2 Establishing Correspondence between Two Consecutive Point Clouds and Basic Cell Construction 83

4.4.3 Consideration of Critical Points 90

4.4.3(a) Consideration of Intermediate Empty Space 90

4.4.3(b) Transition between Different-numbered Regions during Cell Construction 91

4.4.3(c) Distinguishing Connected / Disconnected Regions across the Direction of Traversal of Slices 93

4.5 Implementation and Evaluation 95

4.6 Empirical Evaluation 101

4.7 Discussion and Analysis of Complexity 109

4.8 Summary 111

CHAPTER 5 – SHAPE TRANSFORMATION OF MULTIPLE OBJECTS USING SLICES

5.1 Introduction 112

5.2 Algorithm Overview 114

5.3 Algorithm in Detail 115

5.3.1 Data Traversal and Slicing of Data 115

5.3.2 Boundary Extraction 116

5.3.3 Boundary Projection 116

5.3.4 Boundary Traversal and Extraction of Contour Points 117

5.3.5 Interpolation of Point Clouds 117

5.3.6 Orientation and Translation of Interpolated Point Clouds 118

5.4 Surface Reconstruction 119

5.5 Multiple Shape Transformation Results and Evaluation 120

5.6 Empirical Evaluation for Shape Transformation of Multiple Objects 126

5.7 Multiple Influence Shapes 135

5.8 Discussion and Analysis of Complexity 139

5.9 Summary 142

CHAPTER 6 – FURTHER AND POSSIBLE EXTENSIONS

6.1 Introduction 143

6.2 Parallel Morphing 144

6.2.1 Proposed Approach 144

6.3 Feature-based Volume Morphing 150

6.4 Discussion 153

6.5 Summary 154

CHAPTER 7 – CONCLUSION AND FUTURE WORK

7.1 Conclusion 155

7.2 Limitation of the Approach 158

7.2.1 Adjustment between the Number of Steps (Slices) and the Number of Traversal Interval Intervals 158

7.2.2 Use of Data Cleaning Filter 158

7.3 Future Work 159

7.3.1 Consideration of Graphics Properties 159

7.3.2 Parallel Morphing 159

7.3.3 Reconstruction of Surface from Slices 160

7.3.4 Feature-based Volume Morphing with Slices 161

7.4 Concluding Remarks 161

References 163

APPENDICES 166

APPENDIX A – OUTPUT OF EMPIRICAL EVALUATION ON MORPHING 167

APPENDIX B – OUTPUT OF EMPIRICAL EVALUATION ON SURFACE RECONSTRUCTION 196

APPENDIX C – OUTPUT OF EMPIRICAL EVALUATION ON SHAPE TRANSFORMATION WITH MULTIPLE OBJECTS (WHEN THE NUMBER OF INPUT OBJECTS IS THREE) 206

APPENDIX D – OUTPUT OF EMPIRICAL EVALUATION ON SHAPE TRANSFORMATION WITH MULTIPLE OBJECTS (WHEN THE NUMBER OF INPUT OBJECTS IS FOUR) 219

List of Publications 224

LIST OF TABLES

		Page
Table 2.1	Comparative Analysis of Some of the Existing 3D Morphing Algorithms.	**27**
Table 2.2	Comparative Analysis of Interpolation Methods and Surface Reconstruction Methods of Some of the Existing 3D Morphing Algorithms.	**32**
Table 3.1	Aims/Objectives of Implementation and Evaluation.	**53**
Table 3.2	Variation in Runtime, the Number of Slices and the Average Number of Cells with Variation in the Number of Steps and the Number of Traversal Intervals.	**67**
Table 3.3	Table for Figure 3.33.	**71**
Table 3.4	Table for Figure 3.34.	**73**
Table 4.1	Variation in Runtime and the Number of Cells with Variation in the Number of Slices and the Number of Traversal Intervals.	**102**
Table 4.2	Table for Figure 4.23.	**105**
Table 4.3	Table for Figure 4.24.	**107**
Table 5.1	Variation in Runtime, the Number of Slices and the Number of Cells with Variation in the 'Number of Steps' and the 'Number of Traversal Intervals' Involving Three Objects.	**127**
Table 5.2	Table for Figure 5.15.	**130**
Table 5.3	Variation in Runtime, the Number of Slices and the Number of Cells with Variation in the 'Number of Steps' and the 'Number of Traversal Intervals' Involving Four Objects.	**132**
Table 5.4	Table for Figure 5.17.	**134**

LIST OF FIGURES

		Page
Figure 1.1	A box (a) Twisted and (b) Bent [1].	**1**
Figure 1.2	Data Partitioning in (a) Surface-based Shape Transformation and (b) Volume-based Shape Transformation.	**2**
Figure 1.3	Various Steps of Research Methodology.	**8**
Figure 2.1	Projection of One Object onto the Other Object [5].	**15**
Figure 2.2	(a) Original Source and Target Objects and (b) Conversion of Source and Target Objects to their Base Domain Using Point and Line Control Fields [7].	**17**
Figure 2.3	Minkowski Sum Combines the Shape Characteristics of its Arguments A and B [4].	**17**
Figure 2.4	(a) No Self-intersection and (b) Self-intersection on Projection of a Morphing Patch onto a 2D Plane (Adapted from [9]).	**18**
Figure 2.5	Graphical Representation of Fourier (Left) and Wavelet (Right) Transforms [20].	**20**
Figure 2.6	Data Flow in Feature-based Volume Morphing (Adapted from [14]).	**21**
Figure 2.7	Use of Disk Fields on 3D Object in Volume Morphing [15].	**22**
Figure 2.8	Movement of the Source Object 'A' along its Surface Normals to Fit the Target Object 'B' [17].	**23**
Figure 2.9	(Left) Pairs of Boundary and Normal Constraints Defined as Circles and Pluses and (Right) Resulting Variational Implicit Function as a Height Field (Adapted from [18]).	**24**
Figure 2.10	Morphological Difference between Two Images is Defined by Regions I and II while Region III is the Overlapping Region [19].	**24**
Figure 2.11	(Left) Use of Bounding Object for Blending Two Objects and (Right) Blended Object (Adapted from [22]).	**25**
Figure 2.12	Trilinear interpolation in volume rendering [25].	**29**
Figure 3.1	Flowchart of the Proposed Algorithm	**36**
Figure 3.2	(a) Normal Bounding Box (NBB) and (b) Oriented Bounding Box (OBB).	**37**

Figure 3.3	Division of the End Data when Alignment is within Maximum or Medium Direction of the Oriented Bounding Box (OBB).	**40**
Figure 3.4	Division of the Middle Data.	**40**
Figure 3.5	Flow Chart for the Recursive Binary Subdivision of the End Part and the Middle Part.	**41**
Figure 3.6	Data Traversal and Slicing of Given Data.	**42**
Figure 3.7	Extraction of Source and Target Boundaries.	**43**
Figure 3.8	Boundary Traversal and Extraction of Contour Points.	**44**
Figure 3.9	(a) Region Demarcation in a Particular Contour Point Cloud and (b) Compression of a 'Number Array' into Two Arrays	**45**
Figure 3.10	(a) Region Demarcation in a Particular Contour Point Cloud and (b) Compression of a 'Number Array' into Two Arrays with One Intermediate Empty Region.	**45**
Figure 3.11	Interpolation of Points with Equal Number of Regions in Source and Target Point Clouds.	**47**
Figure 3.12	Insertion of Empty Spaces during Interpolation.	**47**
Figure 3.13	Interpolation when both Source and Target Point Clouds Contain Equal Number of Intermediate Empty Spaces.	**48**
Figure 3.14	Interpolation when Source and Target Point Clouds Contain Unequal Number of Intermediate Empty Spaces.	**49**
Figure 3.15	Orientation and Translation of a Single Interpolated Point Cloud.	**50**
Figure 3.16	Orientation and Translation of All Interpolated Point Clouds.	**50**
Figure 3.17	Reconstructed (a) Rectangular Mesh and (b) Surface from Interpolated Point Clouds Shown in Figure 3.16.	**51**
Figure 3.18	Gradual Morphing Sequence between Source and Destination Objects Using the OBB to Determine the Initial Alignment of Objects.	**54**
Figure 3.19	Gradual Morphing Sequence between Source and Destination Objects Using the NBB to Determine the Initial Alignment of Objects.	**55**
Figure 3.20	Gradual Morphing Sequence between Source and Destination Objects Using the OBB to Determinine the Initial Alignment of Objects.	**55**
Figure 3.21	Two Different Morphing Sequences between the Same Pair of Source and Destination Objects Using Different Direction of the OBB as the Initial Alignment of the Source Object.	**56**

Figure 3.22 Gradual Morphing Sequence between Source and Destination Objects Using Minimum Direction of the OBB as the Initial Alignment for Both Objects. **57**

Figure 3.23 Gradual Morphing Sequence between Source and Destination Objects Using the OBB as Initial Alignment for Both Objects. **58**

Figure 3.24 A Number of Morphing Sequences between the Same Source and Destination Objects Generated by Varying the 'Number of Slices' and the 'Number of Traversal Intervals' Keeping the Initial Alignment Unchanged. **59**

Figure 3.25 Different Morphing Sequences between the Same Pair of Source and Target Objects Using Different Directions of the OBB to Determine the Initial Alignments. **60**

Figure 3.26 Different Morphing Sequences between the Same Pair of Source and Destination Objects Using (a) the NBB and (b) the OBB to Determine the Initial Alignments for Source and Destination Objects. **61**

Figure 3.27 Three Morphing Sequences are generated between the Same Source and Destination Objects by Varying the 'Number of Slices' and the 'Number of Traversal Intervals' Keeping the Initial Alignment Unchanged. **63**

Figure 3.28 Variation in the Morphing Output with Variation in the 'Number of Slices' and the 'Number of Traversal Intervals'. **64**

Figure 3.29 Gradual Morphing Sequence between Source and Destination Objects Using the NBB to Determine the Initial Alignments of Source and Destination Objects. **65**

Figure 3.30 Gradual Morphing Sequence between Source and Destination Objects Using the NBB to Determine the Initial Alignments of Source and Destination Objects. **66**

Figure 3.31 Variation in (a) Runtime and (b) the Average Number of Cells with Variation in the 'Number of Slices' for Different Morphing Sequences for a fixed 'Number of Traversal Intervals'. **69**

Figure 3.32 Variation in (a) Runtime and (b) the Average Number of Cells with Variation in the 'Number of Traversal Intervals' for Different Morphing Sequences for a fixed 'Number of Slices'. **70**

Figure 3.33 Variation in Runtime and the Average Number of Cells with Variation in 'Number of Slices' and 'Number of Traversal Intervals' for 'Teapot vs 'C" of Figure 3.21(a)). **72**

Figure 3.34 Variation in Runtime and the Average Number of Cells with Variation in the 'Number of Slices' and the 'Number of Traversal Intervals' for 'Cow vs Conic spiral' of Figure 3.24. **74**

Figure 4.1	Region Demarcation during Contour Traversal.	**81**
Figure 4.2	Orientation and Translation of Contour Point Clouds.	**82**
Figure 4.3	Mapping Separated Regions: (a) Point Clouds with One Region (Top Point Cloud) and Three Regions (Bottom Point Cloud) and (b) Vertical mapping (Middle) and Lateral Mapping ((Left) and (Right)).	**83**
Figure 4.4	Mapping of Corresponding Regions between Two Consecutive Point Clouds when Disconnected Regions Exist in the Original Point Clouds.	**86**
Figure 4.5	Pseudocode for Optimum Subarrays when there is no Disconnected Region.	**87**
Figure 4.6	Establishing Correspondence between Adjacent Contour Points and Cell Construction.	**89**
Figure 4.7	(a) Interpolated Point Clouds and (b) Surface Reconstruction when one boundary has empty space (Top) and both boundaries have empty spaces (Bottom).	**91**
Figure 4.8	Pseudocode for Transition between Different-numbered Regions.	**92**
Figure 4.9	Surface Reconstruction while Transiting between Different-numbered Regions.	**93**
Figure 4.10	Surface Reconstruction Considering the Number of Intermediate Empty Spaces across the Slice: (a) Two Empty Spaces and (b) One Empty Space.	**94**
Figure 4.11	Surface Reconstruction Considering the Number of Intermediate Empty Spaces across the Slice.	**95**
Figure 4.12	Reconstruction of the Surface of a “teapot” from Slices Arranged along the Y-Axis.	**96**
Figure 4.13	Reconstruction of the Surface of a “Cow” from Slices Arranged along the Y-axis.	**96**
Figure 4.14	Reconstruction of Two Parametric Surfaces (a) “Cross Cap” and (b) “Boy” from Slices Arranged along the Y-axis.	**96**
Figure 4.15	Reconstruction of (a) “Heart” and (b) “Head” from Slices Arranged along the Z-Axis and the Y-Axis Respectively.	**97**
Figure 4.16	Reconstruction of “Cactus” from Slices Arranged along the Y-Axis.	**98**
Figure 4.17	Variation in the Output with Variation in the Number of Traversal Intervals and the Number of Slices.	**98**
Figure 4.18	Reconstruction of a Surface from Slices Arranged along the Z-axis: (a) Front View and (b) Side View.	**99**

Figure 4.19	Adjustment of Discrepancies in the Output by Varying the Number of Slices.	**100**
Figure 4.20	Variation in the Output with Variation in the Number of Slices and the Number of Traversal Intervals: (Top) Front View and (b) (Bottom) X-sectional View.	**101**
Figure 4.21	Variation in (a) Runtime and (b) the Number of Cells with Variation in the 'Number of Slices' for the Surface Reconstruction of Various Objects for a fixed 'Number of Traversal Intervals'.	**103**
Figure 4.22	Variation in (a) Runtime and (b) the Number of Cells with Variation in the 'Number of Traversal Intervals' for the Surface Reconstruction of Various Objects for a fixed 'Number of Slices'.	**104**
Figure 4.23	Variation in Runtime and the Average Number of Cells with Variation in 'Number of Slices' and 'Number of Traversal Intervals' for 'Figure-8 klein' shown in Figure 4.17.	**106**
Figure 4.24	Variation in Runtime and the Average Number of Cells with Variation in 'Number of Slices' and 'Number of Traversal Intervals' for 'Cow'.	**108**
Figure 5.1	Orientation and Translation of an Interpolated Boundary when the Number of Input Objects is more than Two.	**113**
Figure 5.2	Flowchart of the Proposed Shape Transformation Algorithm Involving More than Two Objects.	**115**
Figure 5.3	Extraction of Boundaries when the Number of Input Data is Three.	**116**
Figure 5.4	Interpolation of Three Boundaries after Region Separation when the Boundaries have the Same Number of Regions.	**117**
Figure 5.5	Orientation and Translation of a Single Interpolated Point Cloud for the Three Objects Shown in Figure 5.3.	**118**
Figure 5.6	Orientation and Translation of All Interpolated Point Clouds for the Three Objects Shown in Figure 5.3.	**119**
Figure 5.7	Constructed Surface from Oriented and Translated Point Clouds of the Three Objects in Figure 5.3.	**119**
Figure 5.8	Shape Transformation Involving Three Objects.	**121**
Figure 5.9	Shape Transformation Involving Four Objects.	**123**
Figure 5.10	Shape Transformation for the Same Input Objects with Different Initial Directions of Traversal for the First Data.	**124**
Figure 5.11	Generating Aesthetically More Pleasant Output with Increase in the Number of Slices and the Number of Traversal Intervals.	**125**

Figure 5.12	Shape Transformation by Blending Different Objects in Different Proportions.	**126**
Figure 5.13	Variation in (a) Runtime and (b) the Number of Cells with Variation in the 'Number of Slices' for a given 'Number of Traversal Intervals' for Shape Transformation Involving Three Objects.	**128**
Figure 5.14	Variation in (a) Runtime and (b) the Number of Cells with Variation in the 'Number of Traversal Intervals' for a given 'Number of Slices' for Shape Transformation Involving Three Objects.	**129**
Figure 5.15	Variation in Runtime and the Average Number of Cells with Variation in the 'Number of Slices' and the 'Number of Traversal Intervals' for 'Teapot, Ellipsoid and 'C" of Figure 5.8(a).	**131**
Figure 5.16	Variation in Runtime and the Average Number of Cells with Variation in the 'Number of Slices' and the 'Number of Traversal Intervals' for Shape Transformation Involving Four Objects.	**133**
Figure 5.17	Variation in Runtime and the Number of Cells with Variation in the 'Number of Slices' and the 'Number of Traversal Intervals' for Shape Transformation Involving Four Objects of Figure 5.9(a).	**135**
Figure 5.18	Variation in Runtime with the 'Number of Slices' for a Given 'Number of Traversal Intervals' by Varying the Number of Input Objects.	**136**
Figure 5.19	Gradual Morphing Sequence between Source Object(Left) and Destination Object(Right) Controlled by Two Yellow-coloured Polyhedra Set as Two Control Points in Bezier Curve [39].	**136**
Figure 5.20	Gradual Morphing Sequence between Source Object (Left) and Destination Object (Right) Controlled by a Single Influence Shape [18].	**137**
Figure 5.21	Generating a New Morphing Sequence (dashed line) between the Same Source Object (Corner Left) and Destination Object (Corner Right) by Inserting Three Separate Influence Shapes at Three Stages of the Original Morphing Path.	**138**
Figure 5.22	Transformation of Multiple Objects to One Single Object to Influence the Original Path of Morphing.	**139**
Figure 6.1	Extraction of Equal Number of Boundaries from All Input Objects.	**144**
Figure 6.2	Communication between Master and Slave during Parallelisation.	**145**
Figure 6.3	Assignment of Data between Masters and Slaves During Data Traversal and the Slicing of Data.	**146**
Figure 6.4	Parallel Interpolation of Source and Target Point Clouds.	**147**
Figure 6.5	Parallel Surface Reconstruction from Interpolated Point Clouds.	**148**

Figure 6.6	Pseudocode for the Parallel Version of the Basic Morphing Algorithm.	**149**
Figure 6.7	Gradual Morphing Sequence between a Human Face and a Skull.	**151**
Figure 6.8	Establishing Correspondence between Source (Left) and Destination (Right) Objects in Feature-based Morphing.	**151**
Figure 6.9	(a) Corresponding Source and Target Portions and (b) Corresponding Control Points in Source and Destination Boundaries.	**152**
Figure 6.10	Pseudocode for Feature-based Volume Morphing from Slices.	**153**
Figure A.1	Gradual Morphing Sequences between a 'Teapot' and 'C' by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'.	**168**
Figure A.2	Gradual Morphing Sequences between a 'Teapot' and 'C' by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'.	**169**
Figure A.3	Gradual Morphing Sequences between a 'Teapot' and 'C' by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'.	**170**
Figure A.4	Gradual Morphing Sequences between a 'Teapot' and 'C' by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'.	**171**
Figure A.5	Gradual Morphing Sequences between a 'Bent Pipe' and a 'Torus' by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'.	**172**
Figure A.6	Gradual Morphing Sequences between a 'Bent Pipe' and a 'Torus' by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'.	**173**
Figure A.7	Gradual Morphing Sequences between a 'Bent Pipe' and a 'Torus' by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'.	**174**
Figure A.8	Gradual Morphing Sequences between a 'Bent Pipe' and a 'Torus' by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'.	**175**
Figure A.9	Gradual Morphing Sequences between a 'Bent Pipe' and a 'Torus' by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'.	**176**
Figure A.10	Gradual Morphing Sequences between a 'Bent Pipe' and a 'Torus' by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'.	**177**
Figure A.11	Gradual Morphing Sequences between a 'Bent Pipe' and a 'Torus' by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'.	**178**

Figure A.12 Gradual Morphing Sequences between a ‘Bent Pipe’ and a ‘Torus’ by Varying the ‘Number of Slices’ for a fixed ‘Number of Traversal Intervals’. **179**

Figure A.13 Gradual Morphing Sequences between Two Tori with Different Radii by Varying the ‘Number of Slices’ for a fixed ‘Number of Traversal Intervals’. **180**

Figure A.14 Gradual Morphing Sequences between Two Tori with Different Radii by Varying the ‘Number of Traversal Intervals’ for a fixed ‘Number of Slices’. **181**

Figure A.15 Gradual Morphing Sequences between a ‘Conic Spiral’ and a ‘Torus’ by Varying the ‘Number of Slices’ for a fixed ‘Number of Traversal Intervals’. **182**

Figure A.16 Gradual Morphing Sequences between a ‘Conic Spiral’ and a ‘Torus’ by Varying the ‘Number of Traversal Intervals’ for a fixed ‘Number of Slices’. **183**

Figure A.17 Gradual Morphing Sequences between a ‘Cow’ and a ‘Conic Spiral’ by Varying the ‘Number of Traversal Intervals’ for a fixed ‘Number of Slices’. **184**

Figure A.18 Gradual Morphing Sequences between a ‘Cow’ and a ‘Conic Spiral’ by Varying the ‘Number of Slices’ for a fixed ‘Number of Traversal Intervals’. **185**

Figure A.19 Gradual Morphing Sequence between a ‘Cow head’ and a Parametric Surface ‘Dini’ by Varying the ‘Number of Traversal Intervals’ for a fixed ‘Number of Slices’. **186**

Figure A.20 Gradual Morphing Sequence between a ‘Cow head’ and a Parametric Surface ‘Dini’ by Varying the ‘Number of Slices’ for a fixed ‘Number of Traversal Intervals’. **187**

Figure A.21 Gradual Morphing Sequences between a ‘Human Head’ and a ‘Sphere’ by Varying the ‘Number of Traversal Intervals’ for a fixed ‘Number of Slices’ **188**

Figure A.22 Gradual Morphing Sequences between a ‘Human Head’ and a ‘Sphere’ by Varying the ‘Number of Slices’ for a fixed ‘Number of Traversal Intervals’ **189**

Figure A.23 Gradual Morphing Sequence between ‘Schwarz’ and a ‘Bent Pipe’ by Varying the ‘Number of Traversal Intervals’ for a fixed ‘Number of Slices’. **190**

Figure A.24 Gradual Morphing Sequence between ‘Schwarz’ and a ‘Bent Pipe’ by Varying the ‘Number of Slices’ for a fixed ‘Number of Traversal Intervals’. **191**

Figure A.25 Gradual Morphing Sequence between a 'Teapot' and a Parametric Surface 'Dini' by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'. **192**

Figure A.26 Gradual Morphing Sequence between a 'Teapot' and a Parametric Surface 'Dini' by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'. **193**

Figure A.27 Gradual Morphing Sequence between a 'Cow head' and a Parametric Surface 'Dini' by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'. **194**

Figure A.28 Gradual Morphing Sequence between a 'Cow head' and a Parametric Surface 'Dini' by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'. **195**

Figure B.1 Reconstruction of Human Face from Slices by Varying (Left) the 'Number of Traversal Intervals' and (Right) the 'Number of Slices'. **197**

Figure B.2 Reconstruction of the Surface of a Cow from Slices by Varying (Left) the 'Number of Traversal Intervals' and (Right) the 'Number of Slices'. **198**

Figure B.3 Reconstruction of the Surface of a Cactus from Slices by Varying (Left) the 'Number of Traversal Intervals' and (Right) the 'Number of Slices'. **199**

Figure B.4 Reconstruction of the Surface of a Teapot from Slices by Varying (Left) the 'Number of Traversal Intervals' and (Right) the 'Number of Slices'. **200**

Figure B.5 Reconstruction of the Surface of "Heart" from Slices by Varying (Left) the 'Number of Traversal Intervals' and (Right) the 'Number of Slices'. **201**

Figure B.6 Reconstruction of a Parametric Surface "Figure-8 Klein" from Slices by Varying (Left) the 'Number of Traversal Intervals' and (Right) the 'Number of Slices'. **202**

Figure B.7 Reconstruction of a Surface called "Schwarz" from Slices by Varying (Left) the 'Number of Traversal Intervals' and (Right) the 'Number of Slices'. **203**

Figure B.8 Reconstruction of a Parametric Surface called "Cross Cap" from Slices by Varying (Left) the 'Number of Traversal Intervals' and (Right) the 'Number of Slices'. **204**

Figure B.9 Reconstruction of a Parametric Surface called "Boy" from Slices by Varying (Left) the 'Number of Traversal Intervals' and (Right) the 'Number of Slices'. **205**

Figure C.1 Shape Transformation between a Teapot, a Super Ellipsoid and "C" by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'. **207**

Figure C.2 Shape Transformation between a Teapot, a Super Ellipsoid and "C" by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'. **208**

Figure C.3 Shape Transformation between Three Tori with Different Radii by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'. **209**

Figure C.4 Shape Transformation between Three Tori with Different Radii by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'. **210**

Figure C.5 Shape Transformation between a Parametric Surface "Roman", a Torus and a Parametric Surface "Cross Cap" by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'. **211**

Figure C.6 Shape Transformation between a Parametric Surface "Roman", a Torus and a Parametric Surface "Cross Cap" by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'. **212**

Figure C.7 Shape Transformation between a Parametric Surface "Klein", an X-shaped Cylinder and a Conic Spiral by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'. **213**

Figure C.8 Shape Transformation between a Parametric Surface "Klein", an X-shaped Cylinder and a Conic Spiral by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'. **214**

Figure C.9 Shape Transformation between 'Cow Head', 'Dini' and 'C' by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'. **215**

Figure C.10 Shape Transformation between 'Cow Head', 'Dini' and 'C' by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'. **216**

Figure C.11 Shape Transformation between 'Cow Head', 'Dini' and 'C' by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'. **217**

Figure C.12 Shape Transformation between 'Cow Head', 'Dini' and 'C' by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'. **218**

Figure D.1 Shape Transformation between Three Tori with Different Radii and a Bent Object by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'. **220**

Figure D.2 Shape Transformation between Three Tori with Different Radii and a Bent Object by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'. **221**

Figure D.3 Shape Transformation between a 'Star', a 'Ellipsoid', a Parametric Surface 'Dini' and a Bottle by Varying the 'Number of Slices' for a fixed 'Number of Traversal Intervals'. **222**

Figure D.4 Shape Transformation between a 'Star', a 'Ellipsoid', a Parametric Surface 'Dini' and a Bottle by Varying the 'Number of Traversal Intervals' for a fixed 'Number of Slices'. **223**

LIST OF ABBREVIATIONS

NBB Normal Bounding Box

OBB Oriented Bounding Box

PEMORFAN DAN TRANSFORMASI RUPA BENTUK 3D MENGGUNAKAN HIRISAN

ABSTRAK

Secara umum istilah 'Transformasi Rupa Bentuk' bermaksud transformasi sebuah objek kepada sebuah objek yang lain. Apabila transformasi rupa bentuk melibatkan dua buah objek iaitu objek sumber dan objek destinasi, dan terdapat pelenyapan rupa bentuk objek sumber kepada objek destinasi secara beransur-ansur, berterusan dan serentak maka transformasi rupa bentuk ini dipanggil 'Pemorfan'. Kebanyakan algoritma transformasi rupa bentuk sedia ada melibatkan banyak campur tangan pengguna, tidak boleh diskalakan dengan rapi apabila bilangan objek input lebih daripada dua dan tidak boleh diskalakan dalam persekitaran perkomputeran selari dan teragih apabila diperlukan. Tesis ini menyentuh isu-isu tersebut dengan mencadangkan algoritma pemorfan 3D yang novel menggunakan hirisan. Pada mulanya objek sumber dan destinasi disusur, dan hirisan dijana daripada penyusuran berkenaan. Penyusuran data tertentu mungkin menjana hirisan yang tidak semestinya selari antara satu sama lain. Bilangan hirisan yang sama dijana bagi kedua-dua objek sumber dan destinasi. Sempadan kedua-dua objek berkenaan diekstrak. Sempadan berkenaan kemudiannya disusur pada satah 2D dan ditransformasikan kepada masing-masing satu awan titik kontur. Interpolasi awan titik kontur sumber dan destinasi yang sepadan dilakukan, dan permukaan yang ditransformasikan dibina semula.

Algoritma berkenaan cukup dinamik dan fleksibel kerana bilangan hirisan dan bilangan penyusuran boleh diubah-ubah. Oleh sebab Kotak Pembatas Berorientasi (KPB) digunakan untuk menentukan penjajaran objek berkenaan, algoritma berkenaan mengambil kira secara

automatik kelengkungan/herotan di dalam sesebuah objek. Algoritma berkenaan melibatkan paling sedikit campur tangan pengguna. Di samping mengikat konsep transformasi rupa bentuk hanya antara objek sumber dan destinasi, kaedah berkenaan juga boleh diperluaskan kepada konsep transformasi rupa bentuk yang melibatkan lebih daripada dua buah objek. Objek yang berbeza boleh juga diadun dalam nisbah yang berbeza dalam output yang ditransformasikan. Sebagai hasil sampingan kerja asas dalam transformasi rupa bentuk, satu algoritma yang novel dalam pembinaan semula permukaan daripada hirisan juga dibangunkan. Kaedah berasaskan hirisan berkenaan juga ditunjukkan dapat diskalakan dengan rapi dalam persekitaran selari/teragih. Akhir sekali, ditunjukkan juga bagaimana mudahnya kaedah berasaskan sifat sepenuhnya boleh digabungkan dalam algoritma berkenaan. Kekompleksan algoritma berkenaan juga dianalisis dan hasilnya menunjukkan prestasi yang lebih baik berbanding dengan algoritma transformasi rupa bentuk yang lain.

3D MORPHING AND SHAPE TRANSFORMATION USING SLICES

ABSTRACT

Generally the term 'Shape Transformation' means transformation of one object into another object. When shape transformation involves two objects i.e. source object and destination object and there is gradual, continuous and simultaneous dissolvement of the shape of source object to destination object, it is called 'Morphing'. Most of the existing shape transformation algorithms involve a lot of user intervention, do not scale well when the number of input objects is more than two and are not scalable in parallel and distributed computing environment whenever needed. This thesis addresses the above-mentioned issues by proposing a novel 3D morphing algorithm using slices. Firstly source and destination objects are traversed and slices are generated from this traversal. Traversal of a particular data may generate slices which may not be parallel to each other. Equal number of slices is generated for both source and destination objects. Boundaries of both objects are extracted. The boundaries are then traversed on a 2D plane and are transformed into a cloud of contour points each. Interpolation of the corresponding source and destination contour point clouds takes place and transformed surface is reconstructed.

The algorithm is dynamic and flexible enough as the number of slices and the number of traversals can be varied. As Oriented Bounding Box (OBB) is used to determine the initial alignment of the object, the algorithm automatically takes into account of the curvature/distortion within an object to a certain extent. The algorithm involves the least amount of user intervention. Instead of binding the concept of shape transformation only between 'source'

and ‘destination’ objects, the method is also extended to the concept of shape transformation involving more than two objects. Different objects can also be blended in different proportions in the transformed output. As a spin-off of the basic work on shape transformation, a novel algorithm on surface reconstruction from slices is also developed. It has also been shown that this slice-based method scales well in parallel/distributed computing environment. Lastly, the ease in which fully feature-based method can also be incorporated in the algorithm is shown. Complexity of the algorithm is also analysed and the result shows better performance in comparison with other shape transformation algorithms.

CHAPTER 1

INTRODUCTION

1.1 Overview and Motivation

Generally the term 'Shape Transformation' means transformation of one object into another object. Shape transformation may involve only one object as through some manipulation, the shape of an object can be changed [1]. Figure 1.1 shows how a box is deformed by user-defined warping. When shape transformation involves two objects i.e. source object and target object, and there is gradual, continuous and simultaneous dissolvement of the shape of source object to its target, it is called 'Morphing'. Shape Transformation can also be defined as incorporating the overall characteristics of a number of different objects in one output or incorporating the characteristics of different objects in different proportions in one output. Hence shape transformation can involve more than two objects and morphing can be defined as a subcategory of shape transformation when the number of objects is confined to two.

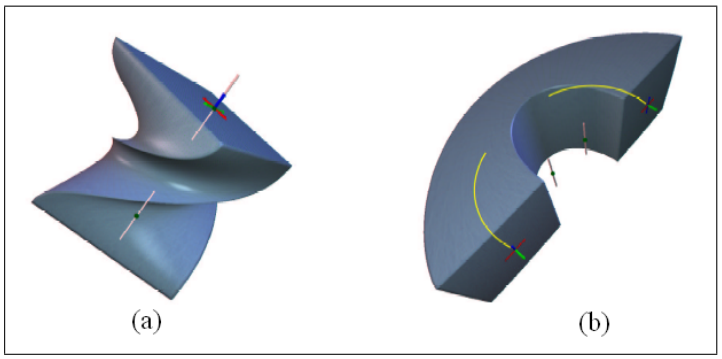

Figure 1.1: A box (a) Twisted and (b) Bent [1].

Starting from the late eighties numerous algorithms on shape transformation have been developed. Shape transformation algorithms mainly fall into two major categories:

1. Surface-based shape transformation and

2. Volume-based shape transformation.

In surface-based methods, data partitioning takes place along the surface of the object whereas in volume-based methods data partitioning happens across the surface of the object. Figure 1.2(a) illustrates data partitioning in surface-based shape transformation while Figure 1.2(b) illustrates data partitioning in volume-based shape transformation.

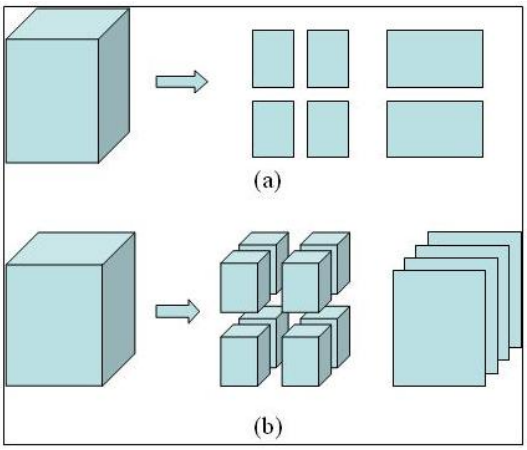

Figure 1.2: Data Partitioning in (a) Surface-based Shape Transformation and (b) Volume-based Shape Transformation.

Surface-based shape transformation involves dividing the surface of both source and destination objects into a number of corresponding morphing patches ([2], [3], [4], [5], [6], [7], [8], [9], [10], [11]). Surface-based approach uses user-defined control fields such as point fields and line fields during transformation to map key features of the input objects. Surface-based methods are important because of their ability to morph between objects of different types of genus, but these methods also require a significant amount of user intervention. Another troubling feature of surface-based methods is the problem of self-intersection. It cannot guarantee that polygonal surfaces will not pass through each other, creating self-intersecting intermediate result as found in [3]. Surface-based methods also do not extend well when shape transfor-

mation involves more than two objects because of the need for excessive user intervention. Requirements of too much interprocess communication have also made surface-based methods totally inadaptable to parallel as well as in distributed computing environment.

Volume-based shape transformation modifies voxel values of a dataset. Volume-based shape transformation can be as simple as cross dissolving method ([12], [13]) where no user intervention is involved. Both datasets are just merged to get the intermediate output without considering warping/distortion within the objects. The method firstly transforms volume data from spatial domain to frequency domain, then interpolates the volume in frequency domain and again transforms back to spatial domain.

User intervention is required when special care is needed to be taken such as warping within a volume. Sometimes user-guided warp is applied on the input objects so that warping is also exhibited in the transformed objects [14]. To take into account of warping in volume-based shape transformation, equal number of disk fields is applied on each input object with each disk having its own normal direction [15]. Shape transformation using 'distance field' method [16] or 'level set' method [17] measures the distance of each corresponding surface voxels within a volume. Shape transformation using implicit functions [18] dissolves a 3-D object into a number of 2D implicit functions. Lee et al. [19] use morphology-based scheme to perform 3D volume morphing by creating morphological difference between source and target objects. These methods involve a little or no user intervention, but at the same time fail to take care of warping in a rigid body. Also when more than two objects are involved in shape transformation, these methods may fail to properly preserve the features of all input objects in the transformed output. These methods may not also extend well when the number of input objects is more than two, and may not be scalable in parallel/distributed computing environment.

From the above discussion, it is obvious that volume-based approach has got some advantages over surface-based approach though each approach has its own advantages. Volume-based shape transformation involves less user intervention although in order to preserve the rigidity within a volume, a certain amount of user intervention is needed. It is imperative that we strive to develop a new 3D morphing algorithm which optimises user input, automatically considers orientation/distortion of rigid body during morphing, preserves smooth transition between source and destination objecs, extends to the concept of shape transformation using more than two objects and is adaptable to parallel/distributed computing environment in order to take into account of faster generation of shape transformation sequences.

1.2 Problem Statements

From our survey we have found that

- Surface-based methods involve a lot of user intervention in comparison with volume-based methods;
- Existing shape transformation algorithms do not consider the orientation/distortion in a rigid body without user intervention;
- Most of the existing methods do not scale well as the number of input objects increases;
- Most of the methods are not extendible in parallel/ distributed computing environment.

Hence we are interested in designing a novel algorithm which involves the least amount of user intervention and at the same time can take into account of the features of the input objects i.e. warping/distortion whenever it is needed. The method should also be easily scalable as the number of objects increases. Also in order to achieve faster generation of morphing sequences, the method should be extendible in parallel as well as in distributed computing environment.

1.3 Objectives

From our survey we have found that volume-based methods involve less user intervention. Also from our observation we believe that among the volume-based approaches, slice-based approach ([15], [18]) is more efficient and flexible than other approaches ([14], [17]). Orientation as well as distortion can also be easily considered with slices as each slice can have its own normal/orientation depending on how it is cut across. Slices also scale well depending on the number of objects as well as their complexity. Hence our main objective is to design a new 3D morphing algorithm using slices which:

- optimises user input;
- considers orientation/distortion of rigid body during shape transformation;
- considers smooth transition between source and target during morphing;
- is dynamic and simple enough to be extendible to the concept of shape transformation involving more than two objects;
- is adaptable to the parallel/distributed computing environment in order to take into account of realtime shape transformation;
- is able to work as a fully feature-based volume morphing algorithm with the incorporation of a little amount of user intervention.

Besides the above mentioned objective, we aim to investigate and develop any associated algorithms such as an algorithm for surface reconstruction for our transformed output which is more suitable for our approach rather than using any other existing algorithms for surface reconstruction.

1.4 Research Scope and Limitations

We wish to highlight that we are in this thesis mainly concerned with designing a novel 3D shape transformation algorithm which can automatically consider orientation/distortion in rigid body to a certain extent. The proposed shape transformation method is a featureless volume-based shape transformation method. Although this method is capable of automatically considering some features (orientation, distortion, empty spaces etc.) of both source and destination objects and we have shown how feature-based shape transformation can also be incorporated in our approach at ease, full exploitation of our proposed method on feature-based shape transformation is recommended as one of our future work.

Among graphics properties, we have only considered 'colour' in our implementation. For the sake of simplicity, other graphics properties such as 'texture' and 'opacity' have not been considered in the implementation. It is asssumed that these graphics properties can also be incorporated in the same way as 'colour' is incorporated during implementation.

Again, for the sake of simplicity, only linear interpolation has been used in our implementation during interpolation of source and destination objects and the result is found to be quite satisfactory. Other interpolation techniques can also be considered in the implementation.

In our research, we also have designed a new method for surface reconstruction using slices. This new method for surface reconstruction was mainly developed to reconstruct surface of interpolated point clouds during morphing where accuracy during surface reconstruction is not so stringent. However, this method has been tested on a number of complex objects and shown to be quite sound as a standalone algorithm for surface reconstruction.

1.5 Methodology

Our methodology consists of a number of steps. A flowchart of the methodology is given in Figure 1.3. Firsly, an extensive literature review on 3D morphing is carried out and good criteria for 3D morphing are identified. These criteria include:

- Less amount of user input;
- Smooth transition between source and target;
- Ease of implementation;
- Capability to consider distortion/orientation of the input objects and
- Ease of extendibility.

At this stage some of the possible extensions of the algorithms such as possibility of involving more than two objects in shape transformation and ease of parallel morphing are also chalked out.

It is found that shape transformation using slices is more dynamic and adaptable in nature than other shape transformation algorithms. In order to extract slices from a certain object, the object needs to be cut across. Hence shape transformation using slices is able to exhibit the properties of a volume-based method. Equal number of slices is generated from each object. Slices may be parallel to each other or may have different orientations. Traversal of a data along its longitudinal direction may generate slices which are differently oriented from each other. This helps in considering the distortion in a rigid body to some extent. Hence our basic implementation is a 3D morphing algorithm using slices which

- considers all the above-mentioned good criteria;

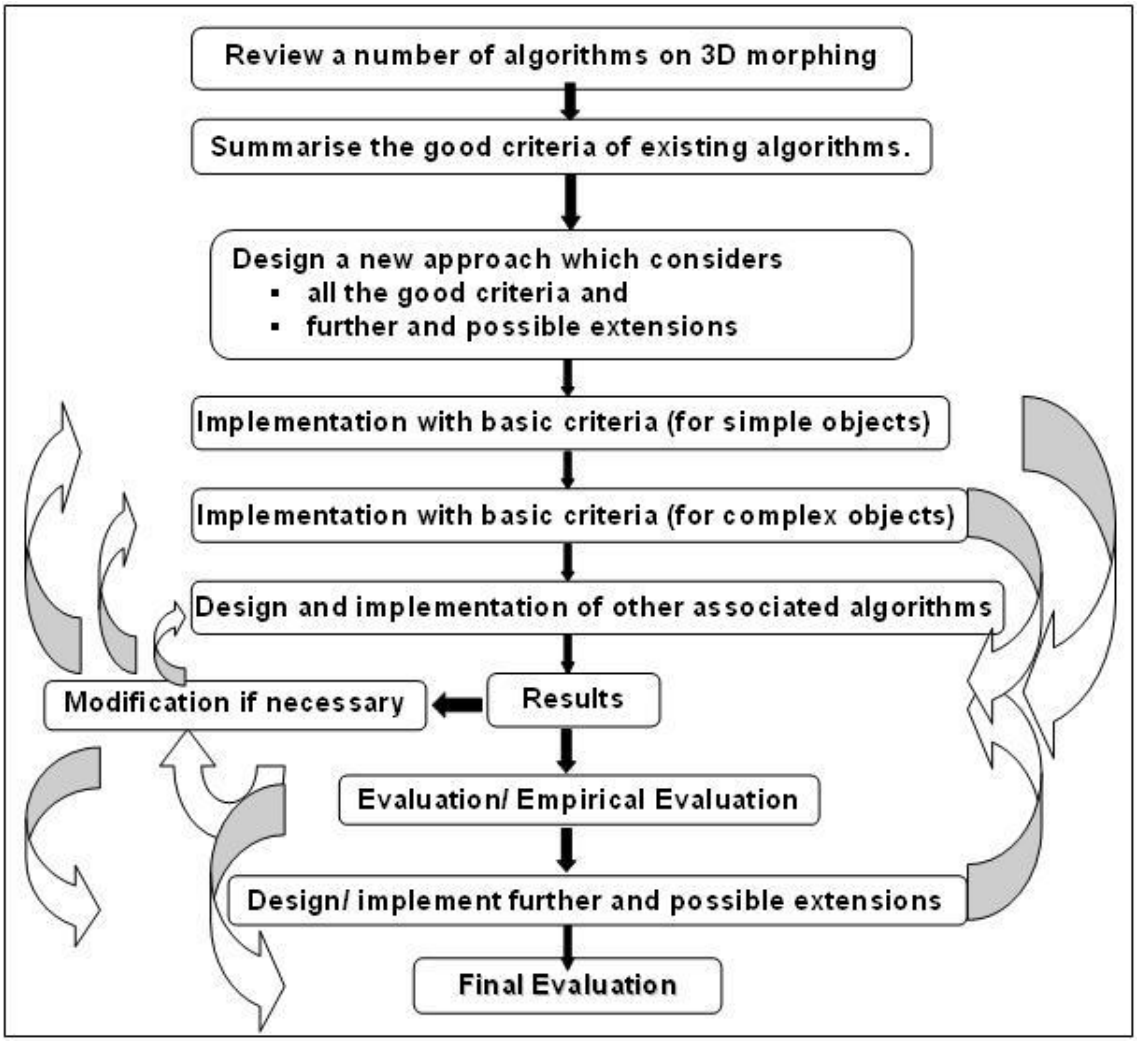

Figure 1.3: Various Steps of Research Methodology.

- is dynamic enough to be extendible to the concept of shape transformation using multiple objects;
- is scalable in parallel/distributed computing environment, and
- is extendible to feature-based volume-morphing algorithm.

The basic algorithm is on morphing which involves two objects i.e. source object and destination object. The basic algorithm is implemented first. The implementation consists of two steps: firstly very simple objects (objects without any hole or empty space) are considered and later, this simple implementation is gradually evolved so that it can consider complex objects (objects with holes, empty space or with any irregular configuration) too. Evaluation is

performed by carrying out extensive testing on different kind of objects and by varying different parameters to prove the flexibilty of the method. While implementing the basic algorithm, some associated algorithms such as a slice-based surface reconstruction algorithm is also developed. Later the basic algorithm is extended so that it can also work for shape transformation involving more than two objects. In these cases evaluation is done in the same way as for the basic algorithm. The basic algorithm, its extended version as well as other associated algorithm are also evaluated empirically. Other possible extensions of the proposed algorithm such as exploitation of the algorithm in parallel/distributed computing environment and incorporation of fully feature-based shape transformation in our method are also explored.

1.6 Applications and Benefits of the Research

Volume morphing and shape transformation can be applied in a number of fields including entertainment, industrial design, medicine and manufacturing. Nowadays morphing has become an indispensable tool in 3D animation industry. When shape transformation involves more than two objects, it can be used for creating special effect in synthetic model design. This can also be useful in inventing a variety of shapes in ceramic industry as well as in designing perfume bottles and other showpieces.

Shape transformation can also be applied in a number of areas in medical as well as geological sciences. Shape transformation from slices can also be used in medical science for surface recovery, reconstruction of joint in human body and volume calculation of human organs. In geological field, shape transformation is useful in developing terrain model of a particular area from a number of contours.

1.7 Contributions

This thesis has three principal contributions. Firstly we present a novel volume morphing algorithm based on slices. The speciality of the algorithm is that it involves less amount of user intervention and at the same time this automated method can take into account of warping in input objects to a certain extent. Initial orientation of the input objects plays an important part of the method. Object can be oriented along any of the principal axes or can be oriented along any of the directions of the eigen vectors or can be aligned along any particular direction intially. Depending on this initial orientation of the input objects, a number of different morphing sequences can be generated for the same input objects.

Our second major contribution is the development of a new surface reconstruction algorithm while working for the construction of surface of morphed objects based on slices. Although some heuristic approach has been adopted in the design, this method still works well as a standalone algorithm for surface reconstruction of fairly complex objects.

Our third major contribution is the extension of the basic algorithm so that the number of input objects can be more than two. Instead of binding the concept of shape transformation between two objects, it has been tested on three or more input objects. In some cases the overall characteristics of all the objects are blended in one transformed output and in another case different objects are blended in different proportions in the transformed output.

Besides these three principal contributions, we also focus our attention on parallel morphing. Our algorithm has been designed in such a way that it can scale well in parallel computing environment as simple data structure of sliced body is suitable for both data as well as functional partioning. Lastly we have shown how fully feature-based morphing/shape transformation can also be incorporated in our algorithm.

1.8 Thesis Organisation

The remainder of this thesis is organised as follows:

- Chapter 2 provides a survey of existing work on morphing and shape transformation which includes a comparison on the existing shape transformation methods.
- Chapter 3 firstly describes a brief overview of the basic algorithm and its various steps. Then it describes the various steps of the basic algorithm in detail and also presents some results and evaluation.
- Chapter 4 describes a new algorithm for surface reconstruction based on slices. This is described and evaluated as a standalone algorithm for surface reconstruction.
- Chapter 5 extends the basic algorithm to incorporate more than two input objects in shape transformation. It also provides some outputs from the implementation and evaluation of the extended version.
- In Chapter 6, we discuss the ease of implementing parallel morphing and parallel shape transformation with our algorithm. Ease of incorporating feature-based volume morphing with our method is also discussed in Chapter 6.
- Finally in Chapter 7, we conclude this thesis and suggest some future work.

CHAPTER 2

LITERATURE REVIEW

This chapter discusses related work on shape transformation. We begin this chapter with a classification on existing algorithms on shape transformation. As mentioned in the previous chapter when shape transformation involves two objects, i.e. source object and target object, and when a gradual sequence of transformation between source and target objects is generated, it is called morphing. So whenever we use the term 'morphing', it signifies 'shape transformation' involving two objects. In this chapter, firstly literature review is carried out bearing in mind that a morphing algorithm which possesses most of the characteristics of a good morphing algorithm is to be developed. Existing morphing techniques are classified and reviewed, and their strengths as well as weaknesses are identified. Then related work on shape transformation involving more than two objects are considered. Surface reconstruction also plays an important part in morphing. Hence existing algorithms on surface reconstruction are also reviewed.

2.1 Classification

Depending on the various approaches, existing morphing algorithms can be classified into the following categories:

1. Surface-based Morphing and
2. Volume-based Morphing.

Surface-based morphing consists of continuous mapping of small pieces of polygonal surfaces of source object to those of target object whereas volume-based morphing modifies voxel values of a volume dataset for smooth transition between source and target shapes. In surface-based approach, data partitioning takes place along the surface of the object whereas volume-based morphing cut the object into slabs, slices or blocks. This may cause the data to be unstructured.

According to the control fields applied in morphing, morphing methods can also be classified as follows:

1. Cross Dissolving;
2. Field Morphing and
3. Mesh Morphing.

Cross dissolving method just merges the volumes. It does not require any control datasets. In field morphing, key features of each object are defined using a set of control fields such as point fields, line fields and disk fields. In mesh morphing, source and target datasets are first subdivided into equal number of meshes and then mapped.

In the following sections, we have mainly tried to categorise the existing morphing algorithms into surface-based approach or volume-based approach. Then while discussing each of these methods, we identify whether it uses cross dissolving technique, field morphing or mesh warping.

2.2 Surface-based Approach

Surface based approach uses user-defined control fields such as point fields and line fields during morphing process. The first step in surface-based approach is to establish correspondence between each point of the source and target objects. Then interpolation technique comes into consideration. Finally surface is reconstructed from the morphing patches.

Hong et al. [3] present the simplest surface-based morphing algorithm based on calculating minimal dynamic distance between corresponding edges of source and target objects. 'Minimal Dynamic Distance' is the distance between the vertices of the corresponding edges that can be rearranged or reversed when necessary in order to find the minimum distance between the edges. Linear interpolation is performed between the corresponding vertices. Lastly surface is constructed by triangulation method. But where objects are a bit complex, this method fails.

Kent et al. [5] consider only star-shaped polyhedrons. A star polygon is a polygon for which there exists a point i.e. 'star point' inside the object from which all other points are visible. Correspondence between source and target objects is established by first projecting the topology of both objects onto a unit sphere. Once both models have been projected, topologies of both models are merged by clipping the projected faces of one object onto the projected faces of the other. Each projected edge of one object intersects with the projected edge of the other object (Figure 2.1). Corresponence between the vertices are established by finding the intersection pairs. For interpolation purpose, linear as well as hermite interpolation technique is used. After interpolation, facets are triangulated and surface reconstruction is performed. However this method may fail when object is overly complex as inaccurate ordering of the pairs of intersection points can generate improper result. Another shortcoming with this method is that it only considers star-shaped polyhedral objects.

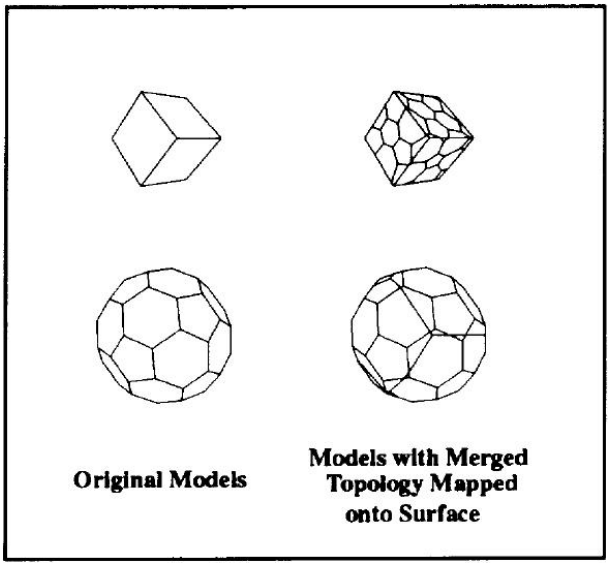

Figure 2.1: Projection of One Object onto the Other Object [5].

Gregory et al. [2] establish correspondence between source and target objects by specifying user-defined corresponding features for both source and target objects. Based on these features, both source and target objects are decomposed into the same number of morphing patches. Corresponding morphing patches are mapped to a 2D polygon and their topologies are merged. From the merging of each pair of morphing patches, new morphed facet is generated. 'Morphing trajectories' are formed in between each feature pair and each of the merged 2D polygons is placed across that trajectory. When all the corresponding morphing patches are merged, a morphed surface is formed. This approach allows a broad class of object to be morphed, however user needs to outline the entire network.

Lazarous et al. [6] create a common parameterised mesh for both source and target during sampling process. Firstly two axes are defined: one for each of the source and target objects so that objects are star-shaped around the axes. Shape transformation is done by dividing each object into three parts: two hemispherical end parts and one cylindrical middle part. Later these three portions are converted into three sheets and vertices of both source and target objects are projected onto the sheets in parametric domain. Sharp edges and conic points from each object are adjusted onto the sheet. Interpolation of the corresponding vertices is performed

in local coordinates. Vertices are linearly interpolated. Both axes of source and target are also interpolated. For axes, linear interpolation is performed for the coordinates as well as quaternion interpolation is performed for the rotation. Interpolated vertices are placed along the radii of the interpolated axes and mapped. This method is capable of considering the warp in rigid body.

Lee et al. [7] use a small set of user-defined point fields and line fields to establish primary correspondence between source and target objects. Based on these control fields, a coarse-grained base map is constructed for both source and target objects which is called 'Base Domain' (Figure 2.2). Base domains for both source and target are constructed by successive removal of independent points while keeping the feature points intact. Next source base domain is projected onto target base domain and aligned with target base domain. Once source base domain is mapped with the target base domain, any point in the orginal source object can also be mapped to the target. Fine-grained mesh of original source and target objects is also constructed. Then any point from fine-grained source mesh can be mapped to any point on fine-grained target mesh through their base domains. This is done in a multiresolution fashion. Next corresponding mapped points from source and target are interpolated and then a morphed mesh is created.

Kaul and Rossignac [4] compute the Minkowski sum of both source and target objects for interpolation purpose. This algorithm is based on a set of propositions such as: "Any face generated from the Minkowski sum of two convex polyhedra A and B may be the Minkowski sum of vertices of A with face of B, face of A with vertices of B, edge of A with edge of B". This proposition is depicted in Figure 2.3.

There are several other propositions. Minkowski sum combines the shape characteristics of both source and target objects based on these fixed set of propositions. A pair of polyhedra is

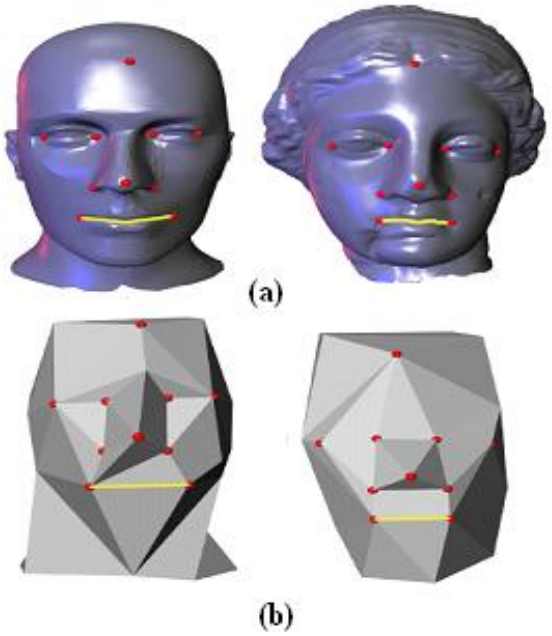

Figure 2.2: (a) Original Source and Target Objects and (b) Conversion of Source and Target Objects to their Base Domain Using Point and Line Control Fields [7].

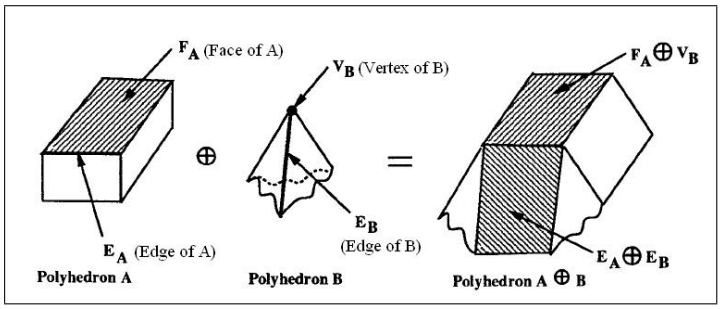

Figure 2.3: Minkowski Sum Combines the Shape Characteristics of its Arguments A and B [4].

transformed by computing the weighted Minkowski sum and a morphing sequence is generated by gradually scaling down one model from 100 percent to 0 percent while simultaneously scaling up the other from 0 percent to 100 percent. This method works for convex objects only. Also dependence on a fixed set of principles has made this algorithm a bit rigid.

Yan et al. [8] firstly decompose each object into isomorphic meshes on the basis of a selected number of corresponding inner points. Then corresponding feature points are identified and both objects are divided into several equivalent patches. Thus a two-level correspondence is used to map source and destination objects. Each pair of patches are bijectively mapped. The

deformation of an object is defined by its strain-field which can be derived from the object's displacement field or position field. Strain-field interpolation is used to map the corresponding patches to avoid shape squeezing. But this method involves a lot of user input and does not scale when shape transformation involves more than two objects.

Huang et al. [9] use the above-mentioned two-level correspondence between source and target. For embedding and merging of both corresponding patches in a 2D plane, a novel technique called 'Structures of Minimal Coverage' is used to avoid self-intersection or fold-overs. Figure 2.4(a) shows the case where no intersection happens on projection of a morphing patch onto a 2D plane whereas self intersection takes place on projection of the patch onto a 2D plane in Figure 2.4(b). 'Structures of Minimal Coverage' classifies the position of edges of source patches according to the overlay of the starting and the ending points of the edges on the target patches and vice versa to avoid fold-over or self-intersection.

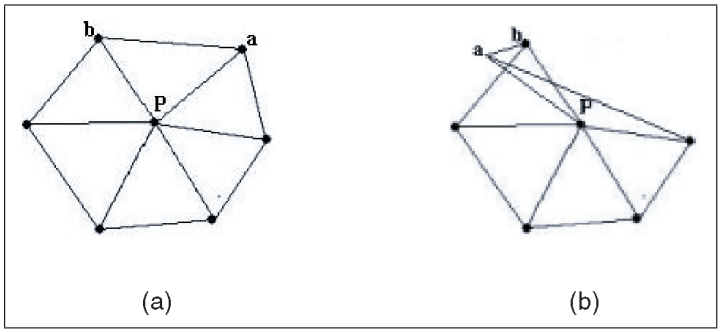

Figure 2.4: (a) No Self-intersection and (b) Self-intersection on Projection of a Morphing Patch onto a 2D Plane (Adapted from [9]).

Later a better version of the above work was chalked out where instead of embedding/merging, some operations like vertex removing, vertex splitting and edge swapping are performed to dynamically add or remove vertices among the corresponding morphing patches and hence to generate a smooth and gradual connectivity from the source object to the target object [10]. This method can only consider convex polyhedral objects and does not scale well for shape

transformation involving more than two objects.

Parent et al. [11] describe shape transformation process from contours. In this method both source and target objects have equal number of contours. Correspondence in established by ray-firing: two points are assumed to be corresponding if they are intersected by the same ray. Before ray firing, each shape is mapped to its convex hull first. Each convex hull is then mapped to a unit square by scaling the hull so that its bounding box coincides with the square. Corresponding vertices are matched on the basis of having the minimum distance between them. If a vertex has more than one correspondence, all of its correspondences are visited first to find out the "best match" which is found out by comparing distances. For the rest of the vertices, new vertices are inserted into the contour which has less number of vertices. At the end of preprocessing, each corresponding contour has unique match for its vertices. After correspondence processing, the shapes are averaged by averaging corresponding points. Three types of averages can be found: the mean, the median and the mode. To process more than two shapes, a base contour is chosen which has got the maximum number of vertices. Then the "base contour" is mapped with the other corresponding contours in the same way as described above. This method only applies to simple objects and it requires too much user intervention especially for correspondence preprocessing.

2.3 Volume-based Approach

The simplest volume-based approach is the cross-dissolving method [12] which involves no control fields. This method at first transforms volume data from spatial domain to frequency domain by Fourier transform, then linearly interpolates volume in frequency domain and again transforms back to spatial domain. To enhance the smoothness of the in-between volumes, Fourier transform has been used by gradually removing high frequencies of source model, interpolating over to the low frequencies of the second model and smoothly adding in the

high frequencies of the second model. However, Fourier transform does not localise in spatial domain. In order to have a smooth transition, voxel values of the entire volume are modified according to the distance of the nearest iso-surface.

The above-mentioned problem can be alleviated by Wavelet transformation [13], which localises both in frequency and spatial domain. Hence decomposition and reconstruction processes are carried out in a multi-resolution fashion so that frequency can be adjusted to the desired level. Different interpolation schedules are applied at different resolution resulting in smooth approximation of intermediate model between source and target. Figure 2.5 shows the difference between Fourier and Wavelet transforms through graphical representation.

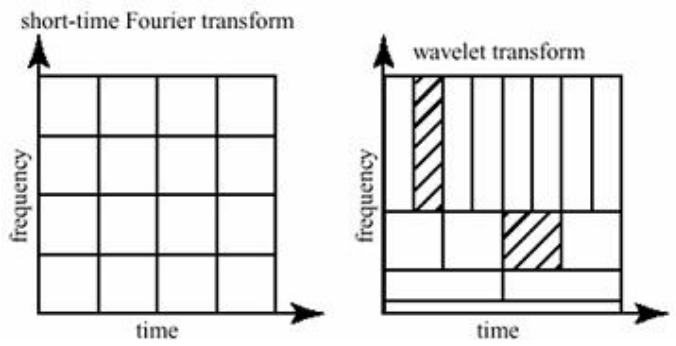

Figure 2.5: Graphical Representation of Fourier (Left) and Wavelet (Right) Transforms [20].

However, both methods have difficulties in specifying slightly complex geometric transformations such as object rotation. By relying on the frequency information, the methods would also have difficulties while associating with some scalars such as colours, opacities and textures. This problem can be alleviated by applying warping before interpolation as found in [14]. Firstly pairs of corresponding feature elements of both source and target volumes are identified by the user. Feature element pairs can be defined as pairs of points, segments, rectangles, boxes and so on. If necessary, user-defined warp is applied on the feature elements of source object and vice versa to resemble each other. Next the warped source and target volumes produced from the original source and target volumes are blended either by linear or non-linear

cross dissolving method. The flowchart of this process is depicted in Figure 2.6. This method has been exploited in distributed computing environment [21], but too much communications are required among the processors.

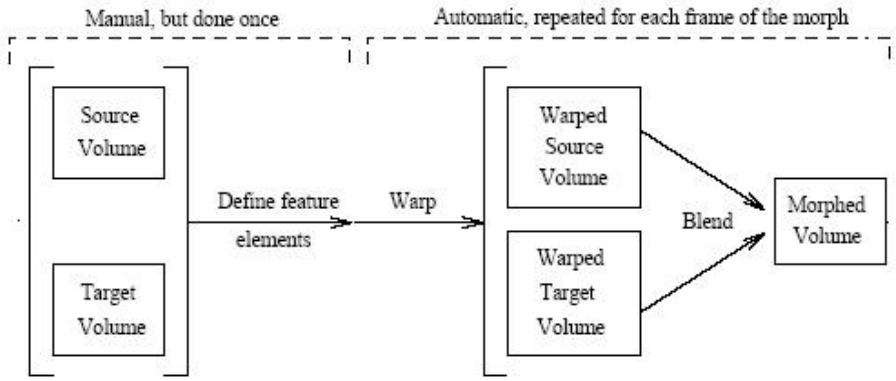

Figure 2.6: Data Flow in Feature-based Volume Morphing (Adapted from [14]).

Instead of using point, line or box control fields in 3-D volume morphing, user-specified disk field can also be used [15]. A disk field is established with a centre point 'c', a normal vector 'N' and a radial vector 'R'. Pairs of corresponding disk fields for source and destination objects in local cylindrical coordinate are used to control the distortion from a source volume to a destination volume. Later the corresponding coordinates are transformed back to the Euclidean space. All kind of transformation i.e. translation, scaling (1D, 2D, 3D), rotation and skewing are performed in disk space. The normal vector 'N' of an inbetween disk field from a pair of starting and finishing disk fields is obtained by linear interpolation of the lengths of normals and their inbetween angles. Figure 2.7 shows the use of disk field on a particular 3D object. Parallelising of this method is also mentioned as possible future work [15].

Payne et al. [16] calculate 'Distance Field' of input objects. A distance field, or a distance volume, is a volume dataset where the value stored at each voxel is the shortest distance to the surface of the object being represented by the volume. The value of the distance field is considered 'positive' inside the object, 'zero' on the surface and 'negative' outside the object. Once

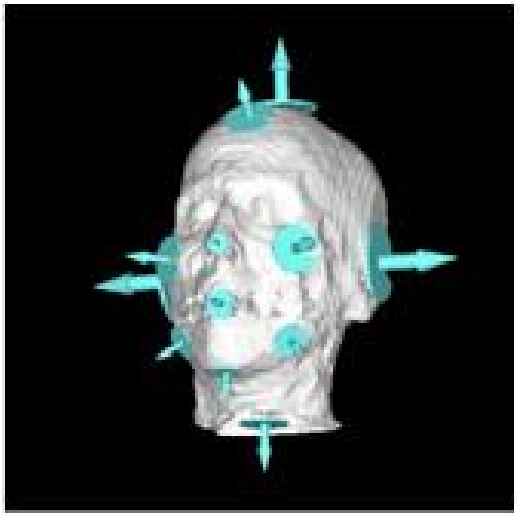

Figure 2.7: Use of Disk Fields on 3D Object in Volume Morphing [15].

distance field is created for both source and target objects, correspondence between distance values are established and distance values are interpolated using linear interpolation method or boolean operation. Payne et al. [16] also introduce the concept of averaging a number of objects (more than two) using boolean operators but no implementation was carried out.

Breen et al. [17] divide the surface of each object into a number of levels (iso-surface). Both source and target objects can be defined as 'level-set' of distance values. Firstly, source volume is initialised so that its furthest level set approximately aligns with that of target object. User-defined warp can be applied at this stage on both source and target objects. Connection is established between the way in which points on the surface moves. Every point on the source surface moves in the direction of the normal at that point with a velocity proportional to the signed distance at that point in 3-D space from target surface. Those part of source that are outside of the target contracts whereas the inside part moves in the direction of surface normals and expands (Figure 2.8). Parallelising of this level-set method of morphing was also suggested as possible future work.

Shape transformation using variational implicit function [18] is also an improved version of 'Distance field' method but here implicit function is created for both source and target objects

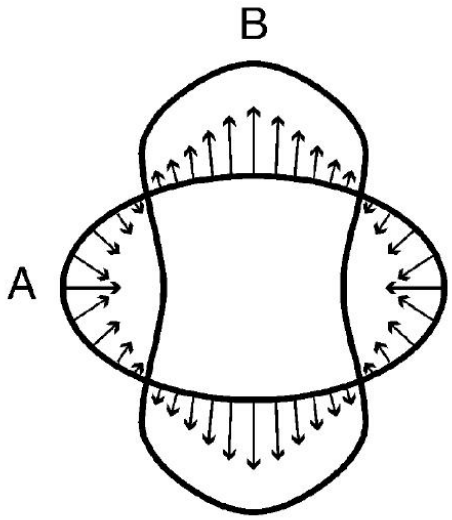

Figure 2.8: Movement of the Source Object ‘A’ along its Surface Normals to Fit the Target Object ‘B’ [17].

with distance values, surface normals at those distance points as boundary constraints. Firstly, equal number of 2D implicit functions are created for both source and target objects from their distance values and surface normals as boundary constraints. These functions are created at particular interval along the height of source and target. Each pair of corresponding implicit functions are interpolated resulting in a function that takes on ‘0’ at zero-value constraints and positive in the direction of normal constraints (Figure 2.9). Later, surface is reconstructed considering all these constraints from all of the interpolated functions creating a single 3D implicit function. Intermediate shapes are generated by interpolating corresponding 2D implicit functions only. Warping is another degree of control that may be added to shape transformation technique. Variational interpolation in implicit function creates plausible function values in places between boundary constraints.

In morphology-based volume morphing, firstly morphological difference between source and target objects is created [19]. In Figure 2.10, morphological difference is represented by regions ‘I’ and ‘II’. Next dilation operation is applied only to the morphologically different part i.e. regions ‘I’ and ‘II’ instead of the entire volume space. Lastly erosion-based interpolation is performed to interpolate regions ‘I’ and‘III’ and regions ‘II’ and‘III’. But this method is a bit

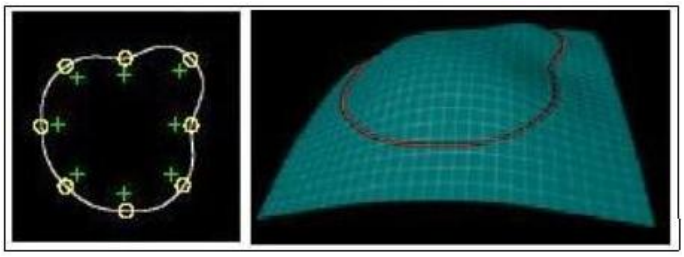

Figure 2.9: (Left) Pairs of Boundary and Normal Constraints Defined as Circles and Pluses and (Right) Resulting Variational Implicit Function as a Height Field (Adapted from [18]).

rigid and does not consider the curvature in rigid body. This method will not scale well when shape transformation involves more than two objects. In this case this method may result in undesirable outputs.

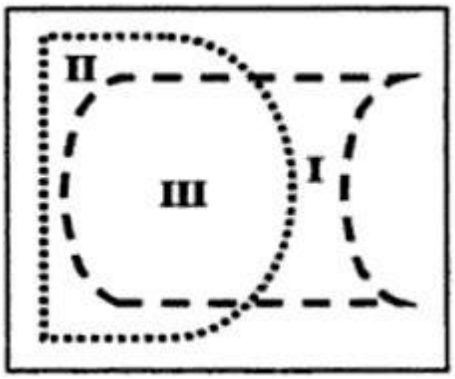

Figure 2.10: Morphological Difference between Two Images is Defined by Regions I and II while Region III is the Overlapping Region [19].

2.4 Shape Transformation Involving more than Two Objects

Very few work have been carried out on shape transformation involving more than two objects. As mentioned earlier, Parent et al. [11] first introduced the concept of averaging more than two objects. But implementation was carried out on some simple objects only. Pasko et al. [22] use boolean operators for blending a number of objects with the help of a third object defined as 'Bounding Object' (Figure 2.11). Bounding object helps in generating control portions within the input objects which are to be blended. Blending is performed by using set-theoretic operations such as intersection, union and difference in control portions of the input objects.