

STATUS OF THESIS

Title of thesis

VISIO-HAPTIC DEFORMABLE MODEL FOR HAPTIC
DOMINANT PALPATION SIMULATOR

I, IRSHAD AHMAD,

hereby allow my thesis to be placed at the Information Resource Center (IRC) of
Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

Confidential

Non-confidential

If this thesis is confidential, please state the reason:

The contents of the thesis will remain confidential for _____ years.

Remarks on disclosure:

Endorsed by

Signature of Author

Signature of Supervisor

Permanent address: Vill: Palosi Maqdarzai,
Teh/Dist Peshawar, P.O Peshawar,
Khyber Pakhtoonkhwa, Pakistan

Name of Supervisor
Dr. Suziah Sulaiman

Date : _____

Date : _____

UNIVERSITI TEKNOLOGI PETRONAS
VISIO-HAPTIC DEFORMABLE MODEL FOR HAPTIC DOMINANT
PALPATION SIMULATOR

By

IRSHAD AHMAD

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance this thesis for the fulfillment of the requirements for the degree stated.

Signature: _____

Main Supervisor: Dr. Suziah Bt. Sulaiman

Signature: _____

Co-Supervisor: Assoc. Prof. Dr. Dayang Rohaya Bt Awang
Rambli

Signature: _____

Head of Department: Dr. Jafreezal Bin Jaafar

Date: _____

VISIO-HAPTIC DEFORMABLE MODEL FOR HAPTIC DOMINANT
PALPATION SIMULATOR

By

IRSHAD AHMAD

A Thesis

Submitted to the Postgraduate Studies Programme

as a Requirement for the Degree of

DOCTOR OF PHILOSOPHY

INFORMATION TECHNOLOGY

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SERI ISKANDAR,

PERAK

SEPTEMBER 2013

DECLARATION OF THESIS

Title of thesis

VISIO-HAPTIC DEFORMABLE MODEL FOR HAPTIC
DOMINANT PALPATION SIMULATOR

I IRSHAD AHMAD,
hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

Signature of Author

Signature of Supervisor

Permanent address: Vill: Palosi Maqdarzai,
Teh/Dist Peshawar, P.O Peshawar,
Khyber Pakhtunkhwa, Pakistan

Name of Supervisor
Dr. Suziah Sulaiman

Date : _____

Date : _____

DEDICATION

I dedicate this research work to my beloved *parents*, *wife*, and *children* who have always been a great moral and practical support in my life.

ACKNOWLEDGEMENT

First and foremost, I would like to thank the Almighty Allah, the most beneficent, the most merciful, and the most gracious for His guidance and providence. I made it possible to come up with this thesis due to the Blessings of Almighty Allah.

It is a great honour and pleasure for me to express my gratitude to my supervisor Senior Lecturer Dr. Suziah Binti Sulaiman for their kind supervision, encouragement, valuable suggestions, and enlightening discussions to conduct this study. She kept me on track during my entire period of PhD and gave me the courage and strength at times when I was giving up. I am also very thankful to my co-supervisor Assoc. Prof. Dr. Dayang Rohaya Binti Awang Rambli for her kind cooperation and support. I am very obliged to Universiti Teknologi PETRONAS, Malaysia for providing me fully funded scholarship to pursue my PhD studies and excellent environment.

Words cannot express the feelings I have for my parents for their love, support, and pray for my success and patience throughout this study. In addition, I am very thankful to my wife for her patience, encouragement, support and taking care of my children when I was busy in my research work and experiments. I would like to acknowledge my lovely daughters Bi Bi Khola, and Bi Bi Barira and my cute little son Atta Ur Rehman, the asset of my life.

ABSTRACT

Vision and haptic are two most important modalities in a medical simulation. While visual cues assist one to see his actions when performing a medical procedure, haptic cues enable feeling the object being manipulated during the interaction. Despite their importance in a computer simulation, the combination of both modalities has not been adequately assessed, especially that in a haptic dominant environment. Thus, resulting in poor emphasis in resource allocation management in terms of effort spent in rendering the two modalities for simulators with realistic real-time interactions. Addressing this problem requires an investigation on whether a single modality (haptic) or a combination of both visual and haptic could be better for learning skills in a haptic dominant environment such as in a palpation simulator. However, before such an investigation could take place one main technical implementation issue in visio-haptic rendering needs to be addressed. Implementing a real-time interaction together with its proper synchronization without using any special computer hardware requires extensive computation due to the complex nature of the mechanical behaviour of the soft tissue. In addition, multimodal cues that include haptic sensation require an update rate of 1000 Hz in order to maintain realism. Thus, a realistic visio-haptic deformable model development is required but up to certain extent as beyond realism, computation is counterproductive.

This thesis focuses on the development of fast rendering algorithms for real-time visio-haptic deformation modelling of soft tissue, in general, and investigation of multimodal cues and its extent in a haptic dominant palpation simulator for the enhancement of learning skills, in particular. Two visio-haptic deformable models were developed. The first is a vertex based visio-haptic deformable model using slope intercept form of a line equation and is implemented on the surface mesh model. The second is an adaptive area based visio-haptic deformable model, implemented on the surface mesh model as well as on a volumetric mesh model in order to capture the internal physical properties of the soft tissue. These models are different in terms of

their visual and haptic realism. The efficiency of the models was improved further by implementing an octree space partitioning, and one and two-neighbourhood techniques. Both models developed are important in this thesis and are meant to be used as test bed for the evaluation.

An evaluation study consisting of four series of experiments was conducted to validate the models and investigate the role of visio-haptic cues in a haptic dominant simulator. The results of these studies show that visio-haptic modality was better in learning skills as compared to haptic only in a haptic dominant environment. Multimodal cues enhanced learning skills in an instructor/ trainee environment as well as enhanced the learning skills of trainee in a short-term and long-term memory for force and position recall. The findings also stress that the extent of including visual and haptic cue into the simulation needs to be considered in order to improve the performance of the simulator.

ABSTRAK

Visi dan haptik adalah kaedah yang sangat penting dalam simulasi bagi bidang perubatan. Isyarat visual mampu membantu seseorang pengendali untuk melihat prosedur perubatan yang sedang dilakukan manakala isyarat haptik membolehkan pengendali merasai sentuhan ke atas objek yang dimanipulasi semasa interaksi berlaku. Walaupun kedua-duanya penting dalam simulasi komputer, gabungan kedua-dua kaedah belum dikaji dengan menyeluruh, terutamanya dalam persekitaran haptik yang dominan. Kesannya, pengurusan peruntukan sumber adalah tidak memberansangkan daripada segi usaha untuk mengubah dua kaedah ini kepada simulator bagi interaksi semasa secara realiti. Oleh itu, penyelidikan perlu dilakukan bagi mengenalpasti sama ada kaedah tunggal (haptik) atau gabungan kedua-duanya yang berkemungkinan lebih baik bagi tujuan pembelajaran dalam persekitaran haptik yang dominan seperti di dalam simulator sentuhan. Walau bagaimanapun, sebelum penyiasatan sedemikian boleh dilakukan, satu isu pelaksanaan teknikal utama dalam pengubahan visi-haptic rendering perlu dijalankan. Pelaksanaan interaksi masa secara realiti bersama-sama dengan jajaran yang betul tanpa menggunakan mana-mana komponen elektronik komputer, memerlukan pengiraan yang meluas disebabkan sifat mekanikal kompleks yang ada pada tisu lembut itu sendiri. Isyarat mod berbilang adalah termasuk dengan sensasi haptik yang memerlukan kadar peningkatan kepada 1000 Hz bagi mengekalkan realisme. Oleh itu, pembangunan model visi-haptik boleh ubah semasa adalah perlu sehingga pada had tertentu sahaja. Langkauan had yang ada boleh menyebabkan keadaan di luar realisme yang sekaligus menunjukkan pengiraan adalah tidak produktif pada sesetengah situasi.

Tesis ini memfokus kepada pembangunan perubahan algoritma pantas visi-haptik-semasa untuk model tisu lembut yang bermasalah secara umum, dan penyelidikan ke atas isyarat berbilang dan tahap haptik dominan simulator sentuhan untuk meningkatkan kemahiran pembelajaran, khususnya. Melalui kaedah visi-haptik, dua

model yang bermasalah telah dibangunkan. Yang pertama adalah puncak yang berasaskan model yang bermasalah (visi-haptik) menggunakan pintasan cerun yang didapati daripada persamaan linear dan dilaksanakan pada model permukaan jaringan. Yang kedua ialah kawasan penyesuaian berdasarkan model bermasalah (visi-haptik), dilaksanakan pada model permukaan jaringan serta pada model isipadu jaringan untuk mengenalpasti sifat fizikal dalaman bagi tisu lembut yang bermasalah tersebut. Model-model ini adalah berbeza dari segi visual dan haptik realiti. Kecekapan model telah dipertingkatkan lagi dengan melaksanakan pembahagian ruang struktur data pokok atau Octree, serta satu dan dua teknik kejiranan. Kedua-dua model yang dibangunkan adalah penting dalam tesis ini dan bertujuan untuk digunakan dalam projek penyelidikan yang besar supaya penilaian yang baik dapat diperolehi.

Satu kajian penilaian yang terdiri daripada empat siri eksperimen telah dijalankan untuk mengesahkan model-model yang ada dan untuk menyiasat peranan isyarat visi-haptik di dalam simulator haptik yang dominan. Keputusan kajian ini menunjukkan bahawa kaedah visi-haptik adalah lebih baik dalam kemahiran pembelajaran berbanding haptik hanya dalam persekitaran haptik yang dominan. Isyarat berbilang meningkatkan kemahiran pembelajaran seorang pengajar/persekitaran pelatih serta meningkatkan kemahiran pembelajaran pelatih dalam memori jangka masa pendek dan jangka masa panjang untuk capaian memori seperti daya kerja dan posisi tertentu. Hasil kajian juga menunjukkan bahawa keboleh-capaian visual dan isyarat haptik terhadap simulasi perlu dititikberatkan dalam usaha untuk meningkatkan prestasi simulator.

In compliance with the terms of the Copyright Act 1987 and the IP Policy of the university, the copyright of this thesis has been reassigned by the author to the legal entity of the university,

Institute of Technology PETRONAS Sdn Bhd.

Due acknowledgement shall always be made of the use of any material contained in, or derived from, this thesis.

© IRSHAD AHMAD, 2013

Institute of Technology PETRONAS Sdn Bhd

All rights reserved.

TABLE OF CONTENTS

| | |
|---|-----|
| STATUS OF THESIS | I |
| DECLARATION OF THESIS | IV |
| ACKNOWLEDGEMENT | VI |
| ABSTRACT | VII |
| ABSTRAK | IX |
| CHAPTER 1 INTRODUCTION | 1 |
| 1.1 Background | 1 |
| 1.2 Problem Formulation | 2 |
| 1.3 Research Objectives | 3 |
| 1.4 Thesis Contribution | 3 |
| 1.5 Scope of the Thesis | 4 |
| 1.6 Thesis Outline | 5 |
| 1.7 Summary | 7 |
| CHAPTER 2 RELATED WORK | 9 |
| 2.1 Multimodal Cues for Learning Skills Enhancement | 9 |
| 2.2 Palpation Simulators | 11 |
| 2.3 Deformation Modelling Techniques | 16 |
| 2.3.1 Nonphysical Modelling | 17 |
| 2.3.1.1 Vertex Based Modelling | 17 |
| 2.3.1.2 Spline Based Modelling | 17 |
| 2.3.1.3 Skeleton Subspace Deformation | 19 |
| 2.3.2 Physical Modelling | 20 |
| 2.3.2.1 Mass Spring Model (MSM) | 20 |
| 2.3.2.2 Finite Element Method (FEM) | 26 |
| 2.3.2.3 Boundary Element Method (BEM) | 35 |
| 2.4 Choice of Modelling Technique | 35 |
| 2.5 Levels of Detail | 36 |
| 2.6 Summary of Related work | 39 |

| | |
|--|----|
| 2.7 Summary | 41 |
| CHAPTER 3 RESEARCH METHODOLOGY | 44 |
| 3.1 Model Development | 45 |
| 3.2 Simulation Based Performance Evaluation..... | 46 |
| 3.3 Experiments..... | 46 |
| 3.3.1 Types of an Experiment | 47 |
| 3.3.1.1 Laboratory / Controlled Experiment | 47 |
| 3.3.1.2 Field Experiment..... | 47 |
| 3.3.1.3 Natural Experiments | 47 |
| 3.3.2 Experimental Design..... | 48 |
| 3.4 Summary | 49 |
| CHAPTER 4 VERTEX BASED VISIO-HAPTIC DEFORMABLE MODEL..... | 50 |
| 4.1 Model Representation | 50 |
| 4.2 Vertex Based Visio – Haptic Deformable Model..... | 52 |
| 4.2.1 Ring of Neighbour Algorithm | 53 |
| 4.2.2 Algorithm for Visual and Haptic Deformation | 56 |
| 4.2.2.1 Visual Deformation..... | 56 |
| 4.2.2.2 Haptic Deformation | 58 |
| 4.2.3 Shape Preservation..... | 59 |
| 4.3 Collision Detection (CD) | 62 |
| 4.3.1 Space Partitioning | 63 |
| 4.3.1.1 Octree Space Partitioning (OSP) | 63 |
| 4.4 Haptic Rendering of an Octant and Two Rings of Neighbours | 64 |
| 4.5 Results on the Performances of the Model | 66 |
| 4.5.1 Octree Implementation for Fast Collision Detection..... | 67 |
| 4.5.2 Efficient Collision Detection and Rendering | 71 |
| 4.6 Summary | 74 |
| CHAPTER 5 ADAPTIVE AREA BASED VISIO-HAPTIC DEFORMABLE MODEL | 76 |
| 5.1 Model Representation | 77 |
| 5.1.1 Surface Model..... | 77 |
| 5.1.2 Volumetric Model..... | 78 |

| | |
|---|-----|
| 5.2 Adaptive Area Based Visio-Haptic Deformable Model | 79 |
| 5.2.1 Mass Spring Model | 79 |
| 5.2.1.1 Stable Numerical Integration | 84 |
| 5.2.2 Adaptive Area Based Refinement and Simplification Technique | 84 |
| 5.2.2.1 Data Structure for Surface Mesh Model..... | 86 |
| 5.2.2.2 Data Structure for Volumetric Mesh Model..... | 89 |
| 5.2.2.3 Description of the Adaptive Area Based Refinement Algorithm..... | 89 |
| 5.2.2.4 Description of the Simplification Algorithm | 92 |
| 5.2.3 Parameter Identification..... | 93 |
| 5.2.3.1 Mass Identification | 94 |
| 5.2.3.2 Spring Stiffness and Damping Coefficient Identification | 96 |
| 5.2.4 Shape Preservation | 98 |
| 5.3 Criteria for Refinement..... | 99 |
| 5.4 Octree Space Partitioning (OSP)..... | 100 |
| 5.5 Results on the Performances of the Model..... | 100 |
| 5.5.1 Surface Mesh Model..... | 101 |
| 5.5.1.1 Triangular Mesh Reduction | 101 |
| 5.5.1.2 Reduction in Number of New Masses, Spring Stiffness and Damping Coefficients..... | 103 |
| 5.5.2 Tetrahedral Mesh Reduction..... | 106 |
| 5.5.2.1 Model Reduction | 107 |
| 5.5.2.2 Reduction in Number of New Masses, Spring Stiffness and Damping Coefficients..... | 108 |
| 5.5.3 Collision Detection Using Space Partitioning Method | 109 |
| 5.5.4 Deformation Using Adaptive Subdivision Algorithm..... | 112 |
| 5.5.4.1 Comparison of the Proposed Algorithm with the Refined Model..... | 113 |
| 5.5.4.2 Comparison of Proposed Algorithm with Conventional Adaptive Subdivision Method | 115 |
| 5.6 Summary..... | 118 |
| CHAPTER 6 EVALUATION STUDY | 121 |
| 6.1 Objectives of the Evaluation Study..... | 121 |

| | |
|---|-----|
| 6.2 EXPERIMENT 1 | 123 |
| 6.2.1 Participants | 123 |
| 6.2.2 Apparatus | 123 |
| 6.2.3 Experimental Design..... | 124 |
| 6.2.3.1 Prototype - A | 125 |
| 6.2.3.2 Prototype - B..... | 125 |
| 6.2.3.3 Prototype - C..... | 125 |
| 6.2.4 Procedure..... | 126 |
| 6.3 EXPERIMENT 2: Short-Term Skill Learning..... | 128 |
| 6.3.1 Participants | 128 |
| 6.3.2 Apparatus | 128 |
| 6.3.3 Experimental Design..... | 131 |
| 6.3.4 Procedure..... | 131 |
| 6.4 EXPERIMENT 3: Long-Term Skill Learning..... | 132 |
| 6.4.1 Participants | 132 |
| 6.4.2 Apparatus | 133 |
| 6.4.3 Experimental Design..... | 133 |
| 6.4.4 Procedure..... | 133 |
| 6.5 EXPERIMENT 4: Extent of Visio-Haptic Cues | 134 |
| 6.5.1 Participants | 134 |
| 6.5.2 Apparatus | 135 |
| 6.6 Results..... | 135 |
| 6.6.1 Experiment 1 | 135 |
| 6.6.2 Experiment 2 | 138 |
| 6.6.2.1 Palpation Force | 138 |
| 6.6.2.2 Exploration Time | 141 |
| 6.6.3 Experiment 3 | 142 |
| 6.6.3.1 Palpation Force | 143 |
| 6.6.3.2 Exploration Time | 143 |
| 6.6.4 Experiment 4 | 144 |
| 6.7 Summary | 145 |
| CHAPTER 7 DISCUSSION..... | 147 |
| 7.1 Visio - Haptic Deformable Models..... | 147 |

| | |
|---|-----|
| 7.1.1 Vertex Based Visio-Haptic Deformable Model..... | 148 |
| 7.1.1.1 Performance Analysis..... | 148 |
| 7.1.2 Adaptive Area Based Visio-Haptic Deformable Model..... | 150 |
| 7.1.2.1 Performance Analysis..... | 150 |
| 7.2 Evaluation Study Analysis..... | 156 |
| 7.2.1 Visio-Haptic Modality in Collaborative Environment..... | 156 |
| 7.2.2 Visio-Haptic Modality in Short-term and Long-term Skill Enhancement..... | 157 |
| 7.2.3 Visual and Haptic Modality..... | 158 |
| 7.3 Summary..... | 162 |
| CHAPTER 8 CONCLUSION AND FUTURE WORK | 165 |
| 8.1 Deformable Modelling Techniques..... | 165 |
| 8.2 Evaluation Study | 166 |
| 8.3 Contribution | 168 |
| 8.3.1 Practical Contribution..... | 168 |
| 8.3.2 Substantive Contribution | 169 |
| 8.4 Recommendations for Future Work..... | 169 |
| PUBLICATIONS..... | 171 |
| REFERENCES | 172 |
| APPENDIX A..... | 190 |
| APPENDIX B..... | 209 |

LIST OF TABLES

| | |
|--|-----|
| Table 2.1: Previous work related to learning skill enhancement due to multimodal cues | 39 |
| Table 2.2: Previous work related to veterinary palpation simulators..... | 40 |
| Table 2.3: Previous work related to visio-haptic deformable models | 40 |
| Table 2.4: Previous work related to levels of detail | 41 |
| Table 4.1: Time taken in (ms) for searching touched node using a linear and an octree method | 69 |
| Table 4.2: Time required for different search and rendering techniques | 73 |
| Table 5.1: Parameters created during Conventional Adaptive Subdivision and Area Based Adaptive Subdivision Algorithms for the same surface mesh model..... | 106 |
| Table 5.2: Parameters created during conventional adaptive subdivision and adaptive area based subdivision algorithms for the same tetrahedral mesh model..... | 109 |
| Table 5.3: Search time for touching node using octree and nearest neighbour method for models of different resolution..... | 111 |
| Table 5.4: Comparison of proposed method with the refined model..... | 114 |
| Table 5.5: Comparison of the proposed method with a conventional adaptive subdivision method based on collision detection techniques | 117 |
| Table 6.1: Assignment of the Instructors to different sessions of the trainees | 127 |

LIST OF FIGURES

| | |
|---|----|
| Figure 1.1: The flow chart of the thesis structure | 5 |
| Figure 2.1: A haptic pulse palpation simulator [49] | 12 |
| Figure 2.2: A virtual haptic back palpation simulator | 13 |
| Figure 2.3: An illustration of virtual patient palpation in PalpSim [56] | 14 |
| Figure 2.4: An instructor is training trainee during palpation [30]..... | 15 |
| Figure 2.5: NURBS (Non-Uniform Rational B-Spline) surface. (a) A light grey surface at the centre is controlled by yellow points. (b) Deformation of surface by moving control points | 18 |
| Figure 2.6: An example of skeletal deformation [68]..... | 19 |
| Figure 2.7: A mass spring model with nodes connected through springs..... | 20 |
| Figure 2.8: A triangular mesh with springs forming triangular elements | 21 |
| Figure 2.9: A T_2 -mesh having every node connected to its three neighbours..... | 21 |
| Figure 2.10: A three-layer mesh representation of skin used by [19]..... | 24 |
| Figure 2.11: An illustration of elements and its boundary nodes [101]..... | 27 |
| Figure 2.12: Left: Surgeon moves a virtual tool via force feedback device, Right: Real-time deformation of liver [104] | 29 |
| Figure 2.13: Left: A Finite element model of Uterus containing 200 elements. Middle and right: Simulated deformation of the corpus and Fallopian tube of the uterus [107] | 30 |
| Figure 2.14: Top Left: Liver model deformation. Top right: Multi-resolution model of liver. Bottom Left: Toy model deformation. Bottom Right: Multi-resolution model of a toy [102]..... | 31 |
| Figure 2.15: Tetrahedral liver model simulated using nonlinear FEM [115]..... | 33 |
| Figure 2.16: Different deformation profiles during a simulation [25]..... | 34 |
| Figure 3.1: Research Methodology Framework | 45 |
| Figure 4.1: A touched node with its first and second ring neighbours | 54 |
| Figure 4.2: A deformation profile due to variable step size for touching node and its neighbours | 56 |
| Figure 4.3: Force computation and rendering mechanism during interaction with the model..... | 58 |

| | |
|--|-----|
| Figure 4.4: Home springs between a fixed and deformable node for shape preservation..... | 60 |
| Figure 4.5: (a) Initial Configuration, (b) Configuration after force applied..... | 60 |
| Figure 4.6: Example of an octree space partitioning method..... | 63 |
| Figure 4.7: Model partitioning using an octree method..... | 64 |
| Figure 4.8: Snapshots of an interactive simulation for rendering the two rings of neighbour in haptic loop..... | 65 |
| Figure 4.9: Snapshots of an interactive simulation for rendering an octant of the touched node in the haptic loop..... | 66 |
| Figure 4.10: An interactive deformation of the stomach model..... | 67 |
| Figure 4.11: An interactive deformation of the sphere model..... | 67 |
| Figure 4.12: Comparison of a linear and an octree search method for different model resolution..... | 70 |
| Figure 4.13: A plot for Table 4.2..... | 74 |
| Figure 5.1: A surface mesh representation of the reproductive tract of a cow..... | 77 |
| Figure 5.2: Volumetric model made of tetrahedra of the reproductive tract of a cow..... | 78 |
| Figure 5.3: Cross section showing internal tetrahedra of the reproductive tract of a cow..... | 78 |
| Figure 5.4: A mass spring model with the nodes connected through spring..... | 80 |
| Figure 5.5: A mass spring model for tetrahedral mesh..... | 80 |
| Figure 5.6: Standard viscoelastic models used to represent the soft tissues (a) Maxwell (b) Kelvin-Voigt or Voigt (c) Kelvin, or standard linear [112] | 82 |
| Figure 5.7: A coarse triangular mesh of having varied triangle areas..... | 85 |
| Figure 5.8: Reproductive tract of a cow. Left: A textured model, Right: A wireframe model..... | 86 |
| Figure 5.9: A second level of subdivision based on triangles' area. In (a), (b) and (c), SLoD =1, (d), (e) and (f), SLoD = 2, (g), (h), and (i), SLoD = 3, (j) SLoD = 4..... | 91 |
| Figure 5.10: Visualizing neighbour triangles during collision of an HCP with two different nodes..... | 92 |
| Figure 5.11: Adjustment of mass in different levels of subdivision..... | 95 |
| Figure 5.12: Model made up of structural springs, and home springs..... | 99 |
| Figure 5.13: Top: The coarse model, Middle: conventional adaptive subdivision, Bottom: adaptive area based subdivision..... | 102 |

| | |
|--|-----|
| Figure 5.14: Triangles produced using the proposed algorithm and conventional adaptive subdivision algorithm | 103 |
| Figure 5.15: A stomach model used in simulation | 105 |
| Figure 5.16:(a) A touched node subdivision using proposed algorithm and (b) using conventional adaptive subdivision | 105 |
| Figure 5.17:(a) A tetrahedral mesh model, (b) Portion of coarse tetrahedral mesh model (c), Adaptive area based subdivision, (d) Conventional adaptive subdivision | 107 |
| Figure 5.18: A stomach model partitioned using an octree method | 110 |
| Figure 5.19: A graph for Table 5.2 | 112 |
| Figure 5.20: Stomach model (Left) Deformation using adaptive area based method (Right) Deformation using refined model | 113 |
| Figure 5.21: Stomach Model (Left) adaptively refined area deformation using the proposed method (Right) adaptively refined area deformation using the | 115 |
| Figure 5.22: Deformation of the touched node with the force feedback device | 116 |
| Figure 6.1: An instructor-trainee experimental setup of the training simulator | 124 |
| Figure 6.2: An experimental setup for visio-haptic modality..... | 129 |
| Figure 6.3: Side view of the reproductive tract of a non-pregnant cow | 130 |
| Figure 6.4: Mean and variance of palpation force for prototype-A, B, and C | 137 |
| Figure 6.5: Accuracy per percent for prototype-A, B, and C | 137 |
| Figure 6.6: Range Box and Mean for palpation force for the three prototypes..... | 140 |
| Figure 6.7: Individual palpation force by the trainees for the three prototypes-A, B, and C in a post-training phase | 140 |
| Figure 6.8: Range Box and Mean of exploration time for prototype-A, B, and C... .. | 142 |
| Figure 6.9: Range Box and mean of exploration time for the three prototypes | 144 |
| Figure 6.10: Palpation force recorded for the participants using prototype-C1, and C2 | 145 |
| Figure 7.1: New nodes created during the proposed and conventional adaptive | 152 |
| Figure 7.2: New springs created during proposed and conventional adaptive model with total springs in coarse model | 152 |
| Figure 7.3: Execution time comparison of different models for stomach in the graphic and haptic loop..... | 154 |

Figure 7.4: Execution time comparison of different models for reproductive tract in the graphic and haptic loop 155

Figure 7.5: An individual palpation force by the participants using the three prototypes-A, B, and C for short-term force recall 159

Figure 7.6: An individual palpation force by the participants using prototypes-A, B, and C for long-term memory force recall 160

Figure 7.7: Effect of the extent of visual and haptic cues on learning skills enhancement..... 161

Figure 7.8: Range Box for short-term, and long-term for palpation force using prototype-A, B, and C..... 162

LIST OF ABBREVIATIONS

| | |
|------|---------------------------------|
| VR | Virtual Reality |
| 3D | Three Dimensions |
| VH | Visual and Haptic |
| MSM | Mass Spring Model |
| FEM | Finite Element Method |
| BEM | Boundary Element Method |
| OSP | Octree Space Partitioning |
| BSP | Binary Space Partitioning |
| CT | Computed Tomography |
| CD | Collision Detection |
| MRI | Magnetic Resonance Imaging |
| HCP | Haptic Contact Point |
| HOPS | Horse Ovary Palpation Simulator |
| AABB | Axis Aligned Bounding Boxes |
| OBB | Oriented Bounding Boxes |
| RK4 | Runge Kutta Order 4 |
| SSD | Skeleton Subspace Deformation |
| VHB | Virtual Haptic Back |
| MEG | Motorized Endoscopic Grasper |

| | |
|-------|-------------------------------|
| TRD | Torsional Resonator Device |
| SLoD | Sub-Levels of Detail |
| LoD | Level of Details |
| ANOVA | Analysis of variance |
| MPF | Maximum Palpation Force |
| SFT | Stopping Force Threshold |
| ST | Sound Threshold |
| V | Visual |
| H | Haptic |
| VHDM | Visio-Haptic Deformable Model |

CHAPTER 1

INTRODUCTION

1.1 Background

A computer simulation enables experiential training systems to take place in the medical field. The simulation allows the practitioners to be trained on virtual patients [1-3]. The practice supports critical skills analysis and provides feedback on the performed procedure. The feedback works as an improvement for the skills before instigation training with patients. Simulations can also provide the user with an opportunity to practice difficult cases without harming the patients [4].

In a medical simulation, vision and haptic are two most important modalities. Visual modality allows the trainee to see his actions while performing the procedures whereas haptic allows feeling the objects during a training simulation. A situation or environment in which haptic feedback is seen as more dominant could be found in the example of a simulation involving palpation. Palpation is a medical procedure in which the expert with the help of their fingers presses the soft body in order to diagnose various abnormalities and/or features through locating the landmarks beneath the patient's skin [5]. This procedure is also used by the veterans for diagnosis of various stages of pregnancy in animals. The driving force behind the computer based palpation simulator are the ethical issues, huge number of experiments per patient, and the non-availability of the patients at the right time [6]. In such computer based palpation simulators, the practitioner learns skills in performing a specific procedure before using his expertise on the real patient especially the internal delicate organs such as reproductive tract. The use of visual and haptic feedback in the palpation simulator signals their importance, particularly in learning the procedures.

Both new and even expert practitioners could benefit from the cues provided. Ideally, this practice supports a collaborative process which experts and novices work together to achieve a common goal: knowledge transfer. To achieve this goal, both parties must build a common ground: a common and consistent representation of the shared situation [7]. Integration of both visual and haptic has a potential to accelerate this aspiration despite the fact that palpation procedure is more of haptic dominant. An understanding on how both modalities could assist in such a training procedure whilst taking into account the constraint on the technical implementation is needed. This is the central element of this thesis.

1.2 Problem Formulation

Multisensory modality is more informative and helps users to learn quickly different shapes or procedure as compared to single modality [8]. It has been reported in [9], that Visual and Haptic (VH) intervention have progressed in learning different shapes as compared to Visual (V) intervention. In addition, Morris et al., [10], showed that visio-haptic modalities enhanced force learning skills. However in the findings of Srimathveeravalli and Thenkurussi [11] for the positional information recall, have stated that visio-haptic is insignificant. Other researchers reported various study findings on the importance of visio-haptic information. The visio-haptic training is not effective but showed promising results in long-term motor skills [12]. Huegel and O'Malley, [13] showed that visio-haptic is effective in the beginning of the learning curve, has no significance at the end of the training. These conflicting results raise a question whether visio-haptic cue enhances the learning skills of the practitioner in a haptic dominant environment such as palpation. Thus, in the field of palpation dealing with elastic objects, a complete evaluation is needed [4].

Even though the visio-haptic modality could enhance skill learning, the implementation and timely coordination of these cues are computationally expensive. The reason being a refresh rate for realistic visualization is 30-40 Hz, while the refresh rate of the realistic haptic rendering is 500-1000 Hz [15]. The two most widely used methods in physics based deformable models are Mass Spring Model (MSM),

and Finite Element Method (FEM) [14]. Mass Spring Model is easy to implement and efficient, but still computationally expensive for high-resolution models [15-22; 14]. Finite Element Method, although accurate due to its continuous nature but computationally more expensive, when compared to the Mass Spring Model, and is therefore not suitable for real time interactive applications in its pure form [16; 23-26]. Using visio-haptic modalities as compared to visual only or haptic only requires more computation and proper synchronization of the two modalities. Therefore, efficient visio-haptic deformable models are required for the realization of visio-haptic cues in the palpation simulators based on the underlying physical properties, and without any special graphics or processing hardware. Further, in addition to the development of visio-haptic deformable models for the visio-haptic modalities, there is also a need for finding the extent of visual cue and haptic cue to be present in a haptic dominant simulator such as for the palpation procedure because beyond such extent, realism is counterproductive [27].

1.3 Research Objectives

The research objectives of this thesis are:

- To develop visio-haptic deformable models for a palpation simulator that satisfies minimum rendering requirements.
- To implement the models into a haptic dominant palpation simulator.
- To evaluate the haptic only and, visio-haptic parameters of the palpation simulator for learning skill enhancement.
- To assess the extent in which the visual and haptic cues are important in the visio-haptic modality in terms of the visual and haptic rendering requirements.

1.4 Thesis Contribution

The main contributions of this thesis are as follows.

1. The development of visio-haptic deformable models that satisfied minimum rendering requirements for interactive haptic dominant palpation training simulator.
2. An empirical study on the haptic only and visio-haptic modalities of learning skills development.
 - That involves both the instructor and trainee
 - For short-term and long- term memory
3. A validation study on the importance of multimodal cues in a haptic dominant palpation simulator.

1.5 Scope of the Thesis

Even though there is a wide variety of palpation simulator, here the focus of this thesis is on palpation simulators in the field of veterinary for teaching and training the skills of finding different features in the reproductive tract and the pregnancy levels. The reason for choosing palpation simulator in the field of veterinary is twofold: firstly, it is a haptic dominant palpation simulator, interacting with the soft objects, and the best candidate to evaluate multimodal cues in such a single cue dominant environment. Secondly, the existing simulators as reported in [28-31], lack the realistic visual and haptic cues based on the underlying physical properties. The visio-haptic deformable models developed in this thesis are specifically for the palpation simulators in the field of veterinary, and the prototype development involves an animal internal organ.

As the main idea is the development of visio-haptic deformable model that is adequate in a haptic dominant simulator, and the evaluation of the multimodal cues for the enhancement of skills in learning a procedure, therefore, it is not necessary to test this model with the vet students in particular. What important is to have a prototype as a test bed for the evaluation. In addition, suturing, cutting, and pinching of the internal delicate organs are not necessary in palpation, and are therefore out of the scope of this thesis.

1.6 Thesis Outline

The contents of this thesis are divided into eight chapters as shown in Figure 1.1.

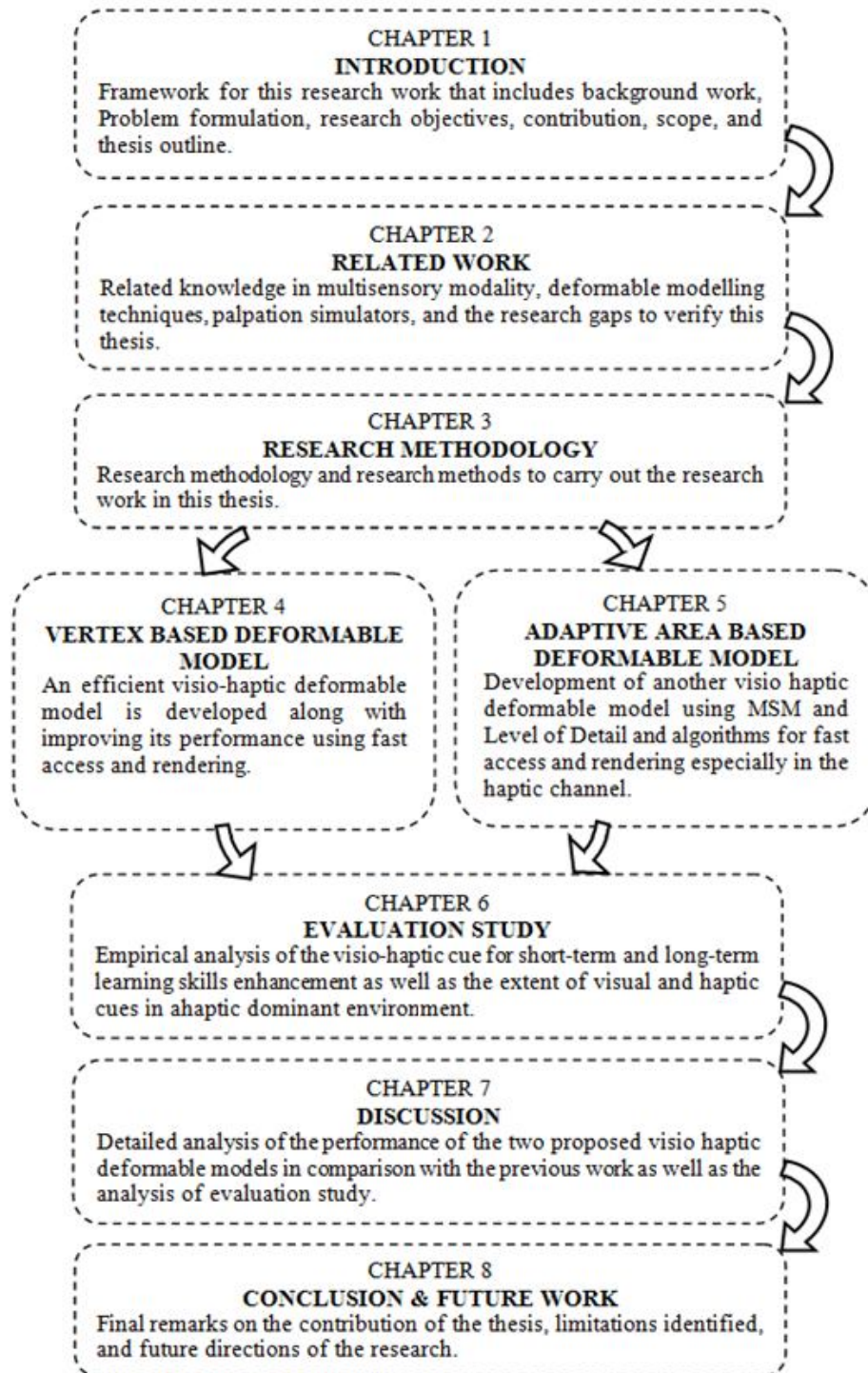


Figure 1.1: The flow chart of the thesis structure

The outline of the subsequent chapters is as follows.

Chapter 2 presents the related work that starts with issues on multisensory modality, followed by palpation simulator, and deformable modelling techniques. The choice of modelling techniques is discussed next. The levels of detail are also discussed in this chapter.

Chapter 3 presents the research methodology for this research work. First section discusses the model development. Second section discusses the simulation based performance evaluation. Third section discusses the experiments, and the experimental design.

Chapter 4 presents the first model that involves vertex based visio-haptic deformable method using slope intercept form of a line equation. Shape preservation springs are introduced to achieve elastic deformation. For fast collision detection between the PHANTOM force feedback device and the model, an octree is implemented only for haptic cue and; furthermore, for fast rendering two-neighbourhood method is used for the haptic cue as well. The visio-haptic deformable model is simple in implementation, yet efficient. The haptic rendering implements Hooke's Law for the haptic sensation when interacting with the deformable model during simulation in order to evaluate the effect of visual cue on the enhancement of learning skills in a haptic dominant palpation simulator.

Chapter 5 presents the second visio-haptic deformable model, namely an adaptive area based visio-haptic deformable model that is implemented on both the surface based as well as volume based mesh models. The region of interest during an interactive simulation is refined based on the triangles' area. Mass Spring Model is used in a physically realistic tissue representation as well as visually realistic. The mass spring model is refined in the region of interest both in the visual and haptic cues in order to feel the model at the point of contact visually and haptically realistic. The new masses, stiffness, and damping coefficients are computed in order to maintain the same physical properties in the refined region as were before refinement. The octree space partitioning method is also implemented for the haptic cue and shape preservation springs are used in order to maintain the overall shape of the model

during palpation in the case of the surface mesh model only. The reaction forces computed from the mass spring model are sent back to the user in order to feel the object in the haptic channel.

Chapter 6 presents four experiments for the evaluation of multimodal cues from simple to more realistic in terms of visio-haptic cue, and haptic only for the enhancement of short-term and long-term learning skills, the effectiveness of visual cue in a collaborative, and the haptic dominant environment, and the extent of visual and haptic cue needed in the simulator. The first experiment presents an empirical study on the importance of visual cue, for the instructor guiding the trainee during skill learning procedure in a collaborative and haptic dominant environment. The second experiment statistically analyses visio-haptic modality and haptic only for short-term learning skills enhancement, while the third experiment evaluates visio-haptic modality and haptic only for long-term learning skill enhancement. The last experiment evaluates the extent of visual and haptic cues, using one of the proposed models with different resolution of multimodal cues for learning skills enhancement, as beyond realism, computation is counterproductive. These experiments also validate the two visio-haptic deformable models proposed in this thesis.

Chapter 7 discusses the work presented in chapters four, five, and six. The first section discusses the results of the two proposed visio-haptic deformable models presented in chapters four and five respectively. The subsequent section discusses the experimental work presented in chapter six.

Chapter 8 presents a summary of the contributions achieved in this work, conclusions and possible future work from the thesis.

1.7 Summary

This chapter introduced the background study, problem formulation, research objectives, main research contribution, and the thesis outline. The main objective of this research is to develop visio-haptic deformable models, and then to evaluate visio-haptic modality versus haptic only modality in a haptic dominant environment as well as the extent of visual and haptic cues to be included for realism, as beyond realism,

computation is counterproductive. The next chapter presents related work in the field of multisensory modality, palpation simulators, and deformable modelling techniques specifically for the simulators involving haptic cues.

CHAPTER 2

RELATED WORK

This chapter presents the related work in the field of multimodal cues, palpation simulator, and deformable modelling techniques. The intention is to provide a basis to examine the reported work in a similar area in preparation to achieve the research objectives specified in chapter 1.

2.1 Multimodal Cues for Learning Skills Enhancement

Humans interact with their environment through several modalities: vision, touch, audition, gustatory, olfactory. Among these modalities, vision and touch play an important role in gathering information from the surrounding environment and store representation of the environmental objects. Eighty percent of learning is done through vision and estimated above fifty percent of the brain activity are devoted to the visual input [32; 33]. Multisensory modality that includes touch is more informative and helps users to learn quickly different shapes or procedure as compared to single modality [8]. Haptic or touch feedback has become an integral component of teaching surgical skills as in [34-36]. Haptics can best contribute to the user's learning of skills [37], and to user perceptions of virtual shape objects [38].

The results of recent work suggest that haptic and vision when used in combination, improved learning skills than any of these modalities used individually [8]. Huang et al., [39] used three learning paradigm containing visual only, haptic only, and visio-haptic modalities. In all these paradigms, the participants were required to excite a virtual oscillator. The finding reveals that visio-haptic modality were superior to the other modalities.

The authors in [38], examined the impact of haptic feedback on perception of unknown objects. They used three treatment groups of students, i.e. visual, haptic, and visual plus haptic feedback to explore a set of virtual objects. They found that the visual plus haptic group was more accurate in identifying objects than the visual or haptic only groups. The authors in [40] found that haptic constraints provide significant benefit for both performing, and learning movement patterns. In another study, Morris et al., [10], used the three paradigms as a visual display of normal force (V), haptic display of normal force (H), and visio-haptic display of normal force (VH) for learning a sequence of forces. The study findings show that using the visual and haptic (VH) modalities, the participants are better able to memorize the instructed force sequences, and therefore, enhanced force learning skills of the participants as compared to visual (V) and haptic (H) modality. Kalenine et al., [9], showed that Visual and Haptic (VH) intervention have progressed in learning different shapes as compared to Visual (V) intervention. In another study, the authors in [41], used two training paradigms, i.e. visual-haptic (VH), and visual (V) for handwriting learning. The result showed the improvements of VH training on letter recognition and handwriting quality were higher than the improvements after V training.

A contrast to the above findings, Srimathveeravalli and Thenkurussi [11], in their work on the positional information recall, stated that visio-haptic is insignificant. The subjects in [42], were asked to reproduce a novel three-dimensional path after practising with the mechanical guidance from a robot using haptic guidance with visual and visual demonstration. The results showed that the ability to reproduce the novel path is improved using both haptic guidance, and visual demonstration. There was nearly significant trend for the visual demonstration to be better than the haptic guidance [42]. In another work, Xing-Dong et al. [12], reported two experimental studies for the investigation of the effectiveness of visio-haptic modality in helping people develop short-term, and long-term motor skills. In the experiment for developing the short-term motor skills, the findings of the experiment stated that visio-haptic is insignificant. The second experiment shows promising results in helping people develop long-term motor skills. Huegel and O'Malley, [13] performed an experiment and showed that a visio-haptic modality is effective in the beginning of the skill learning and gave same learning results in the post-training.

Due to these conflicting results, a complete evaluation of the haptic only and visio-haptic modalities need to be performed in applications, which are only haptic dominant in the physical world such as a palpation simulator for the learning skills enhancement [4]. The studies in the literature have mainly focused on the learning skills enhancement, dealing with the hard objects. Palpation simulators deal with the underlying soft and deformable objects. Therefore, a complete evaluation of the haptic only and visio-haptic modalities are required in the deformable objects. The authors in [43] presented a model that suggests different behaviour using haptic or visual information will be exhibited depending on the degree of discrepancy between haptic and visual information. Therefore, the next concern is to find the effect of the extent of visual and haptic cue to be present in a haptic dominant palpation simulator. These two issues are the last two research objectives mentioned in chapter one, and along with the visio-haptic deformable model development, are the main subject matter of this thesis.

2.2 Palpation Simulators

Palpation is a procedure in which the expert with the help of their fingers presses the soft body in order to diagnose various abnormalities and/ or features through locating the landmarks beneath the patient's skin. The veterans also use this procedure for the diagnoses of various stages of pregnancy in animals. The ethical issues, huge number of experiments per patient, and the non-availability of patients at the right time are the driving forces behind the development of palpation simulators.

The first simulator developed was a knee palpation simulator, reported in [44]. Rutgers Master was used for the force feedback to the simulator. This is then followed by a tumour palpation simulator, developed in [45]. In this simulator, the trainee palpates the liver to find the hard object as tumour using the Rutgers Master II force feedback device. A simulator for prostate cancer was developed in [46]. They used PHANToM haptic force feedback for applying force and providing forces back to the trainee. The device has in fact influenced the development of other training simulators as it is mainly used in the real-time interactions.

A palpation simulator for laparoscopic surgery was developed in [47]. The PHANToM force feedback was used for rendering the reaction forces during interaction with the human liver in the simulator. A haptic pulse palpation simulator was developed in [48]. In this simulator, the user is allowed to explore human body skin. Once the user reaches the virtual skin through the force feedback devices, he/she can feel the pulses generated. In another study, a custom thimble for a PHANToM Omni haptic device has been fabricated in a brachial pulse palpation simulation [49]. This fabricated thimble transfers the force to a single finger. The pulse intensity depends upon the force applied by the user on a particular point on the skin. This is shown in Figure 2.1.



Figure 2.1: A haptic pulse palpation simulator [49]

Other studies that involve palpation training simulators and using the PHANToM haptic device in a more effective way could be seen in [50-52]. A Virtual Haptic Back (VHB) palpation simulator was developed by [50-52], as an aid for teaching and learning palpatory diagnosis. Two PHANToM ® 3.0 force feedback devices were used as a haptic interface through which the trainees are allowed to explore the virtual

model of the human back. Two PHANToM devices were used in order to use both hands in the palpation of the human back. This is shown in Figure 2.2.



Figure 2.2: A virtual haptic back palpation simulator

The authors in [53] developed a remote diagnosis simulator. The simulator designed for diagnosis of malignancy in the head and neck region. A 3D surface model created from computed tomography (CT) is used in the simulation with which the user interacts with the PHANToM desktop force feedback device.

The work on a haptic palpation simulator that supports multiple contact points of interaction could be found in [54; 55]. They developed a breast palpation simulator. In this simulator, they used multi-finger haptic device named Haptic Interface Robot (HIRO), which allow multiple contact points during interaction with the virtual model. Mechanical model of the breast with tumours were implemented using finite element method. Collision detection resolves multiple contact point, which was the focus of this work. The simulation was rendered on a computer monitor but no visio-haptic collocation was provided.

The authors in [56; 29] have developed PalpSim for training femoral palpation and needle insertion. They also developed a novel low cost device for femoral palpation. User can use both of his hands during training through PalpSim with 5

degrees of freedom as shown in Figure 2.3. In this simulator, the haptic devices integrated with augmented reality. This is the first simulator that uses Chroma-key technology for immersive augmented reality environment. The benefit of such environment is that haptic tools are seamlessly collocated with the virtual patient in the simulator.

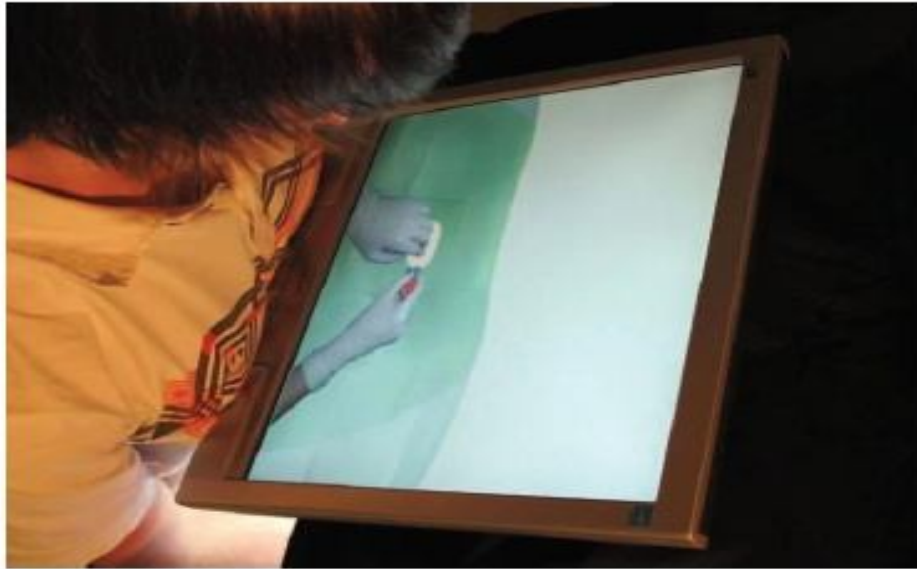


Figure 2.3: An illustration of virtual patient palpation in PalpSim [56]

Recently Sebastian [57] proposed a novel approach for tissue dragging in a haptic palpation for medical simulation. The stylus of the PHANToM force feedback device is modified by adding a lightweight palpation pad. They provided a detailed description of an extensive user study for the face validity of the simulator.

In the field of veterinary medicine, a rectal palpation simulator has been developed in the Royal Veterinary College, London United Kingdom. The first palpation simulator developed was Horse Ovary Palpation Simulator (HOPS) by [28]. The horse ovary is palpated through the PHANToM force feedback device. This training simulator is using a simple graphics interface and is used to palpate and diagnose the ovary only.

Following the development of HOPS from the same university, another simulator called bovine rectal palpation simulator for identifying through palpation the various structures such cervix, ovary, and uterus in the reproductive tract of a cow are

developed by [30]. The PHANToM force feedback device is positioned inside a fibreglass model of a cow as shown in Figure 2.4.



Figure 2.4: An instructor is training trainee during palpation [30]

A mixed reality palpation simulator is developed for Feline abdominal palpation containing the physical model and virtual reality [58]. Two PHANToM force feedback devices are used for palpating the physical model.

The simulators developed in the field of veterinary [30; 31; 28; 58] have a simple graphical model but lack the interactive visual deformation during the simulation. This lack of visual cue, in addition, restrict an instructor in properly guiding the trainee, learning is a collaborative process in which instructors and trainees work together toward a common goal: knowledge transfer [59]. Most importantly, the haptic sensation needs to be based on the physical properties of the soft tissue. A further improvement on the deformable modelling methods in order to improve realism, both in the visual and haptic sensation, allows mimicking the gap between virtual and real. In addition, the haptic sensation is based on the *Hook's Law*, and not on the underlying physical properties [58; 31; 30]. Therefore, realistic visual and

haptic cues are required in the veterinary palpation simulators to enhance the learning skills of the trainees.

In this thesis, a bovine rectal palpation simulator is chosen as a Testbed for the implementation and validation of the proposed visio-haptic deformable models. The simulator is also used for the evaluation of haptic only and visio-haptic modalities for the enhancement of short-term and long-term learning skills. The assessment of the extent of visio-haptic cues for learning skills enhancement is carried out using this palpation simulator. Furthermore, the simulator is also used for the evaluation of multimodal cues due to the lack of interactive visual deformation and without having realistic haptic sensation based on the physical properties of the soft tissue in the existing simulators.

2.3 Deformation Modelling Techniques

Soft tissue has the property of elasticity in which the shape of the tissue is retained when the external forces are removed. Therefore, methods for modelling soft tissue must consider rapid response during such a manipulation. It should closely approximate the behaviour of soft tissue while deforming and should appear realistic, as one has to feel dealing with the real soft tissue. If augmented with haptic feedback, the model also has to calculate the reaction forces. Although there are many techniques for soft tissue simulation in the literature, but here the focus will be on the techniques of real-time in which there is an interaction of users with the soft tissue through force feedback devices.

Deformable models could be defined based on the dimension. It may be one-dimensional (Lines and Curves), two-dimensional (surfaces) and three-dimensional (Solid Objects). Irrespective of the dimensionality of the deformable models, these models can be divided into two broad categories, nonphysical deformable and physical deformable models. The subsequent sections discuss these categories in detail.

2.3.1 Nonphysical Modelling

This category is also known as geometric modelling. In nonphysical modelling, the geometry of the model is directly manipulated, but the internal physics and dynamic interactions are not considered. These techniques are computationally efficient and greatly rely on the skills of the designer rather on the principles of physics. Two approaches in geometry based are vertex based and spline based used for deformation [60].

2.3.1.1 Vertex Based Modelling

In this method, while deforming an object visually, the vertices of the body are manipulated directly. All the points in the region of interest are deformed in such a way that shows realistic deformation visually. This is the most basic and efficient method of visual deformation.

A vertex based deformation is used in laparoscopic simulation for visual deformation [61]. In order to deform the soft tissue locally, they translated all vertices within a certain range, called the radius of influence, of the collision point along the surgical instrument. The magnitude of translation is controlled by the second order polynomial while the shape of the deformation is by the degree and the coefficients of the polynomial. This method is only applied for visual deformation. In real-time interactive simulation, this method is very efficient in terms of ease of computation and implementation but lack the physical properties of the soft tissue.

2.3.1.2 Spline Based Modelling

This method is mainly used for Freeform deformation. In a Freeform deformation, the object to be deformed is embedded in a linear cube of the grid. This grid acts as a handle for the deformation of objects. Changing the handle position deforms the object embedded in it as shown in Figure 2.5.

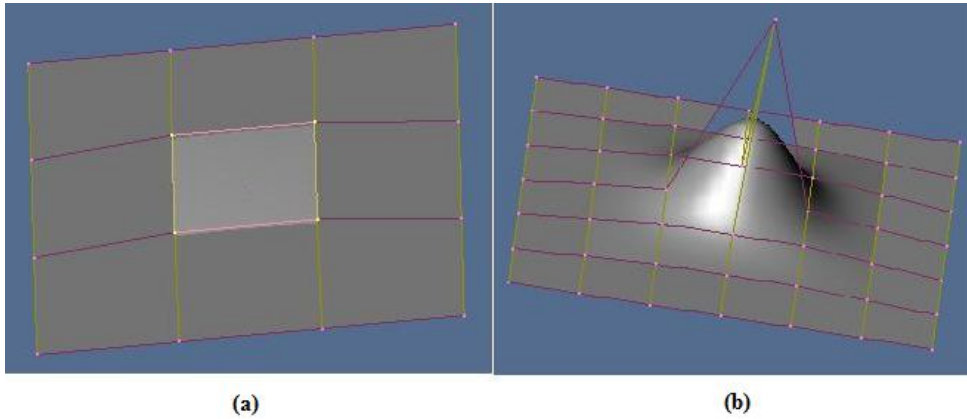


Figure 2.5: NURBS (Non-Uniform Rational B-Spline) surface. (a) A light grey surface at the centre is controlled by yellow points. (b) Deformation of surface by moving control points

Sederberg and Pary [62], who deform objects by distorting the space in which it contains, propose the pioneering work in this field. This method was used both for local and global deformation and the method is based on trivariate Bernstein polynomials. Davis and Burton [63] used Rational Bernstein bases. The authors Lamousin and Waggenpack [56] used Non-Uniform Rational B-Spline (NURBS) to achieve Freeform deformation. Griessmair and Purgathofer [64] used a trivariate basis-spline (B-spline). Hsu et al., [65], proposed a method in which points on the surface of the embedded object is manipulated directly. Song, and Yang [66]; proposed a weighted T- spline free form deformation (w-TFFD) method. The main advantage of this method is that multi-resolution control lattices can approximate any complex shapes tightly and automatically.

Although this technique is useful for model building and for computer animation, it is inappropriate for real-time simulation. The reason is that it does not work on the geometry of the model directly. Soft tissue properties lay in the geometry and hence this technique does not consider these properties in the simulation. To cope with this situation some works have been done as in [15]. It combined the Spline based deformation with the mass spring model. This method incorporates the physical behaviour of the system.

2.3.1.3 Skeleton Subspace Deformation

Skeleton Subspace Deformation (SSD) is the most widely used methods in interactive applications especially in the interactive video game industry. The reason for its popularity in the video game industry is its ease of implementation and computational efficiency and produces good visual results during simulation.

Skeleton Subspace Deformation techniques as shown in Figure 2.6 combine the surface geometry of the model with or the internal vertices with the skeleton by assigning a set of bones and a weight for each influence to each vertex. The position of the deformable vertex is obtained by transforming this vertex rigidly related to each of its effect weights and then these weights are used as coefficients to compute a linear combination of these transformed positions as the final position [67].



Figure 2.6: An example of skeletal deformation [68]

In SSD, two main drawbacks are that deformation is restricted only to the subspace and if some of the bones are out of the subspace, it cannot affect the deformation although they should do so. The second drawback is that users are not allowed to manipulate the vertices directly [67; 69].

2.3.2 Physical Modelling

These models simulate the physical behaviour of objects and consider the internal and external forces. These methods are more appropriate in soft tissue modelling and therefore have been used extensively in the medical application such as training. In physical modelling, there are three popular methods for simulating deformable models in real-time simulation. These are the Mass Spring Model (MSM), Finite Element Method (FEM), and the Boundary Element Method (BEM).

These methods are discussed in detail in the subsequent subsections.

2.3.2.1 Mass Spring Model (MSM)

An MSM is a popular approach in physics based deformation. The reason for its popularity is that it is efficient and easy to implement. The MSM considers a 3D surface or volume as a finite set of discrete mass points, also known as nodes. These nodes are then connected with each other through massless springs forming a lattice as shown in Figure 2.7. The total mass of the object being modelled is distributed among these nodes. The deformation occurs when some external forces such as gravity or user-applied forces displace the nodes, and internal forces in the form of spring force. Stronger the spring force, the stiffer is the object it represents.

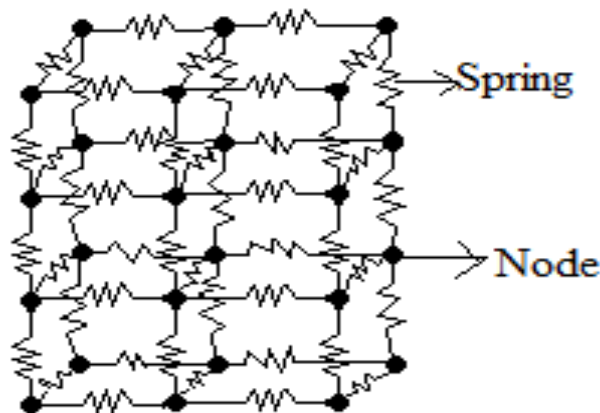


Figure 2.7: A mass spring model with nodes connected through springs

The discretization of the model may be triangular patches in which springs represents the edges of the polygonal surface mesh and, nodes are the vertices as shown in Figure 2.8, [70-76], or T_i - meshes in which each vertex of the mesh is connected with the constant number of $i+1$ next vertices as shown in Figure 2.9 for T_2 - mesh, [77-79]. In the T_i - meshes, T_2 - meshes are chosen because of its resemblance with the triangular meshes and most of the graphics cards are typically optimized for triangular surface meshes [80].

Various numerical methods are used for defining positions of nodes and thus important to retain the shape of the object.

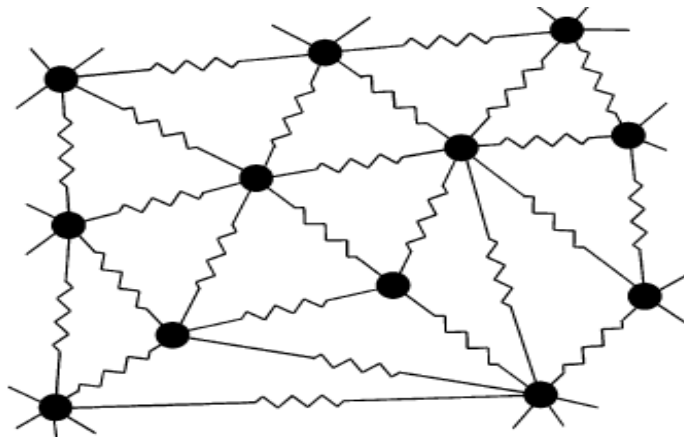


Figure 2.8: A triangular mesh with springs forming triangular elements

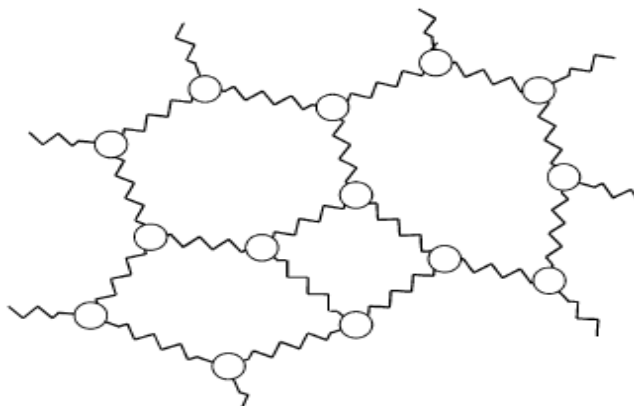


Figure 2.9: A T_2 -mesh having every node connected to its three neighbours

Spring exerts spring force on both of the nodes connected to it. Rest length l is the length of spring in the initial configuration that defines the shape of the body. When a

node is allowed to change its position from u_1 to u_2 and its velocity from v_1 to v_2 then the internal force F_{int} can be calculated according to Hooke's law as [81]:

$$F_{\text{int}}^1 = - \left[k_s (|\Delta u| - l) + k_d \left(\frac{\Delta v \cdot \Delta u}{|\Delta u|} \right) \right] \frac{\Delta u}{|\Delta u|} \quad (\text{Eq. 2.1})$$

$$F_{\text{int}}^2 = -F_{\text{int}}^1 \quad (\text{Eq. 2.2})$$

where F_{int}^1 and F_{int}^2 are the spring forces on nodes 1 and 2 respectively. In an Equation 2.1, k_s is a spring constant and k_d is a damping coefficient for the resistance against the relative motion between nodes 1 and 2. l is the rest length of the spring. $\Delta u = u_1 - u_2$, is the actual distance between two point masses, and $\Delta v = v_1 - v_2$ is the difference of velocities of two point masses. Spring force is proportional to the difference in length of the spring and damping force is proportional to the difference of velocity of the point mass.

As each node is connected through springs to many other nodes in the model therefore, all these neighbours exert forces on this node and this force is the sum of all spring forces acting on it and is calculated using Equation 2.3 [81].

$$F_{\text{int}}^i = \sum_{j \in N_i} k_j \frac{|u^j - u^i| - |u^j - u^i|^0}{|u^j - u^i|} (u^j - u^i) \quad (\text{Eq. 2.3})$$

where N_i is the set of nodes connected through springs with node i , k_j^i is the spring stiffness between node i and node j , while $|u^j - u^i|$ and $|u^j - u^i|^0$ are its current and initial length, respectively.

The dynamics of N nodes in an MSM under the external force F_{ext} is governed by Newton's law of motion and is given in Equation 2.4.

$$M\ddot{u} + C\dot{u} + Ku = F_{\text{ext}} \quad (\text{Eq. 2.4})$$

where M , C , and K are the $3N \times 3N$ mass, damping and stiffness matrices, respectively. The vector F_{ext} is a $3N$ - dimensional vector representing the total

external force on the node. The system has evolved forward through time by rearranging Equation 2.4 as a system of first order differential equations:

$$\dot{u} = M^{-1}(-Cv - Ku + F_{ext}) \quad (\text{Eq. 2.5})$$

$$\dot{u} = v \quad (\text{Eq. 2.6})$$

where v is the velocity vector of the system of nodes. During a simulation, the acceleration, velocity, and position of each node are updated at discrete time points spaced by discrete time steps and therefore known as discrete time system.

There are many numerical integration techniques available which is used to find u and v as a function of time such as finite Euler's differences [71; 73; 82; 83] and Runge Kutta order four methods. The fourth order Runge Kutta solution for the first order differential equation of the form [84]:

$$y'(x) = F(x, y) \quad (\text{Eq. 2.7})$$

is found by iterating the equation:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (\text{Eq. 2.8})$$

Where k_1, k_2, k_3 , and k_4 are constants and calculated as:

$$k_1 = \Delta x F(x_n, y_n) \quad (\text{Eq. 2.9})$$

$$k_2 = \Delta x F(x_n + \frac{\Delta x}{2}, y_n + \frac{k_1}{2}) \quad (\text{Eq. 2.10})$$

$$k_3 = \Delta x F(x_n + \frac{\Delta x}{2}, y_n + \frac{k_2}{2}) \quad (\text{Eq. 2.11})$$

$$k_4 = \Delta x F(x_n + \Delta x, y_n + k_3) \quad (\text{Eq. 2.12})$$

Runge Kutta method is more stable than Euler's methods and because of its stability, larger time steps can be used in the simulation. The acceleration factor of two with respect to Euler's method could be achieved when the Runge Kutta method is implemented in the simulation [85-87].

The pioneering work and giving the name deformable models for the first time goes to [88; 89]. In [19], the author applied dynamic mass spring systems to facial modelling. As a human face have three layers of tissue namely the dermis, a layer of subcutaneous fatty tissue, and the muscle layer. They constructed a three-layer mesh of nodes as shown in Figure 2.10.

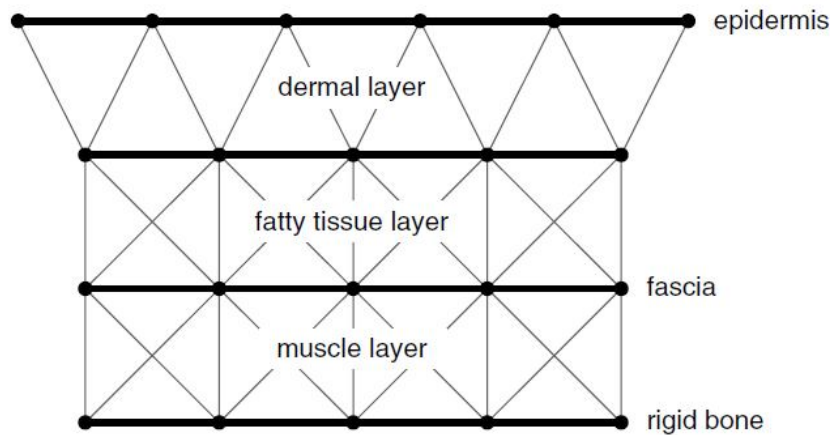


Figure 2.10: A three-layer mesh representation of skin used by [19]

Different spring constants were used to model different tissue layers based on tissue properties.

After the pioneering work done [88; 89; 19], this approach has been used for cloth simulation [90; 91], face animation [92], and importantly for soft tissue behaviour modelling in surgery training simulator.

In the field of surgery and training simulators, the authors in [14], were the first who presented real-time models for surgery simulation. They used a simple surface based MSM for simulating the real-time deformation of the gallbladder. The authors in [93], made a prostate model. This model is based on MSM for the underlying physical properties of prostate. The model simulates tissue deformation in real-time environment due to resection. The authors in [94] and [20] applied the MSM to the simulation of interactive cutting of deformable anatomy. The mass spring model is used for both visual and haptic rendering of progressive suturing in [21]. Real-time simulation of heart deformation is also implemented using the mass spring model in

[22]. Their model consists of an inner and outer surface. The two surfaces are also connected with each other to simulate the forces between surfaces. They later incorporated the model with GPU (Graphics Processing Unit) acceleration, for simulation of operations such as cutting and opening of the heart [95]. They provided a powerful tool for cardiac surgeons to examine the anatomical structure of the patient's heart easily, and to practice possible surgical operations before real surgery.

The authors in [96] proposed an improved mass spring model for surgery simulation. Although the model is a mix of mass spring model and FEM for nonlinear, incompressible, and anisotropic soft tissue but it maintained the advantages of mass spring models such as simple architecture, low memory and fast computation. The model is used to simulate the human kidney at an update rate of 100 Hz for a mesh of about 1000 tetrahedrons (on a PC Intel ® Core™ quad 2.40 GHz, 1GB RAM, and an NVIDIA GeForce 8800GTX GPU).

Basafa and Farahmand [97] used mass spring-damper model for real-time simulation of soft tissue deformation due to laparoscopic surgical indenter and the algorithms developed to achieve an update rate of 150 Hz for the haptic manipulation.

Despite its extended use in various types of deformable models, there are unsolved problems with the mass spring model for modelling soft tissue. There are two main problems in a mass spring model for modelling soft tissue.

- As soft tissue is nonlinear, anisotropic, and inhomogeneous, therefore setting its stiffness parameters is a big challenge.
- Further problems can occur during the simulation of stiffer materials. Large values of stiffness constants if used can lead to numerical instability and small step is a compromise on performance.

Research is still ongoing about addressing the first problem of parameter finding for the mass spring model. Etmuss et al [98], established a link between mass spring model and the classical theory of elasticity. Bianchi et al., [99] used genetic optimization algorithm to find the topology of the mass spring system. D'Aulignac et al., [100] used a layer of 2D linear springs and dashpots supported by nonlinear springs is used in the dynamical deformation of the human thigh and is simulated.

Baran and Basdogan [17], used a novel approach which first conduct static indentation, stress relaxation experiments on the Finite Element (FE) model to record force versus displacement and force versus time responses of the surface nodes, and used this data to calibrate the mass spring model. Natsupakpong and Cenk [18], used a method to determine mass and spring constants of the mass spring model using a FEM as a reference model by minimizing the error of stiffness matrices of FEM and MSM through an optimization. However, practically acquiring such a reference model is also a problem in itself.

The second problem with the mass spring model is numerical instability. Increasing the stiffness value for modelling stiffer materials can lead to instability problems and the system blows up after a few iterations. One solution to this problem is to use small step size but in doing this way will increase computational burden, which is not feasible in real-time interactive simulation.

2.3.2.2 Finite Element Method (FEM)

FEM is the most common and accurate method for modelling soft body behaviour. It formulates soft tissue deformation in a continuous domain. It is directly dependent on the theory of elasticity and therefore provides a good approximation for the complex geometry of deformable models. The main principle of FEM is that it models the stress-strain relationship of soft object by discretizing it into a set of polygonal or polyhedral elements (usually triangular elements in 2D and tetrahedral elements in 3D) as shown in Figure 2.11. At the boundaries of the elements are the nodes.

The displacement of any point within an element is achieved by interpolating the nodal displacements of that element. Deformation of each element is first evaluated based on the stress-strain relationship and then assembled to yield the deformation of the original object [16].

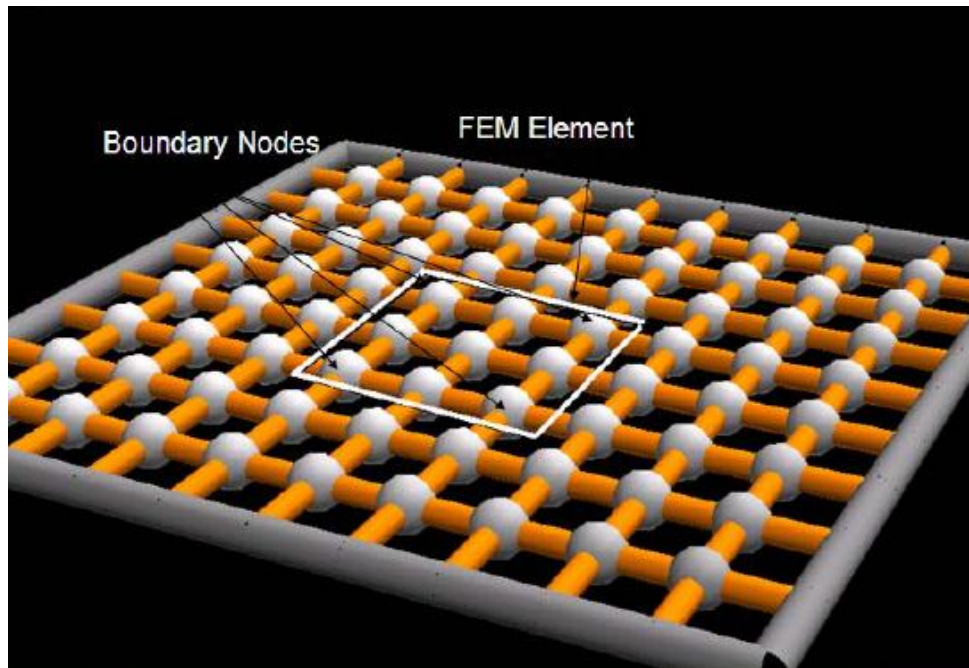


Figure 2.11: An illustration of elements and its boundary nodes [101]

Both the linear and nonlinear FEM have been used for modelling the mechanics of soft tissue. These linear and nonlinear FEMs cannot be simulated in real-time in its pure form due to the steps involved in setting-up and running a finite element calculation. Therefore, researchers in both linear and nonlinear FEM have focused efforts on optimizing finite element based computational techniques to be applicable in real-time simulations.

a) Linear Finite Element Methods

Linear elasticity based FEM are the most widely used techniques for modelling soft tissue deformation in surgical simulations. Linear FEM considers the linear material mechanics. It is computationally fast and stable simulation is possible because the stiffness matrix is constant. In addition, most of the computation is done in the offline before actually starting real-time simulation, which further guarantees efficiency [102]. In this method, only two constants Young's Modulus (E) and Poisson's Ratio (ν) are required to describe isotropic and homogeneous materials, therefore

implementation is simple and computationally efficient and hence enables real-time haptic rendering. This method is most suitable in simulations where small deformation is required because it can model deformation accurately up to 1%.

Bro-Nielsen was one of the first researchers to apply the FEM for real-time soft tissue modelling [103]. In [104], the authors proposed a condensation method. By using the condensation method, he removed the internal nodes from the matrix, which allowed selective matrix multiplication and therefore minimized the computational cost. Although the model was volumetric, they simulated only the surface nodes and ignored the internal nodes. In this they achieved a refresh rate of 20 frames per second for a human leg from the Visible Human Data Set made of 700 system nodes (only surface nodes), when the force was applied to three nodes. The simulation was implemented on Silicon Graphics ONYX with four MIPS R4400 processors using the SGI Performer library. However, in those cases when the topology of the model changes then with this change in topology, the stiffness matrix needs to be updated which affect the performance of the simulation.

A novel method is proposed for deforming a volumetric model in real-time as in [105]. Linear static model is used which allows haptic interaction in real-time. FEM is built on a mesh of tetrahedral elements. They used two types of nodes, fixed which cannot be displaced in space and free nodes, which are free to move upon applying force during haptic interaction. The reaction forces were computed by FEM using the formulation of the elasticity theory. In order to decrease the memory for storage of tensor, only those tensors were kept in the memory whose maximum eigenvalue was greater than the predefined threshold. In this way, for a live model of 940 nodes, 60% memory consumption was reduced. To achieve interactive simulation, the computation of displacement field and the reaction forces were reduced to a linear combination. The real-time interactive deformation of liver during surgery with a force feedback device is shown in Figure 2.12.

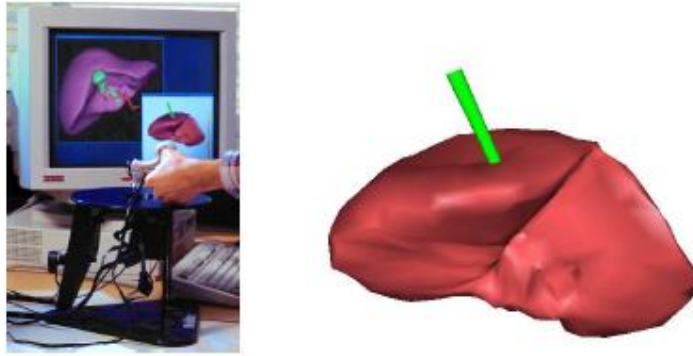


Figure 2.12: Left: Surgeon moves a virtual tool via force feedback device, Right: Real-time deformation of liver [104]

In [104], the authors developed real-time haptic surgical simulation based on a modified FEM. The huge computations were performed during the preprocessing stage of finite element calculation and the results were stored in files to be used directly in the simulator. They also developed three-dimensional anatomical models of liver using data from computed tomography and used linear elasticity based modelling to simulate its real-time deformation. They achieved a high frequency ($>300\text{Hz}$) for touch sensation in the simulator. The surgeons tested the simulator and proved that generated sensations were close to reality.

Rhomberg [106], and Székely, et al., [107], have been proposed a parallel framework for fast soft tissue deformation. The linear system representing internal forces that were assembled and solved at the elementary level. Then the forces between elements were balanced using interconnecting nodes. As the framework is parallel, each finite element model can be assigned to a different processor computing the internal forces of several elements. The different processor has communication with each other therefore if forces between several elements can be balanced by interchanging the data. For an efficient communication between processors, the optimal distribution of elements among processors is required.

The model described above was also implemented in Endoscopic Gynaecology. The main goal was to achieve haptic real-time interaction using a PHANTOM haptic device and a cluster of PCs performing the computations. The prototype of the

simulator was described in [107]. The simulation of interactive deformation is shown in Figure 2.13.

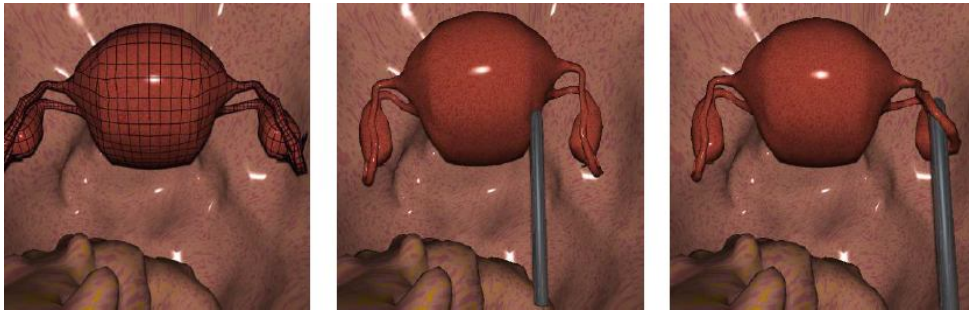


Figure 2.13: Left: A Finite element model of Uterus containing 200 elements. Middle and right: Simulated deformation of the corpus and Fallopian tube of the uterus [107]

The above model was used in the project named as Epidaure and were successfully implemented in the haptic surgery simulator, using Alpha station for computations and Pentium station with Laparoscopic Impulse Engine for interaction [108].

A static model of linear elastic finite element model for simulating real-time interaction in haptic virtual environment was presented in [109]. To achieve the higher refresh rate for the real-time haptic interaction, the small area paradigm is introduced. It is assumed that the area affected by the haptic device is small. This made the computation faster and possible for real-time simulations. They simulated a model made of 10,000 tetrahedra on Silicon Graphics ONYX2 with eight 250MHz processor and showed good results during real-time interaction. The combination of pre-computation and condensation is proposed in [110]. The complex computation of finite element equation was performed offline and the results were stored in a file for on-line usage. This method increased on-line simulation process by a factor of one thousand. The model can be deformed interactively at a graphics speed of 20 frames per second using SGI/Onyx2 300 MHz MIPS R12000. The authors in [111] developed a banded matrix technique for FEM in real-time deformation and force feedback. Banded Matrix is used to optimize finite element model for calculation in real-time interaction with soft tissue. In this technique, only surface nodes were considered and internal nodes were ignored. The author reported that banded matrix

method was superior to conventional finite element equations based on calculation time requirement. The limitation of this technique is that the original mesh could be changed while running the simulation.

A robust adaptive method for simulating dynamic deformations of a viscoelastic object in real-time was developed in [75]. They used a novel automatic space and time adaptive level of detail technique, along with a large displacement (Green) strain tensor formulation. In this method, they partitioned the model into a non-nested multiresolutional hierarchy of tetrahedral meshes. The model is refined at the point of deformation. The user interacts in real-time with the dynamic object through the control of a rigid tool, attached to a haptic device driven by forces derived from the method. Their method ran in the visio-haptic simulator. The technique is shown in Figure 2.14 in which different models are deformed using a force feedback device.

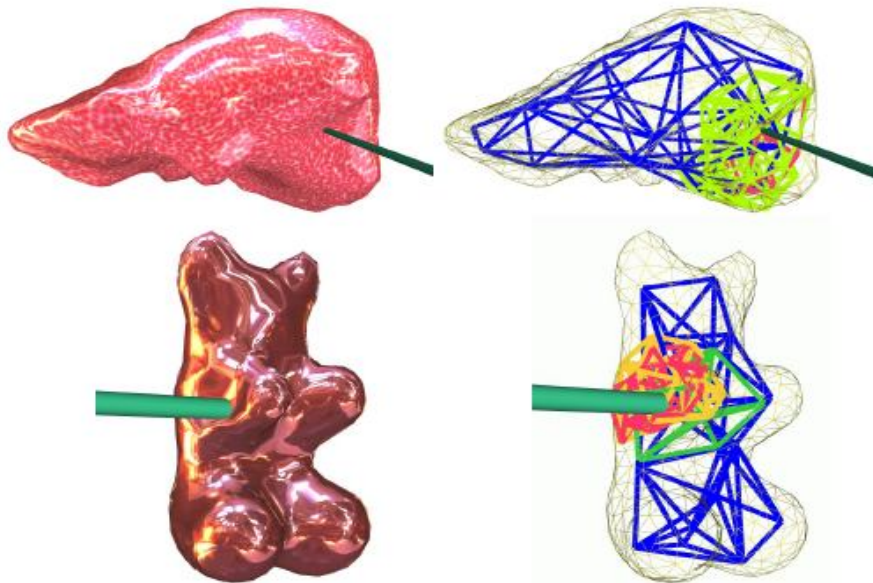


Figure 2.14: Top Left: Liver model deformation. Top right: Multi-resolution model of liver. Bottom Left: Toy model deformation. Bottom Right: Multi-resolution model of a toy [102]

A linear viscoelastic model was developed in [25]. The approach was based on pre-calculation phase: first, the force response of a surface node and the

displacements of the neighbouring nodes were recorded for a unit step displacement, being applied for 30 seconds, the recovery of the nodes during 30 seconds was computed and stored after 1 ms unit step force was applied. The two sets of the data were then used for stable haptic interaction.

b) Nonlinear Finite Element Methods

The physical behaviour of soft tissue may be considered linear if the deformation of soft tissue remains small i.e. less than 10% of the total mesh size [112; 113]. As soft tissue is nonlinear therefore linear FEM are not suitable for soft tissue modelling. Therefore, limitations of linear FEM is that it is not suitable for simulating the nonlinear behaviour of soft tissue and large deformation. To achieve high realism, considering nonlinearity is important although difficulties in implementation.

Authors in [114], proposed a real-time deformable model based on nonlinear elasticity and anisotropic behaviour and finite element method. By introducing this nonlinear elastic component, they overcome the elongation limit of 10% of the total mesh size with reasonable accuracy. The elasticity of soft tissue was modelled using St Venant Kirchhoff elasticity model. To simplify the finite element equations, the mass lumping method was used and the explicit scheme was used for real-time simulation. The results of the deformable model concerning rotation showed that nonlinear model was more realistic than the linear model. The constraint of incompressibility was also added which posed check on the growth of individual elements. In case of force computation, nonlinearity is included in the area of contact with the soft tissue while the rest of the model is treated as linear. The method works as; for each vertex in the model, first linear part of the force is computed and then the nonlinear part is included only if the displacement of the vertex is larger than a threshold. In this way, a refresh rate of 25 Hz for a mesh of 2000 tetrahedra was achieved. The realistic haptic interaction was provided using force extrapolation. The main drawback of this model is however, the computation time required. Going beyond the threshold during simulation there is a shift from a linear model to fully nonlinear model and the update rate due to this shift for the liver model they simulate

dropped from 45Hz to 8Hz, which is an 82% decrease in the update rate. In real-time, however this can be an expensive choice.

The authors in [115], proposed an adaptive refinement scheme based on dynamic progressive meshes for realistic simulation of deformable objects using nonlinear FEM. They considered both geometric and material nonlinearities. Most of the computation is performed offline for a short time to be used during real-time mesh refinement. For this purpose, they used a hierarchical mesh structure, and the geometric and finite element parameters were pre-calculated offline for each level of the hierarchy. In their method for stability, they used explicit time integration scheme with the central difference method. Modelling a liver tissue made up of a mesh of 2200 vertices was simulated at a refresh rate of 20 frames per second on an 800 MHz Pentium III PC as shown in Figure 2.15.

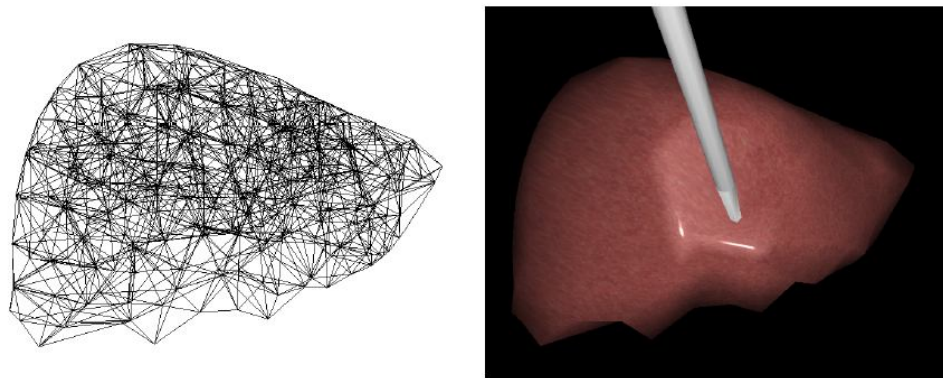


Figure 2.15: Tetrahedral liver model simulated using nonlinear FEM [115]

Although both types of nonlinearity were considered, the main limitation of the concept was redistributing vertices for optimal mesh quality.

The authors in [23] proposed a graphics processing unit (GPU) based algorithm for interactive simulation of nonlinear soft tissue. They implemented nonlinear total Lagrangian explicit dynamics (TLED) algorithm on the GPU. They compared their GPU based algorithm with the CPU (Central Processing Unit) only based algorithm and found speed improvement of 18.6X.

A suite of finite element algorithms for accurate computation of soft tissue deformation is presented in [116]. The main theme of these algorithms is to use the total Lagrangian formulation for the solution of finite element problems. The algorithms also cover important issues that were related to time integration, hour glassing, volumetric locking, and contacts. Nonlinear model was used for large deformation. Reaction forces of 1000 Hz for a mesh containing 2000 hexahedral elements were achieved during simulation on a simple PC workstation.

Interactive deformation using modal analysis was presented in [24]. It transforms the nonlinear equations of a system into a good linear approximation and then finding a coordinate system that diagonalises the linear approximation. This method deformed complex models in interactive manner. However, for long, thin, and highly deformable objects where nonlinearity dominates, this method is not feasible.

An approach based on pre-computation is proposed in [117]. This method allows interacting with complex objects having nonlinear geometric properties in real-time using force feedback devices. Although geometrically nonlinear but the model material properties they used were linear. A new computational approach was presented in [25], for visio-haptic interaction with a finite element model of a human liver. The model has nonlinear material and geometric properties. This method works in two steps: a pre-computation of the configuration space of all deformation configurations of the model and the interpolation of the precomputed data for the calculation of the nodal displacement and reaction forces that are displayed to the user through the real-time interaction through a visual display and haptic device, respectively. The simulation using this method is shown in Figure 2.16.

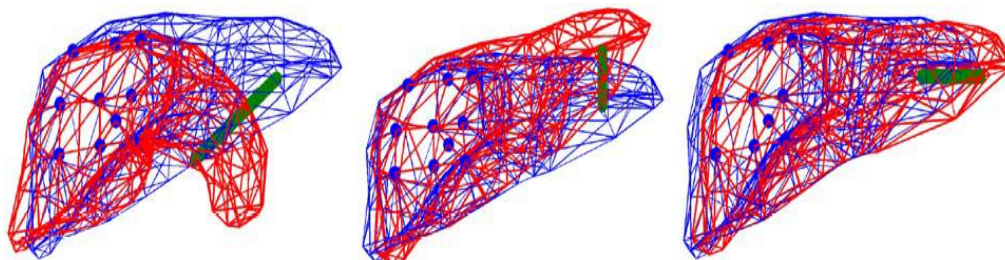


Figure 2.16: Different deformation profiles during a simulation [25]

The main limitation of this method is computational time required for the pre-computations. Also computing configuration spaces for each model and parameter setting is inconvenient.

A hybrid technique for real-time haptic interaction with the nonlinear human liver model is presented in [26]. In this method model order reduction (MOR) known as Karhunen – Loeve was used to reduce system complexities. The refresh rate for visual and haptic was achieved for liver simulation.

2.3.2.3 Boundary Element Method (BEM)

In this method, the surface of the deformable object is discretized into a finite number of elements. The interior of the object is not discretized which results in reducing the size of equations to be solved as compared to FEM. The BEM has an edge in the VR applications that requires real-time interaction with the forces being applied to the deformable objects.

The authors in [118], used BEM for the real-time simulation of linear elastic model incorporating the force feedback device and suggested surgical simulation as an application. In [119], the authors designed a VR simulator for neurosurgery based on BEM also with the force feedback device. They implemented operations such as prodding, pinching, and cutting of deformable objects.

Although many techniques for deformable model exist, there are no one techniques which has all of the characteristics i.e. speed, robustness, physiological realism, and topological flexibility which is needed in virtual reality applications such as surgical simulation [120].

2.4 Choice of Modelling Technique

The above section discussed many techniques, but mass spring model and finite element method are two main techniques for modelling soft tissue behaviour. Choosing any one technique depends upon the application. Each method has its own advantages and disadvantages. However, in case of interactive deformable models

only computation time is the most important factor to consider. The other factors are accurate responding, mechanical feedback, real-time simulation, nonlinear and non-homogenous model, and type of user interaction such as deformation, prodding, and cutting.

If an accurate response and mechanical feedback are the requirement in the simulation, then both finite element method and mass spring model can be used. Also for linear and nonlinear material properties, both methods could be used. However, the key factor in choosing a modelling technique is the computation time required in real-time interaction. The model must fulfill real-time performance while not comprising on the accuracy as well.

To compare the performance of the mass spring model with linear FEM, Zhou et al., [121] showed that computation time for a linear elastic finite element was 45% more than the one observed in mass spring model for small deformation. Now to compare the mass spring model with nonlinear FEM, we need to see the work of Rasmusson, Mosegaard, and Sorensen [122], and Comas et al., [123]. In [122] for a heart modelled using mass-spring-damper system with 88347 degrees of freedom, their implementation of GPU reached 5515 frames per second. On the other hand in [123], a GPU implementation of a TLED using a cube with 89373 degrees of freedom, obtained a simulated performance of 1000 frames per second. These two results show that the mass spring model performs 5.5 times faster than the nonlinear FEM under the same conditions.

Therefore considering the above facts it is clear that the mass spring model is a good choice for soft tissue modelling. In this thesis, mass spring model is considered for soft tissue modelling as well.

2.5 Levels of Detail

Due to the ease of implementation and efficient, mass spring model has been used extensively in the animations and real-time interactive applications [124]. Simulation and the accuracy of the model in mass spring models are largely dependent on the number of nodes and springs used in the simulation. In each time step these nodes and

springs needs to be updated. In an interactive real-time simulation using the force feedback device, there is a very short time for updating nodes and springs. Therefore, if the number of nodes and springs are increased in the model, then the realistic simulation using the mass spring model is even a challenge.

In case of complex mass spring model, the nodes affected by the external force are only a subset of the total nodes. The rest of the nodes usually remain stationary or oscillate invisibly. When all the nodes of the model are used for the computation of deformation, the numerical computation becomes larger and becomes a hinder in the real-time interactive applications. In most of the real-time interactive applications, the user is mainly interacting with the small portion of the model through a force feedback device. Therefore, levels of detail (LoD) methods are combined with the physics based methods in order to reduce the computational load [125].

The first attempt made in surface based deformable modelling is the simulation of draped cloth by refining the cloth in the region of high curvature [126]. An adaptive mass spring model was used for cloth representation. In this method, there was no simplification method. The authors in [92] used loop subdivision to refine the region of high deformation in the rendering of physics based facial animation system. They applied their proposed method to surface based deformation.

In volumetric based deformable models, authors in [127; 128; 75] used to combine linear finite volume based mechanical model with the non-hierarchical refinement method, which was based on a Voronoi's concept and was adaptive in nature in both time and space. They used ghost nodes, which transmitted the mechanical behaviour of the model from one resolution level to the next different resolution level. An adaptive scheme used with nonlinear FEM is proposed in [115]. The proposed scheme for mesh adaptation was based on an extension of the progressive mesh concept. A diagonal mass matrix was produced to allow real-time computation. The authors in [129], used a similar strategy as in [127] to represent the deformation of complex objects using a hierarchical basis that was constructed using the volumetric subdivision. The authors in [130], also used a hierarchical approach in which a hierarchy of mass spring models was built in order to manipulate deformable objects. In this method from coarse to refine representation of the mass spring model

were used based on the degree of deformation. The authors in [125], used an adaptive deformable surface model using the mass spring model and butterfly subdivision scheme for the mesh refinement. They also proposed a model simplification method for bringing back the deformable model to the coarse model once the external force is relaxed. The refined model used changes the topology of the deformable model. In addition, the area is refined irrespective of the triangular area in the region of interest. The authors in [131], proposed a Delaunay triangulation -based online re-mesh deformation model. However, the computation complexity of the model is $O(n^3)$, where n is the number of nodes in the mesh, is therefore, not suitable for real time interaction. The authors in [132], proposed a multi-rate rigid body simulator. However, this approach was not applied to elastic objects.

In this thesis, two visio-haptic deformable models are proposed. One is the vertex based visio-haptic deformable model using slope intercept form of a line equation. The model mainly developed for visual interactive deformation while having the same haptic sensation as in [30; 31; 58], and is used on the surface based triangular mesh model. A slope intercept form of a line equation is used which computes the deformation quickly in the haptic rendering as compared to [52; 61]. Second visio-haptic deformable model is based on the levels of detail, namely, an adaptive area based visio-haptic deformable model, which refines the region of interest based on the triangle's area in that region. The current methods refine the region of interest uniformly [133; 125]. Furthermore, the visio-haptic deformable model works on the volumetric mesh model made up of tetrahedral mesh, in order to represent the internal physical properties of the soft tissue as compared to [125; 133], which works on the surface based mesh models. Once after refinement of the region of interest, a modified scheme based on [125; 133] is presented which adjust the physical parameters of the model. In this thesis, the scheme of parameter identification is extended and is applied to the volumetric mesh model. The parameter for refinement of the region of interest depends on the threshold of the external force applied by the trainee using force feedback devices during the real-time interaction with the model. In addition, when the external force is relaxed in the region of interest, this region is returned to the original coarse model and physical parameters are readjusted.

2.6 Summary of Related work

The following tables summarize the previous work in visio-haptic modalities for learning skills enhancement (Table 2.1), veterinary palpation simulators (Table 2.2), Visio-haptic deformable models (Table 2.4), and levels of detail (Table 2.4).

Table 2.1: Previous work related to learning skill enhancement due to multimodal cues

| Multimodal Cues | Author | Year | Modality Comparison | Result | Hard Object | Soft Object | Short-term | Long-term |
|-----------------|-------------------------------------|------|---------------------|---|-------------|-------------|------------|-----------|
| | Hang et al. [13] | 2004 | V,H, & VH | VH better | √ | X | √ | X |
| | Jones et al. [38] | 2005 | V,H & VH | VH better | √ | X | √ | X |
| | O'Malley et al. [40] | 2006 | V, H & VH | VH better | √ | X | √ | X |
| | Morris et al. [10] | 2007 | V, H & VH | VH better | √ | X | √ | X |
| | Kalenine et al. [9] | 2011 | V, H & VH | VH better | √ | X | √ | X |
| | Bara & Gentaz [41] | 2011 | V & VH | VH better | √ | X | √ | X |
| | Srimathveervalli & Thenkurussi [11] | 2005 | V, H & VH | VH not better | √ | X | √ | √ |
| | Liu et al. [42] | 2006 | H, V | Same | √ | X | √ | X |
| | Xing-Dong et al. [12] | 2008 | V, H & VH | Long-term VH not sig Short-term VH sig | √ | X | √ | √ |
| | Huegel et al. [13] | 2010 | V, H & VH | VH sig in short-term, VH not sig in long-term | √ | X | √ | √ |

Table 2.1 shows that most of the work done for the learning skills enhancement dealing with the hard objects. Soft objects, which are highly deformable, are different from the hard object during real time interaction. Furthermore, most of the previous work considers the short-term learning skill enhancement. A few discussed the long-term learning skills enhancement [11], [12], and [13], and showed contrast results for long-term skill enhancement. Therefore, visio-haptic modalities need to be evaluated in palpation simulator, which is deformable, for the long-term learning skill enhancement.

Table 2.2: Previous work related to veterinary palpation simulators

| | Author | Year | Simulator Name | Limitation |
|----------------------------|---------------------|-------------|----------------------------|---|
| Palpation Simulator | Crossan et al. [28] | 2001 | HOPS | 1. Non-Deformable visual model 2. Haptic sensation is based on Hooke's Law |
| | Baillie et al. [30] | 2005 | Haptic Cow | 1. Non-Deformable visual model 2. Haptic sensation is based on Hooke's Law |
| | Parkes et al. [58] | 2009 | Feline Abdominal Palpation | 1. Non-Deformable visual model 2. Haptic sensation is based on Hooke's Law |
| | Baillie et al. [31] | 2010 | Automated Palpation | 1. Non-Deformable visual model 2. Haptic sensation is based on Hooke's Law |

In Table 2.2, the work done in the field of palpation simulator is presented. The previous work has the non-deformable interactive visual model, and the haptic sensation is based on Hooke's Law. Further investigation is required in order to evaluate the effect of interactive visual deformation, and the haptic sensation based on the underlying physical properties on the learning skills enhancement.

Table 2.3: Previous work related to visio-haptic deformable models

| | Author | Year | Limitations |
|---------------------------------------|--------------------------|-------------|--|
| Visio-Haptic deformable models | Song & Yang [66] | 2005 | 1. Not suitable for real time interaction |
| | Ji [52] | 2008 | 1. More computation for visio-haptic deformation 2. Surface based model |
| | Taylor et al. [23] | 2008 | 1. Used special hardware for processing (GPU) |
| | Baran & Basdogan [17] | 2010 | 1. Calibration of Mass Spring Model is time consuming |
| | Natsupakpong & Cenk [18] | 2010 | 1. Mass spring model is calibrated from Finite Element Model. Practically acquiring such a model is a problem itself |
| | Peterlik et al. [25] | 2010 | 1. Computational time required to pre-computations 2. Computing configuration spaces for each model and parameter is inconvenient |

In Table 2.3, the limitations of the previous visio-haptic deformable models are presented. For the evaluation of visio-haptic cue, efficient visio-haptic deformable models are required in order to satisfy minimum rendering requirements of the visual (30 Hz) and haptic (1000Hz). In this thesis, two visio-haptic deformable models are presented, one is vertex based visio-haptic deformable model for surface based object representation, and second is adaptive area based visio-haptic deformable model for volumetric model representation.

Table 2.4: Previous work related to levels of detail

| | Author | Year | Limitations |
|-------------------------|----------------------|-------------|--|
| Levels of Detail | Hutchinson [126] | 1996 | 1. No simplification method |
| | Kahler[92] | 2001 | 1. Only in graphics rendering 2. Surface based model |
| | Debunne et al.[75] | 2001 | 1. Different resolution models are connected with share nodes, additional computation cost is required |
| | Payandeh et al.[133] | 2005 | 1. Uniform refinement 2. Surface based model |
| | Choi et al.[20] | 2005 | 1. Uniform refinement 2. Surface based model |
| | Paloc et al.[131] | 2006 | 1. Computation complexity is $O(n^3)$ |
| | Susa[132] | 2011 | 1. Not applicable to elastic objects |

Although refinement techniques in the field of computer graphics are very mature, Table 2.4 only shows the previous work of refinement techniques in the field of graphics and haptic. An adaptive area based technique is proposed that refines the region of interest based on the area of a triangle in the region. This produces a lesser number of faces, node, and spring and greatly improves the performance.

2.7 Summary

This chapter discussed related work in three different areas, which are multisensory modality, palpation simulator, and deformable modelling techniques. In the first section, multisensory modality is discussed. Multimodal cues are informative and help in skill learning [8], but the previous work shows conflicting results especially for

haptics and vision modalities. Therefore, multimodal cues need further evaluation in a haptic dominant environment for the skills learning enhancement. In the second section, a detailed literature review of palpation simulators is presented. Palpation simulators, in general, are reviewed as well as in the field of veterinary. For the veterinary students, currently available training simulators are HOPS, and bovine rectal palpation simulator. These simulators have a static visual model, and simple haptic sensation based on the Hooke's law, and during training the focus is on the haptic sensation. The interactive visual deformation part is missing which could allow the instructor to better control the palpation forces of the trainee during an interactive simulation. Although multimodal cues, including visual and haptic sensation are computationally expensive, however, multimodal cues are more helpful than any single cue for skills learning enhancement Fredembach et al., [8]. Furthermore, despite its single cue dominance, an investigation of multimodal cues for enhancing learning skills is required in general for the palpation simulators, and in particular in the bovine rectal palpation simulator. Such an investigation has not yet reported in the domain of palpation simulator.

In the third section, various modelling techniques were presented. The techniques mainly discussed here are in the field of real-time simulation. The most widely used techniques are MSM and FEM. Finite element is the most accurate method for soft tissue modelling but computationally more expensive. Therefore, it is not a candidate for real-time simulation in its pure form. Many optimization techniques such as condensation, graded mesh, and dynamic adaptive mesh has been proposed for making finite element modelling to work in the real-time simulation. For pure deformation such as palpation, finite element is too costly in terms of computation, as it requires constant updating matrices. On the other hand, the mass spring model is very efficient in terms of computation and this feature makes a good candidate for real-time simulation. In real-time visual and haptic simulation, performance is the most pronounced factor. Therefore, the mass spring model has been widely used for real-time interactive simulation.

In computer based palpation simulation, to ensure realistic visio-haptic modalities, the models need to be deformable visually as well as haptically based on the underlying physical properties of the real model it represents. Efficient algorithms

must be developed for the deformable models having visual and haptic modality in order to improve realism in the virtual reality (VR) environment without using any costly special hardware. The algorithms developed so far are application specific and there are no such universal algorithms suitable for most of the applications [52]. Finite Element Method (FEM), [116; 110; 23], and a Mass Spring Model (MSM) [134-136; 96; 97] are the two most widely used techniques for representing and simulation of the soft tissue deformation. These techniques can provide realistic deformations, but have poor computational efficiency.

Although the mass spring model has an edge over FEM but the main drawback of an MSM considered is its lack of proper parameter identification. It is shown that current research has mainly focused on this issue and many techniques proposed have been discussed. Research is still open in this area to find proper parameters for the MSM in general.

In the last section, an overview of the refinement techniques was discussed. These techniques combined with the physics based deformable model greatly improve the computation time in the real-time interactive simulation. Previous techniques are for surface based deformable models and volumetric models. In real-time interactive simulation, the user interact with the soft tissue, therefore during interaction with the region of interest, that region is refined. Once the user left the region of interest, that portion needs to be reverted to the original coarse model. Most of the techniques available mainly focus on the refinement approach and often neglect the simplification methods. In addition, when the levels of detail are combined with the physics based method; such techniques require the dynamic adjustment of the parameters of the mass spring model among different levels of detail during simulation. In the next chapter, the research methodology and approach used in this thesis are discussed in detail.

CHAPTER 3

RESEARCH METHODOLOGY

The purpose of this chapter is to indicate the research methodology and research method for achieving the research objectives discussed in chapter one. In this chapter, the overall methodology of the thesis is described. In this thesis, mixed paradigm is used which allows choosing research methodologies from both of the basic paradigm. The underlying principle provided by Leech and Onwuegbuzie [137] suggests that the main benefits of mixed methods research come from capitalizing on the strengths of the methodologies and reinforcing one's techniques with the techniques of the other, in comparison to either quantitative or qualitative methodologies. Research approaches used in the mixed paradigm are literature - review, formulative - algorithm, evaluative - interpretive, experimentation [138].

An appropriate research methodology and research method used in this work will be selected based on the research objectives outlined in chapter one. Research methods used in this thesis are developmental, simulation based performance evaluation, experiments and statistical analysis techniques. These are shown in Figure 4.1, and are discussed in the subsequent sections.

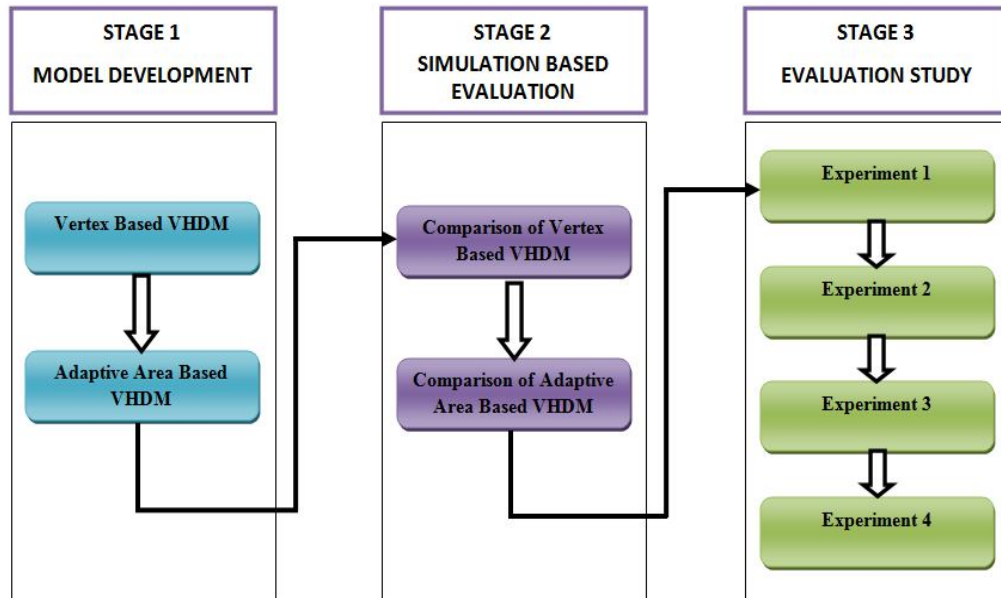


Figure 3.1: Research Methodology Framework

3.1 Model Development

This phase of the research methodology deals with the visio-haptic deformable model development. The simulators developed for training the veterinary students are the HOPS [28], Haptic Cow [30], and Feline abdominal palpation simulators [58]. These simulators lack the interactive visual deformation. In addition, the haptic sensation needs to be based on the underlying physical properties of the soft tissue. In the domain of the palpation simulators, which are haptic dominant, further investigation is required to evaluate the effectiveness of the multimodal cues [4]. In order to develop realistic vision-haptic deformation, and investigate the effect of multimodal cues on learning skills enhancement, two visio-haptic deformable models developed are (i) a vertex based visio-haptic deformable model, and (ii) an adaptive area based visio-haptic deformable model and are discussed in detail in chapters four and five respectively of this thesis. Reason for the two model development is the assessment of visio-haptic cues for the learning skills enhancement. For fast access to the node in the region of interest, efficient methods are required. For this purpose, an octree space partitioning (OSP) technique is implemented in order to quickly access the desired node in the 3D model. Further, for improving fast rendering of the model in the haptic

loop, the neighbourhood algorithm is developed, and used for improving the performance in terms of rendering in the haptic loop.

Two of the research objectives as stated in chapter one, i.e. (i) To develop visio-haptic deformable models for a palpation simulator that satisfy minimum rendering requirements, and (ii) To implement the models into a haptic dominant palpation simulator could be achieved upon completion of the research method in this phase.

3.2 Simulation Based Performance Evaluation

In this step the evaluation of the proposed visio-haptic deformable models is conducted based on the execution time in the haptic and graphics loop. The visio-haptic deformable models developed are implemented in the real-time interactive simulator and the results are compared with the corresponding existing techniques. The results are presented in chapters four, and five, and then are discussed in chapter seven.

3.3 Experiments

According to [139], experiments are "*only experience carefully planned in advance, and designed to form a secure basis for new knowledge*". The most important characteristics of an experiment are (i) manipulation of one or more independent variables; (ii) use of controls such as randomly assigning participants or experimental units to one or more independent variables; and (iii) careful observation or measurement of one or more dependent variables [140].

3.3.1 Types of an Experiment

There are three types of experiments [141].

3.3.1.1 Laboratory / Controlled Experiment

This type of experiment is conducted in a controlled environment, not necessarily a laboratory. However, accurate results are possible. The advantage of this type is easy to replicate, and controlled the extra independent variables.

3.3.1.2 Field Experiment

Such experiments are performed in real life, while still manipulating the independent variable. The advantage is the experiment real reflection due to its natural setting. The disadvantage is that it could suffer from contamination [141].

3.3.1.3 Natural Experiments

These experiments are conducted in the real life environment of the participants but the experimenter has no control over the independent variables. The main limitation is not controlling the extra independent variable that may affect the results of the experiment [141].

In this thesis, laboratory experiment is used. Four experiments have been conducted for the last two research objectives outlined in chapter one, i.e. (i) to evaluate the haptic only and, visio-haptic parameters of the simulator for learning skill enhancement, and (ii) to assess the extent in which the visual and haptic cues are important in the visio-haptic modality in terms of the visual and haptic rendering requirements. In three of these experiments, three prototype simulators will be used. One containing haptic only, and named as prototype-A, the other two contains visio-haptic modalities, named prototype-B, and prototype-C. In prototype-B, the visio-haptic deformable model discussed in chapter four, namely, a vertex based visio-haptic deformable model is implemented. In prototype-C, the visio-haptic deformable

model discussed in chapter five, namely, an adaptive area based visio-haptic deformable model is implemented. The first experiment is conducted for evaluating the visual cue effect on skills learning enhancement in an instructor-trainee interactive environment when the instructor as a guidance tool uses visual cue for controlling the palpation forces applied by the trainee. The second experiment evaluates the effect of visio-haptic cue for short-term memory as compared to the haptic only for learning skills enhancement. The third experiment evaluates the visio-haptic cue for long-term memory as compared to the haptic only for learning skills enhancement. The fourth experiment is conducted to evaluate the effect of the extent of visual cue on the enhancement of learning skills.

In the first three experiments, three prototype simulators will be used, and ten participants will be using each of the three prototypes. In the first experiment, only the palpation force applied by the participant is recorded, while in the second and third experiments, the palpation force as well as exploration time will be recorded for each participant. The fourth experiment will use prototype-C with two different resolutions of the model, in which one prototype has a model with twice the resolution of the other. Ten participants will be used for each of the two prototypes in this experiment. This experiment evaluates the effect of the extent of the visual cue.

3.3.2 Experimental Design

An experimental design is a plan for assigning experimental units to treatment levels and the statistical analysis associated with the plan [142].

In order to achieve the last two objectives, it is necessary to select an appropriate experimental design, and appropriate method of statistical analysis. According to Miller (1974), in the experimental control, there are three possible methods: (i) the repeated measures design, (ii) the matched participants design, and (iii) the independent group design. The experimental control selected for the evaluation of multimodal cues is the independent group design where the participants in the groups for using the three prototypes are entirely different, thereby eliminating the biases in the results.

From the statistical analysis techniques, two- sample independent tests will be used. Data collected in the experiments will be statistically analysed and the main study will be descriptive-statistics such as mean, standard deviation etc., and inferential-statistics such as two sample independent tests. The significance among data will be evaluated using an ANOVA (Analysis of Variance). An independent sample t-test will be used for the evaluation of the data of the two groups.

3.4 Summary

In this chapter, the research methods used to conduct the work in this thesis are discussed. The research methods are model development, simulation based evaluation, and experiments. Data collected will be analysed using statistical analysis techniques. In a statistical analysis, the techniques used will be a descriptive statistical analysis for measuring the performance significance and an inferential statistical analysis for measuring the significant differences between each of the two groups.

The next chapter presents the first proposed visio-haptic deformable model based on the slope intercept form of a line equation. The proposed method uses a surface based triangular mesh model. In this model, the haptic sensation remains the same available in the currently available bovine rectal palpation simulators, while the interactive visual deformation is achieved using the proposed algorithm.

CHAPTER 4

VERTEX BASED VISIO-HAPTIC DEFORMABLE MODEL

In this chapter, the first proposed model that involves visio-haptic deformable model is presented, namely, a vertex based visio-haptic deformable model using a slope intercept form of a line equation for deformation in the region of interest. In Section 4.1, the model representation is discussed. In Section 4.2, a visio-haptic deformation based on the slope intercept form of a line equation is presented in detail. As the deformation in the proposed work is local, therefore two-neighbourhoods of the touched node are used for deformation along the touched node itself using different slopes. The ring of neighbour searching and storing algorithm is presented in Subsection 4.2.1. In the next Subsection 4.2.2, the proposed algorithms are presented for visual and haptic deformation. When the external force is relaxed in the region of interest then the deformable model comes to the original position. This is achieved using shape preservation method and is presented in Subsection 4.2.3. Collision detection (CD) plays an important role in the real-time interactive simulation and this is discussed in Section 4.3. As the haptic loop runs approximately 20 times faster than the visual loop, therefore rendering few triangles in the haptic loop improves performance and such technique proposed in this thesis are discussed in Section 4.4. Section 4.5, presents the results and Section 4.6, summarizes the chapter.

4.1 Model Representation

The model is made up of a surface triangular mesh containing nodes and edges. Separate lists are used for nodes and triangles in the model.

Information stored in the node class as.

```
class Node
{
Vertex3D position;
int numNodes;
std::vector<int>firstRingNeigh;
std::vecor <int> secondRingNeigh;
int numFirstRingNeigh;
int numSecondRingNeigh;
};
```

In addition, the class declaration for a triangle is

```
class Triangle
{
int vertIndex [3];
double area;
...
};
```

In this representation, all of the information is stored in the node as we are mainly dealing with the nodes in this vertex based visio-haptic deformation technique. Each node of the model has 3D position in the model. It has the first ring of neighbours which are directly connected with this node (one - neighbourhood) and second ring of neighbours (two - neighbourhood) that are directly connected to its first ring of neighbours. The Triangle class is used to collect information for the node in the preprocessing step. It is also used to render the model in the graphics as well as in haptic loop.

Two types of nodes are used, one is fixed node that contains only position information. When the model is loaded then fixed node are filled with the original position of each node. This node is used for maintaining the shape of the model and is discussed in the next section. Another type of node is a deformable node that contains position information as well as information about its first and second rings of neighbours. This node is free to move when the force is applied to it. Therefore, two lists are maintained in the simulation, one contains *fixed node* and the other list contain *deformable nodes*.

The triangle list uses nodes from the list of *deformable nodes* for graphic and haptic simulation. *Fixed nodes* are only used for maintaining the shape or bringing the deformable nodes back to its original position once the external force is removed.

4.2 Vertex Based Visio – Haptic Deformable Model

In a vertex based visio-haptic deformation, the vertices, or nodes of the model are manipulated directly. The advantage of vertex based deformation is its efficiency and ease of implementation. In this method, the nodes of the model are manipulated directly upon the interaction of the user with the force feedback device. For this purpose, a slope intercept form of a line equation is used as a constraint that restricts the movement of nodes along the direction of external force applied through a haptic force feedback device.

The slope intercept form of a line equation is given as

$$y = mx + b \quad (\text{Eq. 4.1})$$

An Equation 4.2 is modified form of an Equation 4.1, and is given as

$$P^{i+1}(x, y, z) = P^i(x, y, z) + m_1 * F(x, y, z) + b_1 \quad (\text{Eq. 4.2})$$

An Equation 4.2 calculates the new position of the node as a function of external force applied to it. Here $P^i_{(x,y,z)}$ is the i^{th} node current position and $P^{i+1}_{(x,y,z)}$ is its next position. Here m_1 acts as step size for the new position of the contacted node and b_1 is the intercept and it is set to zero. $F_{(x,y,z)}$ is the force applied to the node $P^i_{(x,y,z)}$ through a haptic force feedback device, PHANToM Desktop [143]. Equation 4.2 calculates the new position for the node contacted through the haptic force feedback device only.

When an external force is applied to the one of the nodes in the model, then it changes its position and this position is determined using the slope intercept form of a line equation. However, for a realistic deformation, the surrounding nodes of the touched node needs also be deformed in such a way that it looks realistic. For this purpose, the first ring of neighbours and the second ring of neighbours of the touched node is also deformed using the slope intercept form of a line equation. For each node of the model, these first and second rings of neighbours are calculated and stored in the preprocessing step and is performed before the simulation starts. These two rings of neighbours are deformed with different slopes. In the proposed algorithm, these two rings of neighbour are sufficient for visually appealing deformation of the soft tissue. Equation 4.3 is used for first ring of neighbour and Equation 4.4 is used for the second ring of neighbour and is given as

$$PN^{i+1}(x, y, z) = PN^i(x, y, z) + m_2 * F(x, y, z) + b_2 \quad (\text{Eq. 4.3})$$

$$PNN^{i+1}(x, y, z) = PNN^i(x, y, z) + m_3 * F(x, y, z) + b_3 \quad (\text{Eq. 4.4})$$

In Equation 4.2, Equation 4.3, and Equation 4.4, $b_1 = b_2 = b_3 = 0$, and m_1 , m_2 , and m_3 are negative, and in the range of $[0, -1]$, exclusive of 0 and -1, and must satisfy the condition: $m_1 > m_2 > m_3$.

To find the next node position on the first and second ring of neighbours of the contacted node, the same modified slope intercept form is used but with different slopes.

4.2.1 Ring of Neighbour Algorithm

When a user touches the model with the force feedback device then it affects one of the nodes of the model on which the user force is concentrated. In order to deform the model in a realistic way, all other surrounding nodes need to be deformed. To achieve a realistic deformation in this region of interest, the neighbours of the touched node

need to be deformed as well. For this purpose, there is a need for calculating the neighbours of this touched node. Such neighbours are calculated for each node of the model in the preprocessing step before the interactive simulation starts.

The neighbours are divided into two rings, first ring, and second ring. The first ring also known as one-neighbourhood contains those nodes that are directly connected with the touched node and the second ring also known as two-neighbourhood contains those nodes that are one node distance apart from the touched node. These two rings are sufficient for the local deformation during simulation.

The touched node along with its first ring (nodes $A1$ to $A6$) and second ring (nodes $B1$ to $B12$) of neighbours is shown in Figure 4.1.

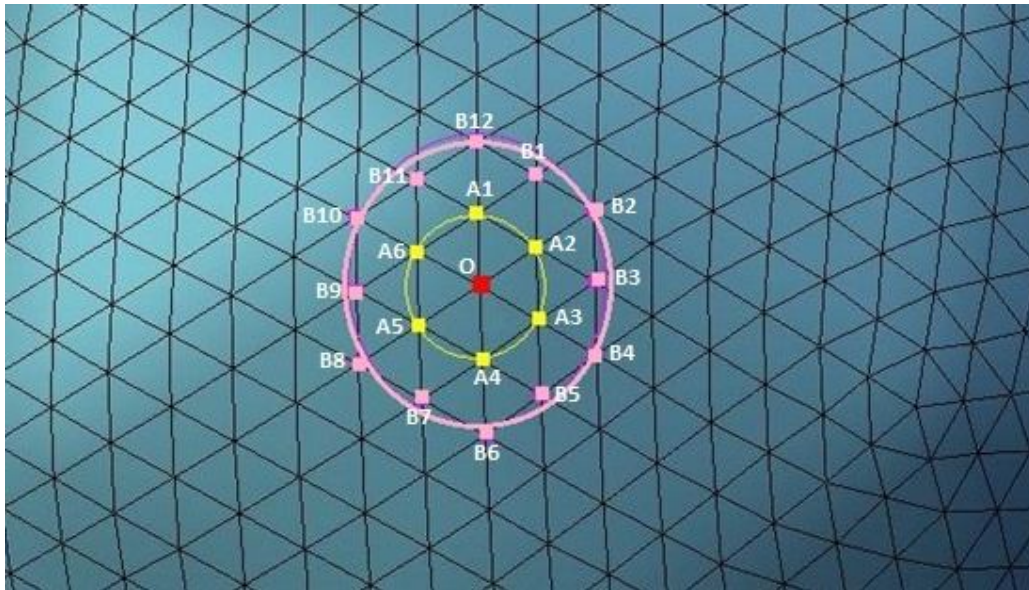


Figure 4.1: A touched node with its first and second ring neighbours

In Figure 4.1, node O is the touched node, nodes $A1$ to $A6$ are the nodes in the first ring of neighbours, and $B1$ to $B12$ represent the nodes in the second ring of neighbours.

The algorithm to find and store the first ring of neighbours and the second ring of neighbour is given as follows in a pseudo-code form. These two rings of neighbour are calculated for every node of the model.

Vertex3D position;

FINDFIRSTRINGNEIGHBOUR (maxNodes)

INPUT: An array of maxNodes

OUTPUT: An array of firstringneighbour nodes

for $i \leftarrow 1$ to maxNodes-1 **do**

for $j \leftarrow 1$ to maxNodes-1 **do**

 search(i, j)

 node[i].neighbour $\leftarrow j$

CALL FINDSECONDRINGNEIGHBOUR ()

FINDSECONDRINGNEIGHBOUR (maxNodes)

INPUT: An array of maxNodes

OUTPUT: An array of secondringneighbour nodes

for $i \leftarrow 1$ to maxNodes-1 **do**

for $j \leftarrow 1$ to nodes[j].maxNeighbour **do** // first ring of neighbours for
 // each node

 index=0 //Index for
 //secondringneighbour array

for $k \leftarrow 1$ to maxNodes-1 **do**

 search (node[i].Neighbour[j], k) // search k in the
 // firstringneighbour

 node[i].SecondNeighbor[index++] $\leftarrow k$ // store as a
 //secondringneighbour

This pseudo-code finds the two rings of circular neighbours for every node in the model. As mentioned above, this algorithm runs and gathers the two rings of neighbour (two-neighbourhood) for each node of the model, before the actual simulation starts. Therefore, the performance of this algorithm is not an issue.

When the node of the model is touched with the force feedback device and palpation force is applied then the touched node deforms along the force vector. In a similar way, its first and second ring of neighbours also deforms. Deformation profile due to the different step size of touching node and its first and second rings of neighbour is shown in Figure 4.2.

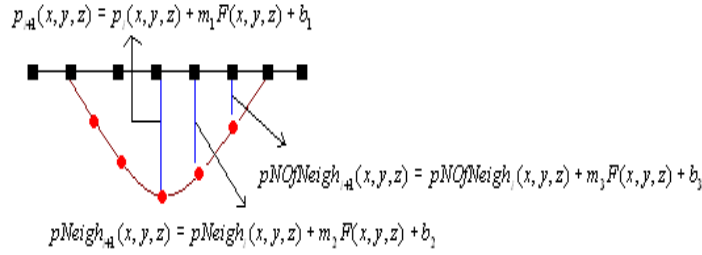


Figure 4.2: A deformation profile due to variable step size for touching node and its neighbours

In Figure 4.2, the black squares represent the original node positions and the red circles are its deformed positions. The depth of deformation for each node is controlled by the slope value m and is known as the step size.

4.2.2 Algorithm for Visual and Haptic Deformation

As mentioned there are two separate loops, one is for visual channel and the other is for haptic channel. Therefore, the code for both the loops is different, and is given as:

Preprocessing Step: Find out all neighbours in the first and second ring of every node using the algorithm discussed above and store to get a constant searching time both for haptic and visual channel.

4.2.2.1 Visual Deformation

Detect collision between a Haptic Contact Point (HCP) and node in the model

if (Collision between HCP and Model)

{

Get the force applied $F(x, y, z)$ through the haptic probe.

If collision between HCP & Model

node[i] $\leftarrow F(x, y, z)$ // Apply force to the ith node

$p_{i+1}(x, y, z) \leftarrow p_i(x, y, z) + m_1 F(x, y, z) + b_1$ // Get the new
//position for the
//contacted node

for $i \leftarrow 1$ to nodes[p].numFirstRingNeigh-1 do

$pN_{i+1}(x, y, z) \leftarrow pN_i(x, y, z) + m_2 F(x, y, z) + b_2$ //Loop

//through all first ring of
//nodes of p and update
//position

for $i \leftarrow 1$ to nodes[p].numSecondRingNeigh-1 do

$pNN_{i+1}(x, y, z) = pNN_i(x, y, z) + m_3 F(x, y, z) + b_3$

//through all first ring of
//nodes of p and update
//position

else if Touched && !Collision

$f_s^i \leftarrow k_s(x_1 - x_2) + k_d(v_1 - v_2)$ // Bring the touched node to rest position

for $i \leftarrow 1$ to nodes[p].numFirstRingNeigh-1 do

$f_s^{N_i} \leftarrow k_s(x_1 - x_2) + k_d(v_1 - v_2)$ //Bring the first ring nodes to
//rest position

for $i \leftarrow 1$ to nodes[p].numSecondRingNeigh-1 do

$f_s^{NN_i} \leftarrow k_s(x_1 - x_2) + k_d(v_1 - v_2)$ //Bring the second ring nodes to
//rest position

4.2.2.2 Haptic Deformation

In haptic deformation, the forces generated are rendered to the user so that he/she can feel the object being palpated. When the user touches, the model then fixed node acts as an initial node and the deformable node act as a final node. The force is calculated using Hooke's law and rendered to the user as shown in Figure 4.3.

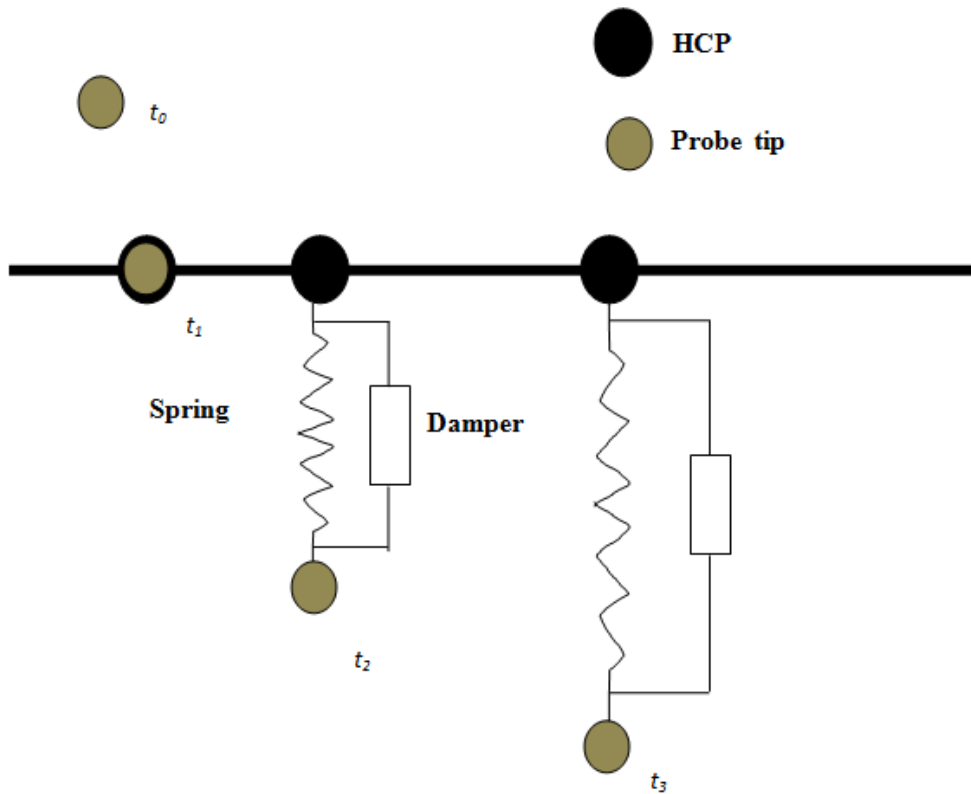


Figure 4.3: Force computation and rendering mechanism during interaction with the model

Haptic deformation pseudo-code for haptic loop is given as.

if touchedModel

$initPos \leftarrow node[Touched]$ // Get the position of fixed node touched

$finalPos \leftarrow p_i(x, y, z) + m_1 F(x, y, z) + b_1$ // Get the final position using
// Equation 4.2

```

renderForce(x,y,z) ← -k (finalPos – initPos) // Render force to the
//user calculated using
//Hooke's Law

user ← renderForce(x,y,z) // Apply render force to the user

```

In this visio-haptic deformable model, the haptic sensation is the same as in the currently available bovine rectal palpation simulator.

4.2.3 Shape Preservation

The description mentioned above achieves a plastic deformation in which the nodes once deformed does not come back to its original position. As the soft tissue is elastic meaning when the external force is removed then the nodes need to come to its resting position in order to maintain the shape of the model. Therefore, to maintain the shape while external force is removed, shape preservation springs are used. This spring known as *home spring* brings the deformable node to fixed node, which is its resting position once the external force is relaxed. A *home spring* ensures that the distance between a *fixed node* and *deformable node* is zero. In an initial setup, both fixed node and deformable node are at the same position in space and its initial spring length is zero. When the external force is applied to the node then the deformable node is free to move while there is no effect on the fixed node. Once the external force is relaxed then home spring comes into play and brings the *deformable node* to its resting position. Home spring connecting deformable node with the fixed node is shown in Figure 4.4 for the surface based triangular mesh model of the reproductive tract.

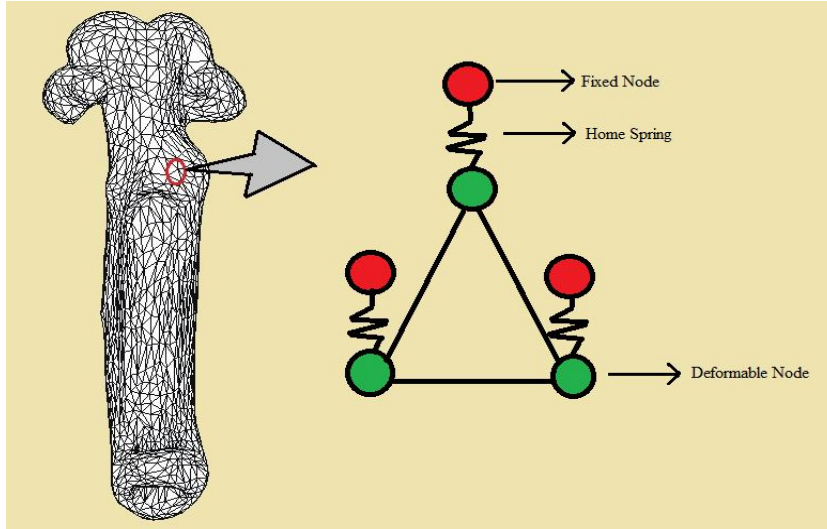


Figure 4.4: Home springs between a fixed and deformable node for shape preservation

As these two nodes are connected through a spring, the spring applies force only on the *deformable node* to bring it to the initial configuration. Figure 4.5 (a), shows the initial configuration of *fixed* and *deformable nodes* which shares the same space. Figure 4.5 (b), shows the configuration after an external force is applied. The springs are connecting these two nodes.

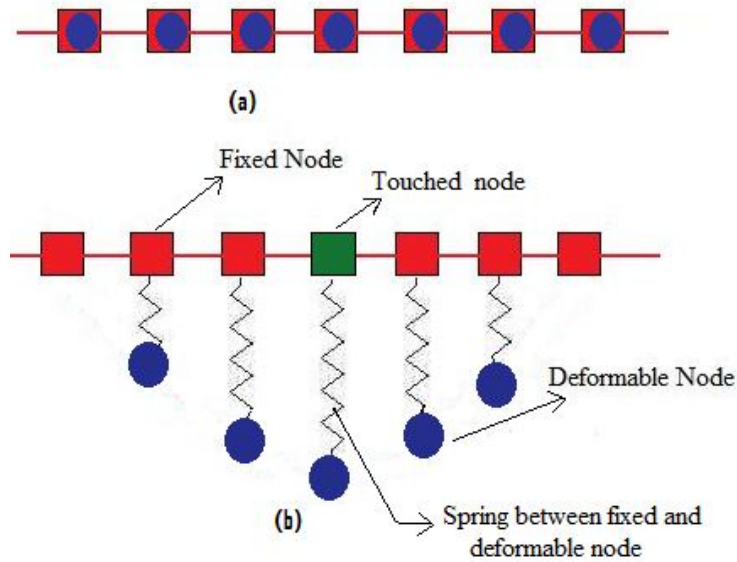


Figure 4.5: (a) Initial Configuration, (b) Configuration after force applied

The spring force between a *fixed node* and a *deformable node* is unidirectional and from a *fixed node* towards a *deformable node*. This spring force will cease to exist when the distance between fixed and deformable node is zero and comes into play otherwise.

As the spring connecting two nodes oscillates, therefore mathematical formula needs to be derived for a *home spring*, which bring a *deformable node* to its original position in time and avoid oscillating it. For this purpose, there is a need of a function C such that [81]

$$C(x_1, x_2) = x_1 - x_2 \quad (\text{Eq. 4.5})$$

Which says two points x_1 and x_2 needs to be in the same position in space. In an Equation 4.5, x_1 represents a fixed node and x_2 represents a deformable node. We need to convert this equation into force law. However, when two points are stretched then kinetic energy is converted into potential energy. This is given as

$$E = \frac{1}{2} k_s C.C \quad (\text{Eq. 4.6})$$

Where k_s is the spring stiffness coefficient. Since force due to scalar potential is minus the energy gradient, the force on node x_i due to C is

$$f_i = \frac{-\partial E}{\partial x_i} = -k_s C \frac{\partial C}{\partial x_i} \quad (\text{Eq. 4.7})$$

Here $\frac{\partial C}{\partial x_i}$ is the transpose of the *Jacobian Matrix*. This force will oscillate the system about $C=0$, therefore we need damping force. To add damping force, then Equation 4.7 becomes as:

$$f_i = (-k_s C - k_d \dot{C}) \frac{\partial C}{\partial x_i} \quad (\text{Eq. 4.8})$$

As shown in Equation 4.5 that $C = x_1 - x_2$, so

$$\frac{\partial C}{\partial x_1} = I \text{ and } \frac{\partial C}{\partial x_2} = -I$$

Where I is the identity matrix. Putting these values in Equation 4.8, we get

$$f_1 = -k_s(x_1 - x_2) - k_d(v_1 - v_2) \quad (\text{Eq. 4.9})$$

$$f_2 = k_s(x_1 - x_2) + k_d(v_1 - v_2) \quad (\text{Eq. 4.10})$$

where f_1 and f_2 are the two opposite spring forces that attracts fixed and deformable nodes toward zero-length spring [81]. In this work, the fixed node is not movable so there is no need to apply the spring force on it. The only spring force is on the deformable node that brings it to its rest position.

Therefore the spring force needs to bring a deformable node to its original position is given in Equation 4.11.

$$f_s = k_s * (x_1 - x_2) + k_d * (v_1 - v_2) \quad (\text{Eq. 4.11})$$

This is the force law for the zero-rest-length damped spring. This spring force is used to bring the deformable node to its original position once the user relaxes the external force.

4.3 Collision Detection (CD)

Collision detection is an important component in the medical simulators. A better collision detection technique greatly improves the performance of the simulator. In this thesis, the model is made up of nodes and triangles. The HCP is also treated as a 3D node. Checking an HCP with every node for collision is a very slow process and furthermore, as the haptic loop runs at 1000Hz, then it becomes impractical to use the linear searching method, available in the OpenHaptics API (Application Programming Interface) [144]. To avoid the slip through of the HCP into the model during interaction, an efficient method is required for searching the touched node. Therefore, there is always a need for fast and efficient collision in order to make it faster. In this thesis, a space partitioning technique is used in the haptic channel, and is discussed in the next section.

4.3.1 Space Partitioning

In this technique, the space occupied by the model is divided. During collision most of the unwanted region is quickly removed to narrow down the region of interest. There is a different space partitioning techniques such Binary Space Partitioning (BSP) and octree space partitioning (OSP). In this thesis, an OSP is used and its brief overview is given in the next section.

4.3.1.1 Octree Space Partitioning (OSP)

In this method, the whole space is divided into sub-spaces. First, the whole model is considered as a bounding box and then it is divided into eight sub-spaces and each is called an octant. The process repeats until a certain threshold for stopping the recursion is reached. In the proposed work, stopping condition is the number of nodes per octant and level of subdivisions. The nodes are stored in the leaves only. This technique is implemented only in the haptic rendering loop for quick access of a particular node of the model. The octree subdivision mechanism is shown Figure 4.6.

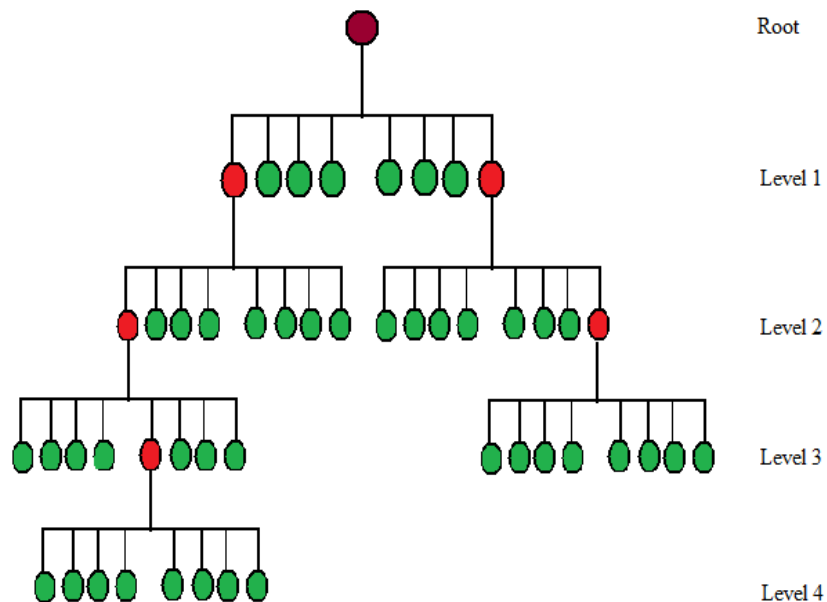


Figure 4.6: Example of an octree space partitioning method

As a starting point, the whole model is bounded by one box and is stored in the root node of the octree. Then the process of subdivision starts which divide the space into eight sub-spaces in its first level and then recursively continued. The model shown in Figure 4.7 contains 1500 nodes and 3000 triangles, and these nodes are stored in the leaves of an octree. The threshold value set for stopping this recursive subdivision is 100 nodes per leaf. This is shown in Figure 4.7.

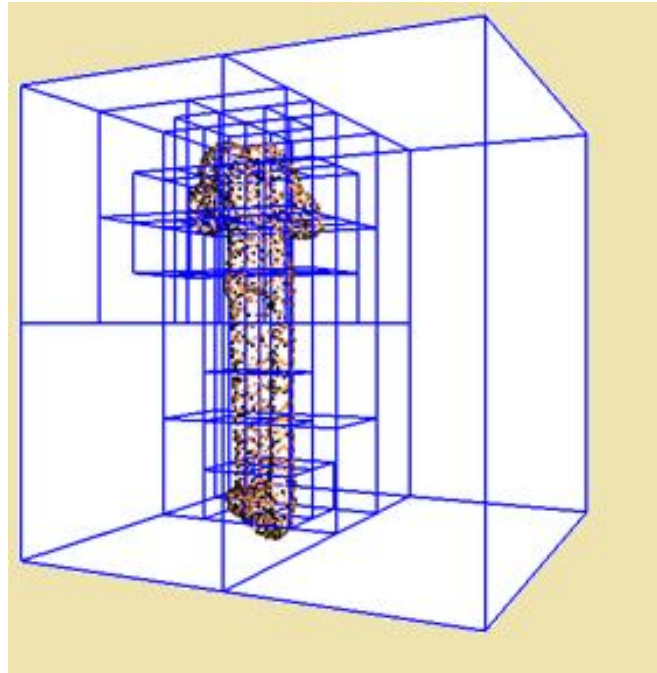


Figure 4.7: Model partitioning using an octree method

The HCP is constantly checked with an octree for collision in the collision thread. In this collision, the main task is to find the node of the model with which HCP collides. Once found the desired node, then information about its first and second ring of neighbours is already stored in it. This node position is then changed using the slope intercept form as well as its first and second ring of neighbours each with different slope.

4.4 Haptic Rendering of an Octant and Two Rings of Neighbours

In a haptic channel, only the octant which is touched by the haptic force feedback is rendered for fast rendering. All the triangles belonging to this octant are rendered in

the haptic loop for the haptic sensation of the model. Another method is to render the two rings of neighbours in order to further reduce the number of triangles rendered in the haptic loop. This further reduces the rendering time as only the neighbours are haptically rendered and not the whole octant. Figure 4.8 shows the different snapshots taken during an interactive real-time simulation for rendering the two rings of neighbours of the touched node.

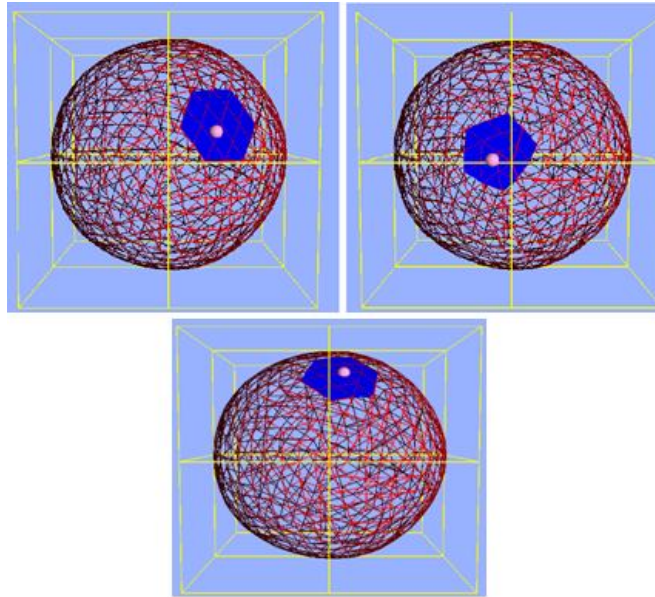


Figure 4.8: Snapshots of an interactive simulation for rendering the two rings of neighbour in haptic loop

Similarly, Figure 4.9 shows snapshots taken during an interactive real-time simulation for rendering an octant of the touched node.

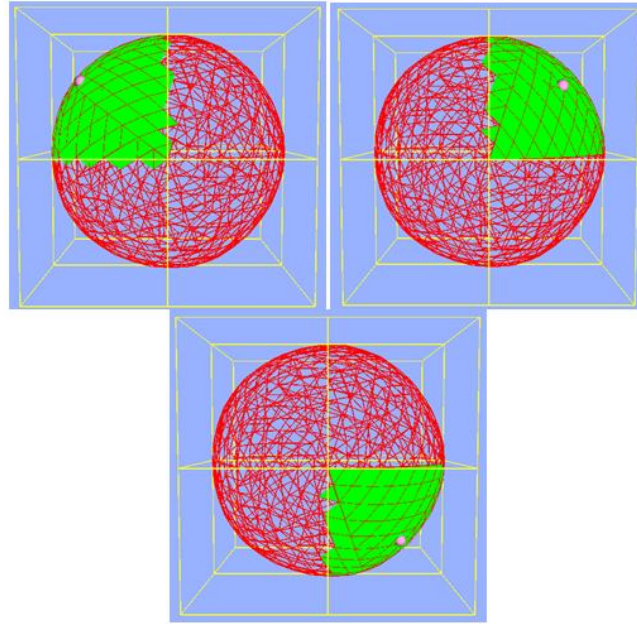


Figure 4.9: Snapshots of an interactive simulation for rendering an octant of the touched node in the haptic loop

4.5 Results on the Performances of the Model

A vertex based visio-haptic deformable model using slope intercept form of a line equation is first implemented for visual and haptic deformation. The deformation of the soft model is local meaning that only the region near the haptic force feedback device is deformed. Deformation using proposed algorithm is shown in Figure 4.10 for stomach model and in Figure 4.11 for sphere model. The algorithm is fast in visual deformation without needing efficient data structures for mesh. The proposed vertex based visio-haptic deformable model is best suitable for tissue whose properties are linear which means force versus displacement relationship is linear, isotropic means force versus displacement relationship is the same in all directions, and homogeneous which means the behaviour of soft objects does not change with the volume.

Without an octree space partitioning method, and using linear search for the touched node of the model with the force feedback device, the algorithm works fine on the models resolution of up to 20,000 triangles.

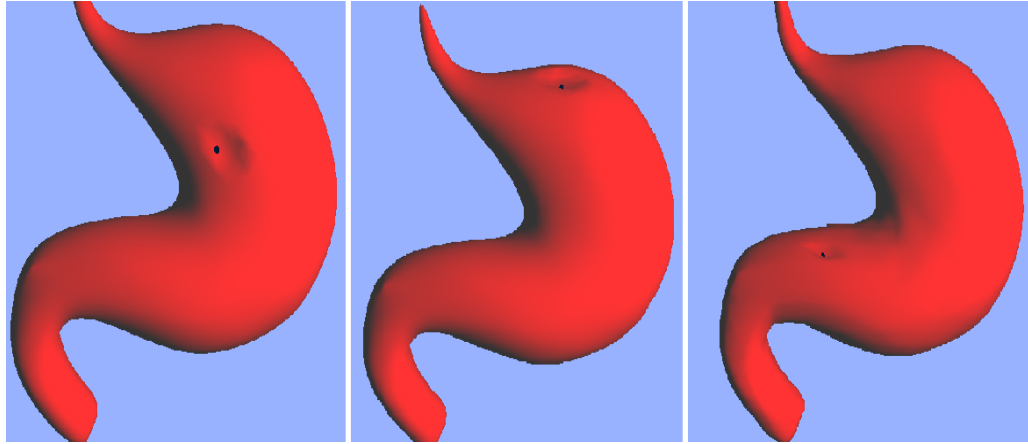


Figure 4.10: An interactive deformation of the stomach model

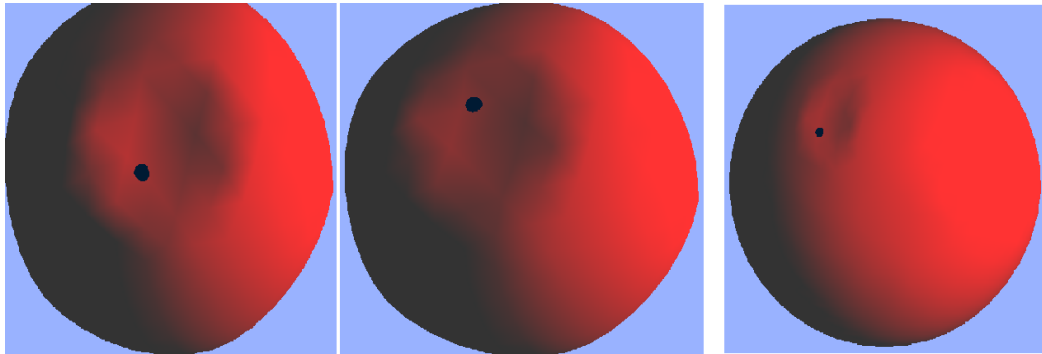


Figure 4.11: An interactive deformation of the sphere model

4.5.1 Octree Implementation for Fast Collision Detection

As mentioned above, the algorithm works realistic for model resolution of up to 20,000 triangles in the model. Beyond this limit, rendering becomes slower for haptic sensations and lapses occur during the interactive simulation. The reason for these lapses is not the computation complexity of the proposed algorithm, but the time required for linearly searching for the touched node during an interactive simulation. Further improvement is required on the collision detection part of the simulator, and hence, an efficient technique is required to access quickly the touched node of the model.

In order to get to the node of the model touched with the force feedback device, an OSP technique is implemented. For the haptic sensation in this thesis, an OpenHaptics API (Application Programming Interface) [144] is used and it does not provide octree implementation therefore in this thesis an OSP is implemented for accessing the node touched with the force feedback device within the haptic loop. In this way, the touched node is accessed quickly. Once reached to the contacted node, the slope intercept form of the line equation is used to deform the touched node and its two-neighbourhood based on the force applied.

In Table 4.1, accessing time for the touched node is presented using linear search and octree search method. The linear search uses nearest neighbour algorithm to find the node from the list of model close to HCP, which is available in the OpenHaptics APIs. In an octree, the same algorithm is applied once the correct octant with which the HCP collided is found.

Searching in an octant instead of searching the whole list is much faster. In this thesis, during the implementation of an octree method, fifty nodes of the model are stored in each leaf node, and the subdivision level is three. The condition set is whichever comes first.

Table 4.1: Time taken in (ms) for searching touched node using a linear and an octree method

| Model Resolution | Time Taken in (ms) | |
|-----------------------------------|------------------------------------|----------------------|
| | (Only Finding Touched Node) | |
| | Linear Method | Octree Method |
| Vertices: 994 Triangles:1985 | 1.05 | 0.17 |
| Vertices: 3970 Triangles:7937 | 4.08 | 0.24 |
| Vertices: 5762 Triangles:11521 | 5.94 | 0.30 |
| Vertices: 8664 Triangles:17325 | 8.94 | 0.92 |
| Vertices:14392 Triangles:28781 | 14.84 | 1.10 |

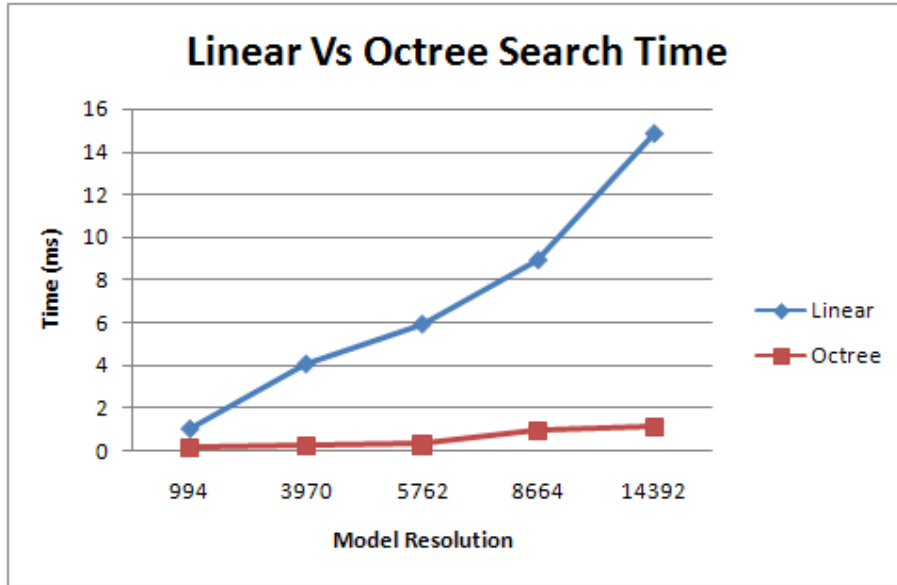


Figure 4.12: Comparison of a linear and an octree search method for different model resolution

Figure 4.12 compares the search time using a linear search method and an octree search method for accessing the touched node. The access time using an OSP is much smaller than linear search method, which uses nearest neighbour algorithm for searching the touched node. Although the search time is much smaller, still increasing the number of nodes increases the time required for searching a touched node. This could be improved further by increasing subdivision levels. In Figure 4.11, results were collected for three levels of subdivisions. If the subdivision levels are increased further, then the time required will be further reduced.

Access time is directly related to the number of nodes stored in the leaf and indirectly related to the number of subdivision levels. This relationship is shown in the Equation 4.12 as.

$$t_{access} = \frac{NoofNodesperleaf}{Noofsubdivisionlevels} \quad (\text{Eq. 4.12})$$

Increasing the subdivision levels greatly improves the performance in terms of quickly reaching the collided node. Although increased levels of subdivision are very efficient, care must be taken while combining this method with haptically rendering the model.

An octree method is implemented for searching the touched node. Once located the touched node in the model, the proposed algorithm is applied to deform the model. The whole model is rendered in haptic channel for the haptic sensation, while manipulating the object in order to avoid slipping inside the object. When the model is complex, then rendering it in the haptic loop consumes many haptic cycles. An octree method for searching improves the overall performance but if the whole model is rendered in the haptic loop, then still it degrades the performance. Therefore, methods need for efficient rendering as well. If only the touched octant is used then rendering is greatly improved. As this is a soft object simulation, therefore the node deforms during simulation and changes its position. The octree needs to be dynamic in order to accommodate this change of position of the nodes. Nevertheless, recalculating the octree in real-time simulation is itself computationally expensive.

4.5.2 Efficient Collision Detection and Rendering

An octree method for accessing the touched node is advantageous in terms of speed but if the whole model is rendered in the haptic loop for haptic sensation, then this rendering still becomes computationally expensive. Therefore, we need to render as few triangles as possible in the haptic rendering pipeline while maintaining the realism. As the user moves over the object then it is sufficient to render only the triangles in the close vicinity of the HCP.

A new method is developed for searching and rendering in the haptic loop to improve efficiency. This method uses an octree method for searching the touched node during an interactive simulation. Once found the touched node, only two circular rings of this touched node are rendered in the haptic loop for improving efficiency while maintaining the same haptic sensation of the object. The search method is fast enough to locate quickly the touched node and as well the rendering method which runs quickly to feel the object at that point.

Table 4.2 compares the results of a combination of various searching and rendering techniques for improving the performance. In the first type of simulation, a linear search using nearest neighbour node is used for searching the touched node while whole model is rendered in the haptic loop for the haptic sensation. In the second type of simulation, an octree method is used for searching the touched node and the whole model is rendered in the haptic loop [50; 69]. While in the third type of simulation, a proposed method, which uses an octree method for searching and two rings of the neighbour of the touched node for rendering in the haptic loop, is presented. One level of subdivision using an octree is used for this simulation and the two rings of neighbour are calculated and stored in each node in a preprocessing step. All these simulations use the proposed algorithm for deformation and force generation.

Table 4.2: Time required for different search and rendering techniques

| Model Resolution | Time Taken In Microseconds (Haptic Rendering Loop) | | |
|-----------------------------------|---|---|--|
| | Linear Search, Whole Model In Haptic Rendering, Force Generation | Octree Method, Whole Model In Haptic Rendering, Force Generation | Octree Method, Rings Of Neighbour In Haptic Rendering, Force Generation |
| Vertices: 1442 Triangles:2880 | 4.8 | 3.8 | 0.90 |
| Vertices: 3970 Triangles:7937 | 13.50 | 10.10 | 1.40 |
| Vertices: 5762 Triangles:11521 | 19.50 | 12.50 | 1.85 |
| Vertices: 8664 Triangles:17325 | 30.10 | 21.50 | 2.70 |

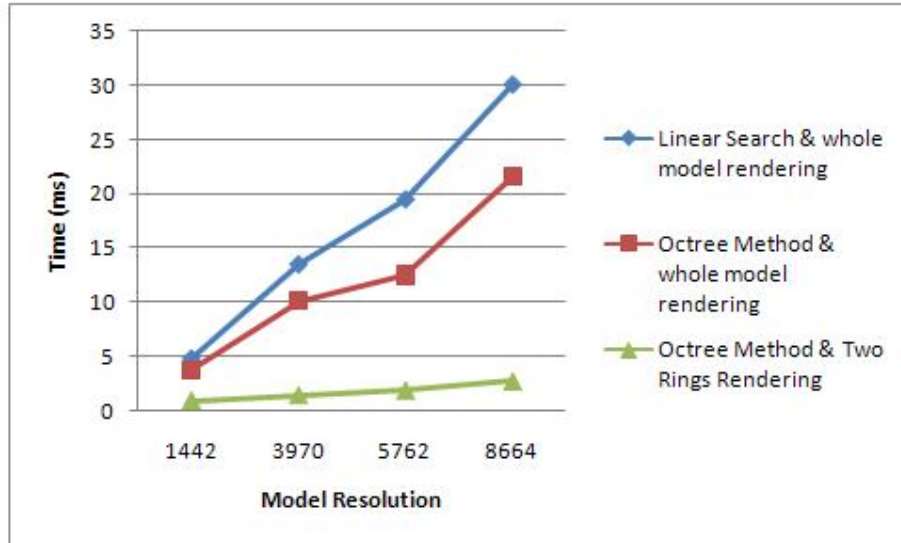


Figure 4.13: A plot for Table 4.2

It is clear from Figure 4.13 that time required for rendering increases with the increase of model resolution while rendering time is greatly improved when rendering only the triangles that are in close vicinity of the touched node in the haptic loop. The running time complexity of the proposed model is $O(\log^2 N)$, running time complexity of Octree method, and whole model rendering is $O(N \log N)$. Similarly, running time complexity of the linear search, and whole model rendering in the haptic loop is *quadratic* $O(N^2)$. Increasing the levels of space partitioning using an octree method, further reduces the time required for searching the touched node.

4.6 Summary

In this chapter, a vertex based visio-haptic deformable model based on the slope intercept form of a line equation is presented for an interactive deformation of soft tissue using the force feedback device. This slope intercept form of a line equation is used to manipulate directly nodes of the model. An algorithm is also discussed for finding neighbours of every node in the model. For every node, two rings named as first ring and second ring of neighbours are calculated and stored in a preprocessing step. The node is deformed along with its circular first and second rings of neighbours whenever touched by the user through the force feedback device. For a realistic

deformation, the touched node, and its first and second ring of neighbours are deformed using the slope intercept form but with different slopes.

As the soft tissue is viscoelastic, therefore when the external force is removed the nodes come to its original position. To achieve this during visio-haptic deformable model simulation, shape preserving springs are used that connects the deformable nodes with the fixed nodes. When the user manipulates the deformable nodes then these springs brought it back to the original position once the external force is relaxed.

As the model is getting complex, searching for the touched node is also becoming time consuming. Therefore, to improve the performance an octree method is implemented to get to the touched node quickly. An octree method divides the space occupied by the model into octants and stores the nodes in the leaves. This helps quickly locating the touched node when searching.

If the whole model is rendered in the haptic rendering then even octree implementation for searching is less pronounced. Therefore, to make rendering faster in the haptic loop, two rings of neighbours are used to render in the haptic loop. This method along with the octree method for searching is implemented and the results shown here provide improved performance.

In the next chapter, a second visio-haptic deformable model proposed is discussed which is based on volumetric mesh model for incorporating the physical properties of the soft tissue and is more realistic in terms of visual and haptic sensations. The proposed also works on surface mesh models, and presented in the next chapter, but in this thesis, volumetric model is used for the evaluation study.

CHAPTER 5

ADAPTIVE AREA BASED VISIO-HAPTIC DEFORMABLE MODEL

In this chapter, the second proposed visio-haptic deformable model using levels of detail in the region of interest, based on the area of a triangle is presented. The algorithm is applied to both the triangular mesh representing the surface of the model, and tetrahedral mesh for considering the internal volume of the model for realistic visual and haptic sensation. This method is more realistic than the method discussed in chapter 4 for the volumetric models. This method, using the volumetric models, is not only different in realistic visual deformation, but also in realistic haptic sensation calculated based on the physical properties of the soft tissue. These physical properties are implemented using mass spring model (MSM) implementation of a tetrahedral mesh model that considers the volume of the soft tissue. Section 5.1 presents the surface and volumetric model representation. In Section 5.2, an adaptive area based visio-haptic deformable model is presented which is implemented using the surface based mesh model as well as volumetric mesh model. This model includes four main components, which are an MSM, refinement, simplification in the region of interest, and parameter identifications for the different levels of detail, and shape preservation upon interaction, and are presented in the subsections of 5.2 respectively. In a real-time interactive simulation, the area is refined dynamically based on the user interaction with the portion of the model. There is a need for criteria of refinement and this is discussed in Section 5.3. To ensure real-time interactivity with the complex model during simulation, fast access methods are required and for this purpose in the proposed model, an OSP is used, discussed in Section 5.4. Results are discussed in Section 5.5, and Section 5.6, summarizes the chapter.

5.1 Model Representation

Model representation is important in order to capture the physical properties. Two model representations are surface model and volumetric model. A surface model is efficient in rendering but has no internal physical properties, while volumetric model capture the internal physical properties of the model it represent. The two models are discussed in the next section.

5.1.1 Surface Model

The reproductive tract is modelled as a triangular mesh representing the surface of the model. In this way, few triangles are rendered. This model is faster in rendering but did not capture the internal physical properties correctly due to the fact that it is hollowed from inside. Therefore this type of model is best for the soft tissue that is hollowed from inside such as vessels and the gallbladder [145]. Surface model of the reproductive tract of a cow is shown in Figure 5.1.

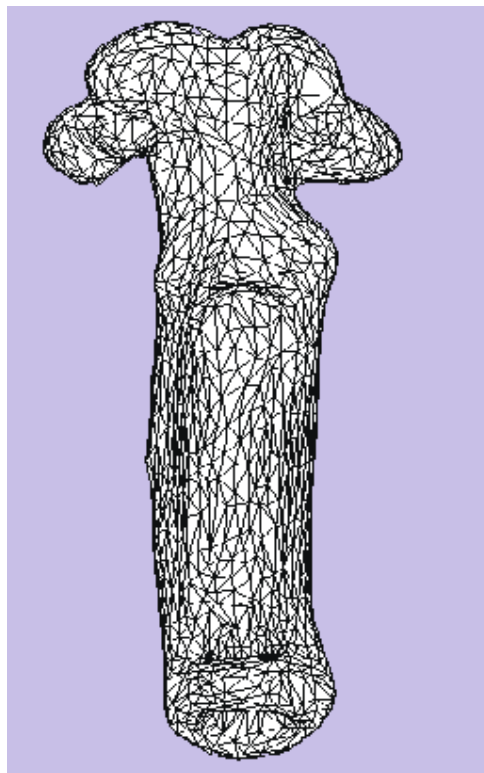


Figure 5.1: A surface mesh representation of the reproductive tract of a cow

5.1.2 Volumetric Model

This model is more accurate and can capture the internal physical properties of the soft tissue than the surface model but computationally expensive due to the complexity of the model. In this thesis, tetrahedral model is used to capture the internal physical properties of the soft tissue. The tetrahedral model of the reproductive tract of the cow in a wire mesh is shown in Figure 5.2, and the cross section in Figure 5.3. This tetrahedral model is created using the tetgen [146] software.

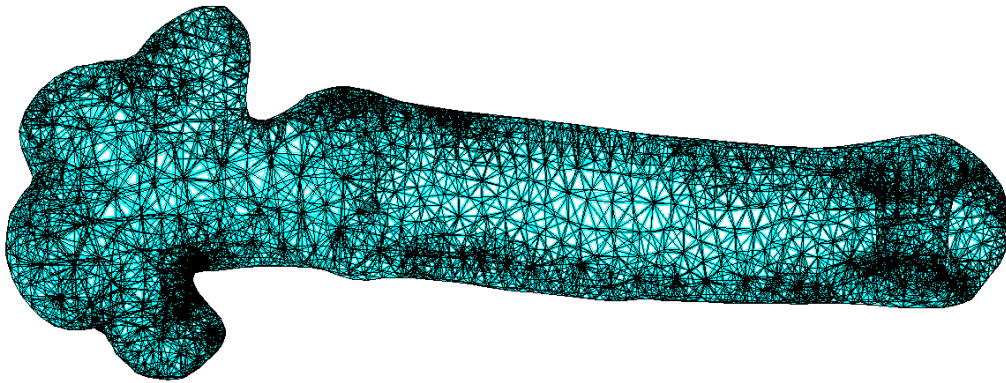


Figure 5.2: Volumetric model made of tetrahedra of the reproductive tract of a cow

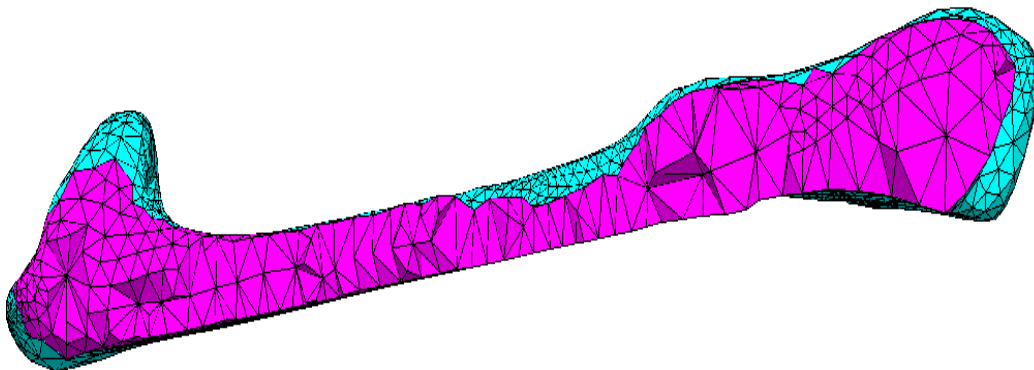


Figure 5.3: Cross section showing internal tetrahedra of the reproductive tract of a
cow

5.2 Adaptive Area Based Visio-Haptic Deformable Model

An adaptive area based visio-haptic deformable model is proposed for surface mesh model that comprises of four main components and for a volumetric mesh model that comprises three components. The first component is the MSM, which is used for capturing the physical properties of the soft tissue being modelled. The second component is the proposed method of refinement and simplification of the region of interest based on its triangle's area during real-time interaction with the model. The third component is the proposed method for adjustment of physical parameters among different levels of detail during simulation. The fourth component used only in the surface mesh model is the shape preservation, which maintains the shape of the model during simulation due to the fact that the model is hollow inside. The following subsections discuss these components.

5.2.1 Mass Spring Model

A Mass Spring Model is of one the physics based method and is famous for its ease of implementation and efficiency. These two properties make this technique more pronounced for the soft tissue modelling. A mass spring model is a discrete alternative to the continuous model. An MSM consist of nodes and massless springs that connects these nodes to form a lattice. The mass of the model is concentrated and distributed in these nodes. The spring keeps the two nodes at its ends a fixed distance apart. The deformation occurs when the nodes are displaced by some external forces such as the force applied by the user. The stiffness of the spring determines the hardness of an object.

The proposed visio-haptic deformable model is made up of triangular mesh both for the surface and for volumetric representation of the model. Each triangle has three vertices and three edges. This triangular mesh is converted to MSM in order to incorporate the physical properties of the soft tissue. Each vertex is then converted into a node and each edge is converted into a spring. Each spring has two nodes, one at each end. When an external force is applied, it disturbs the nodes and these springs come into play to bring the nodes back to its rest position. An MSM with nodes and springs forming a lattice is shown in Figure 5.4.

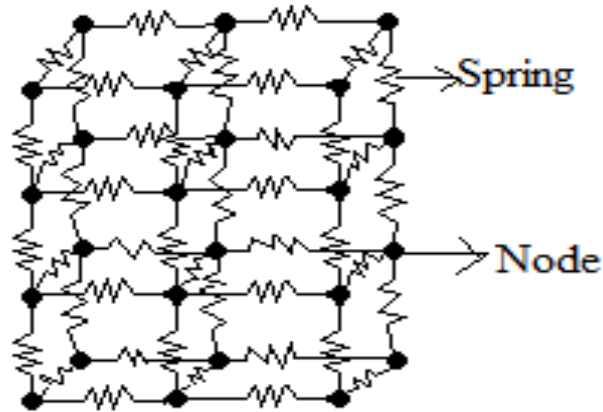


Figure 5.4: A mass spring model with the nodes connected through spring

In the visio-haptic deformable model, for the surface mesh model, and volumetric model, the structure springs are used. Each spring connects two nodes. In the surface mesh model, there are only springs on the surface of the model while in the volumetric mesh model, as it is made up of tetrahedra, so the internal springs are also present. This configuration is shown in Figure 5.5.

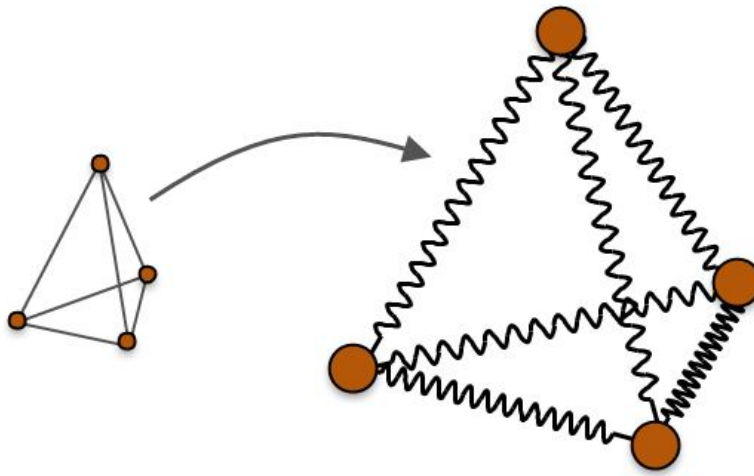


Figure 5.5: A mass spring model for tetrahedral mesh

Each node in the model is exerted with the mutual spring force. In the initial configuration, rest length l is the length of the spring that defines the shape of the body. When a node is allowed to change its position from u_1 to u_2 then the internal

force F_{int} can be calculated according to Hooke's law, given in an Equation 5.1 [84; 81].

$$F_{int}^1 = k_s(|\Delta u| - l) * \frac{(u_2 - u_1)}{|u_2 - u_1|} \quad (\text{Eq. 5.1})$$

$$F_{int}^2 = -F_{int}^1 \quad (\text{Eq. 5.2})$$

where F_{int}^1 and F_{int}^2 are the spring forces on nodes u_1 and u_2 respectively. In the Equation 5.1, k_s is a spring constant, and l is the rest length of the spring. $\Delta u = u_1 - u_2$, is the actual distance between two nodes. Spring force is proportional to the difference in length of the spring.

The spring force alone exhibits an elastic behaviour. Soft tissue exhibits viscoelastic behaviour. Therefore, for a realistic deformation of soft tissue based on the mass spring model, damping component (dashpot) is introduced. Soft tissue is modelled with the springs and the dashpot, in which spring models the elastic behaviour, and dashpot models the viscous behaviour. Three models used for soft tissue modelling are (i) Maxwell, (ii) Kelvin-Voigt or Voigt, and (iii) Kelvin, or standard linear [112]. Each of these models differs in the arrangement of springs and dashpots as shown in Figure 5.6.

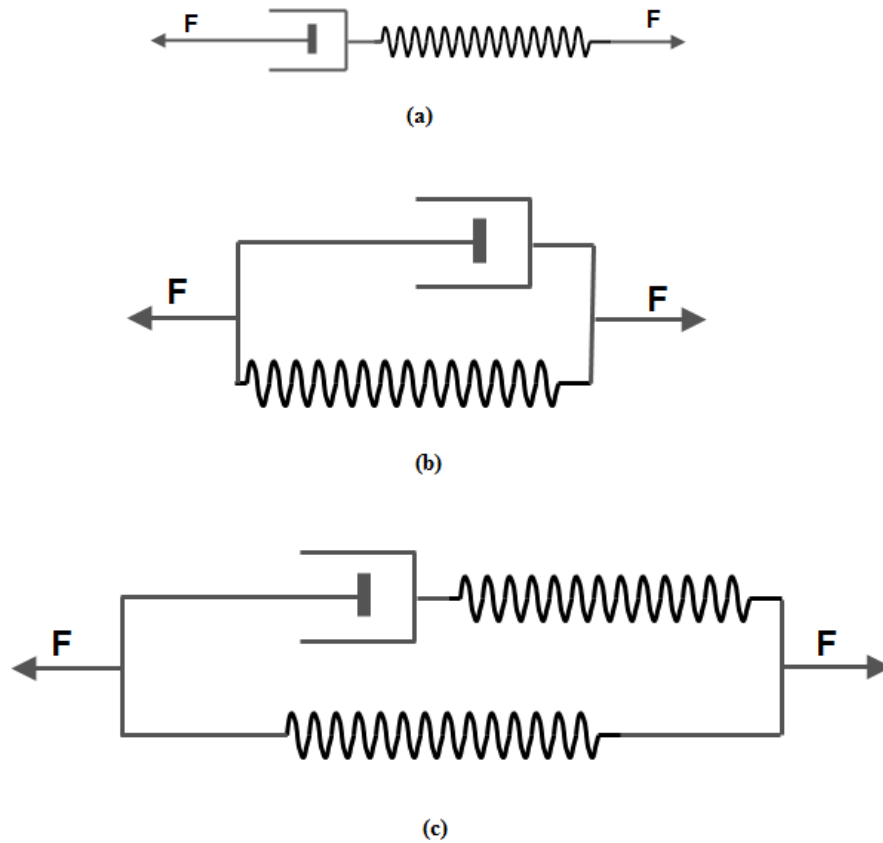


Figure 5.6: Standard viscoelastic models used to represent the soft tissues (a) Maxwell (b) Kelvin-Voigt or Voigt (c) Kelvin, or standard linear [112]

For modelling the biological tissue, the authors in [147; 148] used Maxwell model for *ex-vivo* bovine liver. Kelvin, or standard linear model is used by the authors in [149], for *ex-vivo* kidney, and authors in [150] for *in-vivo* porcine liver. The authors in [151] used Voigt model for bovine liver. The authors in [152] used Voigt model for viscoelastic deformable objects.

In this thesis, the interactive deformation is local. Therefore, the mass spring model based on Voigt model is used for soft tissue modelling. The Voigt model is more appropriate for the viscoelastic material [153]. In addition to the force computed with Equation 5.1, each spring exerts a viscous force, which can be computed as the difference of velocities at its ends weighted by a damping coefficient k_d . Therefore, Equation 5.1, after adding the damping component becomes as given in Equation 5.3 [81].

$$F_{\text{int}}^1 = - \left[k_s (|\Delta u| - l) + k_d \left(\frac{\Delta v \cdot \Delta u}{|\Delta u|} \right) \right] \frac{\Delta u}{|\Delta u|} \quad (\text{Eq. 5.3})$$

Where $\Delta v = v_1 - v_2$ is the difference of velocities of two nodes. Spring force is proportional to the difference in length of the spring and the damping force is proportional to the difference in velocity of the node. The damping coefficient k_d is the resistance against the relative motion between the two nodes attached at the end of the spring.

As each node is connected through springs to many other nodes in the model, therefore, all these neighbours springs exerts force on this node, and this force is the sum of all spring forces acting on it and is calculated as given in Equation 5.4.

$$F_{\text{int}}^i = \sum_{j \in N_i} k_j^i \frac{|u^j - u^i| - |u^j - u^i|^0}{|u^j - u^i|} (u^j - u^i) \quad (\text{Eq. 5.4})$$

Where N_i is the set of nodes connected through springs with the node i , k_j^i is the spring stiffness between node i and node j , while $|u^j - u^i|$ and $|u^j - u^i|^0$ are its current and initial lengths, respectively.

The dynamics of N nodes in a mass spring model under the external force F_{ext} is governed by Newton's law of motion and is given as in Equation 5.5.

$$M\ddot{u} + C\dot{u} + Ku = F_{\text{ext}} \quad (\text{Eq. 5.5})$$

Where M , C , and K are the $3N \times 3N$ mass, damping and stiffness matrices, respectively. The vector F_{ext} is a $3N$ - dimensional vector representing the total external force on the node. The system has evolved forward through time by rearranging Equation 5.5 as a system of first order differential equations:

$$\dot{u} = M^{-1}(-Cv - Ku + F_{\text{ext}}) \quad (\text{Eq. 5.6})$$

$$\dot{u} = v \quad (\text{Eq. 5.7})$$

where \mathbf{v} is the velocity vector of the system of nodes. During simulation, the acceleration, velocity, and position of each node are updated at discrete time points spaced by discrete time steps and therefore known as discrete time system.

5.2.1.1 Stable Numerical Integration

There are many numerical integration techniques available to find \mathbf{u} and \mathbf{v} as a function of time such as Euler's explicit method, Euler's implicit method, and Runge Kutta order 4 methods (RK4).

The explicit Euler integration method is a fast and easy choice for numerical integration. However, the system becomes unstable if the spring stiffness coefficient or the time step of the simulation is very large. If the time step is kept small in the explicit Euler integration, then it takes too much time for producing an animation results of the mass spring model [154]. The solution for this instability problem is to use implicit Euler integration method [90; 155].

To update the velocity and position of each node in the mass spring model, this thesis uses an implicit Euler integration and the equations are given below [81].

$$\mathbf{v}_i^{t+h} = \mathbf{v}_i^t + \mathbf{F}_i^{t+h} \frac{h}{m_i} \quad (\text{Eq. 5.8})$$

$$\mathbf{u}_i^{t+h} = \mathbf{u}_i^t + \mathbf{v}_i^{t+h} * h \quad (\text{Eq. 5.9})$$

Using the implicit Euler's integration method guarantees stability of the system, therefore this allows large time steps to be taken in the animation. In this work, Euler's implicit method has been used due to its stability in during simulation.

5.2.2 Adaptive Area Based Refinement and Simplification Technique

In an adaptive area based refinement technique, the triangles whose area greater than the predefined area are subdivided. The predefined area also known as *thresholdArea* is chosen based on hit and trial method to achieve realism. Figure 5.7 is the simplified

version of the reproductive tract model taken using a 3D digital scanner. It is clear that not all triangle's area is about the same size. In many modelling software the smoothing technique is available which smoothen the triangles and after some iteration all the triangles achieves the same triangular area but this is at the expense of loosing topological details.

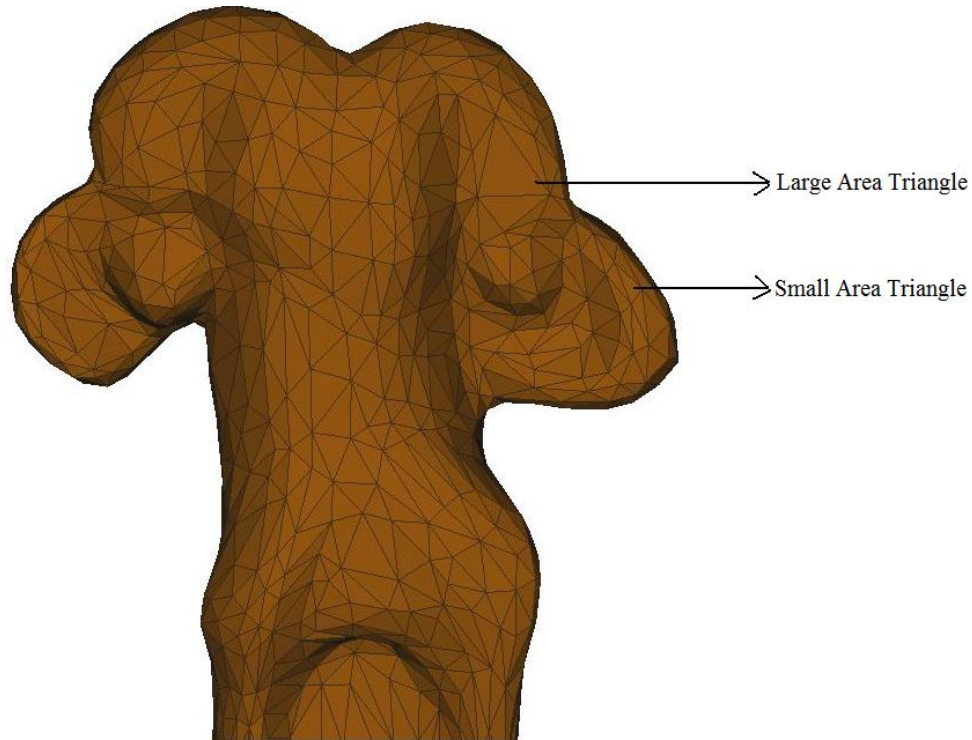


Figure 5.7: A coarse triangular mesh of having varied triangle areas

It is clear in Figure 5.7, that some triangles have a larger area while the other has small area. If such a mesh is divided by any existing subdivision techniques then all triangles will be divided irrespective of its area. Therefore, shortcoming of applying the existing techniques is that extra triangles need to be rendered and new nodes, extra spring force computation, and integration are required, which affects the performance of the simulation.

A simplification technique is the process in which the refined region of interest is returned to the original position, when an external force is relaxed in that region.

5.2.2.1 Data Structure for Surface Mesh Model

A 3D model of reproductive system of a Malaysian cow is scanned through a 3D digital scanner and then simplified and edited at different resolution levels using 3D modelling software such as Blender [156], and MeshLab [157]. The 3D model is shown in Figure 5.8.

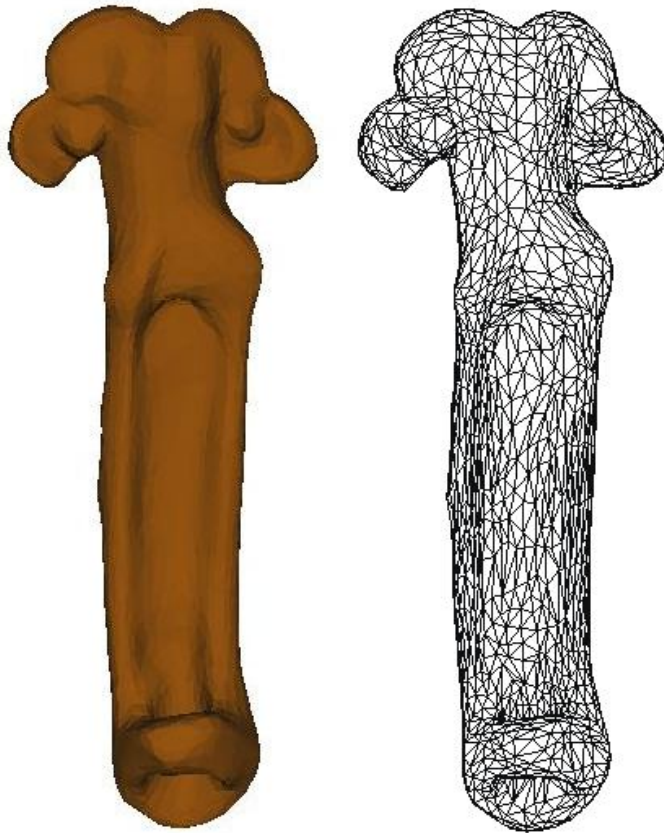


Figure 5.8: Reproductive tract of a cow. Left: A textured model, Right: A wireframe model

The above model comprises of a triangular mesh that contains 3D Points known as nodes and triangles. Following is the description of these for rendering.

```
Class Node
{
hduVector3Dd position;
hduVector3Dd force;
```

```
hudVector3Dd velocity;  
Edge* edge;  
int nodeID;
```

For triangles the class is as

```
Class Triangle  
{  
int nodeIndex[3];  
double triangleArea;  
...  
};
```

Another class called the Spring class is also used which is shown here and will be discussed in the next section.

```
Class Spring  
{  
int nodeA;  
int nodeB;  
double KS;  
double KD;  
double restLength;  
};
```

The Edge class is used to maintain the connectivity information during the subdivision process. Each edge has its starting and ending nodes, next edge in the list, opposite edge in the list and the face to which this edge belongs. This class plays an important role during the subdivision process. Half-edge implementation guarantees constant search time for one-neighbourhood of any node in the model that is touched with the force feedback device at any level of resolution of the model.

```
Class Edge  
{  
unsigned int vert1;  
unsigned int vert2;  
unsigned int next;  
unsigned int opposite;  
unsigned int face;
```

```
};
```

The last class is the Mesh class that contains all of the nodes and triangles in the model along with the springs connecting these nodes. This is a simple mesh representation method in which three lists are maintained for nodes, triangles, and springs in the model.

```
class Mesh
{
std::vector<Triangle> triangleList;
std::vector <Node> nodeList;
std::vector <Spring> springList;
std::vector<Edge> edgeList;
int numTriangles;
int numNodes;
int numSprings;
int numEdges;
};
```

The connectivity information is important during the subdivision process in order to subdivide adaptively the model during simulation.

The following is the snapshot of an octree class that encompasses the model.

```
class Octree
{
Mesh * m_Mesh;
Octree * m_ChildNode[8];
creatNode()
createNewNode()
AssignFacesToNode()
PointBasedCollision()
}
```

The two meshes of the same model are maintained during simulation of the surface based model, while for the volumetric model only one mesh is maintained. For the surface mesh model, one is the original mesh and the other is the deformable mesh. The deformable mesh is allowed to deform the interactive simulation. In the

preprocessing step, the coarse model is refined based on the proposed algorithm and all the information regarding masses, stiffness and damping coefficients are calculated and stored. Only new masses, stiffness, and damping coefficients are calculated during the interactive simulation.

5.2.2.2 Data Structure for Volumetric Mesh Model

The surface mesh model is converted into a volumetric mesh model using tetgen software. The tetgen software creates the volumetric model and stores the data in different text files. These files have the extensions as node, face, and neigh. The files are read in the program. The node file contains the nodes present in the model, the face files contain all the faces in each tetrahedron, and the neigh file contains the neighbour of each tetrahedron in the model. Each tetrahedron contains four triangles (faces). The remaining data structure is almost the same as discussed in the previous section.

5.2.2.3 Description of the Adaptive Area Based Refinement Algorithm

The Haptic Contact Point (HCP) is used as an interaction tip in the simulation. The user applies force with the force feedback device. The phantom cursor is modelled as an HCP. We check constantly the collision between HCP and the nodes present in the model. When it collides with any node in the model and concentrates the force at the collided node, and such concentrated force goes beyond the threshold force set for the refinement algorithm then this node's one-neighbourhood is locally refined. Increasing the model complexity increases searching time for touching the node on the model, and this degrades the performance of the overall system. Therefore, to optimize this collision, an OSP method is implemented. This method is discussed later. In the preprocessing step, the triangle area of all the triangles in the model is calculated and stored.

The description of the steps in this algorithm is as follows.

1. During simulation, the collision between Haptic Contact Point and nodes of a model is detected.
2. Loop through one-neighbourhood faces of this touched node.
 1. Check the area of each face against the *thresholdArea*. If it is greater than the *thresholdArea* then subdivides it into four triangles
 - Increase the subdivision level l by one and SLoD (Sub-levels of Detail) = 4.
 - Now check each of the four newly created triangle's area with the *thresholdArea*.
 - If all four triangles are chosen for subdivision, increase subdivision level l by one and SLoD =4.
 - Else set SLoD = i where $i = 1,2$, or three triangles subdivided.
 - Stop when the subdivision condition is false.
 - Deactivate its old masses, stiffness constant and damping coefficients.
 - Deactivate old time step.
 - Load its new masses, stiffness constant and damping coefficients.
 - Activate and change its time step.
 2. Next face
 3. End loop.

During the subdivision process, Sub-levels of Detail (SLoD) are maintained. If in the next subdivision only one triangle is subdivided further, then its SLoD is one. If

two triangles are subdivided then SLoD is two. If three triangles are subdivided then its SLoD is three and if all triangles are subdivided then SLoD is four. SLoD increase with the number of triangles subdivided irrespective of the position of the triangles. This is shown in Figure 5.9. In Figure 5.9, (a), (b) and (c), only one triangle is subdivided therefore its SLoD is one. This is also shown for two, three, and four triangles subdivisions.

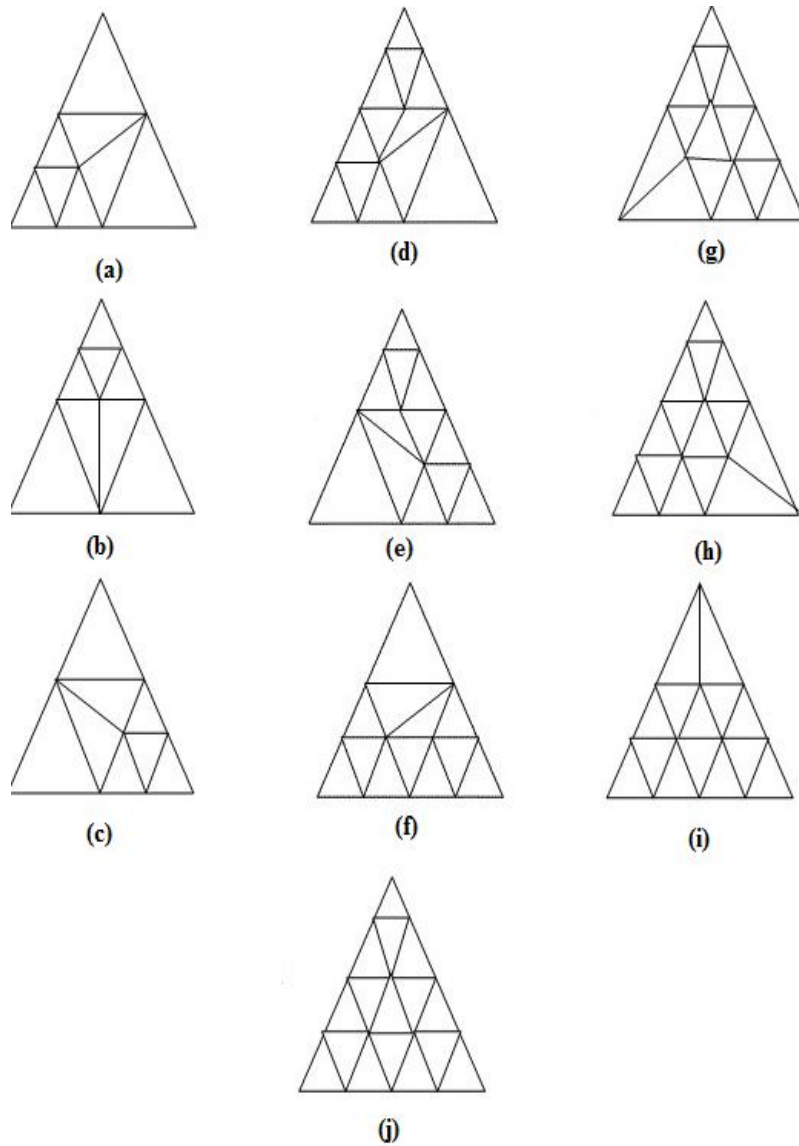


Figure 5.9: A second level of subdivision based on triangles' area. In (a), (b) and (c), SLoD = 1, (d), (e) and (f), SLoD = 2, (g), (h), and (i), SLoD = 3, (j) SLoD = 4

During interactive simulation, the user touches the nodes of the model with the force feedback device. The touched node and its one-neighbourhood is refined based on the area of a triangle. This is shown in Figure 5.10 for the various nodes touched during an interaction with the model using the force feedback device.

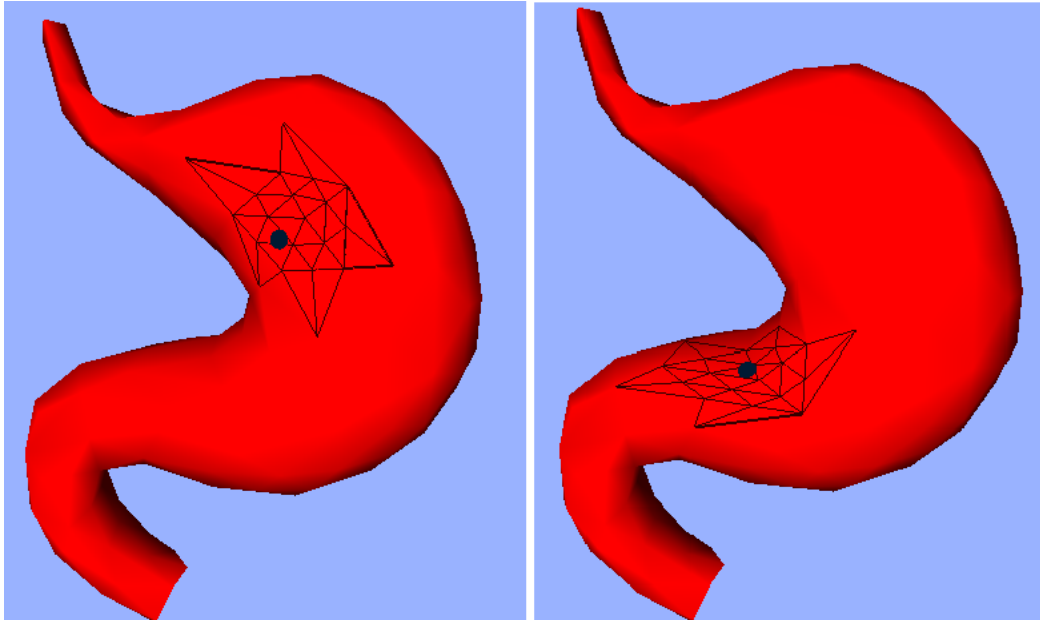


Figure 5.10: Visualizing neighbour triangles during collision of an HCP with two different nodes

5.2.2.4 Description of the Simplification Algorithm

If the user interacts with a particular node then this touched node and its one-neighbourhood is refined. However, once the user moves from this node then this node and its one-neighbourhood needs to return to the coarse mesh in order to lighten the computational burden. For this purpose a simplified algorithm is applied which revert the mesh back to the original coarser mesh.

The steps in the simplification algorithm are as under.

1. During simulation, the collision between Haptic Contact Point and nodes of a model is checked.

2. If (isSubdivided && notTouched)
 - i. Loop through one-neighbourhood faces of this touched node.
 1. Load old Face.
 2. Deactivate new masses, stiffness constants, and damping coefficients.
 3. Deactivate new time step.
 4. Load old masses, stiffness constants, and damping coefficients.
 5. Load old time step.
 - ii. Next Face
 - iii. End loop.
3. End if

5.2.3 Parameter Identification

In an adaptive area based visio-haptic deformable model, when the region of interest is refined then new nodes and springs are produced. In order to maintain the physical properties of the soft tissue after refining the region of interest, new nodes, and springs created must be adjusted so that it does not affect the physical properties of soft tissue in that region. For this purpose new masses, spring stiffness, and damping coefficients are recalculated such that physical properties remain the same before and after subdivision.

The physical properties of the original model are calculated based on the Young's Modulus of the soft tissue. These physical properties are represented by the mass, spring stiffness and damping constant of the mass spring model. Therefore, during real-time interaction with the addition of new nodes and new springs, these physical

properties changes, and thus needs proper readjustment. The following sections describe the method for adjusting these parameters during real-time interaction among the different levels of detail (LoD).

5.2.3.1 Mass Identification

In an adaptive subdivision process, each level of subdivision creates new nodes that need to be assigned new masses such that the total mass of the model remains the same. Assigning the same mass to the newly created nodes will increase the total mass of the model. Therefore, after each subdivision level, it is necessary to recalculate the mass of each node in order to preserve the overall mass of the model.

In the subdivision process, there are two types of nodes, old nodes and newly created nodes. To find new mass for the old nodes, the formula is given in Equation 5.10.

$$M^i_{new} = M^i_{old} * \frac{N}{N'} \quad (\text{Eq. 5.10})$$

To find the mass of newly created nodes, Equation 5.2 is used as

$$M^i_{new} = \frac{N * (M^i_{old} + M^{i+1}_{old})}{2 * N'} \quad (\text{Eq. 5.11})$$

Whereas N is the number of old nodes and N' is the number of new nodes.

These equations are used for finding the masses of old and new nodes in the preprocessing step. During the simulation as only selected faces are subdivided therefore, some nodes especially the boundary nodes could be at the different levels of detail. The mass of such nodes needs to be recalculated in order to maintain the overall mass of the model. Consider the red node shown in Figure 5.11; it is connected with nodes that exist at different levels of detail. Therefore, although it has recalculated mass using Equation 5.10 and Equation 5.11 but these equations does not guarantee the stability of the model during an interactive simulation. So the mass of

such a node is recalculated again during the simulation. For the red node in Figure 5.11, the mass is recalculated as given in Equation 5.12, [125].

$$M_{new} = (M_{L_0}) * 1 + M_{L_2} * 4 / 5 \quad (\text{Eq. 5.12})$$

Where M_{L_0} is the mass of a node at zero-level of the subdivision and M_{L_2} is the mass of a node at 2nd level of subdivision. Red node has one edge with zero-level node and four edges with 2nd level node.

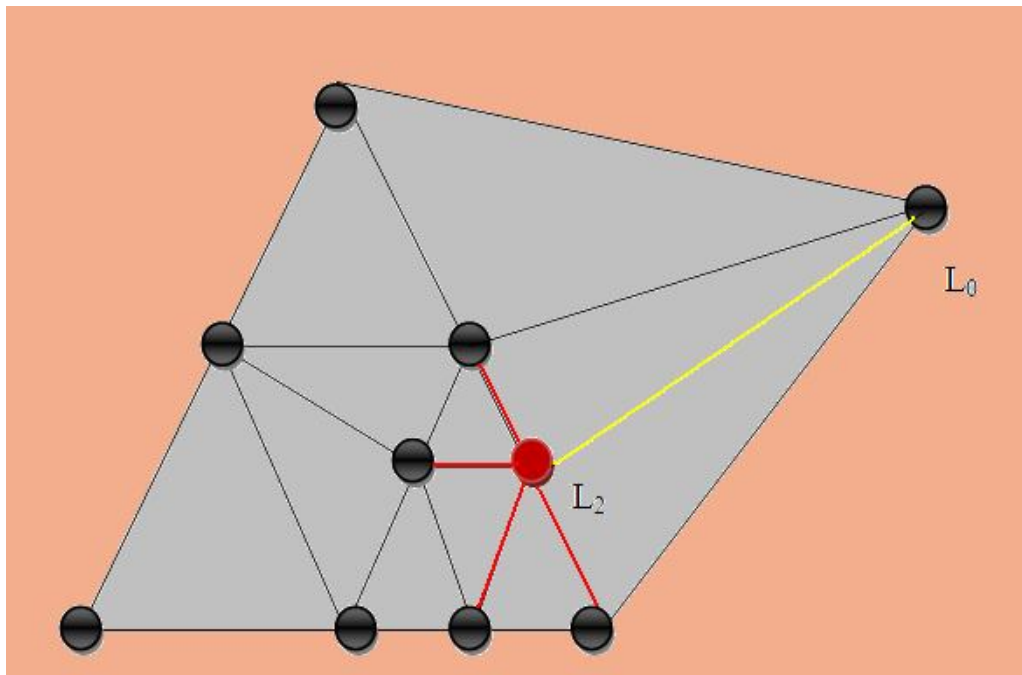


Figure 5.11: Adjustment of mass in different levels of subdivision

The general equation is

$$M_{new} = \frac{M_{L_0} * \text{NoofEdgesatL}_0 + M_{L_i} * \text{NoofEdgesatL}_i}{\text{NoofEdgesatL}_0 + \text{NoofEdgesatL}_i} \quad (\text{Eq. 5.13})$$

In this case, as the subdivision is based on the area of a triangle therefore during simulation, the nodes exist at various levels of subdivision and not only either at the

zero-level or at the highest level. Therefore, we modified and generalized Equation 5.13 for finding the new mass of nodes present in various levels of subdivision other than zero-level. The general equation becomes then is

$$M_{new} = \frac{M_{L_i} * NoofEdgesatL_i + M_{L_{i+1}} * NoofEdgesatL_{i+1}}{NoofEdgesatL_i + NoofEdgesatL_{i+1}} \quad (\text{Eq. 5.14})$$

Here L_i is the i^{th} level of subdivision while L_{i+1} is the next level of subdivision.

5.2.3.2 Spring Stiffness and Damping Coefficient Identification

An adaptive refinement process also creates new springs. The spring stiffness k_s , and damping coefficient k_d need to be recalculated so that the surface reflects the same force through haptic force feedback as before the subdivision, otherwise the surface will feel stiffer. In the proposed algorithm, when the haptic force feedback device touches a node, then its one-neighbourhood triangles are refined based on the threshold area set as a refinement criterion. The subdivision process creates variable numbers of springs in different parts of the region of interest. These new springs must be assigned proper spring stiffness and damping coefficients so that it requires the same force for deformation as before subdivision and same force reflection during interaction as before the subdivision.

The acceleration of a node is important in the subdivision process. The acceleration of a node should be kept the same in all levels of subdivision as was in the coarse model. Acceleration depends upon the external force, the spring force, and mass of the nodes in the model. The external force applied is the same for the coarse model as well as for the refined models of all levels. The spring force changes, if the spring stiffness and damping coefficient remain the same in the refined model as was in the coarse mass spring model. Therefore, to maintain the same acceleration of the node before, and after the subdivision process, the spring stiffness and damping stiffness need to be recalculated.

According to [125], the spring stiffness and damping coefficients are calculated as

$$k_{s(refine)} = 2^l * k_{s(coarse)} * \frac{(\sum_{v \in P_{refine}} Mass(v))/N_{refine}}{(\sum_{v \in P_{coarse}} Mass(v))/N_{coarse}} \quad (\text{Eq. 5.15})$$

In the proposed algorithm, in each level of subdivision, one triangle is subdivided into four using the midpoint subdivision based on the threshold area. As mentioned in the algorithm description section, if a triangle is divided into four we increase the subdivision level l . Each subdivision level l contains four sub-levels denoted as SLoD. If in any level l , one triangle is further subdivided then its sub-level SLoD = 1. If two triangles are divided then SLoD = 2 and so if three triangles are subdivided then it's SLoD = 3 and similarly if all the triangles are subdivided then it's SLoD = 4 and this is equal to level $l + 1$. There are four sub-levels with each level of subdivision. To find the stiffness coefficient for the area based subdivision, an Equation 5.6 is modified, and given in Equation 5.16 as

$$k_{s(refine)} = \frac{SLoD}{NoofSublevel} * 2^l * k_{s(coarse)} * \frac{(\sum_{v \in P_{refine}} Mass(v))/N_{refine}}{(\sum_{v \in P_{coarse}} Mass(v))/N_{coarse}} \quad (\text{Eq. 5.16})$$

Similarly, the damping coefficient becomes as

$$k_{d(refine)} = \frac{SLoD}{NoofSublevels} * (k_{d(coarse)} * \frac{k_{s(refine)}}{k_{s(coarse)}}) \quad (\text{Eq. 5.17})$$

Acceleration of the node also depends upon the time step used during the simulation. This parameter also needs modification. The time step for the refined level is given as ([125])

$$timeStep = \frac{timeStep_{coarse}}{2^l} \quad (\text{Eq. 5.18})$$

where $timeStep_{coarse}$ is the time step used in rendering of the coarse model and l represents the level of subdivision.

5.2.4 Shape Preservation

As discussed in the previous section, Structure springs are used which connects the surface nodes of the model. These structure springs are not enough to maintain the shape of the model during simulation as the model blows up after a few iterations.

In order to preserve the shape of the model during simulation two separate lists of nodes are maintained for the same model. One list contains nodes that are fixed in space and has no effect of any internal or external force on it. These are known as *fixed nodes*. Another list contains nodes that change its position whenever any internal or external force is applied to it. These are known as *deformable nodes*. Springs known as *home springs* are used which connect each of the nodes from the *fixed nodes* to its corresponding node from the *deformable nodes*. These home springs actually maintain the shape of the model and comes into play whenever any external force is added to the deformable node. In the initial configuration, home spring length is zero meaning that the *deformable node* and *fixed node* are at the same position in space. A *deformable node* is allowed to change its place in space upon the external force effect while the fixed node is held in its place, as there is no effect of force on it and acts as an anchor node for the *deformable node*. Structure springs and home springs are shown in Figure 5.12.

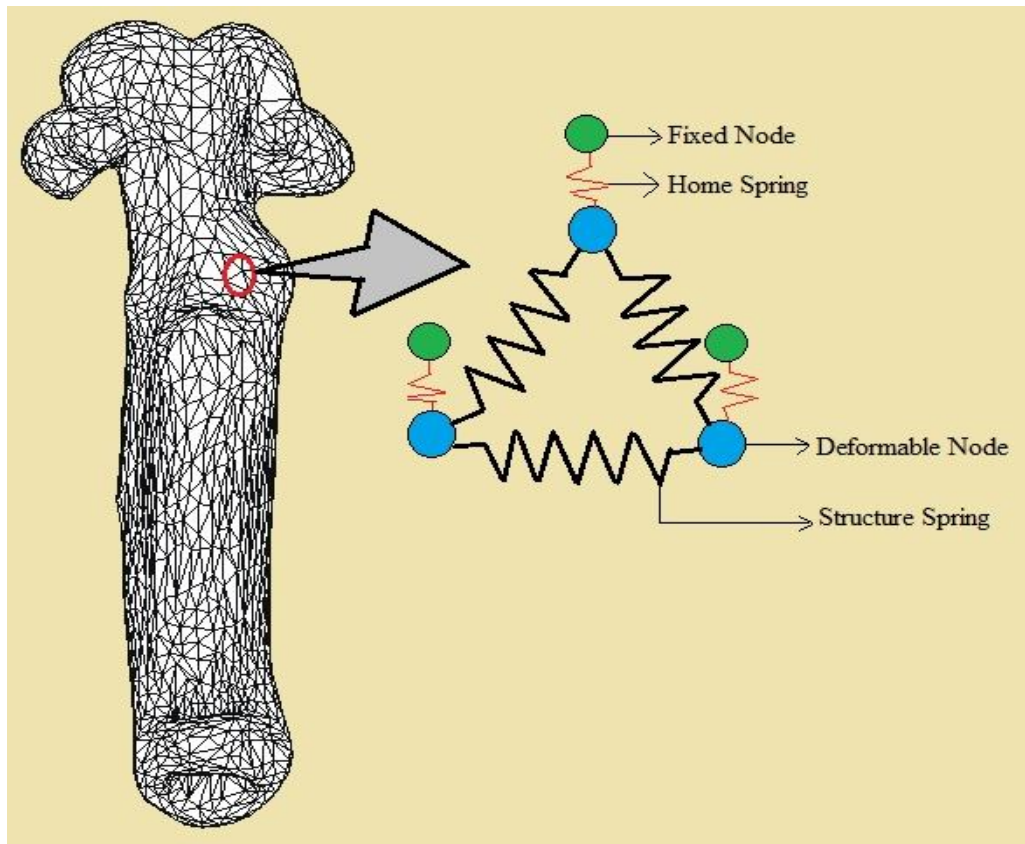


Figure 5.12: Model made up of structural springs, and home springs

Home spring maintains its zero length and tries to maintain this zero length during the entire simulation. When the *deformable node* is deformed then the home spring comes into play to bring it back to its rest position.

Shape preservation springs are used only on the surface based mesh model. The tetrahedral mesh model is an exception from these types of shape preservation springs.

5.3 Criteria for Refinement

During simulation, the user interacts with the soft object through the force feedback device. The user pushes or pulls the soft object during simulation. In palpation simulation the user applies the force on the object whenever trying to palpate different organs. Therefore, an external force is used as a refinement criterion. A threshold

force is set. When the user applies the force, and this force is greater than the threshold force then that particular region is refined for increasing the resolution level to achieve a realistic interactive deformation.

5.4 Octree Space Partitioning (OSP)

An octree space partitioning method is used in the haptic loop to find the touched node in an interactive simulation. This method is discussed in much detail in the previous chapter. The space occupied by the model is partitioned using this OSP, and the nodes of the model are stored in the leaves of an octree. The HCP is constantly searching for collision detection with the model during an interactive simulation, only in the haptic loop. The main idea of an OSP implementation in this research work is to access the touched node quickly in an interactive simulation. Therefore, once the collided node of the model is accessed, then its one-neighbourhood faces could be used for further processing using the proposed method of refinement. As the model is implemented using half-edge data structure, the search time for the one-neighbourhood of any node in the model is constant.

5.5 Results on the Performances of the Model

As the simulator, have many components which all having the potential for improving its performance. In this thesis, the overall performance of the simulator is also considered. For fast and quick collision response, an OSP technique is implemented. For local deformation, only one-neighbourhood of the touched node is considered for further processing. For a realistic visual appearance, an adaptive area based subdivision algorithm is proposed which refines only the region of interest based on the area of a triangle and leaves the rest of the model coarser. All these components greatly improve the performance of the simulator in general. Results are presented for each of these components in the following section. The results are presented using the surface based triangular mesh model and the tetrahedral based model to incorporate the internal physical properties of the model.

5.5.1 Surface Mesh Model

Surface mesh model represents the surface of the real model without considering the interior of the model. This representation is useful for soft tissue that is hollow from inside such as vessels and gall bladder.

5.5.1.1 Triangular Mesh Reduction

Adaptive area based subdivision algorithm dynamically refines the region of interest upon the interaction with the force feedback device. As the model is made up of triangles whose area varies therefore dividing all the triangles smoothly irrespective of the area increase overhead in terms of computation. Here the adaptive area based subdivision is compared with the midpoint subdivision algorithm, which divides the region of interest irrespective of the area of a triangle. Figure 5.13 shows the coarse model on which adaptive area based subdivision and conventional adaptive subdivision algorithms apply to the same region contacted by the force feedback device.

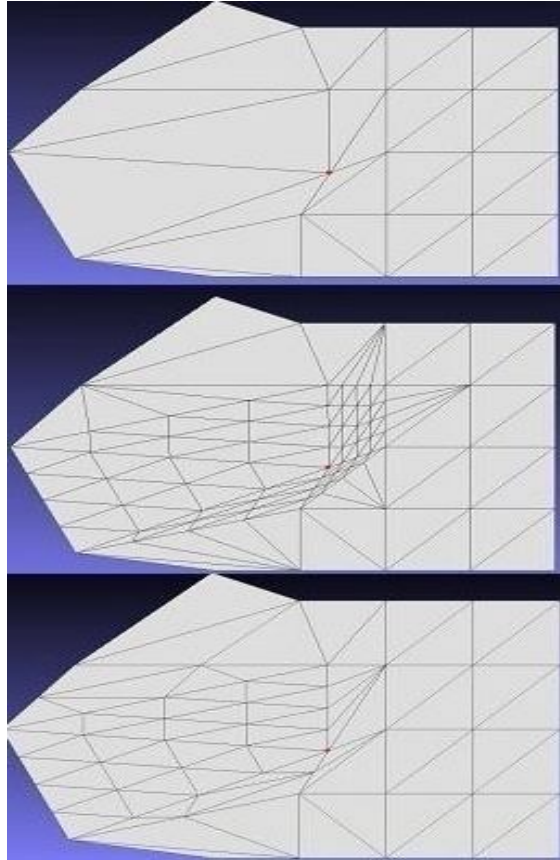


Figure 5.13: Top: The coarse model, Middle: conventional adaptive subdivision,
Bottom: adaptive area based subdivision

In Figure 5.13, only the number of triangles produced after using the proposed method and the conventional adaptive subdivision method is applied to the same model. In this experiment, an MSM is not considered. Figure 5.14 shows the total number of triangles in the coarse model, triangles produced using an adaptive area based subdivision method, and conventional adaptive subdivision method.

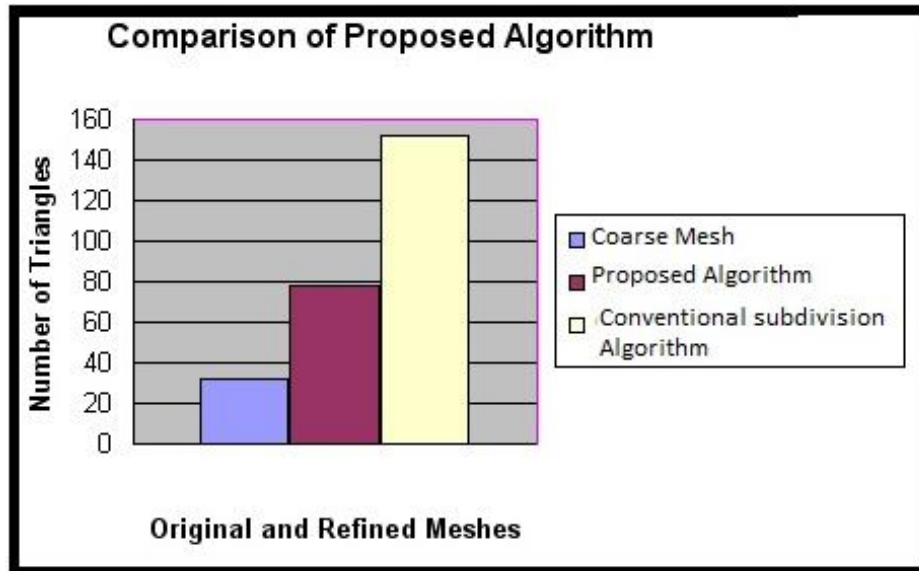


Figure 5.14: Triangles produced using the proposed algorithm and conventional adaptive subdivision algorithm

It is clear from the graph shown in Figure 5.14 for the triangular mesh model given in Figure 5.13 that the proposed algorithm produces a lesser number of triangles (almost 50% lesser) after the same level of subdivision as compared to the conventional adaptive algorithm while maintaining the same resolution. Therefore, for this model, the rendering speed is almost double as compared to the conventional adaptive subdivision. Fewer triangles are produced when the variation among the triangles is high.

5.5.1.2 Reduction in Number of New Masses, Spring Stiffness and Damping Coefficients

In a conventional adaptive subdivision, the region of interest is divided irrespective of the area of a triangle. In this way, smaller triangle divides more and is densely refined as compared to its neighbour triangle with the larger area. The result is that extra and unnecessary triangles need to be rendered. This also produces extra nodes and springs which further consumes time in the computation during simulation for force calculation and node position integration.

As an adaptive area based subdivision is implemented along with the MSM, therefore new nodes, stiffness and damping coefficients are produced. This number is directly proportional to the increase in the variation of the triangle's area and to the threshold area, chosen for the refinement. Figure 5.15 shows the human stomach model made up of triangular mesh and modelled in Blender free 3D modelling software. The coarse stomach model is made up of 192 nodes, 372 faces, and having 564 springs. This model is divided using the proposed adaptive area based subdivision and conventional adaptive subdivision techniques. The number of new masses, stiffness, and damping coefficients created during a simulation of the same touched node are compared in Table 5.1. Reducing the number of these parameters further reduces the rendering time. Due to the lesser number of new nodes, few springs are produced, and therefore this also reduces the integration time for new nodes, and springs while maintaining the same physical properties in that region during the simulation.

As mentioned before, the number of new masses, stiffness coefficients and damping coefficients depend upon the large variation of triangle area in the model and the threshold chosen for a subdivision. A stomach model shown in Figure 5.15 is chosen for comparing the proposed method with the conventional adaptive subdivision for producing new masses, stiffness coefficients and damping constants. Figure 5.16 (a) shows the touched node subdivided using the proposed method and Figure 5.16 (b) shows the same touched node subdivided using conventional adaptive subdivision method. The subdivision level of conventional adaptive subdivision is set to three i.e. $l = 3$. A threshold area in Figure 5.16 (b) is set and it resembles visually the same with the subdivision details shown in Figure 5.16 (a).

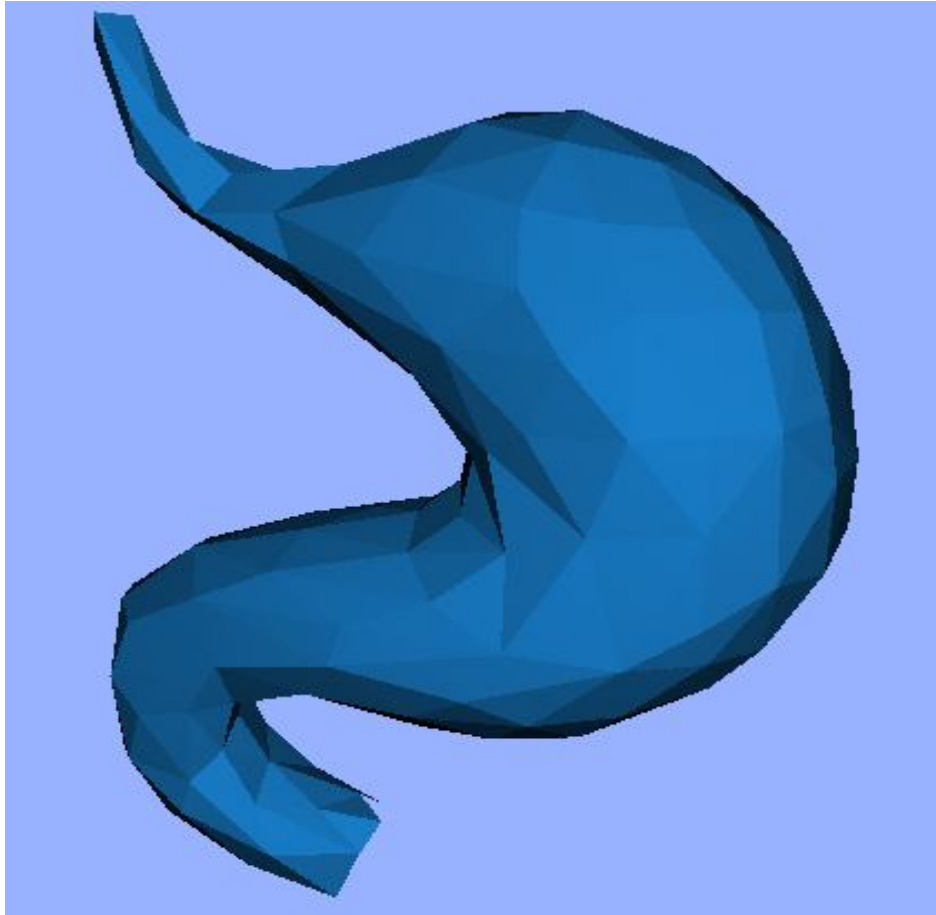


Figure 5.15: A stomach model used in simulation

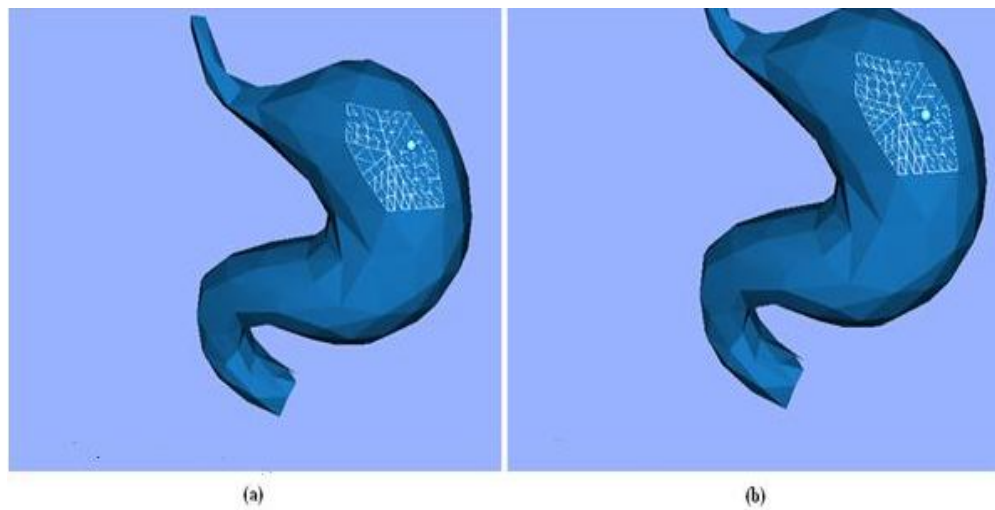


Figure 5.16:(a) A touched node subdivision using proposed algorithm and (b) using conventional adaptive subdivision

Table 5.1, shows the data collected for the same touched node using conventional adaptive subdivision and the proposed algorithm. For the same level of subdivision, lesser number of triangles is produced as compared to the conventional subdivision method. This means that a lesser number of new masses, stiffness, and damping coefficients need to be assigned. This improves the performance of the simulation.

Table 5.1: Parameters created during Conventional Adaptive Subdivision and Area Based Adaptive Subdivision Algorithms for the same surface mesh model

| Parameters | Conventional Adaptive Subdivision | Area Based Adaptive Subdivision |
|--------------------------------------|-----------------------------------|---------------------------------|
| New Masses | 91 | 81 |
| New Stiffness Coefficients (k_s) | 234 | 214 |
| New Damping Coefficients (k_d) | 234 | 214 |

It is apparent from Table 5.1 that the proposed method produced a lesser number of new masses, spring stiffness, and damping coefficients, when compared with the conventional adaptive subdivision. The visual realism is the same in both of these methods. This difference could be improved further having triangles in the neighbourhood of greater variation in its area.

5.5.2 Tetrahedral Mesh Reduction

The tetrahedral mesh model is more complex than the surface mesh model, due to the extra nodes and springs for the internal physical properties. This model is more realistic and suitable for the soft tissue representation. The following sections show that a lesser number of nodes, and springs are produced, using the proposed method on the tetrahedral mesh models.

5.5.2.1 Model Reduction

An adaptive area based subdivision algorithm dynamically refines the region of interest upon the force feedback device. As the model is made up of the tetrahedron whose area varies therefore dividing all the triangles smoothly irrespective of the area increase overhead in terms of computation. In this tetrahedral model, only the surface faces are refined and the internal faces are not considered in the refinement. Here the adaptive area based subdivision is compared with the midpoint subdivision algorithm, which divides the region of interest irrespective of the area of a triangle. Figure 5.17 shows the coarse tetrahedral model on which adaptive area based subdivision and conventional adaptive subdivision algorithms apply to the same region contacted by the force feedback device. The original tetrahedral model contains five tetrahedral with 18 faces. When this model is refined using the proposed method, then for the same visual appearance, it produces 118 faces using the threshold area (for Figure 5.17, $thresholdArea = 0.08$). While for the same visual appearance, conventional adaptive subdivision produces 144 faces using two levels of refinement.

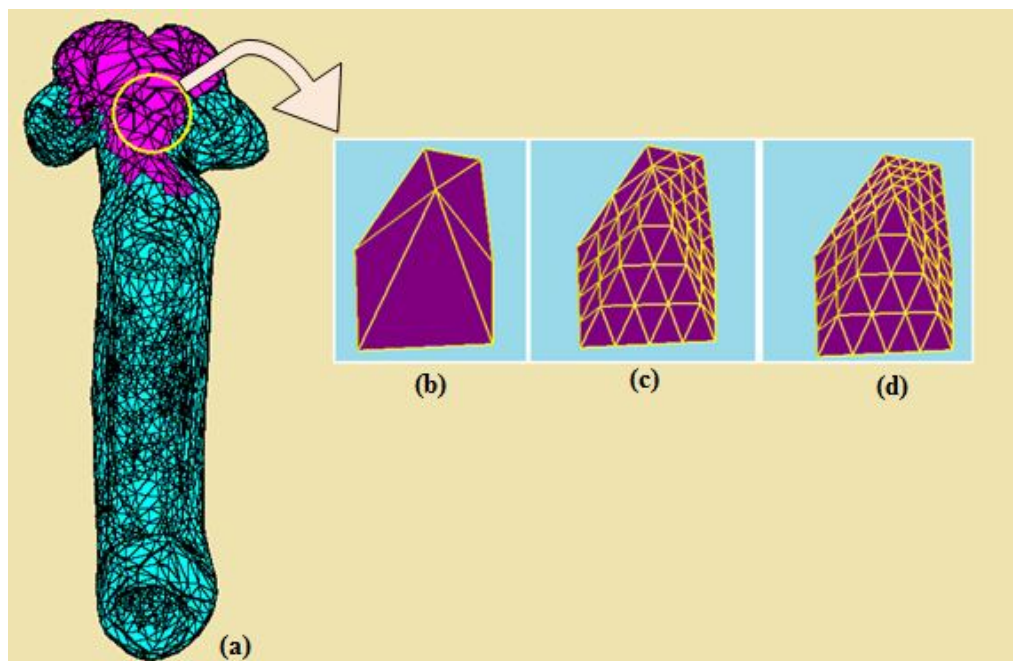


Figure 5.17:(a) A tetrahedral mesh model, (b) Portion of coarse tetrahedral mesh model (c), Adaptive area based subdivision, (d) Conventional adaptive subdivision

5.5.2.2 Reduction in Number of New Masses, Spring Stiffness and Damping Coefficients

In a conventional adaptive subdivision, the region of interest is divided irrespective of the area of a triangle. This way smaller triangle divides more and is more refined as compared to its neighbour triangle with the larger area. The result is that extra and unnecessary triangles need to be rendered. This also produces extra nodes and springs which further consumes time in the computation during simulation for force calculation and node position integration.

As the area based adaptive subdivision is implemented along with the MSM, therefore new masses, stiffness and damping coefficients are produced. The number of new masses, stiffness, and damping coefficients also depends on the variation of the area of a triangle in the model, and value of the threshold area. The surface mesh model of the reproductive tract model contains 1500 nodes, and 3000 faces, is modelled in Blender free 3D modelling software. This surface model is then converted into a tetrahedral mesh using tetgen software. The coarse tetrahedral model contains 8744 tetrahedral, 2487 nodes, and 4974 boundary faces. In Figure 5.16 (b), a portion of tetrahedral mesh is chosen that contains 5 tetrahedral, 8 nodes, 18 faces, and 19 springs. This model is divided using the proposed adaptive area based subdivision method with the threshold area is 0.08, and conventional adaptive subdivision techniques using two levels of subdivision. The number of new masses, stiffness, and damping coefficients created during a simulation of the same touched node are compared in Table 5.2. Reducing the number of these parameters further reduces the rendering time. Besides this, with lesser number of nodes are produced as well as spring length calculation, the spring force calculation is reduced.

Table 5.2: Parameters created during conventional adaptive subdivision and adaptive area based subdivision algorithms for the same tetrahedral mesh model

| Parameters | Conventional Adaptive Subdivision | Area Based Adaptive Subdivision |
|---|--|--|
| New Masses | 62 | 50 |
| New Stiffness Coefficients (k_s) | 199 | 161 |
| New Damping Coefficients (k_d) | 199 | 161 |

5.5.3 Collision Detection Using Space Partitioning Method

Collision detection plays a very important role in the simulation, especially in the visio-haptic simulation. In this thesis, an octree space partitioning method is used for collision detection as shown in Figure 5.18. In addition, point based collision detection is used. As this is a deformable model, therefore, a point based collision detection method is the best method. The reason is that the nodes in the soft tissue changing the position constantly. In this thesis, as HCP act as a 3D point, its collision is detected with the node in the model. For this purpose, we used an OSP to partition the space and then used the nearest neighbour to find the nearest node to an HCP in the corresponding octant.

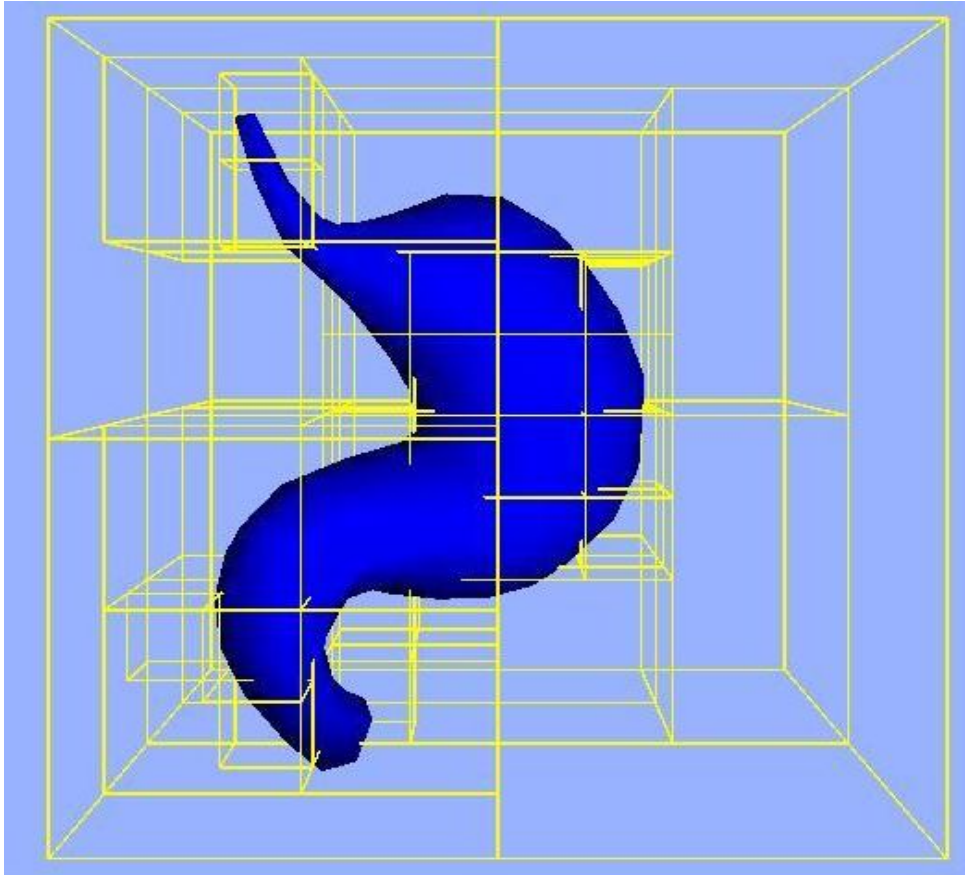


Figure 5.18: A stomach model partitioned using an octree method

In the half-edge based data structure, once the touched node is found then there one-neighbourhood can be found in constant time. Therefore, for subdivision we choose this one-neighbourhood of a touched node once the external force on this node exceeds the threshold force. The octree method is implemented in the haptic loop for locating quickly the touched node with the haptic force feedback device.

Table 5.2 shows a comparison of the collision detection using an OSP and nearest neighbour, which is available in the OpenHaptics API for models having different resolution. An OpenHaptics provides nearest neighbour algorithm, which searches the entire list of nodes in the model for finding the closest node with HCP. If the model is complex then this technique becomes too expensive in terms of time, which makes unrealistic real-time interactive simulation. Therefore, algorithms are required, which reduce time required for searching the touched node in the model. In Table 5.2, for an OSP technique, three levels of subdivisions are used, and the number of nodes stored

at each leaf of an octree is 25. The levels of subdivision, and the number of nodes are ANDed for stopping the subdivision process, and the storage of nodes per leaf whenever whichever comes first.

Table 5.3: Search time for touching node using octree and nearest neighbour method for models of different resolution

| Model Resolution | Search Time (Microseconds) | | | |
|-------------------------------------|-----------------------------------|------|------|--------------------------|
| | Octree Space Partitioning | | | Nearest Neighbour |
| | (Nodes per leaf =25) | | | |
| | Level of Subdivision | | | |
| | 1 | 2 | 3 | |
| Vertices: 642 Triangles: 1281 | 0.15 | 0.12 | 0.11 | 0.60 |
| Vertices: 1442 Triangles: 2881 | 0.83 | 0.20 | 0.12 | 1.25 |
| Vertices: 1552 Triangles: 3041 | 0.84 | 0.22 | 0.15 | 1.33 |
| Vertices: 2282 Triangles: 4561 | 1.22 | 0.37 | 0.21 | 2.06 |
| Vertices: 9122 Triangles: 18241 | 2.72 | 1.35 | 0.48 | 8.54 |
| Vertices: 23522 Triangles: 47041 | 13.50 | 4.92 | 1.35 | 22.26 |

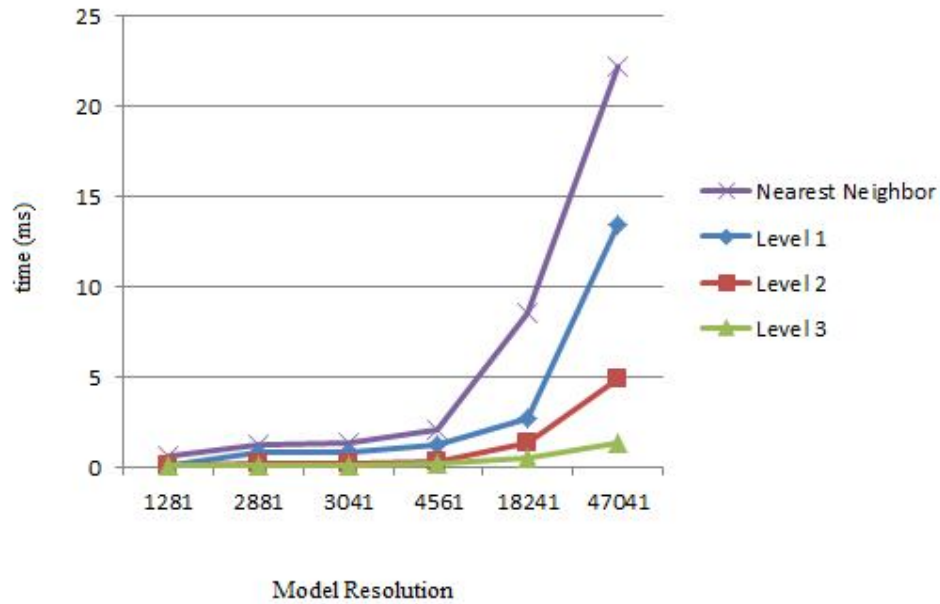


Figure 5.19: A graph for Table 5.2

Figure 5.19 that depicts graphically the data of Table 5.3 shows that the space partitioning method gives faster access to the collided node during the simulation. This is shown only for the three levels of the subdivision process. The rendering time could be reduced further if more subdivision levels are used. The choice of node per leaf and levels of subdivision also depends on the model resolution. For the last model in Table 5.3 that is more complex, the rendering time could be reduced further, when the levels of subdivision are increased.

5.5.4 Deformation Using Adaptive Subdivision Algorithm

The proposed algorithm is compared with refined model and with the conventional adaptive method. The following sections show the results obtained during the simulation.

5.5.4.1 Comparison of the Proposed Algorithm with the Refined Model

The proposed algorithm is compared with the refined model. The proposed algorithm adaptively subdivided the region of interest whenever touched with the force feedback device. Deformation of the model occurs in this region of interest when touched by the force feedback device.

The refined method, subdivides the whole model, irrespective of the region of interest. This way more faces are produced and need to be rendered that also involves extra computation for the mass spring model. In Table 5.4, two models are used, one is a human stomach model, and the other is the reproductive tract of the cow. In these models, number of faces, nodes, and springs produced due to refined method, and the proposed method are shown. The next column in Table 5.4 shows the execution time measured in microseconds for the refined and the proposed methods. The execution time is shown separately in the graphic loop, and the haptic loop. Figure 5.20 shows the interactive deformation using the refine model, and using the proposed.

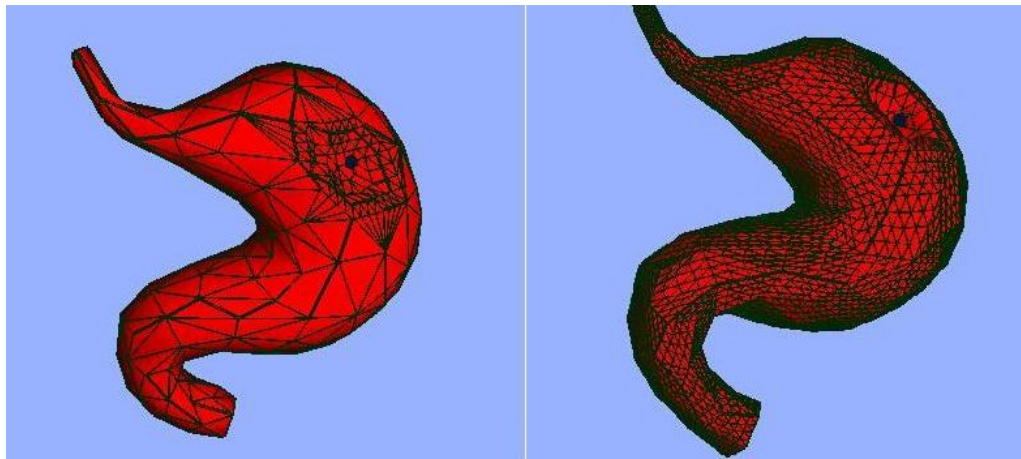


Figure 5.20: Stomach model (Left) Deformation using adaptive area based method
(Right) Deformation using refined model

Both the refined method and the proposed method implements an octree space partitioning (OSP) method for fast collision detection of an HCP with the nodes of the model. In addition, both methods use one-neighbourhood of the touched node for the haptic rendering and for simulating the local deformation.

Table 5.4: Comparison of proposed method with the refined model

| Model Type | Model Resolution | | | | | | Execution Time (ms) | | | |
|--------------------------|------------------|-------|---------|-----------------|-------|---------|---------------------|-------------|-----------------|-------------|
| | Refined Method | | | Proposed Method | | | Refined Method | | Proposed Method | |
| | Faces | Nodes | Springs | Faces | Nodes | Springs | Graphics Loop | Haptic Loop | Graphics Loop | Haptic Loop |
| Stomach Model | 5952 | 3000 | 2244 | 522 | 263 | 783 | 29.40 | 3.97 | 3.43 | 0.06 |
| Reproductive Tract Model | 24000 | 12000 | 36000 | 1550 | 775 | 2325 | 110.6 | 15.45 | 5.40 | 0.08 |

Further analysis of the data in Table 5.4 shows that the execution time difference in the refined and the proposed method in the graphics loop for the simulation of the stomach model is 25.97 microseconds, and in the haptic loop the difference is 3.91 microseconds. Similarly, the execution time difference in the refined method and the proposed method in the graphics loop for the simulation of the reproductive tract model is 105.26 microseconds, and in the haptic loop the difference is 15.37 microseconds.

What is interesting in the data given in Table 5.4 is that the execution time increases exponentially with the increase in the resolution of the model for the refined method, while there is a slight change in the execution time for the proposed method. In Table 5.4, using the proposed method, the execution time difference between the stomach and reproductive tract model in the haptic loop is 0.02 microseconds, while this difference using the refined method is 11.78 microseconds. Similarly, the

execution time difference between these two models in the graphic loop using the proposed method is 1.97 microseconds, while this difference using the refined method is 81.20 microseconds.

5.5.4.2 Comparison of Proposed Algorithm with Conventional Adaptive Subdivision Method

The proposed method is compared with the conventional adaptive subdivision method available in the literature [125]. Two simulators were used on the same platform. One simulator used the proposed method for the refinement, while the other simulator used conventional adaptive subdivision method. A mass spring model is used and implemented in both the simulators. The two methods were used on two models, the human stomach model, and the bovine reproductive tract model, having different resolution. The models were interactively refined and deformed, using the proposed, and conventional adaptive method. Figure 5.21 shows the interactive deformation, and the refinement of the stomach model using the proposed and conventional adaptive method. The triangles are shown here for visualizing the refined region during an interactive deformation of the stomach model.

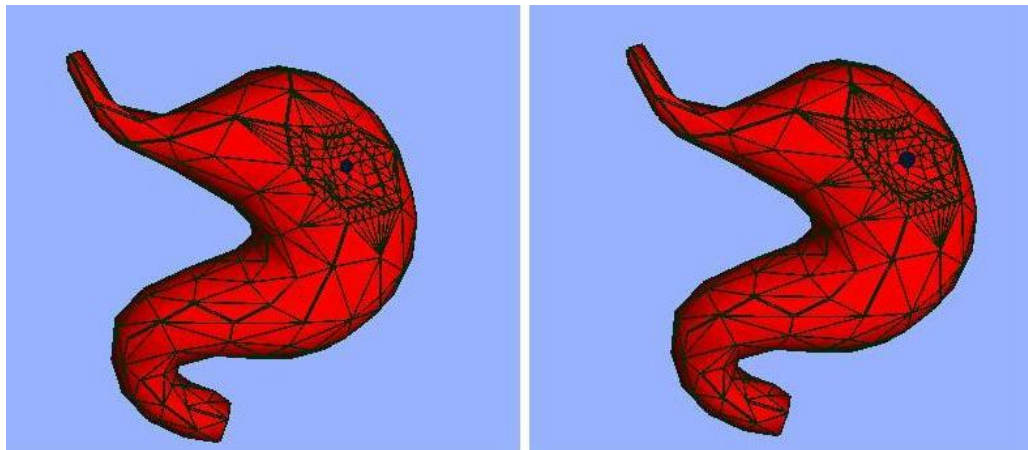


Figure 5.21: Stomach Model (Left) adaptively refined area deformation using the proposed method (Right) adaptively refined area deformation using the conventional method

Both the proposed method and the conventional adaptive refinement method use one-neighbourhood of the touched node for the refinement. In addition, both methods use the one-neighbourhood of the touched node for rendering in the haptic loop for touch sensation. Figure 5.22 shows the real-time interactive deformation of the touched node along with its one-neighbourhood with the force feedback device.

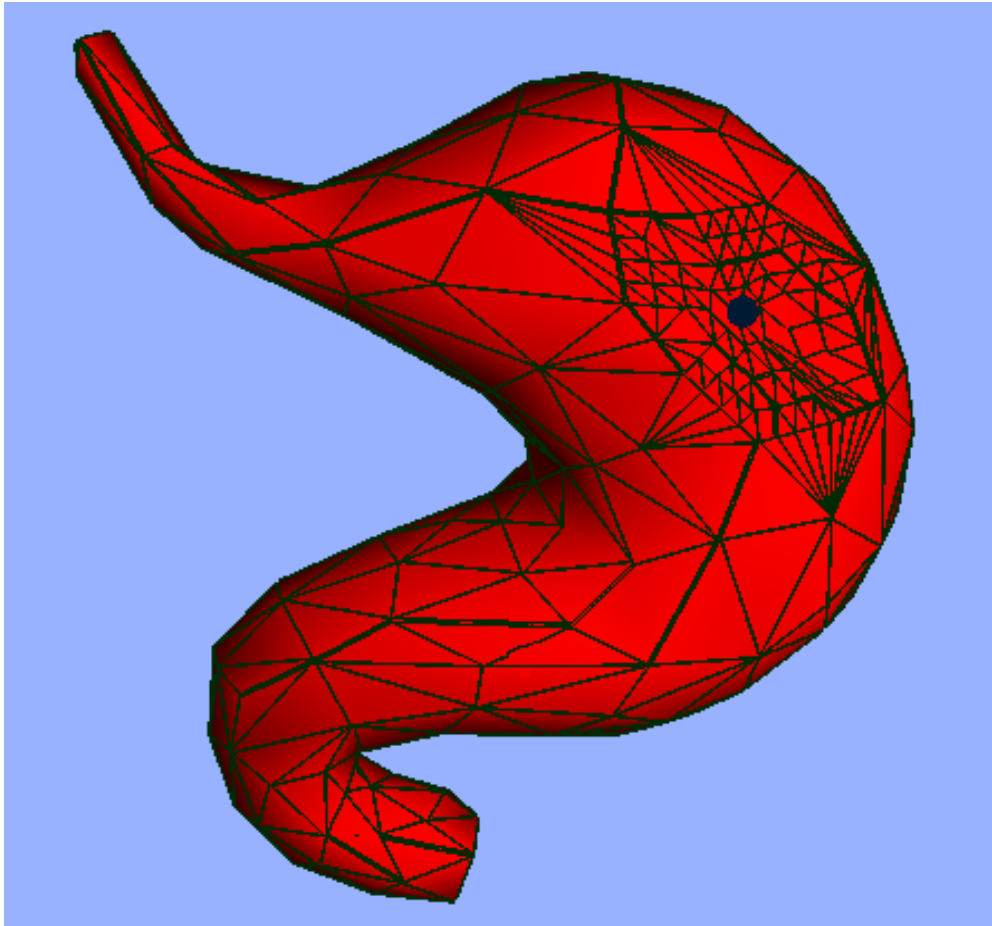


Figure 5.22: Deformation of the touched node with the force feedback device

The execution time includes the time required for the refinement, the time required for the new nodes, and spring creation, the integration time for computing the new nodes position, and the spring force computation time. The execution time is recorded separately in the graphic and haptic loops for both the models. Table 5.5 shows the rendering time in the graphic and haptic loops for the proposed, and conventional adaptive refinement method. Both these methods were applied to the

human stomach model, and reproductive tract model. The two models were different in the resolution.

Table 5.5: Comparison of the proposed method with a conventional adaptive subdivision method based on collision detection techniques

| Model Type | Model Resolution | | | | | | Execution Time (ms) | | | |
|--------------------------|--|-------|---------|-----------------|-------|---------|--|-------------|-----------------|-------------|
| | Conventional Adaptive Subdivision Method | | | Proposed Method | | | Conventional Adaptive Subdivision Method | | Proposed Method | |
| | Faces | Nodes | Springs | Faces | Nodes | Springs | Graphics Loop | Haptic Loop | Graphics Loop | Haptic Loop |
| Stomach Model | 534 | 273 | 807 | 522 | 263 | 783 | 4.93 | 1.05 | 3.43 | 0.06 |
| Reproductive Tract Model | 1644 | 822 | 2466 | 1550 | 775 | 2325 | 11.8 | 2.79 | 5.40 | 0.08 |

Further analysis of the data in Table 5.5 shows that the execution time difference in the refined and the proposed method in the graphics loop for the simulation of the stomach model is 1.50 microseconds, and in the haptic loop the difference is 0.99 microseconds. Similarly, the execution time difference in the refined method and the proposed method in the graphics loop for the simulation of the reproductive tract model is 6.40 microseconds, and in the haptic loop the difference is 2.71 microseconds

What is interesting in the data given in Table 5.5 is that the increase in execution time with an increase in the model complexity is higher for the conventional adaptive subdivision method as compared to the proposed method. In Table 5.5, using the

proposed method, the execution time difference between the stomach and reproductive tract model in the haptic loop is 0.02 microseconds, while this difference using the conventional adaptive subdivision method is 1.7 microseconds. Similarly, the execution time difference between these two models in the graphic loop using the proposed method is 1.97 microseconds, while this difference using the refined method is 6.87 microseconds.

5.6 Summary

In this chapter, an adaptive area based visio-haptic deformable model was introduced. This model uses the mass spring model for representing the physical properties of the soft tissue. An adaptive area based refinement algorithm was used in this deformable model in order to refine the region of interest based on the area of a triangle present in that region. In this way, few triangles were produced during rendering while maintaining the realistic view. This reduced the rendering time in the graphic as well as in the haptic loops. Another main advantage of few triangles productions was that lesser new masses, stiffness coefficients, and damping constants were produced. In addition, the calculation involved with the masses, stiffness coefficients, and damping constants were reduced.

The proposed method was applied to the surface based triangular mesh as well as to the tetrahedral mesh for encoding the physical properties of the soft tissue. In using both the surface based mesh model, and the tetrahedral mesh model, the refinement during an interactive visio-haptic deformation produced new nodes, and new springs. In order to maintain the physical properties of the soft tissue after the refinement, during the simulation new masses were assigned to the newly created nodes and new spring stiffness and damping coefficients were assigned to the newly created springs. The formulae were presented in the assignment of new masses, spring stiffness, and damping coefficients that guaranteed the same physical properties of the soft tissue before and after the refinement of the region of interest. The conventional adaptive technique divided the region of interest without considering the triangle area and therefore produced more triangles to be rendered and more computation required in terms of parameter calculation and re-assigning as compared to the proposed method.

The proposed adaptive area based refinement method, divided the region of interest based on the area of a triangle. Due to the varied area of the triangle in the region of interest, the refinement is not uniform. This produced more triangles in one part and few in the other part in the region of interest. This issue was also considered in the formulae presented for the new masses, spring stiffness, and damping coefficients in such non-uniform region of interest.

A simplification algorithm was also presented that was applied after the refinement of the region of interest. During the interactive deformation, when the user left the region of interest by relaxing the external force, the simplification algorithm was used. The simplification algorithm brought back the old masses, spring stiffness and damping coefficients in that region in order to maintain the same physical properties in the region as was before the refinement method applied.

When the user interacts with the soft tissue during the simulation, OpenHaptics API provides a nearest neighbour method to search linearly the node of the model touched by the user with the force feedback device. For the complex model with high resolution, this method degrades the performance and becomes impractical for the real-time interactive applications. In order to make this process quick and practical for the real-time interactive simulation, an octree space partitioning method was used. This method is faster than the method available in OpenHaptics API as shown by experiments in the results section above.

The octree space partitioning method was implemented in the haptic loop only for quickly detecting the collision between the node of the model and the HCP. Once found the collided node of the model then its one-neighbourhood was used for the refinement and further processing.

In the last section of this chapter, the performance results were presented. In the first part of the results, it was shown that the proposed method created a lesser number of faces, while maintained the same visual appearance of the model during an interactive deformation. This reduced the time required for its rendering in the graphics and haptic loop during each cycle as shown next in the results. As this method was implemented in the mass spring model, therefore, further time was reduced that was required for the computation of new nodes position, velocity, and

stiffness and damping coefficients. In the OpenHaptics API, the available method for searching a node is a linear search method. In this research work, an octree method is implemented and the result showed its applicability in the real-time interactive environment, and outperformed the methods available in the OpenHaptics APIs. Lastly, the proposed subdivision was compared with the refined model of the stomach and reproductive model of the cow and the results showed higher performance for the proposed method. The proposed method was also compared with the currently available adaptive subdivision methods and shows improvement in the performance both in the haptic and graphic loops.

In the next chapter, an evaluation study is conducted for the investigation of multimodal cues, and the extent of visual and haptic cues in a haptic dominant palpation simulator. The evaluation study also validates the proposed visio-haptic deformable models presented in chapters four, and five, respectively.

CHAPTER 6

EVALUATION STUDY

The evaluation study presented in this chapter aims to address the last two research objectives in chapter one, which are as follows:

- To evaluate the haptic only and, visio-haptic parameters of the palpation simulator for learning skill enhancement.
- To assess the extent in which the visual and haptic cues are important in the visio-haptic modality in terms of the visual and haptic rendering requirements.

Section 6.1 presents the objectives of the evaluation study. Section 6.2, Section 6.3, Section 6.3, and Section 6.4, discusses experiment1, experiment 2, experiment3, and experiment 4 respectively. The results obtained from the four experiments are presented in Section 6.5. Section 6.6 summarizes the chapter.

6.1 Objectives of the Evaluation Study

The objectives of the evaluation study were formulated to address the last two research objectives of this thesis presented above. The objectives of the study are as follows:

1. *To evaluate the haptic only and visio-haptic modalities under guidance in the haptic dominant palpation simulator*

Experiment 1 is conducted for the evaluation of this objective. Three prototypes discussed in Section 6.2.3 are used. In this experiment, the visual cue is for the instructor and the haptic sensation for the trainee in order to evaluate its effect on the learning skills enhancement.

2. To evaluate the haptic only, and visio-haptic modalities for learning skills enhancement in a haptic dominant palpation simulator

Experiment 2 and 3 are used to evaluate this objective. Experiment 2 is for the evaluation of haptic only and visio-haptic modalities for short-term skills enhancement. This means that the skills learned are evaluated just after the training. However, according to [158], just acquired motor skills can be lost rapidly in the absence of haptic assistance, even after an intensive training phase. It is believed that a continuous practice of a desired task in a long-term is a practical way to gain a permanent motor skill [12]. Therefore, to avoid biased results, an experiment 3 is conducted to evaluate the haptic only and visio-haptic modalities in a one-week long experiment for long-term skills enhancement.

3. To assess the extent of visual and haptic cues

Experiment 4 is conducted to assess the extent of visual and haptic cues necessary in visio-haptic deformable models. One of the proposed visio-haptic deformable is used with different visual and haptic resolution to evaluate the effect on learning skills enhancement.

In all these experiments, an independent variable is the prototype simulators, while palpation force is the dependent variable. All of the four experiments are randomized experiments in which the trainees are assigned randomly (without considering the gender, and age) to the different prototypes used. Random assignment has three purposes. (i) It helps to distribute the idiosyncratic characteristics of the trainees over different prototypes used in these experiments, so they do not selectively bias the outcomes of the experiment. (ii) The unbiased estimation of error effects are not attributable to the manipulation of independent variable. (iii) It helps to ensure that the error effects are statistically independent [140; 159].

The following sections discuss these experiments.

6.2 EXPERIMENT 1

The objective of this experiment is to assess the impact of visual cue on the skills learning enhancement in the haptic dominant palpation simulator in an instructor-trainee interactive training environment. In this experiment, three types of the simulator are used. An instructor guides the trainee to identify the feature and apply proper palpation forces in an interactive environment based on the extent of visual cue provided to the instructor in the three types of the simulator. The visual cue in these simulators is provided only to the instructor to observe visually and guides the trainee while, the trainee is provided with the haptic sensation for practicing the palpation procedure. The extent of visual cue provided is different in these three types of the simulator.

6.2.1 Participants

Forty volunteer participants took part from the Computer and Information Sciences Department of Universiti Teknologi PETRONAS. Out of these forty participants, ten were academic staff (instructors) and thirty were students (trainees). Out of the thirty trainees, twenty were male trainees and ten were female trainees. The trainees were divided into three groups, namely, group-A, group-B, and group-C, and each group consists of ten trainees. All the trainees were briefed on the purpose of the study and were given a training to interact with the haptic device. All the trainees reported a normal sense of touch and vision, and all of them were right-handed.

6.2.2 Apparatus

The training simulator for bovine rectal palpation was developed using VC++ as a development environment using the OpenGL graphics library for the visual cue, and an OpenHaptics APIs for incorporating the haptic sensation. The training simulator was run on Pentium IV 2.0 GHz processor and 1.43 GB of RAM without any special

processing or graphics hardware. A PHANToM force feedback device i.e. PHANToM Omni was used to feel the haptic sensation in the simulator. The experimental setup for an instructor-trainee real-time interactive simulation used in this study is shown in Figure 6.1.



Figure 6.1: An instructor-trainee experimental setup of the training simulator

6.2.3 Experimental Design

The trainee was required to palpate and diagnose an ovary of the reproductive tract of a non-pregnant cow and apply palpation forces. The instructor was required to guide the trainees in identifying the ovary, and in controlling the palpation forces based on the extent of visual modality provided to the instructor. The visio-haptic deformable models developed in chapters four and five were implemented in the bovine rectal palpation simulator. Three prototypes of the visio-haptic deformable simulator were presented to the instructor to train the trainees, namely, prototype-A, prototype-B, and prototype-C, having a difference in the visual and haptic sensation. Three models of the reproductive tract of the healthy cow, having different resolutions were modelled using a Blender [156] free 3D modelling software, and named as a Model-A, Model-B, and Model-C in order to make it similar to the name of the prototypes which are

using it. Model-A is a surface based triangular mesh that contains 1500 nodes, and 3000 faces. Model-B is also a surface based triangular mesh that is more refined than the Model-A, and contains 6000 nodes and 12000 faces. Model-C is a volumetric model containing tetrahedral mesh in order to capture realistic physical properties of the soft tissue. Model-C contains 8744 tetrahedral, 2487 nodes and 4974 boundary faces.

6.2.3.1 Prototype - A

Prototype-A has haptic, audio, and visual cues but no interactive visual deformation for an instructor during the training sessions. In prototype-A, Model-A is used, having the same visual and haptic sensation used in the currently available simulator for the bovine rectal palpation simulator [28; 30].

6.2.3.2 Prototype - B

In prototype-B, there are haptic and visual cues with interactive visual deformation. In this simulator, a visio-haptic deformable model from chapter four, namely, a vertex based visio-haptic deformable model is implemented. The haptic sensation in this prototype is similar to prototype-A. In addition, it has the interactive visual deformation based on the slope intercept form of a line equation, and Model-B is used in this prototype.

6.2.3.3 Prototype - C

In prototype-C, there are visual and haptic cues with interactive visual deformation. Prototype-C involves the visio-haptic deformable model from chapter five, namely, adaptive area based visio-haptic deformable model. In this prototype, realistic visual and haptic cues are present due to the proposed adaptive area based visio-haptic deformable model. In this prototype, Model-C is used for realistic visual and haptic sensation based on the physical properties of the soft tissue.

6.2.4 Procedure

The experiment consists of (i) pre-training, (ii) training, and (iii) post-training phases for all the three prototypes. In a pre-training phase, both the instructors, and trainees were prepared mentally by giving a presentation about the nature of the experiment. The instructors and trainees were informed about their roles during the experiment. The main objective of the pre-training phase was to familiarize the participant with the manipulation of force feedback device and soft objects. An initial training about the usage of the bovine rectal palpation simulator was also given to the instructors. The instructors were trained also with the force versus the deformation curve in the visual cue included in the simulator. The instructors were trained only about the diagnosis and the palpation of an ovary in this study.

The next step in this experiment was the training phase. The training phase consisted of four training sessions practiced by each trainee of all the three groups. The duration of each training session was five minutes. The trainees from group-A were allowed to use prototype-A, trainees from group-B were allowed to use prototype-B, and the trainees from group-C were allowed to use prototype-C. Instructors were the same for the three groups of the trainees. Each trainee from group-A performed randomly four training sessions with prototype-A, each trainee from group-B performed four training sessions with prototype-B, and similarly each trainee from group-C performed four training sessions with prototype-C. The instructors were different in all four sessions for each trainee. The assignment of instructors (I1 to I10) to different sessions (S1, S2, S3, and S4) for the trainees (Trainee 1 to Trainee 10) in all of the three groups is shown in Table 6.1.

Table 6.1: Assignment of the Instructors to different sessions of the trainees

| Sessions Trainees | Group-A Sessions | | | | Group-B Sessions | | | | Group-C Sessions | | | |
|----------------------|------------------|----|----|----|------------------|-----|----|----|------------------|----|----|----|
| | S1 | S2 | S3 | S4 | S1 | S2 | S3 | S4 | S1 | S2 | S3 | S4 |
| Trainee 1 | I1 | I2 | I3 | I4 | I9 | I10 | I3 | I6 | I5 | I6 | I7 | I8 |
| Trainee 2 | I1 | I2 | I3 | I4 | I9 | I10 | I3 | I6 | I5 | I6 | I7 | I8 |
| Trainee 3 | I1 | I2 | I3 | I4 | I9 | I10 | I3 | I6 | I5 | I6 | I7 | I8 |
| Trainee 4 | I1 | I2 | I3 | I4 | I9 | I10 | I3 | I6 | I5 | I6 | I7 | I8 |
| Trainee 5 | I1 | I2 | I3 | I4 | I9 | I10 | I3 | I6 | I5 | I6 | I7 | I8 |
| Trainee 6 | I1 | I2 | I3 | I4 | I9 | I10 | I3 | I6 | I5 | I6 | I7 | I8 |
| Trainee 7 | I1 | I2 | I3 | I4 | I9 | I10 | I3 | I6 | I5 | I6 | I7 | I8 |
| Trainee 8 | I1 | I2 | I3 | I4 | I9 | I10 | I3 | I6 | I5 | I6 | I7 | I8 |
| Trainee 9 | I1 | I2 | I3 | I4 | I9 | I10 | I3 | I6 | I5 | I6 | I7 | I8 |
| Trainee 10 | I1 | I2 | I3 | I4 | I9 | I10 | I3 | I6 | I5 | I6 | I7 | I8 |

The forces applied by the trainees during palpation were recorded in all of the training sessions. A Stopping Force Threshold (SFT) and Sound Threshold (ST) values were set in prototype-A, with the condition that $ST < SFT$. An SFT is the maximum force, which is applied on a particular part of the reproductive tract and is different for different parts. Similarly, ST is different for different parts as well. For prototype-B and prototype-C only an SFT value was set and same as in prototype-A. In prototype-A, when the trainee reaches ST, a sound is generated and the instructor guides the trainees to release the force. In prototype-B, and prototype-C, the instructor guides the trainees according to real-time interactive visual deformation implemented in the simulator, and there is no audio cue. In all of the three prototypes, the trainees were restricted by an instructor to reach the SFT and not exceed it as it is the upper bound of the force limit.

6.3 EXPERIMENT 2: Short-Term Skill Learning

The objective of Experiment 2 is to assess the impact of the visio-haptic modality on the participant's ability to identify the feature and learn the required palpation forces. Participants were presented with the three prototypes - A, B, and C as discussed above. In prototype-A, only the haptic cue is used. The participants were allowed to use the three prototypes for feature identification, and palpation of the ovary, and were asked to recall those position, and forces.

6.3.1 Participants

Thirty participants from the Al Jouf University participated in this study. All participants were male and in between the age of 22 to 27. All of the participants reported a normal sense of touch and vision, and all of them were right-handed. The experiment took about 30 minutes. All the participants were informed about the purpose of the experiment, possible risks, benefits and their role in the experiment. Participants were divided into three groups namely group-A, group-B, and group-C, and each contained ten participants. Group-A used prototype-A, group-B used prototype-B, and similarly group-C was allowed to use prototype-C.

6.3.2 Apparatus

A reproductive tract of a Malaysian healthy non-pregnant cow was modelled and used in this experiment. The reproductive tract is 35.5 cm long, 5 cm wide with the width of 14.5 cm at the top of the reproductive tract including the two uterine horns.

The PHANToM Omni force feedback device was used for thematic sensation, and placed 90 cm from the ground suitable for the participants in a standing position and visual feedback was provided using 15-inch CRT (Cathode Ray Tube) monitor, which was at the distance of 40 cm away from the participants in a horizontal position. An experimental setup for visio-haptic modality is shown in Figure 6.2.



Figure 6.2: An experimental setup for visio-haptic modality

The side view of the reproductive tract of the Malaysian healthy non-pregnant cow is shown in Figure 6.3 for clarity. The green sphere represents an HCP in an enlarge form for clear visualization.

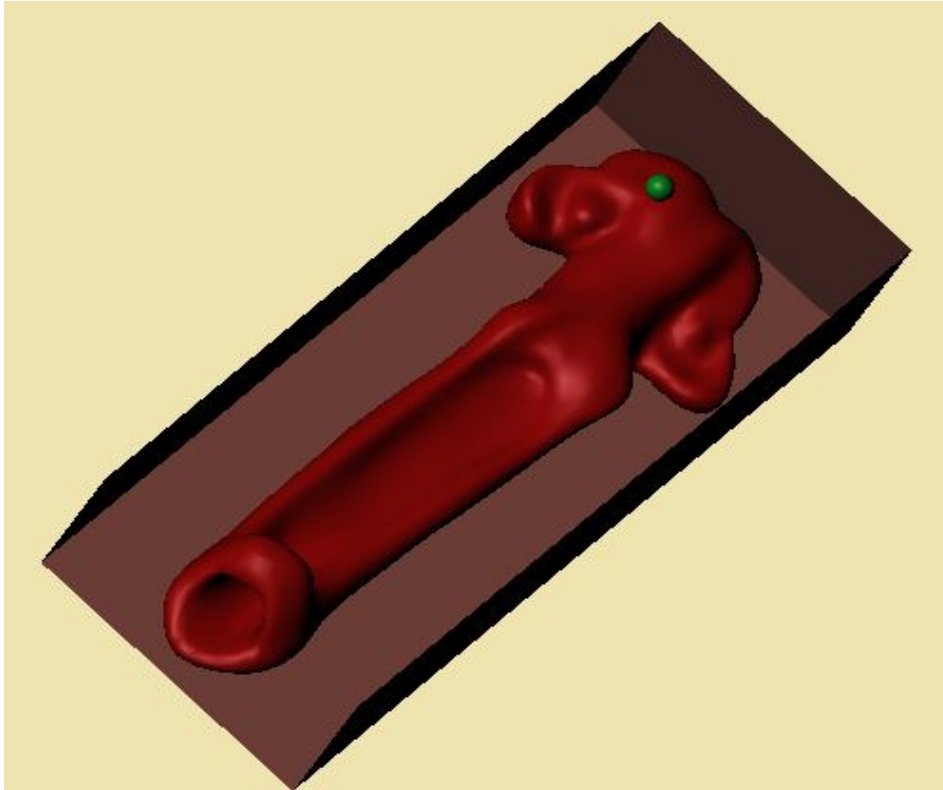


Figure 6.3: Side view of the reproductive tract of a non-pregnant cow

The reproductive tract was scanned through a 3D digital scanner, which was then refined and modelled using Blender [156] free 3D modelling software, and then further remishing and final touch using MeshLab [157] to obtain a surface based mesh model. The volumetric model was obtained using tetgen [146] free tetrahedral remishing software. The training simulator was developed using VC++ as a development environment with the OpenGL graphics library for the visual part, and OpenHaptics APIs for incorporating the haptic sensations and the whole simulator was run on a Pentium IV 2.0 GHz processor with 1.43 GB of RAM.

Three types of the prototype simulator as discussed and presented in Experiment 1 were used in this experiment as well. The only difference was to allow the participants to use both the visual and haptic sensations in the skill learning process.

6.3.3 Experimental Design

All the participants were required to learn the procedure by identifying the feature and by practicing the required palpation force. The two parameters were noted in this experiment for all of the prototypes, which were palpation force and exploration time for reaching to a feature set for all the participants, which in this experiment was the identification of an ovary. Four sessions of 5 minutes were given to each participant of the three prototypes in order to practice the feature identification, and apply the required palpation force once identified the feature.

The training was conducted using three prototypes namely prototype - A, prototype - B, and prototype - C which were already present and discussed in the first experiment of this chapter. Once the trainees practiced and trained in the corresponding simulator, and having learnt the procedure, the trainees were allowed to use a haptic only simulator for the identification of the feature and palpation of the required features.

6.3.4 Procedure

The participants in each group were allowed to use their corresponding training prototype and were allowed to practice in a warm-up session in order to be familiar with the force feedback device and experimental setup.

The experiment consists of (i) pre-training, (ii) training, and (iii) post-training phases for all the three prototypes. In a pre-training phase, the participants were prepared mentally by giving a presentation about the nature of the experiment, Maximum Palpation Force (MPF) that is set differently in the three prototypes, and about the usage of the prototype. In a training phase, the participant practiced in the identification and palpation of the ovary in a reproductive tract of a healthy non-pregnant cow. The training phase consists of four training sessions of five minutes each on its corresponding training simulator with a ten-minute gap between the two training sessions. An MPF value was set in all the simulators, and the participants were required in the training sessions to reach an MPF and avoid applying force

beyond an MPF during palpation. During all the training sessions, three parameters were recorded. The parameters are feature identification, which in this experiment was the identification of an ovary, exploration time, and palpation force. The post-training phase consisted of only one session whose duration was two minutes. The participants were allowed to identify the feature based on their skills developed in the training phase on its corresponding training simulator. The simulator provided to all of the participants in the post-training phase contained haptic only modality. There was no visual or audio cue in this simulator. Exploration time and palpation forces were recorded for all the participants.

6.4 EXPERIMENT 3: Long-Term Skill Learning

The objective of Experiment 3 is to assess the impact of the three prototype using different cues on enhancing the long-term learning skills of the participants. Participants were presented with the three prototypes - A, B, and C as discussed above. In prototype-A, only the haptic cue is used. The participants were allowed to use the three prototypes for feature identification, and palpation of the ovary, and were asked to recall those position, and forces.

6.4.1 Participants

Twenty-four volunteer participants from the Al Jouf University participated in this study. None of these students took part in experiment 1. All participants were male and in between the age of 22 to 27. All of the participants reported a normal sense of touch and vision, and all of them were right-handed. All the participants were informed about the purpose of the experiment, possible risks, benefits and their role in the experiment. Participants were divided into three groups namely group-A, group-B, and group-C, and each contained eight participants. Group-A used prototype-A, group-B used prototype-B, and similarly group-C was allowed to use prototype-C.

6.4.2 Apparatus

The apparatus used in this experiment was the same as used in experiment 1.

6.4.3 Experimental Design

All the participants were required to learn the procedure by identifying the feature and by practicing the required palpation force. The two parameters were noted in this experiment for all of the prototypes, which were palpation force and exploration time for reaching to a feature set for all the participants, which in this experiment was the identification of an ovary. Four sessions of five minutes were given to each participant of the three prototypes in order to practice the feature identification, and apply the required palpation force once identified the feature.

The training was conducted using three prototypes namely prototype - A, prototype - B, and prototype - C which were already present and discussed in the first experiment of this chapter. Once the trainees practiced and trained in the corresponding simulator, and having learnt the procedure, the trainees were allowed to use a haptic only simulator for the identification of the feature and palpation of the required features.

6.4.4 Procedure

The participants in each group were allowed to use their corresponding training prototype and were allowed to practice in a warm-up session in order to be familiar with the force feedback device and experimental setup.

The experiment consists of (i) pre-training, (ii) training, and (iii) post-training phases for all the three prototypes. In a pre-training phase, the participants were prepared mentally by giving a presentation about the nature of the experiment, Maximum Palpation Force (MPF) that is set using different cues in the three prototypes, and about the usage of the prototype. In a training phase, the participant practiced in the identification and palpation of the ovary in a reproductive tract of a

healthy non-pregnant cow. The training phase started on the next day of the pre-training phase. The training phase consists of four training sessions of five minutes each on its corresponding training simulator on separate days. Four training sessions were conducted in four separate days. An MPF value was set in all the simulators, and the participants were required in the training sessions to reach an MPF and avoid applying force beyond an MPF during palpation. During all the training sessions, three parameters were recorded. The parameters are feature identification, which in this experiment was the identification of an ovary, exploration time, and palpation force. The post-training phase, conducted on the seventh day, consisted of only one session whose duration was two minutes. The participants were allowed to identify the feature based on their skills developed in the training phase on its corresponding training simulator. The simulator provided to all of the participants in the post-training phase contained haptic only modality. There was no visual or audio cue in this simulator. Exploration time and palpation forces were recorded for all the participants.

6.5 EXPERIMENT 4: Extent of Visio-Haptic Cues

In this experiment, an extent of visual and haptic cues necessary for palpation force and its effect on the learning skills was considered. The visio-haptic deformable model from chapter five of this thesis was considered for this experiment.

6.5.1 Participants

Twenty participants from the Al Jouf University participated in this study. All the participants were male and in between the age of 22 to 27. All of the participants reported a normal sense of touch and vision, and all of them were right-handed. The experiment took about 30 minutes. All the participants were informed about the purpose of the experiment, possible risks, benefits and their role in the experiment. The participants were divided into two groups namely group-A, and group-B.

6.5.2 Apparatus

In this experiment, only a prototype-C is used as in Experiment 1, but with two different resolutions in the region of interest during simulation. One has a resolution which refines the region of interest with the thresholdArea (A_{tc1}) named as prototype-C1 and the other with the resolution which refines the region of interest using thresholdArea(A_{tc2}) named prototype-C2 where

$$A_{tc2} = A_{tc1}/2$$

The rest of the apparatus, the experimental design, and procedure of conducting were the same as discussed in the second experiment.

6.6 Results

This section presents the results of the four experiments conducted to evaluate the effectiveness of the visio-haptic modality in a haptic dominant environment, as well as validate the proposed visio-haptic deformable models, and the significance of the extent of visual and haptic cues to be considered in the design and implementation of visio-haptic simulators. In all these experiments, data were collected and was statistically analysed using the MINITAB [160] statistical software. An alpha level of .05 was used for all statistical tests.

6.6.1 Experiment 1

The maximum palpation forces (MPF) applied by the trainees during different training sessions under the interactive visual guidance of the instructors were pooled to compare the three prototypes used in the evaluation for the learning skill enhancement. A one-way ANOVA was performed using MINITAB statistical software. The hypotheses are as follows:

1. There is no significant effect on learning palpation force using the three prototypes - A, B, and C.

2. There is a significant effect on learning palpation force using the three prototypes - A, B, and C.

A one-way ANOVA using MINITAB statistical software confirmed a significant difference among the three training prototypes $F(2, 117) = 19.83, p < .001$, (see *Appendix B*) and hence rejected the null hypothesis that there was no difference among the three prototypes. To find the prototype that enhanced the skill learning best among the three types of the prototypes, an independent sample t- test using MINITAB statistical software was performed between the

1. Prototype -A and prototype- B
2. Prototype- A and prototype-C
3. Prototype- B and prototype- C

An independent t- test using MINITAB statistical software confirmed that there was a significant difference between prototype - A ($M = 2.824, SD = 0.250$) and prototype-B ($M = 2.635, SD = 0.198$), $t(78) = 3.76, p < .001$ (two-tailed test). Similarly there was a very clear significant difference between the performance improvement using prototype - A ($M = 2.824, SD = 0.250$) and prototype - C ($M = 2.563, SD = 0.092$), $t(78) = 6.20, p < .001$. Similarly there was also a difference in the performance of prototype - B ($M = 2.635, SD = 0.198$) and prototype - C ($M = 2.563, SD = 0.092$), $t(78) = 2.07, p = .042$. (see *Appendix B*)

The mean and variance of all the three prototypes is compared and shown in Figure 6.4. The mean of prototype-C is very close to the SFT and then prototype-B as compared to prototype-A

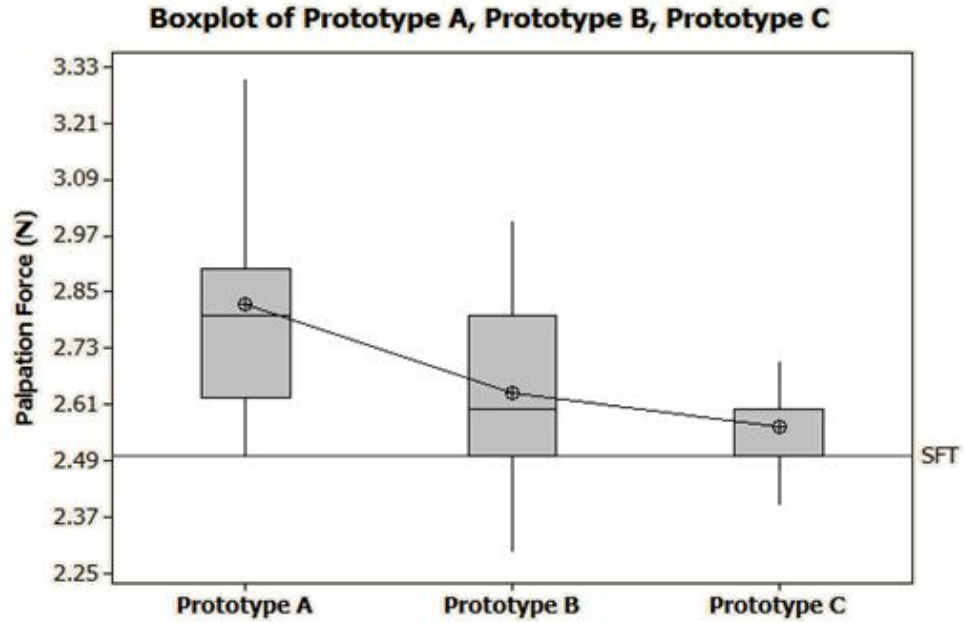


Figure 6.4: Mean and variance of palpation force for prototype-A, B, and C

Similarly, Figure 6.5 shows the accuracy percentage for the three prototypes. The accuracy is higher for prototype-C as compared to prototype-A and B.

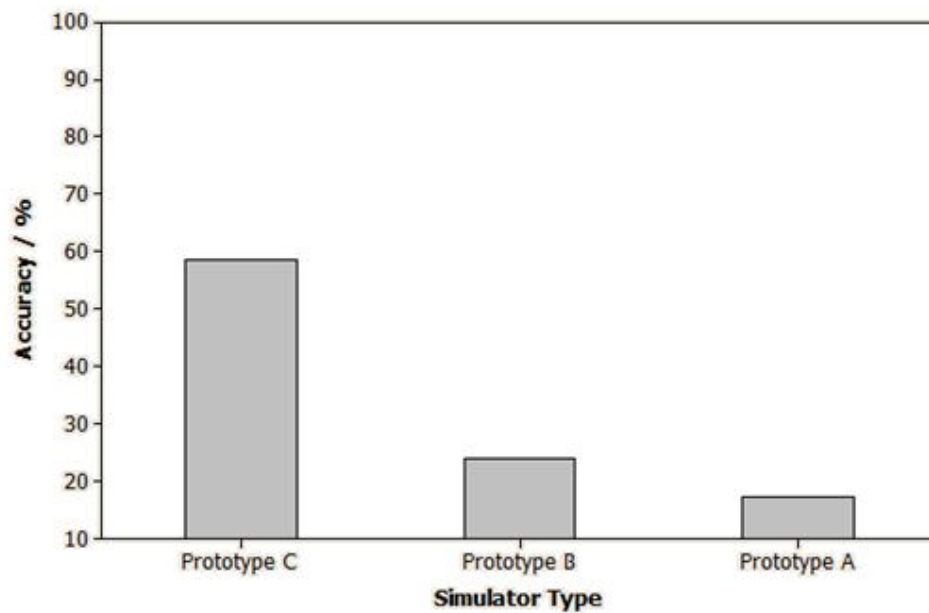


Figure 6.5: Accuracy per percent for prototype-A, B, and C

Prototype-B and C were better than prototype-A due to the fact of interactive visual cue present in these two types. Out of the three prototypes, prototype-C was better due to the realistic visual and haptic deformation as well as proper and realistic coordination between the haptic and visual modality.

Experiment one showed that visio-haptic modality outperformed haptic only modality in an instructor-trainee interactive environment, in which the visual modality for the instructor allowed to better control the palpation forces applied by the trainee.

6.6.2 Experiment 2

Data collected in the post-training phase of prototype-A, B, and, C, were statistically analysed for the skill learning enhancement. An ANOVA was used to examine whether there was any significant difference among the three prototypes in terms of exploration time and palpation force recorded for each participant in a post-training phase. A significant difference was found among the three prototypes for the learning skill enhancement. The palpation force and exploration time were individually analysed in the following subsections.

6.6.2.1 Palpation Force

The maximum value of palpation forces recorded, once the feature was identified, for all the participants in the post-training phase were used for comparing the significance of the prototypes in terms of learning skill enhancement. An ANOVA using the MINITAB statistical software was used to analyse the significance of the data collected. The hypotheses are as follows:

1. There is no significant effect on learning palpation force using the three prototypes-A, B, and C, and
2. There is a significant effect on learning palpation force using the three prototypes-A, B, and C.

A one-way ANOVA using MINITAB statistical software confirmed a significant difference among palpation force for the three training prototypes $F(2, 27) = 18.65$, $p < .001$, and hence rejected the first hypothesis that there was no difference among the three prototypes. Further to find the best among the three prototypes, independent sample t-test using MINITAB statistical software was performed between the following prototypes.

1. Prototype -A and prototype- B
2. Prototype- A and prototype-C
3. Prototype- B and prototype-C

An independent sample t-test confirmed that there was a significant difference between prototype - A ($M = 2.98$, $SD = 0.220$) and prototype-B ($M = 2.67$, $SD = 0.183$), $t(17) = 3.43$, $p = .003$ (two-tailed test). Similarly there was a very clear significant difference between the performance improvement using prototype - A ($M = 2.980$, $SD = 0.220$) and prototype - C ($M = 2.530$, $SD = 0.078$), $t(11) = 6.09$, $p < .001$. Similarly there was also a slight difference in the performance of prototype-B ($M = 2.67$, $SD = 0.183$) and prototype-C ($M = 2.53$, $SD = 0.078$), $t(12) = 2.22$, $p = .046$. The range box and mean of the palpation forces for the three prototypes is shown in Figure 6.6. The mean of prototype-C is closer to the MPF indicating that participants using prototype-C better learnt the skills. In addition, range box in Figure 6.6, shows the variation in the palpation force values of the participants that falls around the MPF for prototype-C, (see Appendix B).

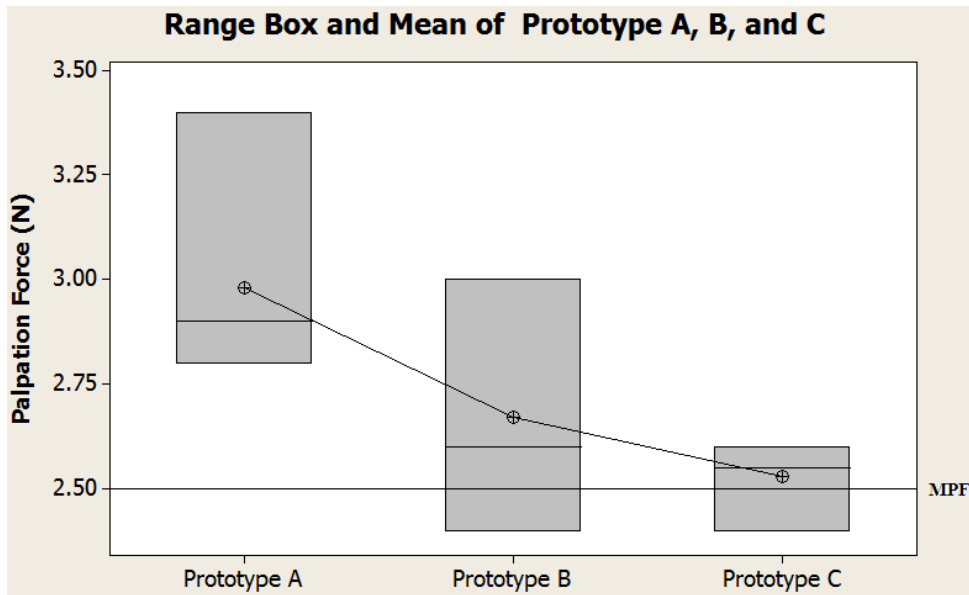


Figure 6.6: Range Box and Mean for palpation force for the three prototypes

Figure 6.7 shows the scatter plot for the palpation forces applied by the trainees during the post-training phase, using the three prototypes-A, B, and C. The graph shows the palpation forces applied using prototype-C is smoother and closer to the threshold palpation force. This indicates that the palpation forces applied by the participants are closer to the MPF using prototype-C as compared to prototype-A, and B.

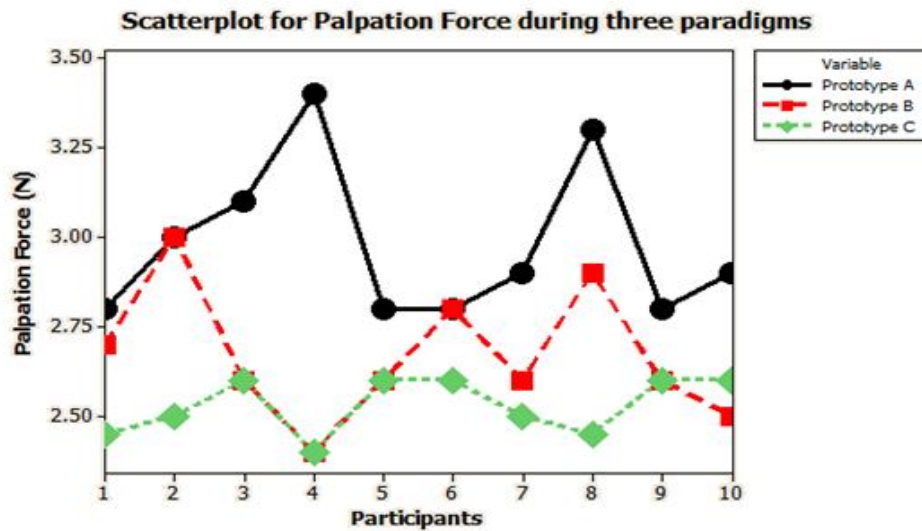


Figure 6.7: Individual palpation force by the trainees for the three prototypes-A, B, and C in a post-training phase

6.6.2.2 Exploration Time

In the reproductive tract of a cow, an ovary was set as a feature for identification discussed in the above section. The time taken from the initial position to the feature (finding an ovary) measured in seconds, and was termed as exploration time. The exploration time ends, once the participant identify the feature and starts palpating the feature. The exploration time was recorded for the three prototypes and was then analysed by an ANOVA using the MINITAB statistical software. The same hypothesis were set here as for palpation force. The hypotheses are as follows:

1. There is no significant difference in the exploration time using the three prototypes-A, B, and C, and
2. There is a significant difference in the exploration time using the three prototypes-A, B, and C.

The analysis of the data recorded using an ANOVA confirmed a significant difference among exploration time for the three training prototypes $F(2, 27) = 23.68$, $p < .001$, and hence rejected the first hypothesis that is there was no difference among the three prototypes. To find the best among the three prototypes, an independent sample t- test for the exploratory time, using MINITAB statistical software was performed between the

1. Prototype -A and prototype- B
2. Prototype- A and prototype-C
3. Prototype- B and prototype-C

There was a significant difference between prototype - A ($M = 43.40$, $SD = 8.32$) and prototype-B ($M = 29.90$, $SD = 3.90$), $t(12) = 4.65$, $p = .001$ (two-tailed test) for the exploration time. Similarly there is a very clear significant difference between the performance improvement using prototype - A ($M = 43.90$, $SD = 8.32$) and prototype - C ($M = 27.20$, $SD = 3.33$), $t(11) = 5.72$, $p < .001$. Similarly, there was no difference in the exploration time between prototype - B ($M = 29.90$, $SD = 3.90$) and prototype - C ($M = 27.20$, $SD = 3.33$), $t(17) = 1.67$, $p > .05$, ($p = .114$), (see Appendix B).

The range box and mean of the exploration time for prototype-A, B, and C is shown in Figure 6.8. The range box shows that prototype-C is better than prototype-A, and B.

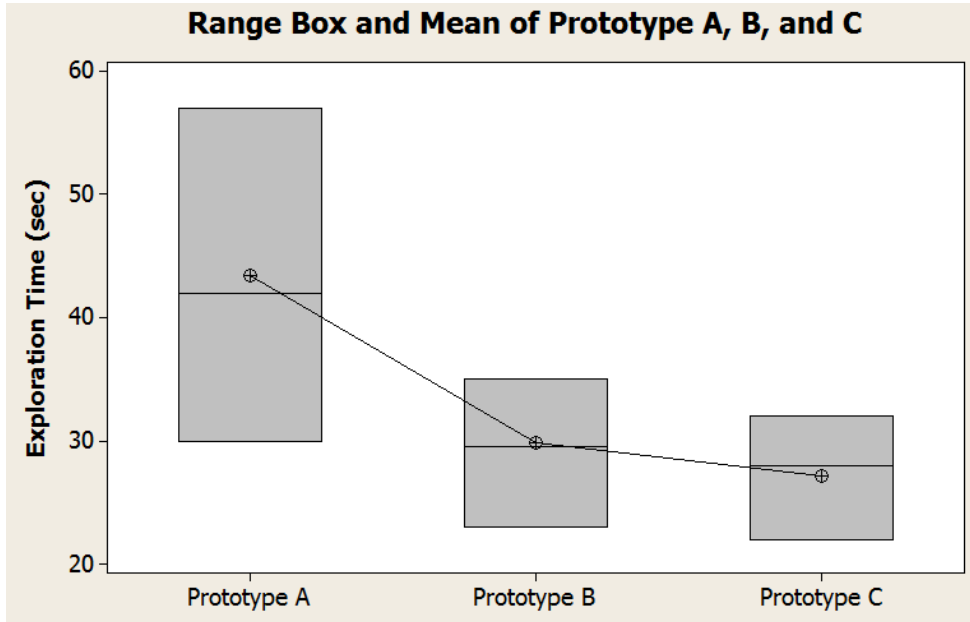


Figure 6.8: Range Box and Mean of exploration time for prototype-A, B, and C

This experiment showed that visio-haptic modality outperformed haptic only modality in learning skills enhancement in a haptic dominant palpation simulator for a short-term memory. This experiment also confirmed that prototype-C was better than prototype-A, and prototype-B, due to the realistic visual and haptic sensation.

6.6.3 Experiment 3

The statistical analysis results for the palpation force and exploration time for the long-term memory are presented in the following sections.

6.6.3.1 Palpation Force

The same hypothesis were set here as in an Experiment 2 discussed above. A one-way ANOVA confirmed a significant difference among palpation force for the three training prototypes $F(2, 27) = 14.56, p < .001$, and hence concluded that there was a significant difference among the three prototypes. Further, an independent sample t-test between prototype - A ($M = 3.01, SD = 0.223$) and prototype-B ($M = 2.81, SD = 0.137$), $t(14) = 2.34, p = .035$, (two-tailed test), shown a significant difference. Similarly there is a very clear significant difference between the performance improvement using prototype - A ($M = 3.01, SD = 0.223$) and prototype - C ($M = 2.61, SD = 0.0966$), $t(12) = 5.01, p < .001$. Similarly there was a significant difference of the palpation force between prototype - B ($M = 2.81, SD = 0.137$) and prototype - C ($M = 2.61, SD = 0.096$), $t(16) = 3.77, p = .002$, (see Appendix B).

6.6.3.2 Exploration Time

The exploration time after one week for the three prototypes were analysed statistically using one-way ANOVA and shown a significant difference among the three prototypes $F(2, 27) = 34.43, p < .001$.

There was a significant difference in the exploration time between prototype - A ($M = 69.50, SD = 15.5$) and prototype-B ($M = 34.10, SD = 6.67$), $t(12) = 6.62, p < .001$ for the exploration time. Similarly there was a very clear significant difference between the performance improvement using prototype - A ($M = 69.95, SD = 15.5$) and prototype - C ($M = 35.20, SD = 8.2$), $t(13) = 6.18, p < .001$. Similarly there was no difference in the exploration time between prototype - B ($M = 34.10, SD = 6.67$) and prototype - C ($M = 35.20, SD = 8.2$), $t(17) = -0.33, p > .05, (p = .746)$, (see Appendix B).

The range box and mean of the three prototypes is shown in Figure 6.9. This also shows the maximum and minimum exploration time for each prototype. Although the range box for prototype-B, and C is different, but this difference is statistically insignificant.

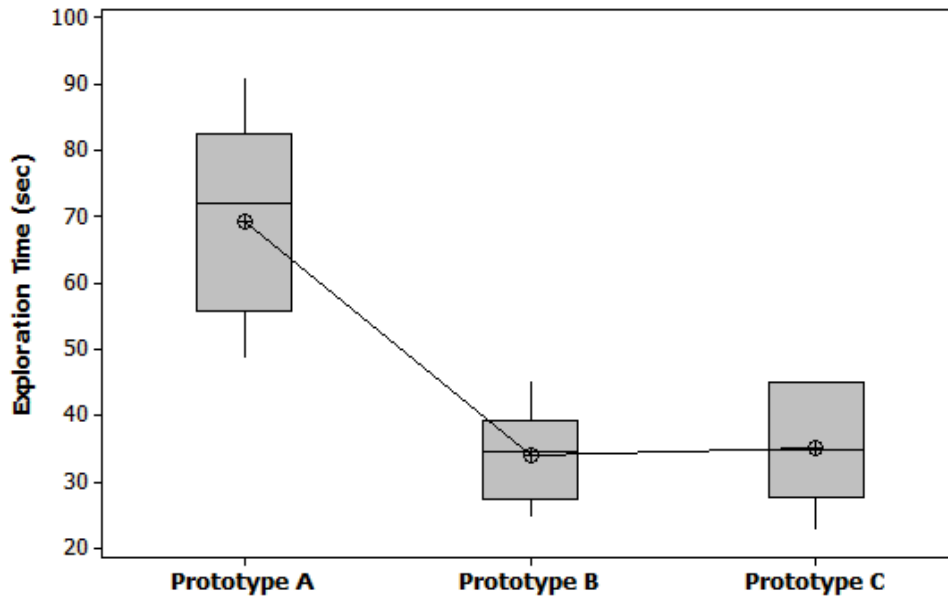


Figure 6.9: Range Box and mean of exploration time for the three prototypes

The exploration time was better using the two proposed visio-haptic deformable model for the enhancement of long-term learning skills as compared to prototype-A. These findings suggest that visio-haptic modality allowed the participants to better recall positional information in a long-term memory.

6.6.4 Experiment 4

The maximum value of the palpation forces recorded, once the feature was identified, for all the participants in the post-training phase were used for comparing the significance of the two prototypes having different extents of visual realism, and touch sensations. An ANOVA using MINITAB statistical software was used to analyse the significance of the data collected. The hypotheses are as follows:

1. There is no significant effect on learning palpation force using prototype-C1, and C2 which was having two different resolutions in the region of interest, and
2. There is a significant effect on learning palpation force using prototype-C1, and C2.

A one-way ANOVA using MINITAB statistical software confirmed there is no significant difference among palpation force for the two resolutions $F(1, 19) = 0.14$, $p > 0.05$, ($p = .714$), and hence could not reject the first hypothesis that there is no difference between the two prototypes. The two prototypes although different in the extent of visual cue in terms of computation but have no effect on the learning of the palpation force. Palpation force recorded for the participants using prototype-C1, and C2 is shown in Figure 6.10.

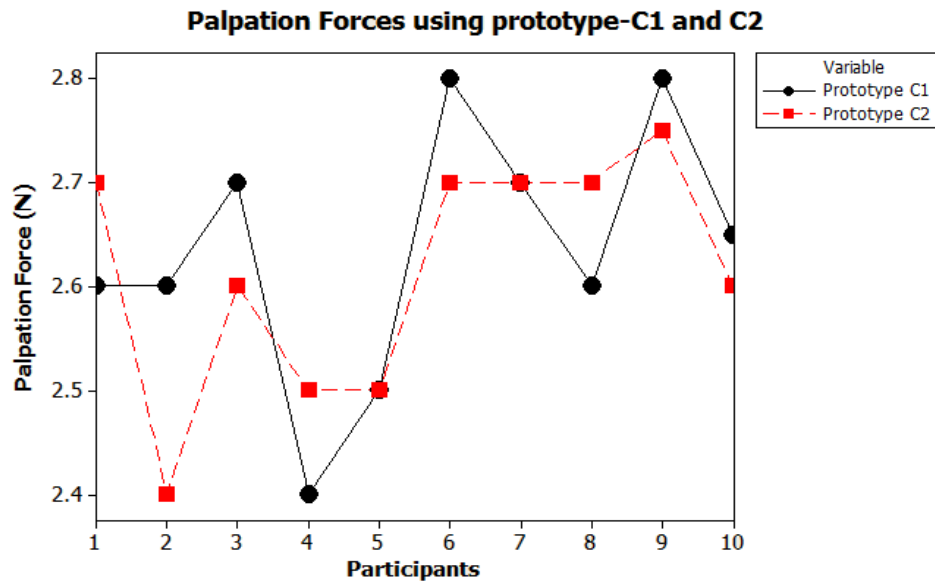


Figure 6.10: Palpation force recorded for the participants using prototype-C1, and C2

6.7 Summary

This chapter presented an evaluation study to validate the proposed visio-haptic deformable models as well as the multimodal cues. The first experiment showed that visio-haptic modality was better at learning a procedure than haptic under guidance. The visual cue allows instructors to better control the movements of the trainee during a procedure and guides better than the haptic only modality. In experiment two, the effect and extent of visual cue on the short-term memory for palpation force and exploration time were evaluated in a haptic dominant palpation simulator. The statistical analysis performed shows that prototype-B and C were better than

prototype - A, meaning that visio-haptic is better than haptic only modality for force and position recall. The experiment also shows the difference between prototype-B and C for palpation force only and there was no difference in exploration time. The third experiment was conducted for the evaluation of the effectiveness of multimodal cue for the enhancement of long-term learning skills. The study findings suggested that visio-haptic modality enhanced the force and position recall of the participants. In the fourth and last experiment, the effect and extent of visual and haptic cue were evaluated using different resolution of the proposed visio-haptic deformable model, and named as prototype-C1, and prototype-C2. The statistical analysis performed shown that there was no significant difference in the learning outcomes of the two models. This suggested that the extent of multimodal cues need to be considered in the development of such simulators.

The next chapter discusses the two visio-haptic deformable models presented in chapters four, and five, and the evaluation study presented in this chapter.

CHAPTER 7

DISCUSSION

This chapter presents the discussion of the two proposed visio-haptic deformable models and an investigation of multimodal cues in the haptic dominant palpation simulator. Section 7.1, discusses the two visio-haptic deformable models proposed in chapters four, and five respectively. Section 7.2 presents discussion of the evaluation study presented in chapter six. Section 7.3 summarizes the chapter.

7.1 Visio - Haptic Deformable Models

Real-time interactive deformation instead of sophisticated hardware has been in focus for the last two decades. For modelling the physical properties of the soft tissue mainly two methods: FEM [116; 110; 23] and MSM [134-136; 96; 97] have been widely used in which an MSM is the best choice because of its ease of implementation, fast response, and realistic deformation. For models of high resolution and the need of interactivity in the simulation including haptic sensation, makes these methods computationally expensive, when considering the whole model. Due to these reasons, a variation of MSM is used in this thesis in order to incorporate the physical properties of the soft tissue and achieve real-time interactive simulation of the soft tissue.

The following subsections discusses the two proposed visio-haptic deformable models, namely, a vertex based visio-haptic deformable, and an adaptive area based visio-haptic deformable.

7.1.1 Vertex Based Visio-Haptic Deformable Model

A vertex based visio-haptic deformable model was used to deform the touched node and its two rings of neighbours (two - neighbourhood) using the slope intercept form of a line equation. This model is very simple in its implementation and efficient in deforming the soft tissue upon applying the external force with the help of force feedback device. Deformation is faster because of fewer calculations required due to the slope intercept form of a line equation as compared to [61; 161]. This model deforms locally when interact in a real-time simulation, and is combined with the shape preserving springs in order to maintain the shape of the model during interactive deformation, once the external force is relaxed. For quickly accessing the touched node, an octree space partitioning technique is implemented in the haptic channel which is not available in the OpenHaptics APIs [144; 52]. Furthermore, a quick haptic rendering was achieved using only the two-neighbourhood of the touched node in the haptic channel. The overall performance of the visio-haptic deformable model is improved as compared to the existing methods [52; 60; 61].

7.1.1.1 Performance Analysis

A novel approach based on the slope intercept form of the line equation is used for deforming the touched node, and all the nodes in the close vicinity of this touched node. For visually realistic deformation, the touched node and its two-neighbourhood nodes were deformed using the same line equation with different slopes, termed as the step size. For achieving an elastic deformation, the shape preserving springs were introduced to restore the soft tissue, once the external force was relaxed.

For the models with high resolution, the interactive simulation degrades and the user could experience the lapses during simulation especially in the haptic sensation. This is because of the searching technique available in the OpenHaptics API for searching a node in the complex model [52; 144]. This searching is based on the nearest neighbour method, which chooses the node from the model closest to Haptic

Contact Point. The time complexity of this method is $O(n)$. In addition, as the model is rendered twice in the haptic and the graphics loop so rendering complex model twice in the real-time interactive simulation counted for the lapses as well. Firstly, to overcome these lapses during the simulation of soft tissue due to searching, an octree space partitioning method has been implemented in the haptic loop for quick access to the touched node. The OSP is not required to be implemented in the graphics loop, as the whole model is needed to be rendered for visual appearance. The time complexity of an octree method is $O(\log(n))$. The results (*see* Table 4.1) of the first proposed method for the same model of different resolutions showed 83% to 92% performance improvement when the octree partitioning method was implemented for searching the node. Secondly, to overcome the lapses during the simulation of soft tissue due to the complex, and whole model rendering, only two-neighbourhood nodes of the touched node were rendered for the haptic sensation. Therefore, when implementing two-neighbourhood rendering in the haptic channel, the simulator performance was improved from 6.31% to 87.44% (*see* Table 4.2) as compared to the whole model rendering. This improvement was in the case when both of the simulators have octree method implemented for the quick access of the touched node. In addition, when one simulator has implemented two-neighbourhood for rendering and octree method for searching and the other simulator has whole model rendering and linear search method for searching, then in such simulators, the results showed the performance improvement from 81.25% to 91.02% (*see* Table 4.2). The percentages are given in the ranges due to the reason that models of different resolution have been used in the simulation.

The results showed that the proposed vertex based visio-haptic deformable model, along with the octree space partitioning method and two-neighbourhood rendering has improved the performance up to 91.02% as compared to the currently available methods in the vertex based deformation for the interactive simulation of the model with the same resolution [61; 60]. The proposed model worked fine in a real-time interactive simulation for palpating the reproductive tract of the cow and the human stomach. In the proposed model, the model considered was hollowed from inside and the tissue properties considered were linear, isotropic, and homogenous. This proposed visio-haptic deformable model could be implemented in general for the soft

tissue, having hollow inside, such as vessels and gallbladder. However, the algorithm is unable to encompass the underlying physical properties of soft tissue, which are highly nonlinear, inhomogeneous, and anisotropic.

7.1.2 Adaptive Area Based Visio-Haptic Deformable Model

Subdivision is used as levels of detail in order to refine locally a deformable model in the region of interest. Many methods have been proposed in the literature, which combines levels of detail with the physics based deformable models for easing the computational load during simulation. The authors in [126] proposed an adaptively refined method for the rectangular shaped lattice of masses and springs. They modelled cloth and during simulation of the cloth, it was refined at the regions of high curvature. In this method only refinement at high curvature were provided but no simplification involved. The authors in [92] used loop subdivision to refine the region of interest in rendering of high deformation physics based facial animation system. Most of the existing soft deformation techniques that are combined with subdivisions for the levels of detail neglect to maintain dynamic behaviour of the model during simulation among different levels of details [125; 92; 126; 133]. The proposed adaptive area based visio-haptic deformable model, not only reduces the number of rendered triangles in the graphics, and haptic loop, but also takes care of the dynamic behaviour of the model during simulation.

The authors in [133; 125; 92] used adaptive subdivision for the surface mesh models. In this thesis, volumetric model is used to capture the internal details of the soft tissue.

7.1.2.1 Performance Analysis

The authors in [125] used an adaptive surface deformable model for hollow objects while maintaining dynamic behaviour of the model among different levels of detail. They used modified butterfly interpolation subdivision method and a physical property adjustment scheme in order to preserve the overall behaviour of the model

during real-time interactive simulation. During simulation the topology of the model changes due to the technique, applied for refinement.

In comparison with the previous models available in the literature, an adaptive area based visio-haptic deformable model enjoys several advantages. Firstly, in the proposed model, region of interest is refined based on the area of a triangle in the triangular mesh model. This allows that fewer triangles are produced and therefore, requires less time in graphics rendering. Comparing with the conventional adaptive subdivision model, the proposed model produced 2% less faces for the low resolution stomach model while 5% less faces for the higher resolution reproductive tract model (*see* Table 5.4) for the same visual appearance. This percentage increased with the increase in the variation of the area among triangles in the model.

Secondly, due to the proposed adaptive area based visio-haptic deformable model, least number of new nodes, spring stiffness, and damping coefficients are created. Table 5.1 shows that 10.98% lesser new nodes are created using the proposed method for the surface based mesh model. Similarly, 8.5% lesser number of spring stiffness and damping coefficients are required to be used in the simulation. Table 5.2 shows that 19.35% lesser new nodes are created using the proposed method for the tetrahedral based mesh model. Similarly, 19.09% lesser number of spring stiffness and damping coefficients are required to be used in the simulation. The least number of nodes reduced the time of its creation, and also ceased to create the corresponding springs. The time required in the computation of the spring forces, nodes position was also eliminated during the real-time interaction, while maintaining the same visual realism and haptic sensation. The performance was improved for the models with higher variation in the area of its faces.

In Figure 7.1, the comparison of the proposed adaptive area based visio-haptic deformable model with the conventional adaptive model for new node creation, along with the total nodes present in the coarse model. New nodes created in the proposed model are smaller than the conventional adaptive model.

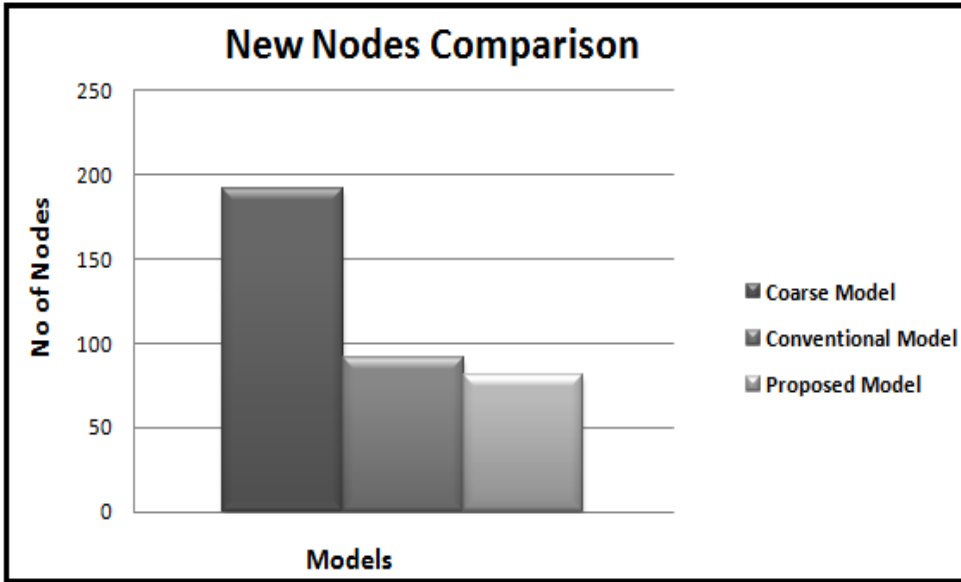


Figure 7.1: New nodes created during the proposed and conventional adaptive model with total nodes in the coarse model

Similarly, Figure 7.2 shows the comparison of the proposed area based visio-haptic deformable model with the conventional adaptive model for the creation of new springs, along with the total springs present in the coarse model. The proposed model creates a less number of springs that has also an advantage of eliminating the force computation for the spring stiffness and damping coefficients in the interactive simulation.

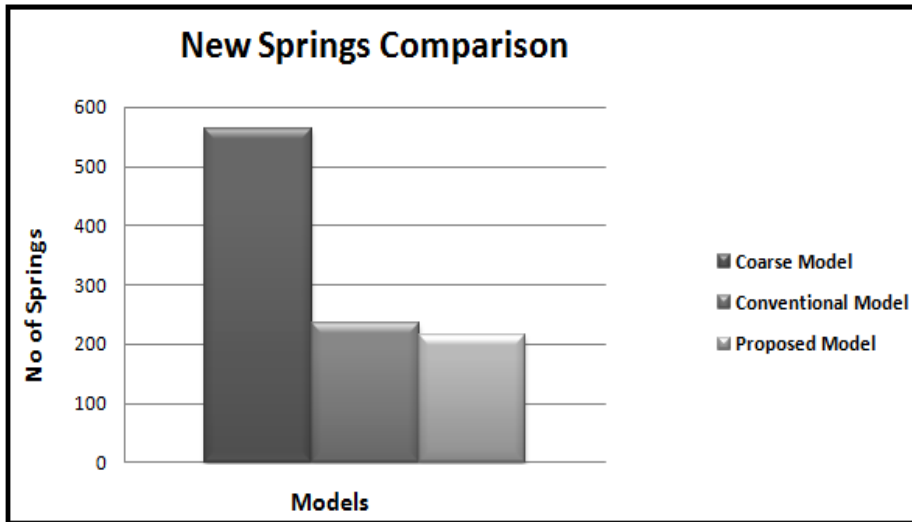


Figure 7.2: New springs created during proposed and conventional adaptive model with total springs in coarse model

The proposed model was compared with the refined model and the results were shown in Table 5.4. In the graphics loop the performance has been improved by 88.3% while in the haptic loop it was improved by 98.25% when rendering the stomach model and for the reproductive tract model, the performance in the graphics loop was improved by 99.48% and in the haptic loop the performance was improved by 95.12%.

Next, the proposed adaptive area based visio-haptic deformable model was compared with the conventional adaptive subdivision model proposed in the literature. In both of these algorithms, MSM is used to incorporate the physical properties of the soft tissue. The results shown in Table 5.5 showed that for the stomach model, the performance in the graphics loop was improved by 30.4%, while in the haptic loop it was improved by 94.28%. Similarly, for the reproductive tract model the performance in the graphic loop was improved by 54.19% and in the graphics loop it was improved by 97.13%.

The results presented showed that the proposed adaptive area based model gave a better performance in real-time visio-haptic interactive simulation of soft tissue deformation. As the resolution of the model increases, there is very little effect on the performance as compared to the refined and conventional adaptive models. In Figure 7.3, the execution time required in the graphics, and haptic loop using the stomach model is separately compared among the proposed adaptive area based model, the refined model, and the conventional adaptive model.

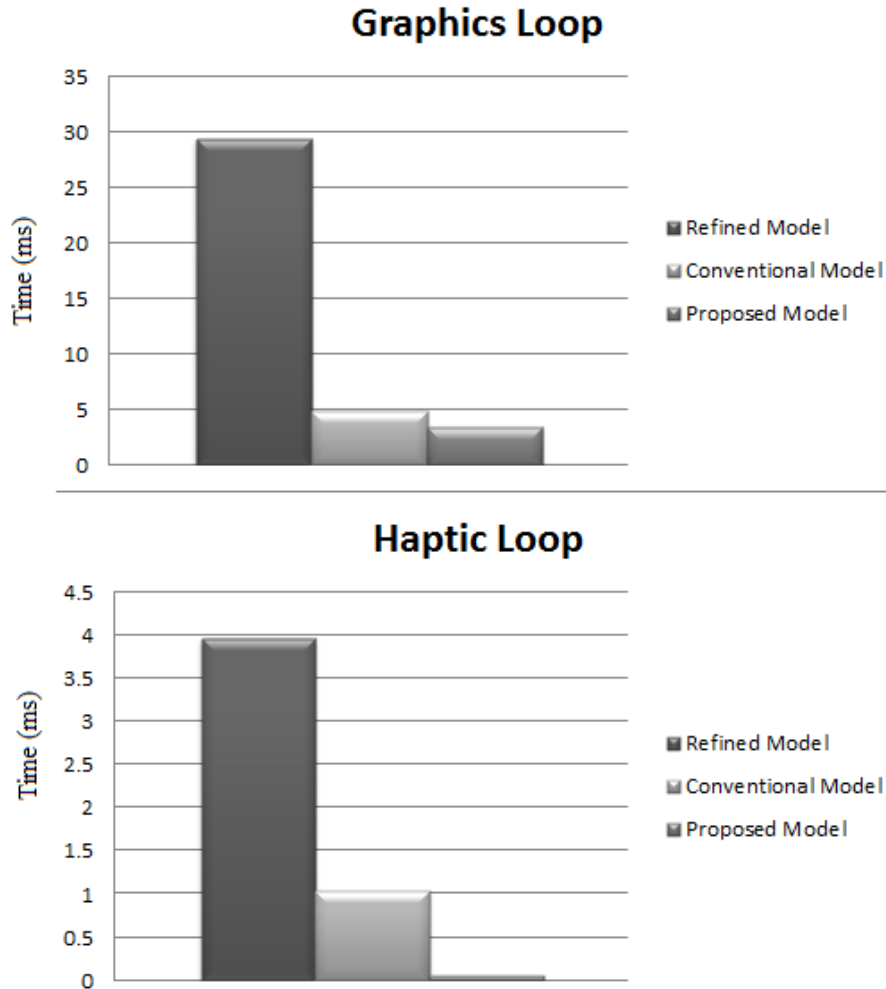


Figure 7.3: Execution time comparison of different models for stomach in the graphic and haptic loop

Similarly, in Figure 7.4, the execution time required in the graphics, and haptic loop using the reproductive tract model is separately compared among the proposed model, the refined model, and the conventional adaptive model.

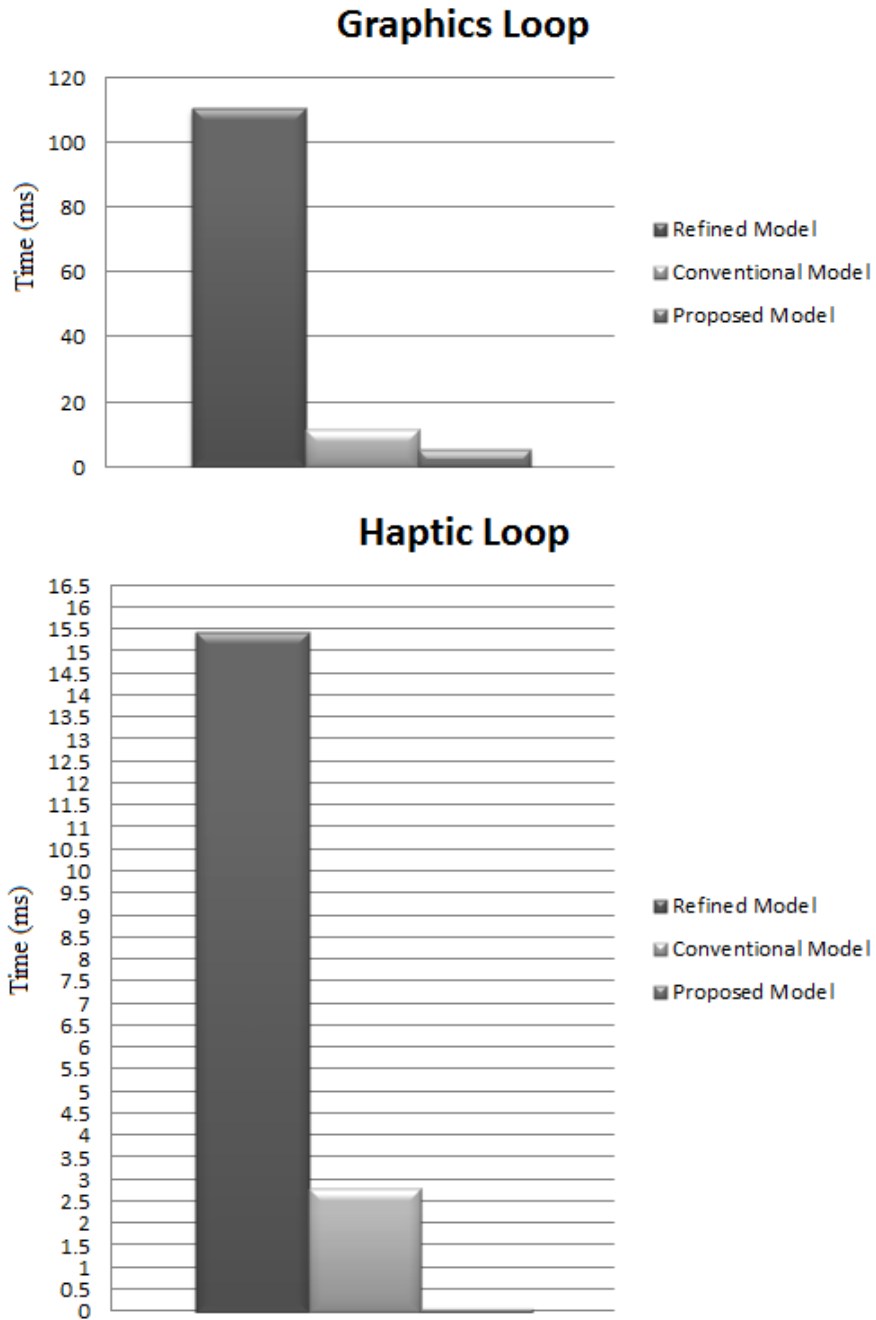


Figure 7.4: Execution time comparison of different models for reproductive tract in the graphic and haptic loop

In Figure 7.4, the coarse model has higher resolution. For the refined model and conventional adaptive deformable model, the execution time increases exponentially while for the proposed model, the execution time is negligible with the increase in model resolution.

7.2 Evaluation Study Analysis

Evaluation of the visio-haptic modality as well as the extent of visual and haptic cues in a haptic dominant environment was presented in chapter six. The evaluation study is meant to meet the last two research objectives stated in chapter one. Four experiments were involved in the evaluation study.

7.2.1 Visio-Haptic Modality in Collaborative Environment

The first experiment in chapter six shows the effectiveness of the visio-haptic modality in a collaborative environment of trainee and instructor. The instructor provided timely information to the trainee due to the interactive visual deformation. Both in the haptic only and visio-haptic environment, the trainees used only the haptic cue for practicing the palpation forces, but in a visio-haptic modality, the visual cue allowed the instructor to better control the palpation forces applied by the trainee during the skills learning process in a haptic dominant environment as compared to the haptic only [30; 28]. This experiment showed that visio-haptic modality allowed to improve the guidance of the participants and better controlled the participants' action during the training and hence enhanced the learning skills of the trainee. There was also a statistically significant difference between the performance of the two proposed visio-haptic deformable models due to implementation of realistic visual and haptic cues and its coordination. In prototype-B, the haptic sensation was similar to that of prototype-A, but with the aid of visual cue, there was statistically significant difference between the performance of prototype-A, and B. The statistical analysis of the data signified the importance of visual cue in a haptic dominant environment. Similarly, prototype-C was better than prototype-A, and B, due to realistic visual and haptic sensation and proper coordination of these cues.

7.2.2 Visio-Haptic Modality in Short-term and Long-term Skill Enhancement

Multisensory modality is more informative and helps quickly learn different shapes or procedure as compared to single modality [8]. However, the study findings of Srimathveeravalli and Thenkurussi [11] stated that visio-haptic is insignificant for the positional information recall. In a similar findings, the authors in [12] reported that visio-haptic training is not effective but showed promising results in long-term motor skills. Huegel and O'Malley, [13] showed that visio-haptic is effective in the beginning of the learning curve only. The second and third experiment in chapter six was conducted to investigate the effect of visio-haptic modality on the enhancement of short-term and long-term skills in a haptic dominant environment.

In the second experiment, the effect of visio-haptic modality was studied in a haptic dominant environment for short-term memory. The results showed that the learning skills were enhanced using visio-haptic modality than the haptic only in a haptic dominant environment. The visio-haptic modality was better in learning palpation force and had better exploration time. The extent of visual cue considered in this experiment by implementing the two proposed visio-haptic deformable models, showed a significant difference in the palpation force but there was no difference in the exploration time. There was also a statistically significant difference between the performance of the two proposed visio-haptic deformable models due to the implementation of realistic visual and haptic cues and its coordination. In prototype-B, the haptic sensation was similar to that of prototype-A, but with the aid of visual cue, there was statistically significant difference between the performance of prototype-A, and B and enhanced the skills learning of the trainee. Similarly, prototype-C, which has the visio-haptic modality, (but these modalities were realistic and based on the physical properties of the soft tissue), showed better performance in learning skill enhancement than prototype - A, and B. In short, the visio-haptic modality enhanced skill learning for better force as well as position recall as compared to haptic only. In addition, the haptic sensation based on the physical properties of the soft tissue was better than the haptic sensation based on the Hooke's law implemented in the currently available bovine rectal palpation simulator [30; 31; 28].

In the third experiment, the effect of visio-haptic modality was studied in a haptic dominant environment for long-term memory. The results in this case also had shown better performance for visio-haptic modality. The participants trained using the proposed visio-haptic deformable models remembered well the palpation force and exploring the feature set during the experiment. The exploration time for the current haptic only modality were much higher for all the participants trained, almost 50% higher than the visio-haptic modality meaning that participants took longer time to explore and reach to the feature set during the experiment. Similarly, the mean palpation force for the haptic only was much higher than the threshold force set as compared to the two proposed visio-haptic deformable models. There was also a statistically significant difference between the two proposed visio-haptic deformable models for the palpation force but the difference in the exploration time was statistically not significant. In short, visio-haptic modality was better in long-term force as well as position recall as compared to haptic only, and hence enhanced the learning skills of the trainee.

The results of experiment two, and experiment three reveals that visio-haptic modality enhanced the learning skills of the trainee as compared to the haptic only modality in a haptic dominant palpation simulators. In addition, the main difference between the two proposed visio-haptic deformable models is the extent of realism that they have in terms of visual and haptic sensation and this resulted in a difference in the performance of the participants.

7.2.3 Visual and Haptic Modality

The statistical results obtained in all experiments have shown that visio-haptic modality enhanced learning skills of the trainees as compared to the haptic only modality. Experiments were also conducted to show the effect of visual and haptic modality on the performance.

In this research, prototype-B has a realistic interactive visual deformation and that haptic sensation was based on Hooke's Law as reported in [30; 31], while in prototype-C, the visual deformation was realistic as well as the haptic sensation which

was based on the reaction forces of the tetrahedral lattice of the mass spring model. The computed forces were provided to the participant through the force feedback device in a realistic way. The statistical results of prototype-B and C have shown a significant difference in learning skills enhancement, and in addition prototype-C was better than prototype-B, although both have the visio-haptic modality. This difference in performance was due to the extent of haptic cue present in prototype-C. The haptic sensation in prototype-C was more realistic than prototype-B, and therefore helped better in learning the palpation forces. The individual palpation force applied by participants using the three prototypes-A, B, and C for short-term learning skills are shown in Figure 7.5. The green square represents the palpation force applied by the participants using prototype-C, and shows that the palpation forces are close to the Maximum Palpation Force vicinity as compared to prototype-A, and B, represented by black circles and red squares respectively.

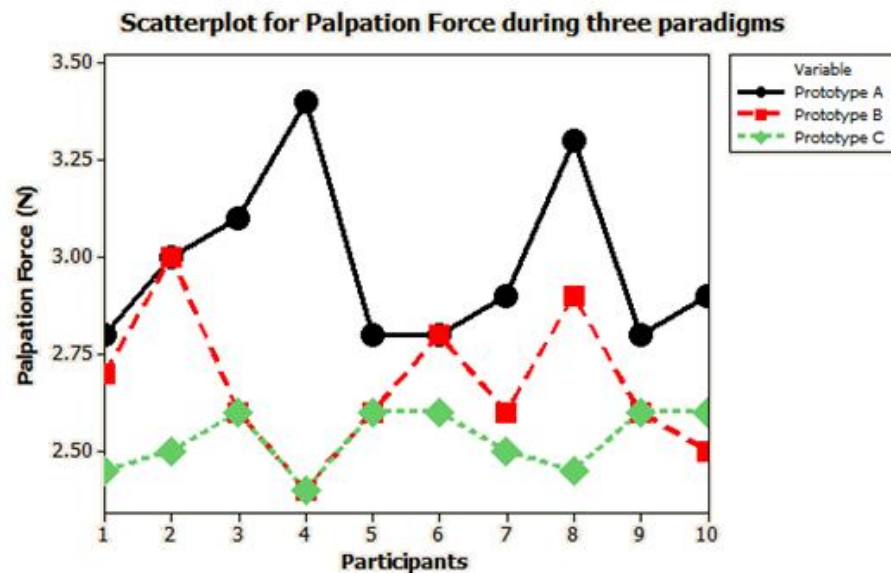


Figure 7.5: An individual palpation force by the participants using the three prototypes-A, B, and C for short-term force recall

The individual palpation forces applied by participants using the three prototypes-A, B, and C for long-term force recall are shown in Figure 7.6. In Figure 7.6, it is clear that palpation forces using prototype-C are closer to the Maximum Palpation Force (MFP) as compared to prototype-A, and B.

Scatterplot of the three prototype-A, B, and C for long-term haptic recall

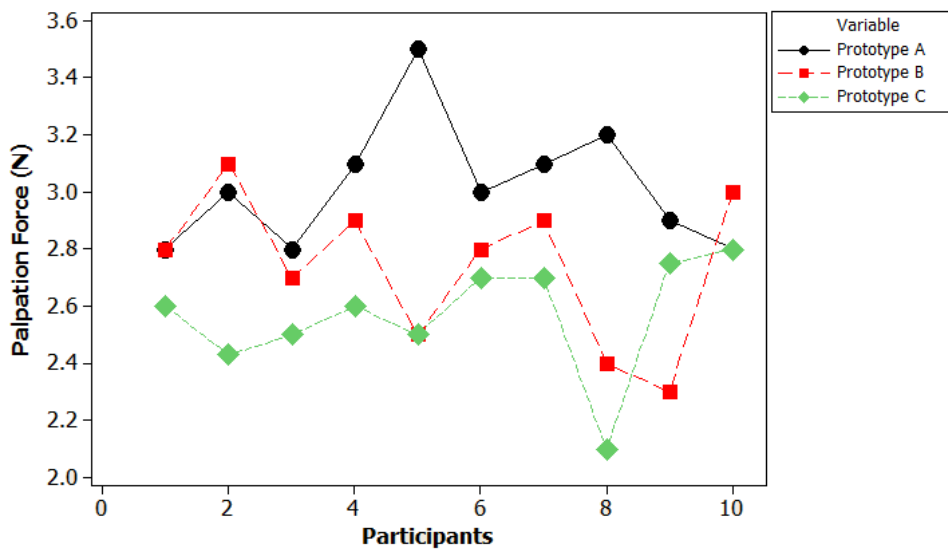


Figure 7.6: An individual palpation force by the participants using prototypes-A, B, and C for long-term memory force recall

In the fourth experiment, prototype-C was used with two different resolutions in the region of interest, in which the resolution of one prototype (prototype-C1) was double than the other (prototype-C2). The aim of this experiment was to evaluate the effect of extent of visual and haptic cues. The statistical analysis of the results obtained showed that the palpation forces recorded in both the simulators were insignificant and hence both trained the participants with the same degree of performance. The statistical analysis of this experiment revealed the importance of the extent of visual and haptic cue for developing visio-haptic simulator that should be considered because beyond the threshold perception, computation is counterproductive and wastage of resources [27]. The results of this experiment suggest the extent of visio-haptic cue is important and need consideration in the development of such simulators but does not quantify the extent of each cue to be present. Figure 7.7, shows the visio-haptic realism versus skill enhancement. Increasing the visual realism and haptic sensation, increases the learning skills. The vertical red line shows the maximum learning skill outcomes at the particular visio-haptic realism. Further increase in visio-haptic realism has no effect on learning

outcome; the computation is counterproductive and hence the wastage of resources. This is shown in Figure 7.7, with a red circle region of the graph. Beyond this region, increasing visio-haptic realism degrades the learning skills outcome, due to the lapses in the visual and haptic sensation.

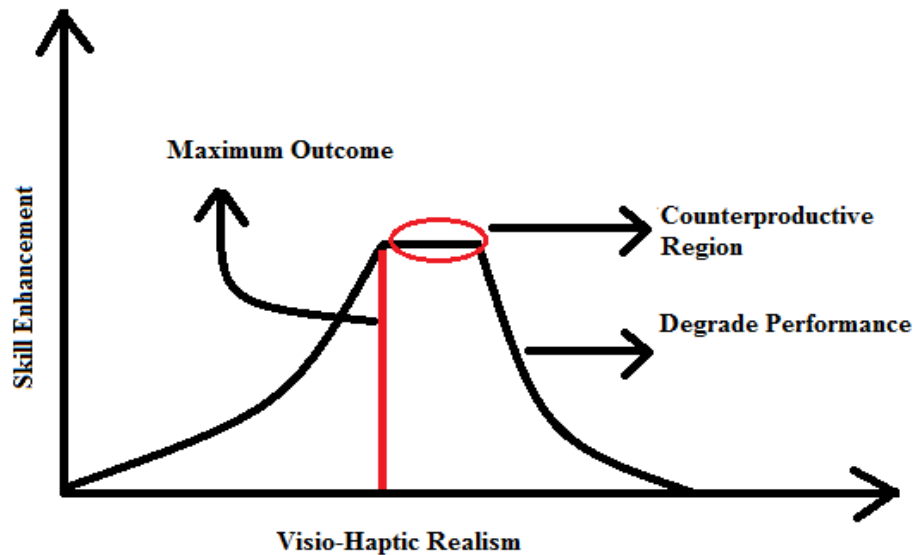


Figure 7.7: Effect of the extent of visual and haptic cues on learning skills enhancement

The statistical results obtained in experiments one, two, and three have shown that visio-haptic modality enhanced learning skills of the trainees as compared to the haptic only modality in a haptic dominant environment. In addition the results have shown the effect of extent of visual and haptic modality on the enhancement of learning skills. The statistical results of experiment four revealed the importance of the extent of visual and haptic cue for developing visio-haptic simulator that should be considered because beyond the threshold perception, computation is counterproductive and wastage of important resources. These results suggest the extent of visio-haptic cue is important and need consideration in the development of such simulators, but these results does not quantify the extent of these cues to be present in the simulators.

The experiments conducted shows that visio-haptic cues enhanced learning skills of the trainees as compared to haptic only. The proposed visio-haptic deformable

models were implemented for the visio-haptic cues, which also validate the proposed visio-haptic deformable models. The difference between the learning outcomes of the two proposed models were due to the implementation of surface based model in one prototype and volumetric based model in the second prototype. As the proposed visio-haptic deformable models enhanced the learning skills of the trainees and hence show the effectiveness of the proposed models. This is shown in Figure 7.8. The range box for short-term and long-term memory for palpation force using prototype-A, B, and C is shown in Figure 7.8.

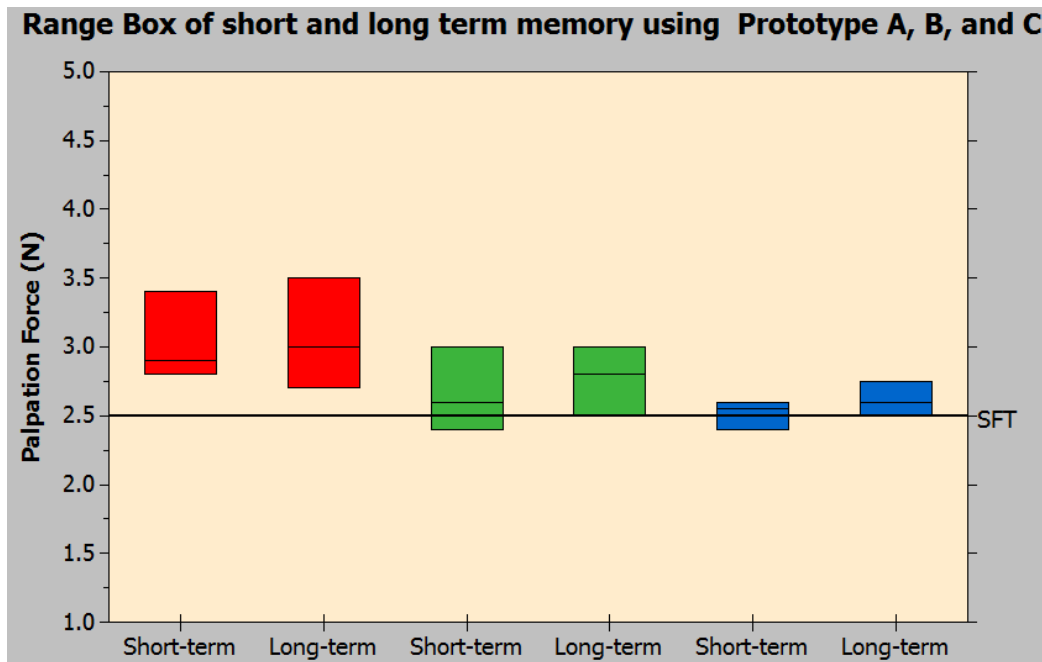


Figure 7.8: Range Box for short-term, and long-term for palpation force using prototype-A, B, and C

Red range box is the palpation forces applied by the trainees using prototype-A, green range box is the palpation forces applied by the trainees using prototype-B, and blue is for prototype-C. The horizontal line represents the Stopping Force Threshold.

7.3 Summary

This chapter discussed the results presented in chapters four, five, and six. In summary, the results from the evaluation study showed that the two proposed visio-

haptic deformable models were efficient in terms of speed and suitable for a real-time interactive simulation.

A vertex based visio-haptic deformable model using the slope intercept form of a line equation for soft tissue simulation, in combination with an octree space partitioning method, and two-neighbourhood method for local deformation has greatly improved the performance. Area based adaptive visio-haptic deformable model combined with an octree space partitioning and MSM for physical properties showed an improvement over the models available in the literature. The results showed that the proposed model gave better performance when compared with the refined model during interactive simulation. Similarly, the proposed model was compared with an adaptive deformable model available in the literature and the results shown improvement in both the graphics and haptic loop. It was also shown that the results could be further improved if the variation in the triangle area is high. It was shown that increasing the resolution of the coarse model for adaptive refinement, the increase in the execution time for the proposed model in the graphics and haptic loop is negligible while this was exponentially increased for the refined model as well as for the conventional adaptive model.

The experiments show that visio-haptic modality is better at learning a procedure than haptic only in the haptic dominated environment. The visual cue was for the instructor to guide the trainee in an interactive environment, providing useful palpation force control to the trainees. This showed that in an instructor-trainee interaction environment visio-haptic modality was more informative than the haptic only modality, and enhanced the learning skills of the trainee. The two experiments for short-term and long-term memory have shown that visio-haptic modality was better than the haptic only. In prototype-B, and C, although visual realism was the same but the haptic sensation was different and showed a difference in the performance. Similarly, findings from the last experiment concluded that the extent of visual cue has an importance in a simulation and need to be considered, as beyond the users' perception of realism, the computation is counterproductive and have no such benefits.

In the next chapter, the conclusion of this thesis is presented, with the limitation, and future work directions from this work.

CHAPTER 8

CONCLUSION AND FUTURE WORK

The aim of this chapter is to conclude the findings of this research work by relating them to the objectives set in the first chapter of this thesis. This chapter also presents future research directions as well as the limitations of the work.

8.1 Deformable Modelling Techniques

In this thesis, two visio-haptic deformable models have been proposed for real-time interactive visio-haptic deformation as presented in chapters four, and five, respectively. Deformable models developed so far mostly are highly application dependent. Therefore, during the proposed work in the thesis, the focus is on real-time visio-haptic deformation in bovine rectal palpation. The two visio-haptic deformable models, developed and implemented are namely vertex based visio-haptic deformable model using slope intercept form of a line equation, and adaptive area based visio-haptic deformable model.

In a vertex based visio-haptic deformable model, deformation occurs along the force vector using slope intercept form of a line equation. This deformation is unrealistic; as the deformed node does not come to the rest position once, the external force has been removed. To make the deformation elastic, shape preservation springs were used to bring it back to the original position once the external force is relaxed. An OSP is used to partition the space occupied by the model and implemented in the haptic loop in order to make the collision of the force feedback device with nodes faster.

The next proposed model is an adaptive area based visio-haptic deformable model, which divides the region of contact dynamically based on the area of a

triangle. This technique is implemented along with the mass spring model for incorporating physical properties of soft tissue. This algorithm has greatly removed unnecessary triangles, which greatly improved performance of graphic and haptic rendering. In addition, it avoids extra spring computation and integration. Furthermore, it also reduces new masses, spring stiffness, and damping coefficient calculations because of selected triangle subdivision. This does not affect the visual appearance of the model during simulation.

In order to get a quick response in the haptic channel, an octree space partitioning method is implemented. An octree space partitioning is also used for fast rendering of selected nodes in the haptic channel. It has been shown that using an octree for rendering and collision detection has greatly improved performance. Furthermore, a one-neighbourhood, and two-neighbourhood are used in the proposed model in the region of interest for quick and efficient deformation.

8.2 Evaluation Study

Four experiments were conducted in order to validate that visio-haptic modality enhanced short-term and long-term learning skills as compared to the haptics only in a haptic dominant palpation simulator as well as to evaluate the extent of visual and haptic cues. The experiments also validate the visio-haptic deformable models proposed in this work.

The first experiment is an evaluation of the effect of visual cue in an instructor-trainee environment with interactive guidance of the instructor for controlling the palpation forces. Three types of simulators were used in which one containing haptic cue and the other two containing visio-haptic cues using the two visio-haptic deformable models presented in chapters four and five, respectively. The finding of this study reported that visual cue enhanced the skill learning process due to the interactive involvement of the instructor during palpation. The results of the two proposed visio-haptic models were also significantly different meant that increasing the haptic sensation improved the performance but up to certain threshold based on

the users' perception, as beyond the threshold perception, the visio-haptic modality has no effect on the learning skills enhancement.

The second experiment was performed to evaluate the effectiveness of visio-haptic modality in a haptic dominant environment for the enhancement of short-term learning skills. The same three prototypes were used as in experiment one, but the findings concerned only one subject. The report of this experiment revealed the effectiveness of the proposed model as well as the potency of visio-haptic cues for the enhancement of learning skills in a haptic dominant environment for short-term memory. The parameters used in this experiment were palpation force and exploration time. The study findings from this experiment showed that visio-haptic modality was better in learning skills as compared to haptic only in a haptic dominant environment.

The third experiment was performed to see the effectiveness of the visio-haptic modality for the enhancement of long-term learning skills. The study findings showed that visio-haptic modality as compared to the haptic only, enhanced the learning skills of the participants.

The fourth experiment was performed to see the extent of visual and haptic cues needed for the learning skills enhancement. Prototype-C was implemented using two different resolutions in the region of interest in which one resolution was twice the other. The results showed no significant difference statistically that advocates the extent of the visual and haptic cues need to be considered in order to improve the performance of the simulator.

The empirical findings statistically analysed, reported that visio-haptic modality was better in learning skills enhancement than the haptic only modality even though in a haptic dominant environment. The effect was on the instructor guiding the trainee as well as in the short-term and long-term memory of the participants for force and position recall. The visual cue both in prototype-B, and prototype-C enhanced the learning skills of the trainees and was better in the recall of force and positional information than the trainees, which were trained with the haptic only modality.

8.3 Contribution

The main contribution of this thesis could be described in the form of practical and substantive contributions. The practical contribution consists of the actual development, implementations of the two visio-haptic deformable models, namely vertex based visio-haptic deformable model, and an adaptive area based visio-haptic deformable model. Substantive contribution consists of the empirical findings of the effectiveness of visio-haptic modality in the haptic dominant palpation environment, the effect of the extent of the visual cue in the skill development process.

8.3.1 Practical Contribution

The practical contribution addresses the first two objectives of this thesis. The first objective is the development of visio-haptic deformable models for interactive simulation of visio-haptic modalities, while the second objective concerns with the implementation of visio-haptic deformable models developed in the haptic dominated environment.

The main practical contributions can be summarized as:

- Development of a visio-haptic deformable model, namely, a vertex based visio-haptic deformable model using slope intercept form of a line equation. The model is combined with the space partitioning techniques and two-neighbourhoods, for further realistic visual and haptic deformation.
- Development of a second visio-haptic deformable model, namely, an adaptive area based visio-haptic deformable model. In this model, the region of interest is refined based on the area of a triangle. If the triangle's area is greater than the threshold area then the region is refined. For switching between the physical properties of the model before and after refinement, new formulae for calculating mass of the new nodes, spring stiffness, and damping coefficients are presented.

8.3.2 Substantive Contribution

A substantive contribution addresses the last two objectives of this thesis mentioned in chapter one, concerning the evaluation of the visio-haptic cues and the extent of visual and haptic cue for learning skills enhancement in a haptic dominant palpation simulator. This has been presented in the form of an evaluation study reporting four experiments in chapter six. The study findings suggest that:

- The interactive visual deformation in a collaborative trainee-instructor environment used by the instructor, enhanced the learning skills of the trainee, and help better control the palpation forces during palpation procedure.
- Visio-haptic modality as compared to haptic only, enhanced short-term and long-term learning skills in a haptic dominant environment.
- The extent of visual and haptic cues is important in a palpation simulator and that the proportion of these two cues in the simulator should be based on the users' perception.
- Prototype-B, and prototype-C, both enhanced short-term and long-term learning skills as compared to prototype-A, and therefore, validates the two visio-haptic deformable model proposed in this thesis.

8.4 Recommendations for Future Work

In this section, recommendations for future work, out of the work presented in this thesis are discussed. Here are few possible goals for research in this field.

- The visio-haptic deformable models proposed in this thesis are used for local deformation. Further work needs to be done to expand it to global deformation for further mimicking the gap between real and virtual.
- In an adaptive area based visio-haptic deformable model, all the refinement levels and its corresponding parameters are calculated in the preprocessing step before actually the simulation starts. Further work is required to make it

dynamic in order to improve the space.

- During a user interaction with the simulation, when the user deforms the soft tissue, then the triangles in the region of interest enlarges due to the movement of its nodes. Such triangles should also be refined dynamically for realistic simulation using a threshold area for its refinement.
- Cache memory can be used to store the most recently visited triangles so that haptic rendering pipeline can be further improved.
- The visio-haptic deformable models developed need to be implemented in a bovine rectal palpation simulator as a whole and needs further exploration based on experts in the field of veterinary. Further evaluation based on having multi-layers collision and force interaction is required.
- Further assessment of the extent of visual cue and haptic cue is required by changing the latency time between visual and haptic modality, and need to find an acceptable extent for the user.

PUBLICATIONS

JOURNAL PUBLICATIONS

- [1] Area Based Adaptive Subdivision of a Triangular Mesh for Visio-Haptic Deformation, (I. Ahmad, and Suziah Sulaiman), Information Technology Journal, (2012), Vol 11, and Issue 1, pp. 171 – 175, (SCOPUS, Index).
- [2] Irshad Ahmad, Suziah Bt. Sulaiman, and Dayang Rohaya Bt Awang Rambli, 2013. Optimized Real Time Vertex Based Deformation Using Octree and Two Neighborhood Method. Research Journal of Applied Sciences, Engineering and Technology, 5(17): 4240-4245, (SCOPUS, ISI, Index).

CONFERENCE PUBLICATIONS

- [1] Real-time Vertex Based Deformation in Training Simulator, (I. Ahmad and S.B. Sulaiman), International Symposium on Ubiquitous Virtual Reality (ISUVR 2011), vol., no., pp. 21-24, 1-3 July 2011 doi: 10.1109/ISUVR.2011.16. (IEEE).
- [2] Evaluation of Real-Time Visio-Haptic Deformable Bovine Rectal Palpation Simulator, (I. Ahmad and S. Sulaiman), In Information Technology (ITSim2010), International Symposium in, 2010, pp. 1 - 4. (IEEE).

REFERENCES

- [1]. Vidal F. P., Bello F., Brodrie K. W., John N. W., Gould D., Phillips R., and Avis N. J. 2006. Principles and Applications of Computer Graphics in Medicine. *Computer Graphics Forum* 25(1):113-137.
- [2]. Liu A., Tendick F., Cleary K., and Kaufmann C. 2003. A survey of surgical simulation: Applications, Technology, and Education. *Presence: Teleoper Virtual Environ* 12(6):599-614.
- [3]. Gallagher A. G., Ritter E. M., Champion H., Higgins G., Fried M. P., Moses G., Smith C. D., and Satava R. M. 2005. Virtual reality simulation for the operating room: proficiency-based training as a paradigm shift in surgical skills training. *Annals of Surgery* 241(2):364-372.
- [4]. Timothy R. C., Dwight M., and Nigel W. J. 2011. The Role of Haptics in Medical Training Simulators: A Survey of the State of the Art. *IEEE Transactions on Haptics* 4:51-66.
- [5]. Palpation. 2007. *Saunders Comprehensive Veterinary Dictionary*.
- [6]. Penny C. D. 2002. Education A University View. *Cattle Practice* 10:255-256.
- [7]. Clark H., and Brennan S. 1991. Grounding in Communication. *Cognition, Perspectives on Socially Shared: A.P. Association*. 1991, Washington D.C: In L. Resnick, J. Levine, & S. Teasley (Eds.).
- [8]. Fredembach B., de Boisferon A. H., and Gentaz E. 2009. Learning of Arbitrary Association between Visual and Auditory Novel Stimuli in Adults: The “Bond Effect” of Haptic Exploration. *PLoS ONE* 4(3):e4844.
- [9]. Kalenine S., Pinet L., and Gentaz E. 2011. The visual and visuo-haptic exploration of geometrical shapes increases their recognition in preschoolers. *International Journal of Behavioral Development* 35(1):18-26.

- [10]. Morris D., Tan H. Z., Barbagli F., Chang T., and Salisbury K. 2007. Haptic Feedback Enhances Force Skill Learning. in Proceedings of the 2007 World Haptics Conference:21-26.
- [11]. Srimathveeravalli G., and Thenkurussi K. 2005. Motor skill training assistance using haptic attributes. Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems: IEEE Computer Society. p 452-457.
- [12]. Xing-Dong Y., Bischof W. F., and Boulanger P. 2008. Validating the Performance of Haptic Motor Skill Training. symposium on Haptic interfaces for virtual environment and teleoperator systems. p 129-135.
- [13]. Huegel J. C., and O'Malley M. K. 2010. Progressive haptic and visual guidance for training in a virtual dynamic task. Haptics Symposium, 2010 IEEE. p 343-350.
- [14]. Cover S. A., Ezquerro N. F., O'Brien J. F., Rowe R., Gadacz T., and Palm E. 1993. Interactively deformable models for surgery simulation. Computer Graphics and Applications, IEEE 13(6):68-75.
- [15]. Hui C., Hanqiu S., and Xiaogang J. 2006. Interactive haptic deformation of dynamic soft objects. Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications. Hong Kong, China: ACM. p 255-261.
- [16]. Gibson S., and Mirtich B. 1997. A Survey Of Deformable Modeling In Computer Graphics. Technical Report NoTr-97-19, Cambridge, Ma: Mitsubishi Electric Research Lab.
- [17]. Baran B., and Basdogan C. 2010. Force-Based Calibration Of A Particle System For Realistic Simulation Of Nonlinear And Viscoelastic Soft Tissue Behavior. Eurohaptics 2010, Lncs 6191, Part 1 6191:23 -28.
- [18]. Natsupakpong S., and Cenk M. C. 2010. Determination Of Elasticity Parameters In Lumped Element (Mass-Spring) Models Of Deformable Objects. Graphical Models 72:61-73.

- [19]. Terzopoulos D., and Waters K. 1990. Physically-based facial modelling, analysis, and animation. *The Journal of Visualization and Computer Animation* 1(2):73-80.
- [20]. Choi K.-S. 2006. Interactive cutting of deformable objects using force propagation approach and digital design analogy. *Computers & Graphics* 30(2):233-243.
- [21]. Lian L. L., and Chen Y. H. 2006. Haptic Surgical Simulation: An Application To Virtual Suture. *Computer-Aided Design & Applications* 3:203-210.
- [22]. Mosegaard J. 2004. LR-Spring Mass model for cardiac surgical simulation. *Stud Health Technol Inform* 98:256-258.
- [23]. Taylor Z. A., Cheng M., and Ourselin S. 2008. High-Speed Nonlinear Finite Element Analysis For Surgical Simulation Using Graphics Processing Units. *IEEE Transactions On Medical Imaging* 27(5):650-663.
- [24]. Hauser K. K., Shen C., and O'brien J. F. 2003. Interactive Deformation Using Modal Analysis With Constraints. In *Graphics Interface*. A K Peters, Cips, Canadian Human-Computer Commnication Society: 247–256.
- [25]. Peterlik I., Sedef M., Basdogan C., and Matyska L. 2010. Real-time visio-haptic interaction with static soft tissue models having geometric and material nonlinearity. *Computers & Graphics* 34(1):43-54.
- [26]. Dogan F., and Celebi M. S. 2011. Real-Time Deformation Simulation Of Non-Linear Viscoelastic Soft Tissues. *Simulation* 87(3):179-187.
- [27]. Batteau L., Liu A., Maintz J., Bhasin Y., and Bowyer M. 2004. A Study on the Perception of Haptics in Surgical Simulation In: Cotin S, and Metaxas D, editors. *Medical Simulation*: Springer Berlin / Heidelberg. p 185-192.
- [28]. Crossan A., Brewster S., Reid S., and Mellor D. 2001. A Horse Ovary Palpation Simulator for Veterinary Training. In: *Interaction IHH-C*, editor: Springer Berlin / Heidelberg. p 157-164.

- [29]. Coles T., John N. W., Gould D. A., and Caldwell D. G. 2009. Haptic Palpation for the Femoral Pulse in Virtual Interventional Radiology. *Advances in Computer-Human Interactions, 2009 ACHI '09 Second International Conferences on.* p 193-198.
- [30]. Baillie S., Crossan A., Brewster S., Mellor D., and Reid S. 2005. Validation of a bovine rectal palpation simulator for training veterinary students. *Studies in Health Technologies and Informatics* 111:33-36.
- [31]. Baillie S., Crossan A., Brewster S. A., May S. A., and Mellor D. J. 2010. Evaluating an automated haptic simulator designed for veterinary students to learn bovine rectal palpation. *Simulation in Healthcare* 5(5):261-266.
- [32]. Louis S. 2000. Optometric clinical practice guidelines. Care of the patient with learning-related vision problems. American Optometric Association.
- [33]. Smiciklas M. 2012. *The Power of Infographics: Using Pictures to Communicate and Connect With Your Audiences: Que.*
- [34]. Basdogan C., Ho C., and Srinivasan M. A. 2001. Virtual Environments for Medical Training: Graphical and Haptic Simulation of Laparoscopic Common Bile Duct Exploration. *IEEE Trans on Mechatronics* 6(3).
- [35]. Morris D., Sewell C., Blevins N., Barbagli F., and K. S. 2004. A Collaborative Virtual Environment for the Simulation of Temporal Bone Surgery. *Medical Image Computing and Computer-Assisted Intervention-MICCAI, Springer-Verlag LNCS* 3217.
- [36]. Webster R., Haluck R., Zoppetti G., Benson A., Boyd J., Charles N., Reeser J., and S. S. 2003. A Haptic Surgical Simulator for Laparoscopic Cholecystectomy using Real-time Deformable Organs. *Proceeding of IASTED International Conference Biomedical Engineering.*
- [37]. Youngblood P., Srivastava S., Curet M., Heinrichs L., Dev P., and Wren S. 2005. Comparison of training on two laparoscopic simulators and assessment of skills transfer to surgical performance. *J Amer Coll Surg* 200(4).

- [38]. Jones M. G., Bokinsky A., Tretter T., and Negishi A. 2005. A Comparison of Learning with Haptic and Visual Modalities. *Haptics-e* 3(6).
- [39]. Huang F., Gillespie R. B., and Kuo. A. 2004. Haptic Feedback Improves Manual Excitation of a Sprung Mass. *Proc IEEE Haptics Symposium*.
- [40]. O'Malley M. K., Gupta A., Gen M., and Li Y. March 2006. Shared Control in Haptic Systems for Performance Enhancement and Training. *Journal of Dynamic Systems, Measurement, and Control* 128(1).
- [41]. Bara F., and Gentaz E. 2011. Haptics in teaching handwriting: The role of perceptual and visuo-motor skills. *Human Movement Science* 30(4):745-759.
- [42]. Liu J., Cramer S. C., and Reinkensmeyer D. J. 2006. Learning to perform a new movement with robotic assistance: comparison of haptic guidance and visual demonstration. *Journal of NeuroEngineering and Rehabilitation* 3(1):495-506.
- [43]. Ernst M. O., and Banks M. S. 2002. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature* 415:429–433.
- [44]. Langrana N. A., Burdea G., Lange K., Gomez D., and Deshpande S. 1994. Dynamic force feedback in a virtual knee palpation. *Artificial Intelligence in Medicine* 6(4):321-333.
- [45]. Langrana N., Burdea G., Ladeji J., and Dinsmore M. 1997. Human performance using virtual reality tumor palpation simulation. *Computers & Graphics* 21(4):451-458.
- [46]. Burdea G., Patounakis G., Popescu V., and Weiss R. E. 1999. Virtual reality-based training for the diagnosis of prostate cancer. *Biomedical Engineering, IEEE Transactions on* 46(10):1253-1260.
- [47]. Kim S. Y., Park J., and Kwon D. 2004. Palpation Simulator for Laparoscopic Surgery with Haptic Feedback. *Int'l Conference on Biomedical Engineering (BioMed 2004)*: 478-482.

- [48]. Ullrich S., Mendoza J., Ntoubas A., Rossaint R., and Kuhlen T. 2008. Haptic Pulse Simulation for Virtual Palpation. In *Bildverarbeitung für die Medizin*:187-191.
- [49]. Alhalabi M. O., Daniulaitis V., Kawasaki H., and Hori T. 2005. Medical Training Simulation for Palpation of Subsurface Tumor Using Hiro. *Eurohaptics 2005*:623-624.
- [50]. Robert L. I., Srivastava M., Howell J. N., Robert R. Conatser J., Eland D. C., Burns J. M., and Chila A. G. 2004. The virtual haptic back for palpatory training. *Proceedings of the 6th international conference on Multimodal interfaces*. State College, PA, USA: ACM. p 191-197.
- [51]. Williams R. L. I., Howell J. N., and Conatser J. R. R. 2008. In *Handbook of Digital Human Modeling for Applied Ergonomics and Human Factors Engineering*. Human Factors and Ergonomics.
- [52]. Ji W., and University O. 2008. *Spatial Partitioning and Functional Shape Matched Deformation Algorithm for Interactive Haptic Modeling*: Ohio University.
- [53]. Stalfors J., Kling-Petersen T., Rydmark M., and Westin T. 2001. Haptic palpation of head and neck cancer patients--implication for education and telemedicine. *Studies in Health Technology and Informatics 81*:471-474.
- [54]. Endo T., Kawasaki H., Mouri T., Doi Y., Yoshida T., Ishigure Y., Shimomura H., Matsumura M., and Koketsu K. 2009. Five-fingered haptic interface robot: HIRO III. *EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems World Haptics 2009 Third Joint*. p 458-463.
- [55]. Endo T., Kawasaki H., Mouri T., Ishigure Y., Shimomura H., Matsumura M., and Koketsu K. 2011. Five-Fingered Haptic Interface Robot: HIRO III. *Haptics, IEEE Transactions on 4(1)*:14-27.

- [56]. Timothy R. C., Nigel W. J., Derek A. G., and Darwin G. C. 2011. Integrating Haptics with Augmented Reality in a Femoral Palpation and Needle Insertion Training Simulation. *IEEE Transactions on Haptics* 4:199-209.
- [57]. Sebastian U. 2012. Haptic Palpation for Medical Simulation in Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics* 18:617-625.
- [58]. Parkes R., Forrest N., and Baillie S. 2009. A mixed reality simulator for feline abdominal palpation training in veterinary medicine. *Studies in Health Technologies and Informatics* 142:244-246.
- [59]. Clark H., and Brennan S. 1991. Grounding in Communication. Washington D.C: In L. Resnick, J. Levine, & S. Teasley (Eds.), 127-149 p.
- [60]. Basdogan C. 1999. Force Reflecting Deformable Objects for Virtual Environments. in *Haptics: From Basic Principles to Advanced Applications, SIGGRAPH' 99, 26th International Conference on Computer Graphics and Interactive Techniques* Course No: 38.
- [61]. Basdogan C., Ho C., Srinivasan M. A., Small S., and Dawson S. 1998. Force interactions in laparoscopic simulations: haptic rendering of soft tissues. *Medicine Meets Virtual Reality VI*:385-391.
- [62]. Sederberg T. W., and Parry S. R. 1986. Free-form deformation of solid geometric models. *SIGGRAPH Comput Graph* 20(4):151-160.
- [63]. Davis O. R., and Burton R. P. 1991. Free-form deformation as an interactive modeling tool. *Journal of Imaging Technology* 17(4):181-187.
- [64]. Griessmair J., and Purgathofer W. 1989. Deformation of solids with trivariate B-splines. In: *Proceedings of Eurographics*, Elsevier, North Holland:137-148.
- [65]. Hsu W. M., Hughes J. F., and Kaufman H. 1992. Direct manipulation of free-form deformations. *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*: ACM. p 177-184.

- [66]. Song W., and Yang X. 2005. Free-form deformation with weighted T-spline. *Visual Computer* 21(3):139-151.
- [67]. Lewis J. P., Cordner M., and Fong N. 2000. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* ACM Press/Addison-Wesley Publishing Co..165-172. .
- [68]. Capell S., Green S., Curless B., Duchamp T., Popovi Z., and #263. 2002. Interactive skeleton-driven dynamic deformations. *ACM Transaction on Graphics (TOG)* 21(3):586-593.
- [69]. Robert I., Williams L., Howell J. N., and Conatser J. R. R. 2008. Digital Human Modeling for Palpatory Medical Training with Haptic Feedback. *Human Factors and Ergonomics*.
- [70]. Keeve E., Girod S., and Girod B. 1996. Craniofacial surgery simulation
Visualization in Biomedical Computing. In: Höhne K, and Kikinis R, editors: Springer Berlin / Heidelberg. p 541-546.
- [71]. Lee Y., Terzopoulos D., and Waters K. 1995. Realistic modeling for facial animation. *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques: ACM*. p 55-62.
- [72]. Brown J., Sorkin S., Latombe J.-C., Montgomery K., and Stephanides M. 2002. Algorithmic tools for real-time microsurgery simulation. *Medical Image Analysis* 6(3):289-300.
- [73]. Mori K., Seki Y., Hasegawa J. I., Toriwaki J. I., Anno H., and Katada K. 1997. A method for shape deformation of organ and its application to virtualized endoscope systems. in: HU Lemke, et al (Eds), *Proceedings of the Computer Aided Radiology and Surgery (CAR: 11)*, Excerpta Medica, International Congress Series 1134:189—194.
- [74]. Delingette H. 1994. Simplex Meshes: A General Representation for 3D Shape Reconstruction. INRIA, Research Report RR2214.

- [75]. DeBunne G., Desbrun M., Cani M.-P., and Barr A. H. 2001. Dynamic real-time deformations using space & time adaptive sampling. Proceedings of the 28th annual conference on Computer graphics and interactive techniques: ACM. p 31-36.
- [76]. Delingette H. 1994. Simplex meshes: a general representation for 3D shape reconstruction. Computer Vision and Pattern Recognition, 1994 Proceedings CVPR '94, 1994 IEEE Computer Society Conference on. p 856-859.
- [77]. Keeve E., Girod S., and Girod B. 1996. Craniofacial surgery simulation. In: KH BÖHME, R KIKINIS (Eds) Proceedings of the Visualization in Biomedical Computing (VBC: 4), Lecture Notes in Computer Science 1131:541—546.
- [78]. Monserrat C., and al. e. 1996. Aproximación mediante Modelos Deformables a la Reconstrucción 3D de la Anatomía Dental. in: Proceedings of the Congreso Español de Ingeniería Gráfica (CEIG: 6), Universidad Politécnica de Valencia, Valencia.
- [79]. Kuhn C., Kühnappel U., krumm H. G., and Neisius B. 1996. A Virtual Reality Based Training System for Minimally Invasive Surgery. In: HU Lemke, et al (Eds), Proceedings of the Computer Assisted Radiology (CAR: 10), Elsevier Science, Amsterdam: 764—769.
- [80]. Meier U., Lopez O., Monserrat C., Juan M. C., and Alcaniz M. 2005. Real-time deformable models for surgery simulation: a survey. Computer Methods and Programs Biomedicine 77:183-197.
- [81]. Witkin A. Physically Based Modeling: Principles and Practice Particle System Dynamics.
- [82]. Cotin S., Delingette H., and Ayache N. 1998. Efficient Linear Elastic Models Of Soft Tissues For Real-Time Surgery Simulation. Inria Technical Report Rr-3510.
- [83]. Monserrat C. 1999. Modelos Deformables de Tejidos Elásticos en Tiempo Real. PhD Thesis, Universidad Politécnica de Valencia, Valencia.

- [84]. Baraff D. Physically Based Modeling Rigid Body Simulation.
- [85]. Downes M., Cavusoglu M. C., Gantert W., Way L. W., and Tendick F. 1998. Virtual environments for training critical skills in laparoscopic surgery. in: JD Westwood, et al (Eds), Proceedings of the Medicine Meets Virtual Reality (MMVR: 6), Studies in Health Technology and Informatics 50:316—322.
- [86]. Hubbard P. M. 1993. Interactive collision detection. in: Proceedings of IEEE Symposium on Research Frontiers in Virtual Reality.
- [87]. Downes M., Cavusoglu M. C., Gantert W., Way L. W., and Tendick F. 1998. Virtual environments for training critical skills in laparoscopic surgery. Stud Health Technol Inform 50:316-322.
- [88]. Terzopoulos D., Platt J., Barr A., and Fleischer K. 1987. Elastically Deformable Models. Siggraph Computer Graphics 21(4):205-214.
- [89]. Terzopoulos D., and Fleischer K. 1988. Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture. Proc 15th Ann Conf Computer Graphics And Interactive Techniques (Siggraph '88):269-278.
- [90]. Baraff D., and Witkin A. 1998. Large steps in cloth simulation. Proceedings of the 25th annual conference on Computer graphics and interactive techniques: ACM. p 43-54.
- [91]. Provot X. 1995. Deformation Constraints In A Spring-Mass Model To Describe Rigid Cloth Behavior. Proceeding of Graphics Interface Conference:147-154.
- [92]. Kahler K., Haber J., and Seidel H. P. 2001. Dynamic refinement of deformable triangle meshes for rendering. Computer Graphics International 2001 Proceedings. p 285-290.
- [93]. Castañeda M. A. P., and Cosío F. A. 2004. Deformable model of the prostate for TURP surgery simulation. Computers & Graphics:767-777.

- [94]. Kühnapfel U., Çakmak H. K., and Maaß H. 2000. Endoscopic surgery training using virtual reality and deformable tissue simulation. *Computers & Graphics* 24(5):671-682.
- [95]. Mosegaard J., Herborg P., and Sorensen T. S. 2005. A GPU accelerated spring mass system for surgical simulation. *Stud Health Technol Inform* 111:342-348.
- [96]. Shaoping X., Liu X. P., Hua Z., and Linyan H. 2010. An improved realistic mass-spring model for surgery simulation. *Haptic Audio-Visual Environments and Games (HAVE)*, 2010 IEEE International Symposium on. p 1-6.
- [97]. Basafa E., and Farahmand F. 2011. Real-time simulation of the nonlinear visco-elastic deformations of soft tissues. *International Journal of Computer Assisted Radiology and Surgery* 6(3):297-307.
- [98]. Etmuss O., Gross J., and Strasser W. 2003. Deriving a Particle System from Continuum Mechanics for the Animation of Deformable Objects. *IEEE Transactions on Visualization and Computer Graphics* 9(4):538-550.
- [99]. Bianchi G., Harders M., and Szekely G. 2003. Mesh Topology Identification For Mass-Spring Models. In: Ellis, RE, Peters, TM (Eds) *Miccai 2003 Lncs*, Springer 2878:50-58.
- [100]. d'Aulignac D., Cavusoglu M. C., and Laugier C. 1999. Modeling the Dynamics of the Human Thigh for a Realistic Echographic Simulator with Force Feedback. *Proceedings of the Second International Conference on Medical Image Computing and Computer-Assisted Intervention: Springer-Verlag*. p 1191-1198.
- [101]. Santhanam A. P. 2006. Modeling, Simulation, And Visualization Of 3d Lung Dynamics. Phd Thesis, School Of Electrical Engineering And Computer Science, University Of Central Florida.
- [102]. Misra S., Ramesh K. T., and Okamura A. M. 2008. Modeling of Tool-Tissue Interactions for Computer-Based Surgical Simulation: A Literature Review. *Presence: Teleoperators and Virtual Environments* 17(5):463-491.

- [103]. Bro-Nilsen M., and Cotin S. 1996. Real Time Volumetric Deformable Models For Surgery Simulation Using Finite Elements And Condensation. *Computer Graphics Forum* 15(3).
- [104]. Cotin S., Delingette H., and Ayache N. 1999. Real-time elastic deformations of soft tissues for surgery simulation. *Visualization and Computer Graphics, IEEE Transactions on* 5(1):62-73.
- [105]. Cotin S., Delingette H., and Ayache N. 1996. Real Time Volumetric Deformable Models for Surgery Simulation. *Proceedings of the 4th International Conference on Visualization in Biomedical Computing: Springer-Verlag.* p 535-540.
- [106]. Rhombert A. C. 2001. *Real-Time Finite Elements: A Parallel Computer Application.* Phd Thesis, Swiss Federal Institute Of Technology, Zurich.
- [107]. Székely G., Brechbühler C., Dual J., Enzler R., Hug J., Hutter R., Ironmonger N., Kauer M., Meier V., Niederer P. et al. . 2000. Virtual Reality-Based Simulation of Endoscopic Surgery. *Presence: Teleoperators and Virtual Environments* 9(3):310-333.
- [108]. Delingette H., and Ayache N. 2005. Hepatic surgery simulation. *Communications of the ACM* 48(2):31-36.
- [109]. Popescu D. C., and Compton M. 2003. A model for efficient and accurate interaction with elastic objects in haptic virtual environments. *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia.* Melbourne, Australia: ACM. p 245-250.
- [110]. Nikitin I., Nikitina L., Frolov P., Goebbels G., G M., Klimenko S., and Nielson G. M. 2002. Real-time simulation of elastic objects in virtual environments using finite element method and precomputed Green's functions. *Proceedings of the workshop on Virtual environments 2002.* Barcelona, Spain: Eurographics Association. p 47-52.

- [111]. Berkley J. W., Gladstone S., Raugi H., Gberg H., and Ganter D. M. 1999. Banded Matrix Approach To Finite Element Modeling For Soft Tissue Simulation. *Virtual Reality: Research, Development, And Applications* 4:203-212.
- [112]. Fung Y. C. 1993. *Biomechanics - Mechanical Properties Of Living Tissues*. Springer-Verlag, Second Edition.
- [113]. Maurel W., Y. Wu Y., Thalmann N. M., and Thalmann D. 1998. *Biomechanical Model For Soft Tissue Simulation*. Springer-Verlag, Second Edition.
- [114]. Picinbono G., Delingette H., and Ayache N. 2003. Non-Linear Anisotropic Elasticity For Real-Time Surgery Simulation. *Graphical Models*:305–321.
- [115]. Wu X., Downes M. S., Goktekin T., and Tendick F. 2001. Adaptive Nonlinear Finite Elements for Deformable Body Simulation Using Dynamic Progressive Meshes. *Computer Graphics Forum* 20(3):349-358.
- [116]. Joldes G. R., Wittek A., and Miller K. 2009. Suite Of Finite Element Algorithms For Accurate Computation Of Soft Tissue Deformation For Surgical Simulation. *Medical Image Analysis* 13(6):912-919.
- [117]. Barbic J., and James D. L. 2005. Real-Time Subspace Integration For St. Venant-Kirchhoff Deformable Models. In *Siggraph '05: Acm Siggraph*:982–990.
- [118]. James D. L., and Pai D. K. 1999. ArtDefo: accurate real time deformable objects. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques: ACM Press/Addison-Wesley Publishing Co.* p 65-72.
- [119]. Wang P., Becker A. A., Jones I. A., Glover A. T., Benford S. D., Greenhalgh C. M., and Vloeberghs M. 2007. Virtual reality simulation of surgery with haptic feedback based on the boundary element method. *Comput Struct* 85(7-8):331-339.

- [120]. Meier U., Lopez O., Monserrat C., Juan M. C., and Alcaniz M. 2005. Real-time deformable models for surgery simulation: a survey. *Computer Methods and Programs in Biomedicine* 77(3):183-197.
- [121]. Zhou X., Zhang N., Sha D., Shen Y., Tamma K. K., and Sweet R. 2009. A Discrete Mechanics Framework For Real Time Virtual Surgical Simulations With Applications To Virtual Nephrectomy. In *Medicine Meets Virtual Reality 17 (Mmvr17)*, Ios Press, Long Beach:459-564.
- [122]. Rasmusson A., Mosegaard J., and Sorensen T. S. 2008. Exploring Parallel Algorithms For Volumetric Mass-Spring-Damper Models In Cuda. In *Springer Berlin/Heidelberg, Ed, Biomedical Simulation 5104*:49-58.
- [123]. Comas O., Taylor Z. A., Ourselin S., Cotin S., and Passeger J. 2008. Efficient Nonlinear Fem For Soft Tissue Modelling And Its Gpu Implementation Within The Open Source Framework Sofa. In *Isbms '08: Proceedings Of The 4th International Symposium On Biomedical Simulation, Berlin, Heidelberg, Springer-Verlag*: 28–39.
- [124]. Choi K.-S., Sun H., Heng P.-A., and Zou J. 2004. Deformable simulation using force propagation model with finite element optimization. *Computers & Graphics* 28(4):559-568.
- [125]. Choi Y.-J., Hong M., Choi M.-H., and Kim M.-H. 2005. Adaptive surface-deformable model with shape-preserving spring: Research Articles. *Computer Animation and Virtual Worlds* 16(1):69-83.
- [126]. Hutchinson D., Preston M., and Hewitt T. 1996. Adaptive refinement for mass/spring simulations. *Proceedings of the Eurographics workshop on Computer animation and simulation '96. Poitiers, France: Springer-Verlag New York, Inc.* p 31-45.
- [127]. Debunne G., Desbrun M., Barr A., and Cani M. P. 1999. Interactive multi resolution animation of deformable models. In *10th Eurographics Workshop on Computer Animation and Simulation*.

- [128]. DeBunne G., Desbrun M., Cani M. P., and Barr A. 2000. Adaptive simulation of soft bodies in real-time. *Computer Animation 2000 Proceedings*. p 15-20.
- [129]. Capell S., Green S., Curless B., Duchamp T., and Popovi Z. 2002. A multiresolution framework for dynamic deformations. *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. San Antonio, Texas: ACM. p 41-47.
- [130]. Kawai H. 2001. Elastic Object manipulation using coarse-to-fine representation of mass-spring models. In *SIGGRAPH 2001 Conference Abstracts and Applications*.
- [131]. Paloc C., Faraci F., and Bello F. 2006. Online remeshing for soft tissue simulation in surgical training. *IEEE Computer Graphics and Applications* 26:24-34.
- [132]. Susa I., Sato M., and Hasegawa S. 2011. Multi-rate multi-range dynamic simulation for haptic interaction. *Proceedings of IEEE Haptics*:233-238.
- [133]. Payandeh S., Dill J., and Zhang J. 2005. A study of level-of-detail in haptic rendering. *ACM Transaction on Applied Perception* 2(1):15-34.
- [134]. Chang Y.-H., Chen Y.-T., Chang C.-W., and Lin C.-L. 2008. Development scheme of haptic-based system for interactive deformable simulation. *Computer-Aided Design*:1-11.
- [135]. Zhong Y.-M., Shirinzadeh B., Smith J., and Gu C.-F. 2009. An electromechanical based deformable model for soft tissue simulation. *Artificial Intelligence in Medicine* 47(3):275-288.
- [136]. Vollinger Y., Stier H., Priensnitz J., and Krause F.-L. 2009. Evolutionary optimization of mass-spring models. *CIRP Journal of Manufacturing Science and Technology* 1(3):137-141.
- [137]. Leech N., and Onwuegbuzie A. 2010. Guidelines for conducting and reporting mixed research in the field of counseling and beyond. *Journal of Counseling and Development (JCD)* 88(1):61-70.

- [138]. Ramesh V., Glass R. L., and Vessey I. 2004. Research in computer science: an empirical study. *The Journal of System and Software* 70:165-176.
- [139]. Fisher R. A., editor. 1935. *The Design of Experiments*: Edinburgh and London: Oliver and Boyd.
- [140]. Kirk R. E. *Experimental Design*.
- [141]. McLeod S. A. 2012. *Experimental Methods in Psychology - Simply Psychology*.
- [142]. Kirk R. E., editor. 1995. *Experimental Design: Procedures for the Behavioral Sciences*. 3rd edn ed: Pacific Grove, CA: Brooks/Cole.
- [143]. PHANToM. 2013. Phantom force feedback device, Accessed March 2013. Sensable Inc, USA.
- [144]. OpenHaptics. 2012. OpenHaptics Toolkit from Sensable. Sensable Technologies Ltd.
- [145]. Delingette H. 1998. Towards Realistic Soft Tissue Modelling in Medical Simulation. *Proc IEEE: Special Issue on Surgery Simulation*:512-523.
- [146]. Si H. 2013. TetGen: a quality tetrahedral mesh generator and three-dimensional Delaunay triangulator. Open Source Project.
- [147]. Liu Z., and Bilston L. 2000. On the Viscoelastic Character of Liver Tissue: Experiments and Modeling of the Linear Behavior. *Biorheology* 37:191-201.
- [148]. Liu Z., and Bilston L. E. 2002. Large Deformation Shear Properties of Liver Tissue. *Biorheology* 39:735-742.
- [149]. Farshad M., Barbezat M., Flueler P., Schmidlin F., Graber P., and Niederer P. 1999. Material Characterization of the Pig Kidney in Relation with the Biomechanical Analysis of Renal Trauma. *Journal of Biomechanics* 32:417-425.

- [150]. Kim J., Tay B., Stylopoulos N., Rattner D. W., and Srinivasan M. A. 2003. Characterization of Intra-Abdominal Tissues from in Vivo Animal Experiment for Surgical Simulation. *Medical Image Computing and Computer-Assisted Intervention-Miccai*.
- [151]. Taylor L. S., Lerner A. L., Rubens D. J., and Parker K. J. 2002. A Kelvin–Voigt fractional derivative model for viscoelastic characterization of liver tissue. *ASME Int Mechanical Engineering Congress and Exposition (NewOrleans,LA)* ed E P Scott.
- [152]. Cai Z., Dill J., and Payandeh S. 2000. Toward deformation modeling with haptic feedback. In *Proc, ASME Dynamic Systems and Control Div, DSC*. p 2.
- [153]. Catheline S., Gennisson J. L., Delon G., Fink M., Sinkus R., Abouelkaram S., and Culioli J. 2004. Measurement of viscoelastic properties of homogeneous soft solid using transient elastography: An inverse problem approach. *The Journal of the Acoustical Society of America* 116(6):3734-3741.
- [154]. Desbrun M., Schroder P., and Barr A. 1999. Interactive animation of structured deformable objects. *Proceedings of Graphics Interface '99*.
- [155]. Breen D., House D., and Wozny M. 1994. Predicting the drape of woven cloth using interacting particles. *Computer Graphics (Proc of SIGGRAPH '94)*:365-372.
- [156]. Roosendaal T. 2012. *The Blender Foundation*.
- [157]. Cignoni P., Corsini M., and Ranzuglia G. 2008. MeshLab: an Open-Source 3D Mesh Processing System. *ERCIM News* 2008 2008(73).
- [158]. Liu J., Emken J. L., Cramer S. C., and Reinkensmeyer D. J. 2005. Learning to Perform a Novel Movement Pattern using Haptic. *International Conference on Rehabilitation Robotics (ICORR)*:37-40.
- [159]. Kirk R. E. *Experimental design: Wiley Online Library*.

- [160]. Minitab. 2010. Minitab 16 statistical software. Minitab Inc, State College, Pennsylvania, USA.
- [161]. Ji W. 2008. Spatial Partitioning and Functional Shape Matched Deformation Algorithm for Interactive Haptic Modeling: Ohio University.

APPENDIX A

Calculating Neighbourhood Of A Touched Node.

Code uses for haptic sensation OpenHaptics APIs [144].

```
void removeDuplicate ();
void removeDuplicate1 ();
void NieghborsofPointN();
void Compare();
void remove();

void PointNeighbour()
{
    int k;
    for(int i= 0;i<numMass; i++)
    {
        k=0;
        Pt[i].numNeighbour =0;
        for (int j=0; j<numFace; j++)
        {
            if (i==tris[j].vIndex [0])
            {
                Pt[i].Neighbour [k++] =tris[j].vIndex [1];
                Pt[i].Neighbour [k++] =tris[j].vIndex [2];
                Pt[i].numNeighbour += 2;
            }
            if (i==tris[j].vIndex [1])
            {
                Pt[i].Neighbour [k++] =tris[j].vIndex [0];
                Pt[i].Neighbour [k++] =tris[j].vIndex [2];
                Pt[i].numNeighbor+=2;
            }
            if (i==tris[j].vIndex [2])
            {
```

```

Pt[i].Neighbour [k++] =tris[j].vIndex [0];

Pt[i].Neighbour [k++] =tris[j].vIndex [1];

Pt[i].numNeighbour+=2;
}
}
removeDuplicate ();
removeDuplicate ();
}
NieghborsofPointN ();
}
void removeDuplicate ()
{
int i, j;

for (int k =0; k<numMass; k++)
{
for (i = 0; i<Pt[k].numNeighbor; i++)
{
for (j = i+1; j<Pt[k].numNeighbor; j++)
{
if (Pt [k] .Neighbour[i] == Pt[k].Neighbour[j])
{
for (int m = j; m<Pt[k].numNeighbor; j++)
{
Pt[k].Neighbour[m] = Pt[k].Neighbour [m+1];
}
Pt[k].numNeighbor--;
}
}
}
}
}
}
void removeDuplicate1 ()
{
int i, j;

for (int k =0; k<numMass; k++)
{

for (i = 0; i<Pt[k].numNeighbor; i++)

```

```

        {
            For (j = i+1; j<Pt[k].numNeighbor; j++)
            {
                if (Pt [k] .NNeighbour [i] == Pt[k].NNeighbour [j])
                {
                    for (int m = j; m<Pt[k].numNeighbor; j++)
                    {
                        Pt[k].NNeighbour [m] = Pt[k].NNeighbour [m+1];
                    }
                    Pt[k].numNNeighbor --;
                }
            }
        }
    }

    Compare ();
}
void NieghborsofPointN ()

{
    int p;
    int k =0 ;
    int i;

    for( i = 0 ; i<numMass; i++)
    {
        p = 0;
        Pt[i].numNNeighbor =0;

        for(int j = 0; j<Pt[i].numNeighbor; j++)
        {

            k = Pt[i].Neighbour[j];

            for(int n = 0; n<Pt[k].numNeighbour; n++)
            {
                if(i != Pt[k].Neighbour[n])
                {
                    Pt[i].NNeighbour [p++] = Pt[k].Neighbour[n];
                    Pt[i].numNNeighbor +=1;
                }
            }
        }
    }
}
removeDuplicate1();
removeDuplicate1();

```



```

    }

}

void remove( int i, int n)
{
    for(int j = n; j<Pt[i].numNNeighbor ; j++)
    {
        Pt[i].NNeighbour [j] = Pt[i].NNeighbour[j+1];
    }
    Pt[i].numNNeighbor --;

}

void Compare()
{
    for(int i = 0 ; i<numMass; i++)
    {
        for(int j = 0; j<Pt[i].numNeighbor; j++)
        {
            for(int n = 0; n<Pt[i].numNNeighbor ; n++)
                if(Pt[i].Neighbour[j] == Pt[i].NNeighbour[n])
                    remove( i, n);
        }
    }

}

```

3D Model Reading Using Half-Edge Data Structure.

```

#ifndef MESH_H
#define MESH_H

class Triangle;
class Edge;
// Class for a 3D point in space
class Mass
{
public:
    // Default Constructor
    Mass(): position(0.0,0.0,0.0),

                velocity(0.0,0.0,0.0),
                force(0.0,0.0,0.0),

```

```

        mass(0.0),fixed(false){}
        // Position of this Vertex in 3D (x, y, z)
        hduVector3Df position;
        // Velocity of this Vertex
        hduVector3Df velocity;
        // Force acted on this Vertex
        hduVector3Df force;
        float mass;
        bool fixed;
        Edge* edge;// edge emanating from this vertex
};

```

```

class Triangle

```

```

{
public:
    Triangle()    {edge = NULL;}
    Triangle( int v0, int v1, int v2)
    {
        vIndex[0] = v0;
        vIndex[1] = v1;
        vIndex[2] = v2;
    }
    void setEdge(Edge * e)
    {
        assert(edge==NULL);
        edge = e;
    }
    Edge* getEdge ()
    {
        assert(edge!=NULL);
        return edge;
    }
    void calculateTriangleArea ();
    //      {
    //          hduVector3Df P1;
    //              //,P2,P3;
    //          P1 = (hduVector3Df)this->edge->getVertex1()->position;
    //
    //          //      P1 = tri->edge->getVertex1()->position;
    //          //      P2 = tri->edge->getNextEdge ()->getVertex1()->position;
    //          //      P3 = tri->edge->getNextEdge ()->getNextEdge ()-
    >getVertex1()->position;
    //      }

```

```

int vIndex[3];

```

```

        float triangleArea;
        Edge* edge; // one of the half edges bordering the face
    };

class Edge
{
public:
    Edge()
    {
        triangle = NULL;
        m_nextEdge = NULL;
        m_opposite = NULL;
        m_vertex1 = NULL;
        m_vertex2 = NULL;
    }
    // Edge(Mass* endvert, Triangle *tri) ;
    void set(Mass * vert1, Mass * vert2, Triangle * tri);
    Mass * m_vertex1; //first vertex of a half-edge //vertex at the end of the half-
edge
    Mass * m_vertex2; // second vertex of a half-edge
    Triangle * triangle; // triangle the half-edge borders;
    Edge* m_nextEdge; //next half-edge around the face
    Edge* m_opposite; // oppositely oriented half-edge
    Edge* getNextEdge ();
    Edge* getOppositeEdge ();
    void setOppositeEdge (Edge * opposite);
    void setNextEdge (Edge* next);
    Mass* getVertex1();
    Mass * getVertex2();
    void setVertex1(hduVector3Df vert1);
    void setVertex2(hduVector3Df vert2);
    Triangle* getTriangle ()
    {
        assert (triangle!=NULL);
        return triangle;
    }
};

class Mesh
{
public:
    Mesh();
    ~Mesh();
    typedef std::vector<Mass *> MeshVertexList;
    typedef std::vector<Triangle *> MeshTriangleList ;
    typedef std::vector<Edge * > MeshEdgeList ;
    // List of Vertex used for storing Points in OBJ file

```

```

MeshVertexList vertices;
    // List of Triangle used for storing Triangles present in the OBJ file
    MeshTriangleList triangles;
    // List of edges present in the OBJ file
    MeshEdgeList edges;
    void drawModelUsingEdge ();
int pointTouched ;
    int numEdges;
    int numVertices ;
    int numTriangles;
void AddVertex(float x, float y, float z);
void AddTriangle ( int v1, int v2, int v3);
void LoadModel ();
void DrawModel ();
void VertexNeighbors ();
//unsigned int getVertexIndex ();
Triangle* GetOneTriangle ( int index);
//void FindNeighbor ();
// Shows total number of neighbours of a vertex
void ShowNeighbor ();
// This Function gives information about total number of vertices and triangles in a
mesh
void MeshModelInfo ();
void CalculateNormal ();
void NeighborofVertexUsingEdge ();
void drawTriangle (hduVector3Df P1,hduVector3Df P2,hduVector3Df P3);
void subdivide(Triangle *tri);
void firstPassSubdivision (Triangle *tri);
void secondPassSubdivision (Triangle* tri);
void drawPoints ();
void drawTriangleGraphics (hduVector3Df P1,hduVector3Df P2,hduVector3Df P3);
void drawModelGraphics ();
};

```

```

#endif

```

```

#include "Mesh.h"
#include <assert.h>
#include <set>
#include <time.h>
Edge * Edge::getNextEdge ()
{
    return(m_nextEdge);
}
Edge * Edge::getOppositeEdge ()
{
    return m_opposite;
}

```

```

}
void Edge::setOppositeEdge (Edge * e)
{
    assert (m_opposite == NULL);
    assert (e != NULL);
    assert (e->m_opposite == NULL);
    m_opposite = e;
    e->m_opposite = this;
}
void Edge::setNextEdge (Edge* e)
{
    assert(m_nextEdge == NULL);
    assert(e!=NULL);
    m_nextEdge = e;
}
Mass * Edge::getVertex1()
{
    return m_vertex1;
}
Mass * Edge::getVertex2()
{
    return m_vertex2;
}
void Edge::set(Mass * vert1,Mass * vert2, Triangle *tri)
{
    m_vertex1 = vert1;
    m_vertex2 = vert2;
    triangle = tri;
}
void Triangle::calculateTriangleArea ()
{
    hduVector3Df P1,P2,P3;
    //P1 = (hduVector3Df)this->edge->getVertex1()->position;

    //      P1 = this->edge->getVertex1()->position;
    //      P2 = this->edge->getNextEdge ()->getVertex1()->position;
    //      P3 = this->edge->getNextEdge()->getNextEdge()->getVertex1()-
    >position;

    //float mag =sqrt ( (P.x * P.x) + (P.y * P.y) + (P.z * P.z) ) ;
    //this->triangleArea = 0.5*mag;
}
Mesh::Mesh()
{
    numVertices =0;

```

```

        numTriangles = 0;
        numEdges = 0;
        pointTouched = -1;
    }
Mesh::~Mesh()
{
    for(int i=0;i<(int)vertices.size ();i++)
        delete vertices[i];

    for( i=0;i<(int)triangles.size ();i++)
        delete triangles[i];

}
void Mesh::AddVertex(float x, float y, float z)
{
    Mass *m = new Mass;
    m->position.set (x, y, z);
    vertices.push_back (m);
}
void Edge::setVertex1(hduVector3Df v1)
{
    m_vertex1->position = v1;
}
void Edge::setVertex2(hduVector3Df v2)
{
    m_vertex2->position = v2;
}
void Mesh::AddTriangle ( int vert1, int vert2, int vert3)
{
    Triangle* tri= new Triangle;
    tri->vIndex[0] = vert1;
    tri->vIndex[1] = vert2;
    tri->vIndex[2] = vert3;
    Edge* e = new Edge;
    int k= 0;

    Mass* v1 =new Mass;
    Mass *v2 = new Mass;
    Mass *v3 = new Mass;

    v1 = vertices[tri->vIndex[0]];
    v2 = vertices[tri->vIndex[1]];
    v3 = vertices[tri->vIndex[2]];

    Edge * edge1 = new Edge;
    Edge * edge2 = new Edge;

```

```

Edge * edge3 = new Edge;

        numEdges++;
        edge1->set(v1,v2,tri);
        numEdges++;
        edge2->set(v2,v3,tri);
        numEdges++;
        edge3->set(v3,v1,tri);

        edge1->setNextEdge (edge2);
        edge2->setNextEdge (edge3);
        edge3->setNextEdge (edge1);

        edges.push_back (edge1);
        edges.push_back(edge2);
        edges.push_back(edge3);
        tri->edge = edge1;
        triangles.push_back(tri);
}
void Mesh::LoadModel()
{
const char *filename = "trisphere.obj";
//      const char *filename = "OvaryModelS.obj";
      FILE *fp = fopen(filename,"r");
// printf("Processing...\n");
      int v;
      char buf[256];

      while (!feof (fp))
      {
          /* Read whole line */
          fgets (buf, sizeof (buf), fp);

          switch (buf[0])
          {
              case 'v':
                  {
                      if (buf[1] == ' ')
                      {

/* Vertex */
                          numVertices++;
                      }
                      break;
                  }

              case 'f':
                  {

```

```

        if (sscanf (buf + 2, "%d", &v) == 1)
        {
numTriangles++;

        }
        else
        {
            /* Should never be there or the model is very crappy */
            fprintf (stderr, "Error: found face with no vertex!\n");
        }

        break;
    }

    default:
        break;
}
}

/* Check if informations are valid */

if (!numVertices)
{
    fprintf (stderr, "error: no vertex found!\n");
}
rewind(fp);
char *pbuf;
int i;
float x,y,z;
while (!feof (fp))
{
    /* Read whole line */
    fgets (buf, sizeof (buf), fp);

    switch (buf[0])
    {
        case 'v':
            {
                if (buf[1] == ' ')
                {
                    /* Vertex */

                    if (sscanf (buf + 2, "%f %f %f", &x,&y,&z) != 3)
                    {
                        fprintf (stderr, "Error reading vertex data!\n");
                    }
                }
            }
        }
    }
}

```



```

        AddVertex(x,y,z);
    }

    break;
}

case 'f':
{
    pbuf = buf;
    // i = 0;
    int index[3];
    for (i = 0; i < 3; ++i)
    {
        pbuf = strchr (pbuf, ' ');
        pbuf++; /* Skip space */
        /* Try reading vertices */
        sscanf (pbuf, "%d", &index[i]);
        /* Indices must start at 0 */
        index[i]--;
    }
        AddTriangle(index[0],index[1],index[2]);

    break;
}
}

// printf("Finished Processing...\n");
// printf("Edge Linking to Vertex...\n");
for(i = 0;i<numVertices;i++)
{
    for(int j=0;j<numEdges;j++)
    {
        if(vertices[i]->position == edges[j]->getVertex1()->position)
            vertices[i]->edge = edges[j];
    }
}

// std::set< Edge*> uedges(edges.begin(),edges.end());
//
// printf("%d ",edges.size());
// printf("\n %d ",uedges.size());
//
// printf("Finished Edge Linking to Vertex...\n\n");
// printf("Now Making Twins...\n\n\n");
for(i = 0;i<numEdges;i++)
{
for(int j = 0;j<numEdges;j++)
{

```

```

        if(edges[j]->m_opposite == NULL && edges[i]->m_opposite == NULL)
        {
            if(edges[i]->m_vertex1 == edges[j]->m_vertex2 && edges[i]-
>m_vertex2 == edges[j]->m_vertex1 )
            {
                edges[i]->setOppositeEdge(edges[j]);
                // edges[i]->m_opposite = edges[j];
                // edges[j]->m_opposite = edges[i];
            }
        }
    }
}
// printf("Finished Twins....\n");

}

```

Mass Spring Model.

Initial Spring Calculation.

```

void initAllSprings()
{
    int sIndex = 0;
    double x1,x2,y1,y2,z1,z2;           // positions of spring points p1, p2
    double r12d;                        // length of p1 - p2 vector
    double vx12;                        // vx1 - vx2
    double vy12,vz12;                  // vy1 - vy2
    double f;                           // hooke force value
    double Fx,Fy,Fz;                   // force vector

    for(int i = 0;i<numFace;i++)
    {

        spring[sIndex].imass1 = tris[i].vIndex[0];
        spring[sIndex].imass2 = tris[i].vIndex[2];
        x1 = Pt[tris[i].vIndex[0]].position[0];
        x2 = Pt[tris[i].vIndex[2]].position[0];

        y1 = Pt[tris[i].vIndex[0]].position[1];
        y2 = Pt[tris[i].vIndex[2]].position[1];

        z1 = Pt[tris[i].vIndex[0]].position[2];
        z2 = Pt[tris[i].vIndex[2]].position[2];
    }
}

```

```

        spring[sIndex].naturalLength = sqrt ( (x1 - x2) *(x1 - x2) + (y1 - y2)
* (y1 - y2) + (z1 - z2)*(z1 - z2) );    // square

```

```

    sIndex++;

```

```

        spring[sIndex].imass1 = tris[i].vIndex[2];
        spring[sIndex].imass2 = tris[i].vIndex[1];

```

```

        x1 = Pt[tris[i].vIndex[2]].position[0];
        x2 = Pt[tris[i].vIndex[1]].position[0];

```

```

        y1 = Pt[tris[i].vIndex[2]].position[1];
        y2 = Pt[tris[i].vIndex[1]].position[1];

```

```

        z1 = Pt[tris[i].vIndex[2]].position[2];
        z2 = Pt[tris[i].vIndex[1]].position[2];

```

```

        spring[sIndex].naturalLength = sqrt ( (x1 - x2) *(x1 - x2) + (y1 - y2)
* (y1 - y2) + (z1 - z2)*(z1 - z2) );    // square

```

```

    sIndex++;

```

```

        spring[sIndex].imass1 = tris[i].vIndex[1];
        spring[sIndex].imass2 = tris[i].vIndex[0];

```

```

        x1 = Pt[tris[i].vIndex[1]].position[0];
        x2 = Pt[tris[i].vIndex[0]].position[0];

```

```

        y1 = Pt[tris[i].vIndex[1]].position[1];
        y2 = Pt[tris[i].vIndex[0]].position[1];

```

```

        z1 = Pt[tris[i].vIndex[1]].position[2];
        z2 = Pt[tris[i].vIndex[0]].position[2];

```

```

        spring[sIndex].naturalLength = sqrt ( (x1 - x2) *(x1 - x2) + (y1 - y2)
* (y1 - y2) + (z1 - z2)*(z1 - z2) );    // square

```

```

    sIndex++;

```

```

}
numSpring = sIndex-1;
sIndex=0;
}

```

Force Accumulation.

```

void accumulateForce()

```

```

{

```

```

    for(int i= 0;i<numMass;i++)

```

```

    {

```

```

        Pt[i].sforce[0]=0.0;

```

```

        Pt[i].sforce[1]=0.0;

```

```

        Pt[i].sforce[2]=0.0;

```

```

    }

double x1,x2,y1,y2,z1,z2;           // positions of spring points p1, p2
double r12d;                        // length of p1 - p2 vector
double vx12;                        // vx1 - vx2
double vy12,vz12;                  // vy1 - vy2
double f;                           // hooke force value
double Fx,Fy,Fz;                    // force vector

for( i=0 ; i < numSpring ; i++)
{
    x1 = Pt[spring[i].imass1].position[0];
    x2 = Pt[spring[i].imass2].position[0];

    y1 = Pt[spring[i].imass1].position[1];
    y2 = Pt[spring[i].imass2].position[1];

    z1 = Pt[spring[i].imass1].position[2];
    z2 = Pt[spring[i].imass2].position[2];
    r12d = sqrt ( (x1 - x2) *(x1 - x2) + (y1 - y2) * (y1 - y2) + (z1 -
z2)*(z1 - z2) );           // square

    // root of the distance

    vx12 = Pt[spring[i].imass1].velocity[0] -
Pt[spring[i].imass2].velocity[0];
    vy12 = Pt[spring[i].imass1].velocity[1] -
Pt[spring[i].imass2].velocity[1];
    vz12 = Pt[spring[i].imass1].velocity[2] -
Pt[spring[i].imass2].velocity[2];
    f = (r12d - spring[i].naturalLength) *2+ (vx12 * (x1 - x2) + vy12 * (y1
- y2)+ vz12*(z1 - z2)) *.01 / r12d;
    hduVector3Dd F;
    F[0] = ((x1 - x2) / r12d) * f;
    F[1] = ((y1 - y2) / r12d) * f;
    F[2] = ((z1 - z2) / r12d) * f;

    Pt[spring[i].imass1].sforce[0]-=F[0];
    Pt[spring[i].imass1].sforce[1]-=F[1];
    Pt[spring[i].imass1].sforce[2]-=F[2];

    Pt[spring[i].imass2].sforce[0]+=F[0];
    Pt[spring[i].imass2].sforce[1]+=F[1];
    Pt[spring[i].imass2].sforce[2]+=F[2];
}

```

```
}
```

Force Integration For Two-neighbourhood

```
void Integrate(double tPrev, double tCurr)
{
    int SCRSIZE =5;
    static double lastTime=tPrev;
    double currentTime=tCurr;

    double timePassed=currentTime-lastTime;

    Pt[massTouched].velocity = Pt[massTouched].velocity+
    (Pt[massTouched].sforce / Pt[massTouched].mass)* timePassed;
    hduVector3Dddrx = Pt[massTouched].velocity* timePassed;

    Pt[massTouched].position= Pt[massTouched].position + drx;
    // First ring of neighbour
    for(int k=0;k<Pt[massTouched].numNeighbor;k++)
    {
        int i = Pt[massTouched].Neighbour[k];
        Pt[i].velocity = Pt[i].velocity+ (Pt[i].sforce / Pt[i].mass)* timePassed;
        hduVector3Dddrx = Pt[i].velocity* timePassed;

        Pt[i].position= Pt[i].position + drx;
    }
    // Second ring of neighbours
    for( k=0;k<Pt[massTouched].numNNeighbor;k++)
    {

        int i = Pt[massTouched].NNeighbour[k];
        Pt[i].velocity = Pt[i].velocity+ (Pt[i].sforce / Pt[i].mass)* timePassed;
        hduVector3Dddrx = Pt[i].velocity* timePassed;

        Pt[i].position= Pt[i].position + drx;
    }
}
```

Volume Calculation Of The Soft Body

```
void calcPressure()
{
    double Volume = 0.0;
    hduVector3Dd Area;

    double Pressure = .0;
    hduVector3Dd Force ;

    for(int i = 0;i<numFace;i++)
    {

        int a,b,c;
        a = tris[i].vIndex[0];
        b = tris[i].vIndex[1];
        c = tris[i].vIndex[2];

        //      Volume= fabs(-Pt[c].Pos.x*   Pt[b].Pos.y*   Pt[a].Pos.z +
Pt[b].Pos.x *   Pt[c].Pos.y*   Pt[a].Pos.z+   Pt[c].Pos.x*   Pt[a].Pos.y*
Pt[b].Pos.z -   Pt[a].Pos.x*   Pt[c].Pos.y*   Pt[b].Pos.z-   Pt[b].Pos.x*
Pt[a].Pos.y*   Pt[c].Pos.z+   Pt[a].Pos.x*   Pt[b].Pos.y*   Pt[c].Pos.z)/6.0;

        Volume= fabs
(Pt[c].position[0]*Pt[b].position[1]*Pt[a].position[2]+Pt[b].position[0]*Pt[c].position[
1]*Pt[a].position[2]+Pt[c].position[0]*Pt[a].position[1]*Pt[b].position[2] -
Pt[a].position[0]*Pt[c].position[1]*Pt[b].position[2]-
Pt[b].position[0]*Pt[a].position[1]*Pt[c].position[2]+Pt[a].position[0]*Pt[b].position[
1]*Pt[c].position[2])/6.0;

        hduVector3Dd e1,e2;

        e1[0] = Pt[b].position[0]- Pt[a].position[0];
        e1[1] = Pt[b].position[1]- Pt[a].position[1];
        e1[2] = Pt[b].position[2]- Pt[a].position[2];

        e2[0] = Pt[c].position[0]- Pt[a].position[0];
        e2[1] = Pt[c].position[1]- Pt[a].position[1];
        e2[2] = Pt[c].position[2]- Pt[a].position[2];

        hduVector3Dd Normal;
        //Vector Normal;
        //      Normal = e1->crossProduct(e2);

        Normal[0] = (e1[1]*e2[2]) -(e1[2]*e2[1]);
        Normal[1] = (e1[2]*e2[0]) -(e1[0]*e2[2]);
        Normal[2] = (e1[0]*e2[1]) -(e1[1]*e2[0]);
```

```

//      Normal.x = (e1.y * e2.z) - (e1.z*e2.y);
//      Normal.y = (e1.z * e2.x) - (e1.x*e2.z);
//      Normal.z = (e1.x * e2.y) - (e1.y*e2.x);

Area[0] = abs((e1[1] * e2[2]) - (e1[2]*e2[1]))/2.0;
Area[1] = abs((e1[2] * e2[0]) - (e1[0]*e2[2]))/2.0;
Area[2] = abs((e1[0] * e2[1]) - (e1[1]*e2[0]))/2.0;
//For trisphere OBJ Object temperature is set as 0.5

//Pressure = (0.5*8.314*0.5)*(1.0/Volume);
Pressure = (0.05*8.314*.5)*(1.0/Volume);
Force[0] = Pressure*Normal[0]*Area[0];
Force[1] = Pressure*Normal[1]*Area[1];
Force[2] = Pressure*Normal[2]*Area[2];

//      Area.x = fabs((e1.y * e2.z) - (e1.z*e2.y))/2.0;
//      Area.y = fabs((e1.z * e2.x) - (e1.x*e2.z))/2.0;
//      Area.z = fabs((e1.x * e2.y) - (e1.y*e2.x))/2.0;
//double NormMag;

//NormMag =
sqrt((Normal.x*Normal.x)+(Normal.y*Normal.y)+(Normal.z*Normal.z));

//Area= fabs(Pt[a].Pos.y*Pt[b].Pos.z - Pt[b].Pos.y*Pt[a].Pos.z -
Pt[a].Pos.x*Pt[b].Pos.y+Pt[a].Pos.y*Pt[b].Pos.x)/2.0;

//      Pressure = (0.5*8.314*26)*(1.0/Volume);
//      Force.x = Pressure*Normal.x*Area.x;
//      Force.y = Pressure*Normal.y*Area.y;

//      Force.z = Pressure*Normal.z*Area.z;

Pt[a].sforce[0]+=Force[0]/3.0;
Pt[a].sforce[1]+=Force[1]/3.0;
Pt[a].sforce[2]+=Force[2]/3.0;

Pt[b].sforce[0]+=Force[0]/3.0;
Pt[b].sforce[1]+=Force[1]/3.0;
Pt[b].sforce[2]+=Force[2]/3.0;

Pt[c].sforce[0]+=Force[0]/3.0;
Pt[c].sforce[1]+=Force[1]/3.0;
Pt[c].sforce[2]+=Force[2]/3.0;

```

}

}

APPENDIX B

Experimental Results

Experiment 1:

1. Palpation Force (N)

1. First Session (S1)

| Trainee No | Prototype A | Prototype B | Prototype C |
|------------|-------------|-------------|-------------|
| 1 | 2.8 | 2.3 | 2.45 |
| 2 | 2.9 | 2.6 | 2.5 |
| 3 | 3.1 | 2.6 | 2.6 |
| 4 | 3.4 | 2.4 | 2.4 |
| 5 | 2.8 | 2.6 | 2.6 |
| 6 | 2.8 | 2.7 | 2.6 |
| 7 | 2.9 | 2.6 | 2.5 |
| 8 | 3.3 | 2.7 | 2.45 |
| 9 | 2.7 | 3 | 2.6 |
| 10 | 2.9 | 2.6 | 2.6 |

2. Second Session (S2)

| Trainee No | Prototype A | Prototype B | Prototype C |
|------------|-------------|-------------|-------------|
| 1 | 2.5 | 2.5 | 2.6 |
| 2 | 2.6 | 2.6 | 2.5 |
| 3 | 3 | 2.6 | 2.8 |
| 4 | 2.8 | 2.4 | 2.7 |
| 5 | 2.9 | 2.9 | 2.5 |
| 6 | 3 | 2.7 | 2.5 |
| 7 | 2.6 | 2.6 | 2.6 |
| 8 | 3.3 | 2.7 | 2.6 |
| 9 | 2.9 | 3 | 2.55 |
| 10 | 2.9 | 2.6 | 2.5 |

3. Third Session (S3)

| Trainee No | Prototype A | Prototype B | Prototype C |
|-------------------|--------------------|--------------------|--------------------|
| 1 | 3.4 | 2.59 | 2.4 |
| 2 | 2.5 | 2.6 | 2.5 |
| 3 | 2.8 | 2.7 | 2.5 |
| 4 | 2.8 | 2.5 | 2.6 |
| 5 | 2.8 | 2.7 | 2.6 |
| 6 | 2.7 | 2.5 | 2.5 |
| 7 | 2.6 | 2.55 | 2.4 |
| 8 | 3.3 | 2.5 | 2.5 |
| 9 | 2.7 | 2.5 | 2.7 |
| 10 | 2.8 | 2.6 | 2.5 |

4. Fourth Session (S4)

| Trainee No | Prototype A | Prototype B | Prototype C |
|-------------------|--------------------|--------------------|--------------------|
| 1 | 2.5 | 2.9 | 2.6 |
| 2 | 2.56 | 2.6 | 2.6 |
| 3 | 2.72 | 2.4 | 2.7 |
| 4 | 2.5 | 3 | 2.5 |
| 5 | 2.7 | 2.3 | 2.55 |
| 6 | 2.5 | 2.5 | 2.6 |
| 7 | 2.7 | 2.6 | 2.6 |
| 8 | 2.7 | 2.6 | 2.4 |
| 9 | 2.6 | 2.4 | 2.7 |
| 10 | 2.6 | 2.65 | 2.5 |

Experiment 2:

1. Palpation Force (N)

| Trainee No | Prototype A | Prototype B | Prototype C |
|------------|-------------|-------------|-------------|
| 1 | 2.8 | 2.7 | 2.45 |
| 2 | 3 | 3 | 2.5 |
| 3 | 3.1 | 2.6 | 2.6 |
| 4 | 3.4 | 2.4 | 2.4 |
| 5 | 2.8 | 2.6 | 2.6 |
| 6 | 2.8 | 2.8 | 2.6 |
| 7 | 2.9 | 2.6 | 2.5 |
| 8 | 3.3 | 2.9 | 2.45 |
| 9 | 2.8 | 2.6 | 2.6 |
| 10 | 2.9 | 2.5 | 2.6 |

2. Exploration Time (Seconds)

| Trainee No | Prototype A | Prototype B | Prototype C |
|------------|-------------|-------------|-------------|
| 1 | 40 | 35 | 30 |
| 2 | 48 | 29 | 26 |
| 3 | 43 | 30 | 28 |
| 4 | 40 | 29 | 30 |
| 5 | 45 | 28 | 22 |
| 6 | 55 | 23 | 24 |
| 7 | 57 | 30 | 28 |
| 8 | 35 | 26 | 23 |
| 9 | 41 | 34 | 32 |
| 10 | 30 | 35 | 29 |

Experiment 3:

1. Palpation Force (N)

| Student No | Prototype A | Prototype B | Prototype C |
|------------|-------------|-------------|-------------|
| 1 | 2.8 | 2.80 | 2.60 |
| 2 | 3 | 2.90 | 2.55 |
| 3 | 2.8 | 2.70 | 2.50 |
| 4 | 3.1 | 2.80 | 2.60 |
| 5 | 3.5 | 2.50 | 2.50 |
| 6 | 3 | 2.80 | 2.70 |
| 7 | 3.1 | 2.90 | 2.70 |
| 8 | 3.2 | 2.80 | 2.50 |
| 9 | 2.9 | 2.90 | 2.75 |
| 10 | 2.7 | 3.00 | 2.70 |

2. Exploration Time (Seconds)

| Student No | Prototype A | Prototype B | Prototype C |
|------------|-------------|-------------|-------------|
| 1 | 80 | 40 | 37 |
| 2 | 90 | 45 | 23 |
| 3 | 75 | 37 | 45 |
| 4 | 49 | 32 | 33 |
| 5 | 76 | 38 | 27 |
| 6 | 69 | 28 | 45 |
| 7 | 58 | 31 | 45 |
| 8 | 58 | 25 | 28 |
| 9 | 49 | 26 | 39 |
| 10 | 91 | 39 | 30 |

Experiment 4:

1. Palpation Force (N)

| Student No | Prototype C1 | Prototype C2 |
|------------|--------------|--------------|
| 1 | 2.60 | 2.70 |
| 2 | 2.60 | 2.40 |
| 3 | 2.70 | 2.60 |
| 4 | 2.40 | 2.50 |
| 5 | 2.50 | 2.50 |
| 6 | 2.80 | 2.70 |
| 7 | 2.70 | 2.70 |
| 8 | 2.60 | 2.70 |
| 9 | 2.80 | 2.75 |
| 10 | 2.65 | 2.60 |