Computer-Assisted Interactive Documentary and Performance Arts in Illimitable Space

Miao Song

A thesis

in

The Department

of

SIP-Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy

Concordia University

Montréal, Québec, Canada

December 2012

© Miao Song, 2012

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Miao Song

Entitled: Computer-Assisted Interactive Documentary and Performance Arts in Illimitable Space

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Danielle Bobker — Chair

Dr. Don Marinelli — External Examiner

Dr. David Howes — External to Program

Prof. Jason Lewis and Dr. Peter Rist — Examiner

Prof. Marielle Nitoslawska and Dr. Maureen Simmonds — Examiner

Dr. Peter Grogono — Thesis Supervisor

Approved by

Chair of Department or Graduate Program Director

December 2012

Dean of Faculty

Abstract

Computer-Assisted Interactive Documentary and Performance Arts in Illimitable Space

Miao Song, Ph.D.

Concordia University, 2012

This major component of the research described in this thesis is 3D computer graphics, specifically the realistic physics-based softbody simulation and haptic responsive environments. Minor components include advanced human-computer interaction environments, non-linear documentary storytelling, and theatre performance. The journey of this research has been unusual because it requires a researcher with solid knowledge and background in multiple disciplines; who also has to be creative and sensitive in order to combine the possible areas into a new research direction. Thus, we summarize the innovative research work surrounding the topic as "Computer-Assisted Interactive Documentary and Performance Arts in Illimitable Space". This work encompasses a lot of research performed in each of the disciplines. It focuses on the advanced computer graphics and emerges from experimental cinematic works and theatrical artistic practices. Some development content and installations are completed to prove and evaluate the described concepts and to be convincing.

More specifically, on one hand, the major research component includes the advanced rendering in real-time of an interactive physics-based softbody object simulation and visualization with OpenGL. Its immediate follow-up work in this thesis extends that system onto an artistic interactive jellyfish simulation controlled with advanced interaction devices, such as Falcon haptics. Some more advanced rendering techniques have been applied, such as stereoscopic effects, GLSL, LOD etc. in order to bring the CG objects to life and to increase the level of realism and speed.

On the other hand, the installation work, *Tangible Memories* transforms the award-winning personal documentary film *I Still Remember* [1], into a non-linear interactive audience-controlled piece using the new media technologies and devices. The audience and society have already appreciated the personal documentary *I Still Remember* with the social values resulting in portrayal of immigration, divorce, reunification with the family, and other memories. Turning it into an interactive new media work makes it a much more profound and sensory in-depth storytelling approach that can be educational as well as help the audience to feel the story by interacting with it and making it available via many media sources. Moreover, the audience's participations and feedback are themselves well-preserved in the new "memory bubbles", so that the same documentary project could be eternal and ever evolving, which may determine the concept of tomorrow's documentary film production. Additionally, another audacious approach extended from *Tangible Memories* installation is for theatrical practice with the same set of technical tools. Theater performers could use their body movements, gestures, and facial expressions to achieve the perceptual and emotional digital effects in sound and images dynamically.

To summarize, the resulting work involves not only artistic creativity, but solving or combining technological hurdles in motion tracking, pattern recognition, force feedback control, etc., with the available documentary footage on film, video, or images, and text via a variety of devices (input and output, projection, stereoscopic viewing) and programming, and installing all the needed interfaces such that it all works in real-time. Thus, the contribution to the knowledge advancement is in solving these interfacing problems and the real-time aspects of the interaction that have uses in film industry, fashion industry, new age interactive theatre, computer games, and web-based technologies and services for entertainment and education. It also includes building up on this experience to integrate Kinect- and haptic-based interaction, artistic scenery rendering, and other forms of control. This research work connects all the research disciplines, seemingly disjoint fields of research, such as computer graphics, documentary film, interactive media, and theatre performance together.

Acknowledgments

I am deeply grateful to my supervisory committee for their valuable advices and support as well in guiding this dissertation and providing all the necessary resources. Choosing an incredible committee might be the best decision I have made.

The first person I have to thank is my principal supervisor, Dr. Peter Grogono. Whenever I approach him with different questions in various disciplines, he could always give me an intelligent and insightful advice not only in Computer Science, but also in theatre, film, music, and interactive media fields. Later I discovered the reason why he is so knowledgeable and has very artistic disposition is because he had been working in electronic music, theatre performance, and even film production early in his career. Dr. Grogono is wise, senseful, and allowed me to be creative in this uncommon research journey with the freedom and space I needed. Moreover, he gives me a lot of encouragement, spiritual, and financial support. There is no doubt that I wouldn't be even on the right track in the research work and would have drowned from the disastrous family situation without his continuing trust, patience, and tolerance for the last several years.

I am grateful to Professor Jason Lewis, whom I knew the earliest among other professors. He always gives me incredible support and very valuable advice. I remember when I was in my undergraduate studies, we had quite friendly and pleasant conversations. From that time I was dedicated to achieve a higher degree, a master's, and now a doctoral degree as well.

I would like to show my appreciation to Professor Marielle Nitoslawska who agreed to be on my supervisory committee when I applied to the SIP doctoral program. It

was thanks to her trust that I could achieve my destination even though at that time when I proposed the research topic, “interactive documentary” really sounded a novel idea. I got a lot of inspirations from her own research work, every meeting and conversation with her.

I am very lucky to have met Dr. Maureen Simmonds three years ago and to have worked in her research lab. It might have been a completely different research experience if I did not have the chance to collaborate with her. Working with her not only broadened my view in the interdisciplinary research, but also was the first time that I could apply some academic knowledge to a real medical research case.

I am very grateful to Dr. Peter Rist, who arrived at my supervisory committee at the later stage of my research. He is so knowledgeable in Chinese cinema and animation and gave me the opportunity to rediscover the beauty of Chinese culture and arts, which I had forgotten about even after having grown up in China.

I was very honoured to be accepted to the SIP program and to have received the support from Dr. David Howes and Ms. Darlene Dubiel. Dr. Howes has been encouraging me and providing with all the possible solutions in front of me when I encountered a dilemma and had to make a hard decision in my interdisciplinary studies.

I believe that it is my fate to meet Dr. Don Marinelli who would become the turning point in my career. I thank him for inviting me to the ETC, Carnegie Mellon to see the “world”. The trip gave me tremendous experience not only with a new way of seeing things, but also a chance to demonstrate my own research projects to peers.

Also, a token of appreciation for helpful advice and comments go to professionals and collaborators on all the different multidisciplinary projects I worked with giving me an enriching experience. Thanks go out to Professors Jean-Claude Bustros, Alison Loader, Robert Reid, Ruru Ding, Guojun Ma, Thomas Waugh, Alice Jim, and Tongdao Zhang. Many thanks to the Montreal Herstory Theatre Performance Group artists—it was a soul exchange experience to collaborate with all of them.

I would express my gratitude to Marco Luna who gave me very precious and valuable advice in documentary filmmaking. Without his encouragement, I wouldn't be brave enough to make the short documentary *I Still Remember*. Thanks to the curator, Ms. Annie Briard who selected *I Still Remember* at the HTMlles festival. I also thank the BJIFF "See the World through Films" Competition Committee to have given me the best short documentary award. I also would like to acknowledge filmmakers Mr. Paul Carvalho and Ms. Beverly Shaffer to collaborate with me on the future fiction film. Many thanks to Joel Taylor, Mark Baehr, Momoko Allard, Phil Hawes from Hexagram and CDA who gave me a lot of technical support. Thanks for their tolerance when every time I had a technical emergency and asked their help at the last minute.

Thanks to Ms. Catherine LeBel and the CSLP team. Without her trust and encouragement, I would not have been in the position to continue my work and studies. Moreover, I need to thank my lawyer Me Raphael Levy, my daughter's lawyer Me Beatrice Clement, and Judge Pierre Béliveau, to having saved a desperate woman's and a poor child's lives. I am truly and deeply indebted to so many friends that helped and saved me and my daughter during our family divorce crisis, and supported and cared about us. They are Xuemei Ye and her family, Yichuan Zhang and his family, Jin Cao and her family, Li Ma and her family, Ying Chen and her family, Yi Zhao and her family, Shulei Guo, Mi Zhou, Helen Wang, Debbie Davis, Sharon Nelson and others whom I could not name them all.

I would like to express my appreciation to my family to support me to bring this project to completion. Thanks to my daughter Deschanel, for her inspiration, collaboration, artistic creativity, and contributions to my life and research work. She shared truthful feelings with me so that we together made *I Still Remember*. Thanks to my son, Timothy, who brings me a lot of happiness, hopes, and courage. Many thanks to my brother Liu Song, for his emotional and financial support to me and my family. I cannot thank enough and am forever indebted to my parents, Lin Song and Fang Liu, for their invaluable support to my life and my family. It is because they

have been helping me to bring up my two kids, I would be able to stay in school late every evening and concentrate to my studies. I need to thank to my husband, Serguei Mokhov, for never leaving me (the poor soul) behind no matter what happened for so many years. Thanks to his love, clinging together in times of trouble, and to his support and belief that I could eventually complete my studies.

In the end, I would like to acknowledge the SIP program, the Department of Computer Science and Software Engineering, Faculty of Engineering and Computer Science, Hexagram Concordia, and Graduate School at Concordia University as well as FQRSC doctoral scholarship, CCSEP scholarship, McGill University, CSLP Concordia University, the Central Academy of Drama (China), China Scholarship Council, for various financial, space, hardware, and software resources provided to make this work possible.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

This chapter begins by stating the research question this thesis answers in Section 1.1. Motivation for this research crossing multiple disciplines is detailed in Section 1.2. Section 1.3 reports on the creative and research achievements, the process, and the thesis scope. Section 1.4 outlines the remaining content of this thesis.

1.1 Summary Statement of Research Questions

Theatre, Film, Television: from the very ancient art form with thousands of years of history to the recent main new media sources of entertainment in the last several decades, whether played by a live performance or a pre-recorded program, all these media have something in common: performer, audience, and space. These media have played a pivotal role in the socialization, civilization, education, and human connection. Moreover, in each of their traditional contexts, the performance is typically linear, the actor is scripted, the audience is observationally passive, and the space is physically constrained.

After the appearance of the first electronic computers developed in the mid-20th century, accompanied by the arrival of computer networking and the Internet, technology has dramatically evolved. Consequently, new media products and devices, such as

digital theatre, video games, Internet TV, computer-graphics-based animation documentary film, "smart" mobile phones, virtual/augmented reality, stereoscopic movies, web based docudrama, building projection, etc., gave audiences more participatory power to the control of artistic works, thus creating more new challenges to the production team and artists.

Today, the new computer technology has already affected the well-established traditional art form, especially in the aspects of actors' theatre performance, filmmaker's production decisions, and the audience's participation, and even in the redefinition of the nature and boundaries of "space" in the digital era.

Will virtual reality (VR), computer generated images (CG) and environment destroy the "holy" aspect in theatre or, conversely, improve its representation? Will the new digital technology subvert the definition of realism of documentary film arts? What does the computation do to change the relation or difference between theatrical and cinematic experiences? What will the next generation interactive theatre and documentary film be like? What unique experience could the live event bring to audience other than pre-recorded media in consideration of HCI? What have the theatre and documentary been? How could we redefine them today in the digital age?

The purpose of this research is to investigate an innovation on how multidisciplinary aspects, such as computer graphics, interactive media, cinema montage, and theatre performance could coherently form and blend in harmony.

1.2 Motivation for Multi-, Inter-, and Transdisciplinary Research Interests

When I first heard about SIP, the Special Individualized Program at Concordia University, which allows students to pursue an interdisciplinary research degree, I was extremely happy that my artistic background I acquired in China would be useful at last! Even though I had already completed another bachelor's and a master's degree in Computer Science at Concordia University, it had not been an easy transition for

me to switch to Computer Science given my background in China was in theatre performance, and following that I became a TV journalist having graduated from Shandong University of Arts in 1997.

Thus, my research gradually and incrementally focuses on the mentioned aspects and includes the following: enhanced softbody simulation and specialist training in virtual reality, interactive media, documentary film production, and theatre arts as the applications of my research. None of these were offered by a single specific Faculty or Department at the PhD level making the SIP (Special Individualized Program) unique and appealing to blend the multiple disciplines throughout my research. I have worked on my proposed research areas and made some contributions and achievements in documentary film making, theatre performance, and computer graphics. I am especially interested in how humans interact with computers and computer-generated graphics not only in science and everyday life, but also in artistic creativity.

It has been already a challenge to work on interdisciplinary research that usually requires collaborations of experts and professionals from specified areas. They have to understand the concepts underlying a discipline other than their own and find a common language to communicate ideas. A lot of courage is required to face up to the highest degree in interdisciplinary research, which becomes more and more a buzzword these days, and have one person with all the related skills and knowledge. Additional difficulty arises when that person's English is not their first language. However, there are also some advantages, such as the innovative ideas could be more naturally, unitedly, and efficiently developed because everything is deeply rooted within *myself*. Meanwhile, the SIP gave me a great opportunity to work with a team of highly qualified professors in computer science, film studies and production, virtual reality, theatre and digital media arts, who have strongly showed their enthusiasm in supporting and guiding me throughout my endeavor.

Regardless of the actual contents of the research itself, my whole research experience shaped up this work from two fronts: I have been studying a technical degree after originally being trained as an artist. Then as a result I could combine the

technical knowledge back to the artistic creativity. Thus, this work has become very valuable to me and my life personally, I believe the society as well, as a pioneer of future interdisciplinary researchers. I am quite confident that standing in between science and art, western and eastern, with hard work, passion, experience, and knowledge definitely makes a great contribution to the rapidly developing era.

To me, a *multidisciplinary study* means that research could be done in a mixture of disciplines, in each of which there exist their own methodologies. These disciplines don't typically integrate or interface with each other. Additionally, there are two other terms, *interdisciplinary* and *transdisciplinary* that have also been increasingly used these years. The former combines two or more academic fields into one single discipline; the later crosses many disciplinary boundaries to create a holistic approach, as to quote:

> *"Multidisciplinarity draws on knowledge from different disciplines but, stays within their boundaries. Interdisciplinarity analyzes, synthesizes and harmonizes links between disciplines into a coordinated and coherent whole. Transdisciplinarity integrates the natural, social and health sciences in a humanities context, and transcends their traditional boundaries."* [4]

My studies started from a multidisciplinary viewpoint and have been expanded to interdisciplinary and transdisciplinary research methods. First of all, the proposed research area of focus for the SIP doctoral research, "Computer-Assisted Interactive Documentary and Performance Arts in Illimitable Space" is leaning towards the soft-body simulation (inherited from my master's research topic [5]) in haptic-enabled environment and specialist-training fused together. In that work I proposed a variety of *inexpensive* techniques for human-computer interaction in various domains as well as advanced 3D modeling and animation in a rich spectrum of application domains. Thus, I explored options to come up with a set of tools, techniques, and methodologies that use haptics- and other-interface-enabled devices such that the

toolbox/framework we develop in this work is affordable and suitable with the least number of modifications to the respective application domains.

Secondly, the thesis work also has promised a research and development on a new method and approach on how technical invention effects artistic production, such as documentary film making and theatre performance. It demonstrates how important the new technology could help develop the contemporary studio art works and satisfy artist's wild imagination compared to the traditional art forms.

1.3 Research, Creation, and Development Process Overview

I can still remember when the first time I proposed this interdisciplinary research, the idea seemed very novel but uncertain. After the past several years of research and study in Computer Graphics (CG), Virtual Reality (VR), Interactive Digital Media, Theatre Performance, and Documentary Film Production, today, I could conclude the research work was worthwhile with some convincing evidence. This is manifested through the publications, film screening, and media installations.

The SIP program required a diverse range of courses taken from different disciplines. From the courses offered from Concordia University's Faculty of Engineering and Computer Science, such as "Advanced Rendering and Animation", "Computer Graphics for Television Production" (offered by my principle supervisor Dr. Peter Grogono), and "3D Graphics and Computer Animation" (offered by Dr. Nizar Bouguila).

I acquired deeper knowledge and understanding in theory of algorithms for advanced computer graphics and animation, advanced rendering, and stereoscopic techniques. The course co-offered by Concordia's same faculty and McGill University's Faculty of Medicine, School Of Physical and Occupational Therapy, "Computer Graphics in Virtual Realities" (co-supervised by Drs. Peter Grogono and Maureen

Simmonds), gave me the precious opportunity to access to fancy motion capture (Mo-Cap) system, advanced virtual reality applications, augmented reality equipment, and to discover a closer relationship among computer technology and its application to medical research.

The courses offered by Concordia University Mel Hoppenheim School of Cinema, "Expanded Documentary" (given by Dr. Marielle Nitoslawska) and "PhD Film Studies Seminar in Film and Moving Image History" (taught by Dr. Thomas Waugh), supplemented my needs in theoretical and practical applications to documentary film and their possible expansion resulted from the rapidly developing computer technology. I started to become aware that traditional documentary form has been more and more affected by computer-generated imagery and focused on how human beings' stories could transform from the traditional linear storytelling to non-linear interactive approach.

I was excited to find out that theatre arts, which have always been a love knot to me, as an art form, actually is extremely rich and powerful through "Enchantment, Matter, and Topological Media" (offered by Dr. Sha Xinwei) at Concordia's Design and Computation Arts Department. Another new discovery will be some research of arts in Asia, more explicitly, the film and animation arts in China. Moreover, I was very impressed by Dr. Peter Rist's knowledge of Asian cinema and Asian traditional theatre. In his course, "Film Studies in Chinese Cinema and Water Ink Animation Film", offered by Film Study Program at Concordia University, we reviewed famous Chinese feature films and international winning animated films and their animation techniques. Furthermore, from September 2011 to July 2012, I have been awarded the CCSEP (Canada-China Scholar's Exchange Program) scholarship and was studying "Traditional Chinese Theatre and its History" (by Dr. Guojun Ma) at Theatre Literature Department and "Theatre Directing, from Ideas to Performances" (with Dr. Ruru Ding) at Theatre Direction Department of the Central Academy of Drama in Beijing, China.

It was a great pleasure to rediscover theatre from a brand new angle and gain

more knowledge in Chinese traditional art form and the Contemporary arts after many years computer science studies and related technical training. Additionally, I have also audited other courses related to my research area, such as Dr. Robert Reid's theatre class in the Theatre Department and Dr. Jean-Claude Bustros's "Film Production" course, and had very deep discussion with them about my proposed research.

Additionally, in the last three years, I also obtained various opportunities to explore and work on different projects with professors from many research disciplines which lasted from several weeks to many months. From collaborations at close quarters with the professors, I derived a lot of first-hand experience from ideas generated from scratch that led into productions, the team cooperation, and their critical and creative approach to the artistic and research work. More importantly, I have also witnessed several excellent examples in the interdisciplinary research, and have been fascinated about the outcome of the combinations.

The very early collaboration was with Dr. Alice Ming Wai Jim, Concordia Art History Professor, whose research focuses on media arts, spatial culture and contemporary Asian and Asian-Canadian art. From the research work done in her project, *Augmented Reality in Asian Contemporary Arts*, I realized that there have been some emerging Asian media artists already experiencing virtual and augmented reality into their creative installations. The short research assistance experience in Dr. Jason Lewis's project *CitySpeak* [6] and Dr. Marielle Nitoslawska's and Alison Reiko Loader's *The Grey Nuns Project* gave ideas about how the programming and computer graphics could be applied to interactive media installation and film production projects. In the *3D Maya Stereoscopic Plugin* [7, 8] project I collaborated with Professor Alison R. Loader; this project was the first that tested my multidisciplinary background because it used the OpenGL and MEL programming skills to implement a stereoscopic tool for 3D Maya [9] software interface in order to help a 3D animation filmmaker to preview stereoscopic effects in realtime prior to final rendering. The *Interactive Cinema* project led by Dr. Jean-Claude Bustros opened up

my mind about how traditional film production could be benefited from interactive media, how the interactive cinema/film could be. I have not only participated in the implementation work as a programmer, but also got close look at the artistic design of their interactive cinema system. The two years of working experience as a research assistant at the "Virtual Reality in Pain, Mind and Movement Lab" [10] originally led by Dr. Maureen Simmonds at McGill University, was extremely important to me and my research as well. From there, I could have access to a very modern MoCap system, sensor-enabled treadmill, HMD (Head Mounted Display) equipment and the software system, where one could implement applications for medical research (and of course entertainment). Moreover, the experience of being a teaching and research assistant and lab instructor in Computer Graphics, Animation for Video Games, and various courses was very instrumental to keep up my knowledge in computer graphics and video games development up-to-date.

1.3.1 Achievements and Contributions

Throughout the past years I have been active in various areas of research in terms of actual investigation, publications, lab research work, and the like. I have published or have accepted-for-publication several related conference papers and posters on each or combined domains included in my proposed research, such as real-time physical based animation, conceptual work on haptic devices and interactive cinema, stereoscopic plug-in and its applications in medical and cinematic research. There were also debate of positions when I presented my work and research results at conferences and public presentations. Moreover, I got my art works and documentary film presented and showcased in different conferences and festivals respectively. See Section 6.3 for a detailed list.

In my primary research direction in computer graphics, I have made some achievements in softbody simulation system framework and 3D game development as well [11, 12, 13, 7, 14]. Throughout the duration of this aspect the knowledge of how to adapt inexpensive haptic devices, algorithms, techniques, and methodologies to

new human computer interaction in order for it to be more accessible to digital media artists, augmented reality researchers, and computer scientists alike to further research in the area, and make it accessible to audiences at home and at school.

Since I have had more opportunities in interdisciplinary studies, including the mentioned expanded documentary film studies, Asian cinema studies, the role of computer graphics in TV and theatre productions, there also have been some achievements in theoretical research and studio works in cinema/documentary film, interactive media [15, 16, 17, 18, 19, 20].

As a studio component of my research, a very recent award is for my documentary film, *I Still Remember*, included in the top ten video art work showcase [16] at HTMlles 2010 festival and, moreover, it won the best short documentary film [1] at 1st Beijing International Film Festival (BJIFF). One of the sub-directions of my SIP PhD research topic centered in Interactive Documentary. The first OpenGL prototype of an interactive media installation, *Tangible Memories*, which is based on my short documentary film, has been implemented and demoed at Concordia Humanities student conference and was also presented at the ICEC 2011 conference in October 2011 [15].

Additionally, I got involved with Dr. Simmonds and the previously mentioned supported course and her McGill lab on virtual reality techniques as a research assistant and a lab technician of the Motek equipment and a head-mounted display VR systems in pain, movement, and VR graphics research. This overall research experience covers, at various levels, conducting a literature research, collection and summarization of the sensor measurements and results, and developing a testable research question and research proposals. It also included development of basic proof-of-concept prototype walk-through game to demonstrate the idea on human-computer interactivity, learn the API, as well as make a more real-life application for patient testing and operating of the VR lab within the context of the research [12]. In the end, I was able to apply back that experience directly to the studies and research in computer graphics and interactive techniques and documentary [12].

Overall, these years of working experience in multidisciplinary and interdisciplinary fields enables me to think critically and open-mindedly on how future cinema and theatre lead the direction of the new media. I am able not only to be very creative in art production, but also to handle the associated technical difficulties. I could see how mature now and how much I have progressed as a researcher compared to three years ago. The knowledge earned from multidisciplinary research is all blended in together in my thoughts. Such knowledge acquired from making the installations, setups, software development and re-use, and the mathematics behind is a contribution made available as a guideline of how to make such setups, the process, and select the appropriate tools and techniques [19]. Such contributions are important in the age of digital media not only for artists and small audiences, but also for educational purposes of children through interaction, as it is widely known that children develop better their abilities when one interacts with them more. Telling interactive stories, teaching decision making in various life situations are specifically important applications of this research [19].

1.3.2 Process

My creative research, development, and production process is thematically depicted and described in this section following one of the main themes of "memory bubbles". Conceptually, the five high-level bubbles in Figure 1 representing each of the elements of the research corresponding to these themes.

Bubble 1 in Figure 1 represents the real-time softbody simulation. This project is physical based simulation to render 1D, 2D, and 3D 2- and 3-layer elastic objects and some applications of it [14]. Bubble 2 represents the Virtual Reality Lab for pain, mind, and movement research that was originally founded by Dr. Simmonds from McGill University, Faculty of Medicine. The lab contained a motion capture (MoCap) system for motion data acquisition, a treadmill as a part of the interactive environment, and an arc screen for realistic computer graphics projective display [10].

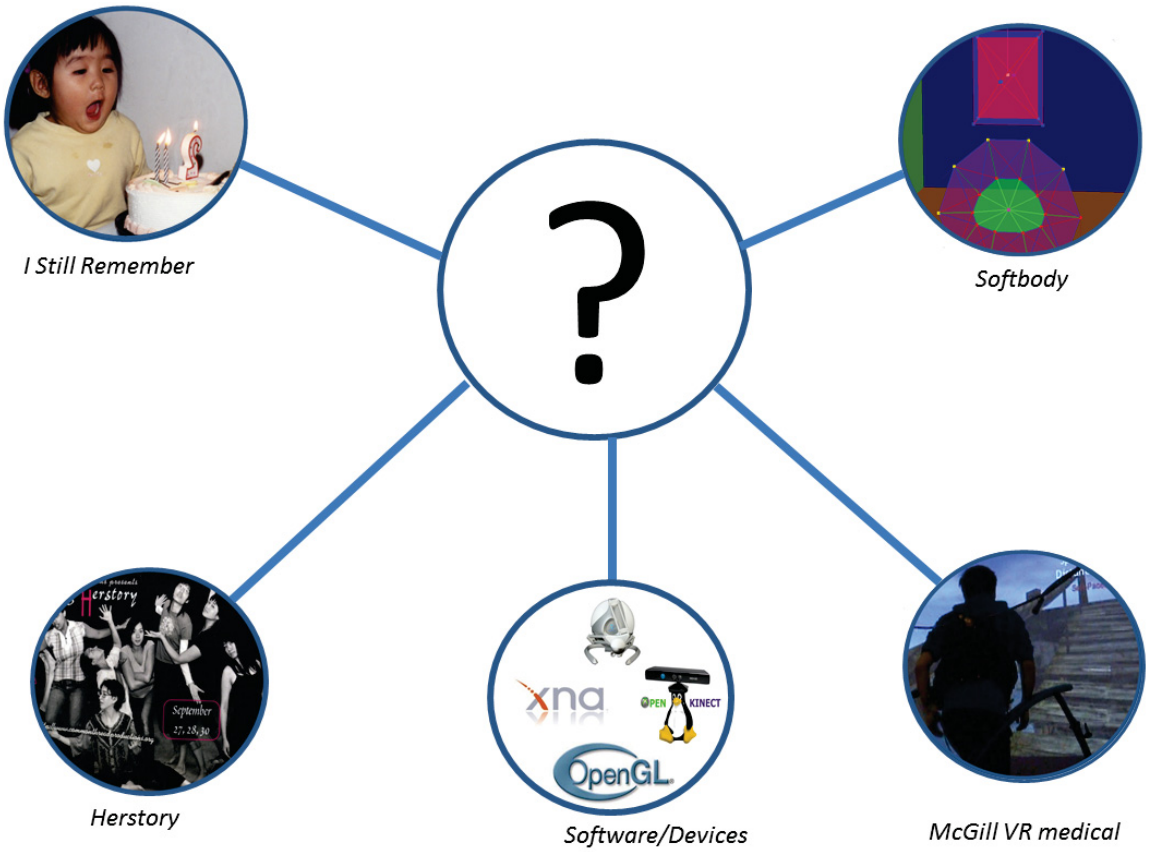


Figure 1: Overall Major Research Points Converging

Bubble 3 represents affordable devices and software APIs used for application implementation. The haptic device acts not only as an interaction tool with computer generated virtual environment, but also a resource of responsive force and feedback of simulated CG objects back into the real. It gives more realistic and advanced interactive perceptual stimulation between a human and a computer. Other possible consumer devices for human-computer interaction are the now well-known Kinect, Wii and Xbox controller and APIs are OpenGL and XNA. Bubble 4 represents *Unraveling Her Story*, which was a theatre performance piece co-produced with a group of female artists including myself in 2007 [21]. Six of us were also the script writers and directors for our own story pieces, which were based on the actresses' ancestors or own experience. Bubble 5 represents the *I Still Remember* documentary film based on my personal story [1]. It is also a project about memories, collected between me and

my daughter. The story was told by the little girl about what she could remember from when she was small. The bubble in the middle is the the research question (detailed in Section 1.1) combining computer graphics, virtual reality, interactive media, theatre performance, responsive environments, and documentary production.

In subsequent figures, specifically in Figure 2, Figure 3, Figure 4, Figure 5, Figure 6, Figure 7, and Figure 8, the process is refined and connections are drawn between media, performance, and technologies.

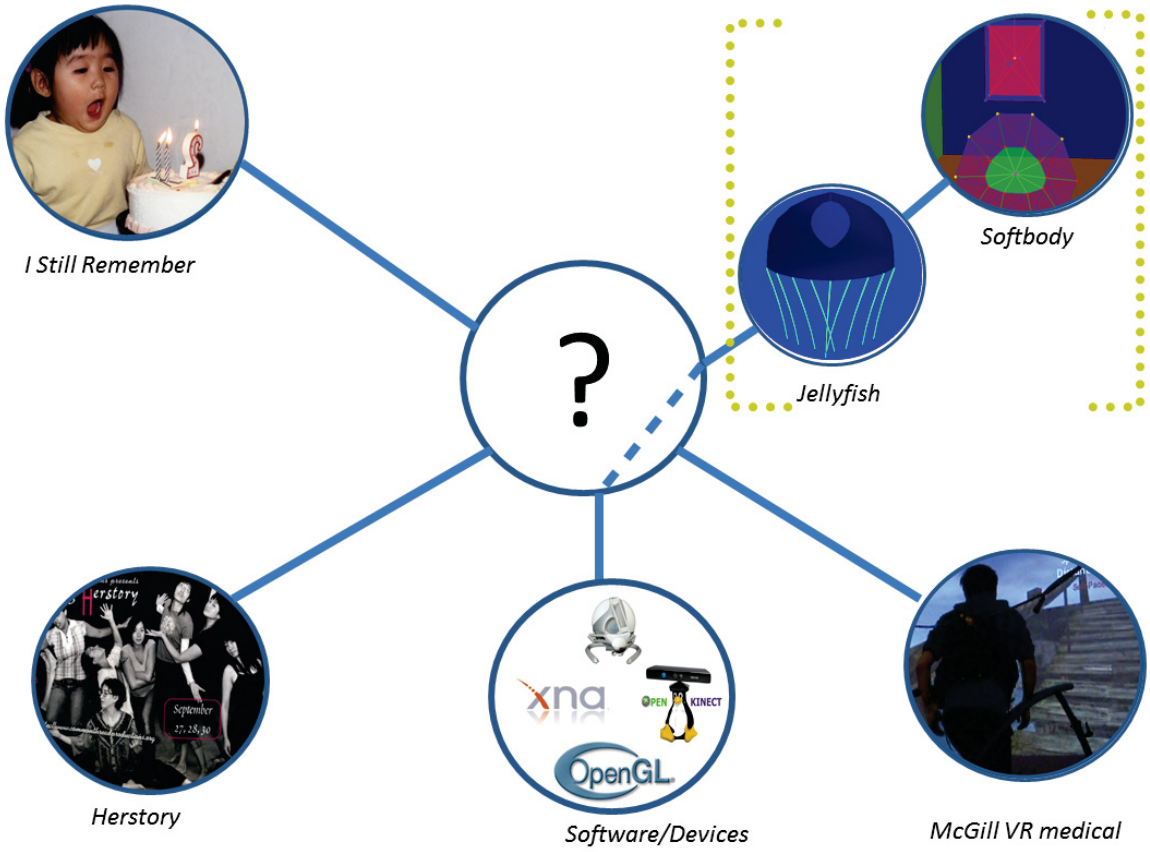


Figure 2: Overall Major Research Points Converging Progression 1

In Figure 2 the earlier contribution of the Softbody Simulation System is extended and applied to the modeling and interactive physical based animation of the jellyfish with added Falcon haptic touch to it.

The concepts of VR interaction and computer graphics are expanded into an interactive documentary depicted by the additional bubble in Figure 3. In the process of fusing the linear documentary footage, CG and VR environments, connections are

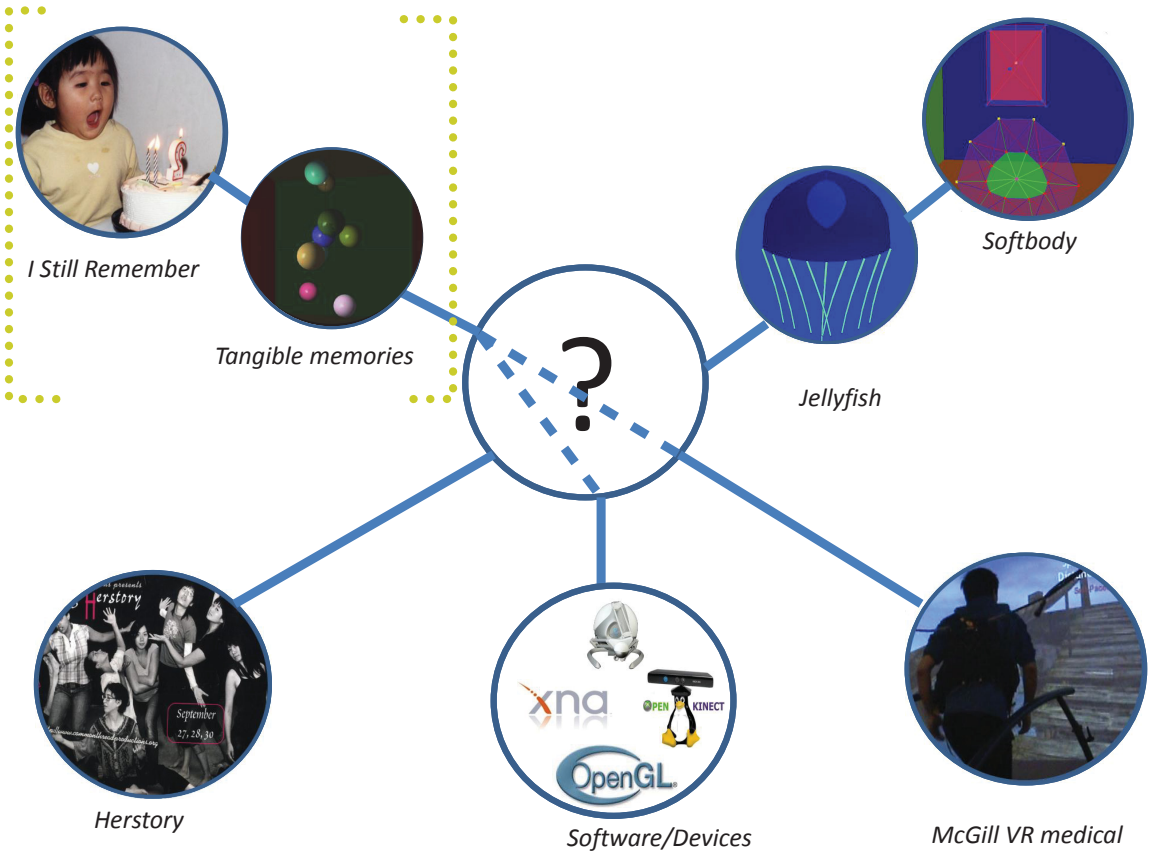


Figure 3: Overall Major Research Points Converging Progression 2

made to make the new media interactive documentary *Tangible Memories* with the memory bubbles (see Section 1.3.3.2) that we detail in this thesis in Chapter 4.

In Figure 4, the bubble of theatrical performance, *Illimitable Performance Space* inspired by previous theatre production *Herstory*, connects to the same interactive computer graphics and VR techniques. This innovation gives the audience not only unlimited imagination space, but also actors real improvisation experience by interacting with computer generated environment.

Figure 5 indicates that the VR system and its applications could be used not only for medical research purpose, but also for education (in progress, outside the scope of this thesis, see Section 1.3.3).

In Figure 6 the process expands further from the short linear documentary *I Still Remember* to a feature length fiction film "Snowflake" (in progress, also outside the

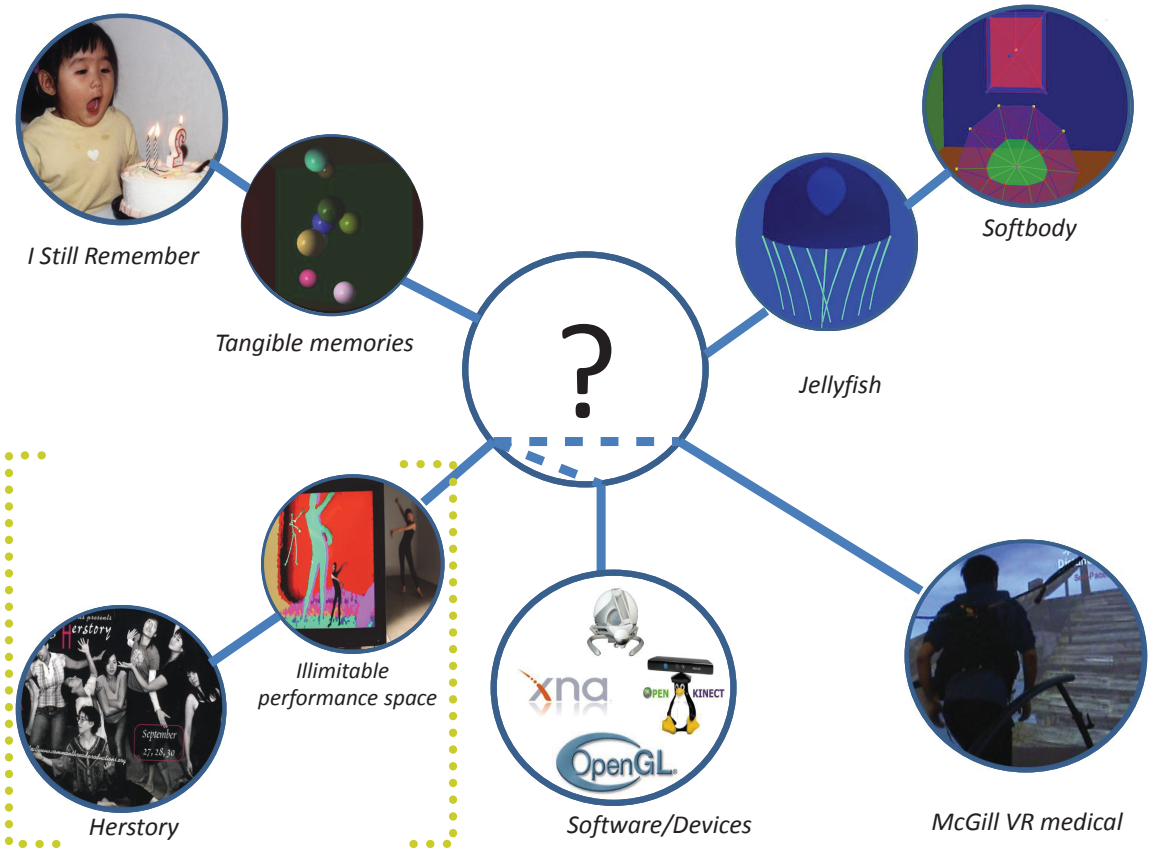


Figure 4: Overall Major Research Points Converging Progression 3

scope of this thesis).

At the same time, in Figure 7 the additional bubble for the projected *Zooming Through the Generations* interactive documentary (which is outside of the scope of this thesis), connects to the the same interactive computer graphics and VR techniques, theatrical performance of real people portraying scenes footage for which can no longer be otherwise recorded augmented with the audience participation.

Figure 8 is the future work in computer graphics research direction. Physical based softbody/jellyfish simulation could be even applied onto skeleton-based characters in order to make their animation more realistic.

Thus the multidisciplinary nature of the process of connecting CG, art, performance and production (Figure 8) pose some interfacing challenges, which this thesis addresses in part and lays ground for the ongoing and future work.

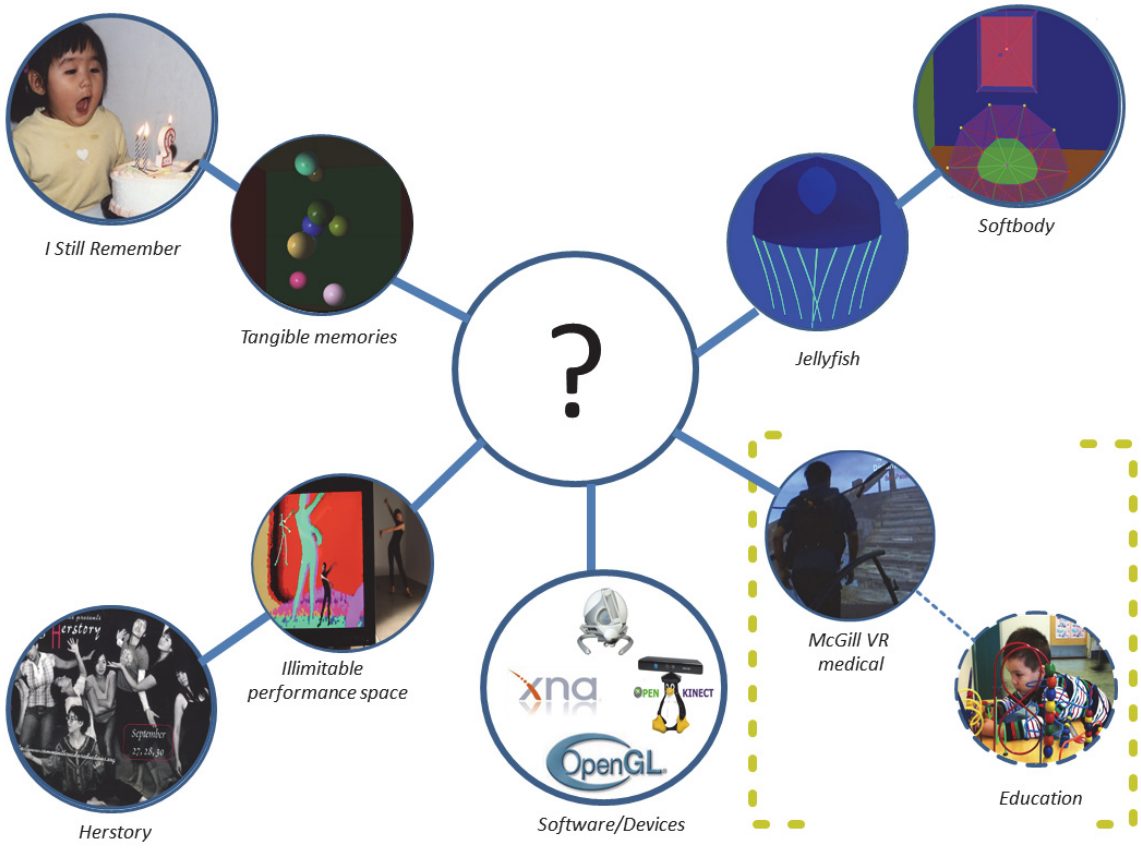


Figure 5: Overall Major Research Points Converging Progression 4

1.3.3 Scope

Here we declare limitations put on the research question based on the feasibility with the time constraints (how much of the proposed work is intended to be done and how much is intended to be left for the future).

1.3.3.1 Softbody Simulation and Jellyfish

This part focuses on the analysis and review of the relevant advanced rendering and animation techniques in computer graphics used for 3D games, some of which we juxtapose with the physically based softbody simulation framework we designed and implemented earlier. We also discuss such implementation results and the future directions. The techniques we touch upon are various algorithms for animation and physical based modeling as well as GPU programming with OpenGL. We break down

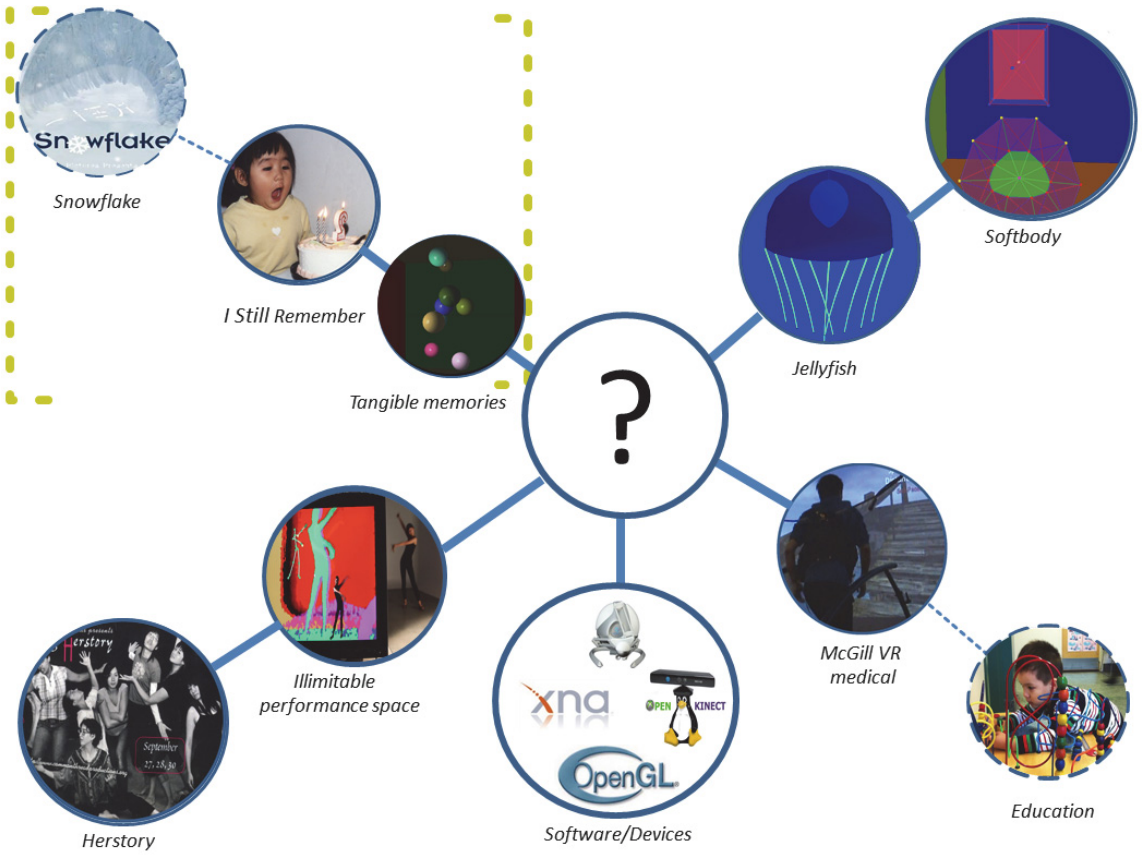


Figure 6: Overall Major Research Points Converging Progression 5

our contribution into two main parts: the analysis and review of the applicability of the techniques to softbody rendering and animation with the summary of the results to date [22].

Then, we derive a set of requirements from the detailed case study of the Softbody Simulation System design and specification in an attempt to come up with the requirements that all similar systems need to have and to further generalize our findings to other physical-based simulation systems involving real-time computer graphics [23].

Moreover, we describe details of modeling and implementation of an interactive jellyfish simulation using OpenGL and the Softbody Simulation System as a result. We outline our process, setup, planned exhibition installation, and the difficulties encountered and the ways we solve some and propose to solve some other difficulties in this work as a possible guideline for others doing similar installation with some

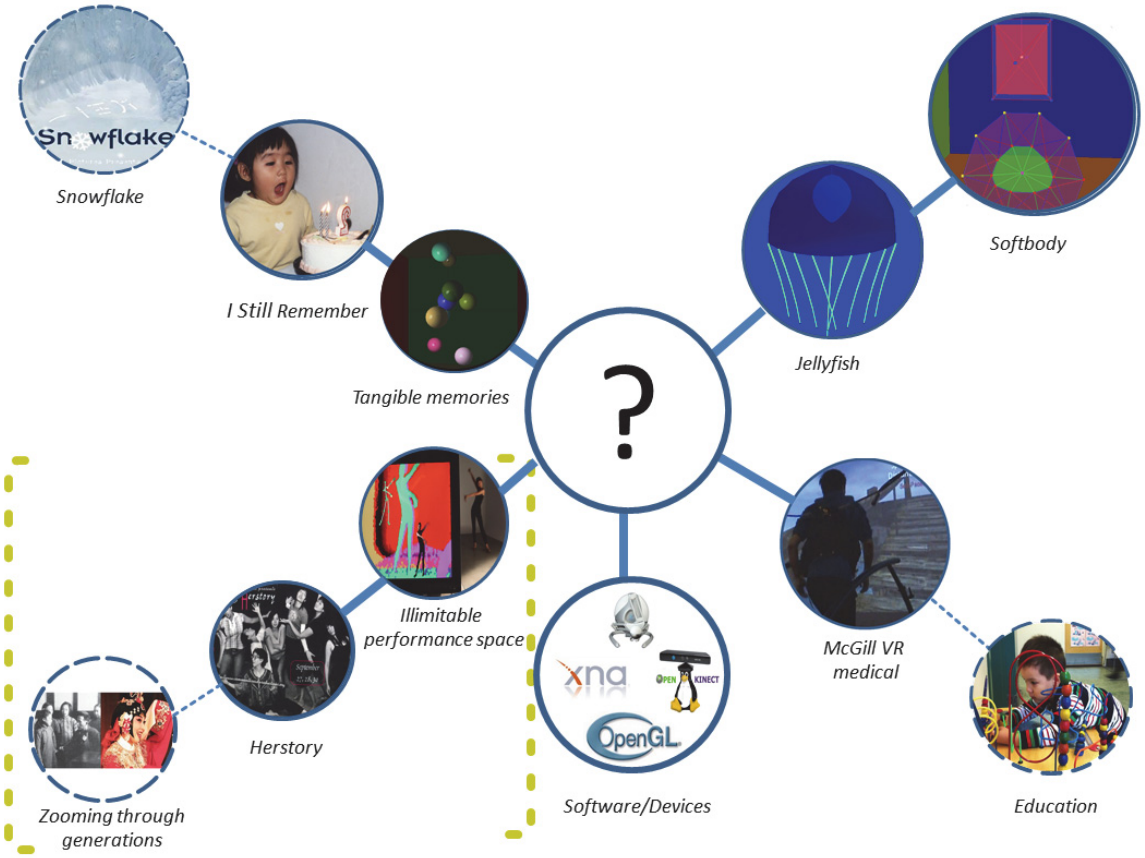


Figure 7: Overall Major Research Points Converging Progression 6

educational value. We review a number of related work items considered in the design and implementation of this work as well as future plans and considerations [11].

1.3.3.2 Interactive Documentary

It is or at least has been difficult in general to make documentary interactive. The interactivity in documentaries converges from several streams of interactive media, such as interactive TV, the Internet, computer graphics, interactive cinema, computer games, and associated realities (virtual, augmented, etc.) and their associated interaction devices. In our research we look into how common interactive TV and film as well as computer graphics techniques may help with the interactive documentary production and what it means to interact with a documentary. We also review

Figure 8: Overall Major Research Points Converging Progression 7

some related literature and existing documentary projects and the logic and techniques behind them. Furthermore, by comparing some documentary projects which are about similar topics, with linear and nonlinear storytelling approach, we present our position on future documentary production, and the importance of interactivity of documentary film making.

Making the *I Still Remember* documentary's floating memory bubbles interactive with audience's participation first used a cross-platform OpenGL (aimed at later enhancements with haptics). We describe a simple process of making a passive documentary interactive using available tools and preserving the aesthetic and emotional appeal. Moreover, careful comparison of the linear conventional film and its nonlinear narrative version with audiences' body movement involvement may give answers to some of the artists who are still hesitant to adapt their projects to the dramatically

developed new technology [15].

The interactive (non-linear) documentary story-telling approach explored here is somewhat different from what was known by interactive documentary as a "web documentary" of database-based compiled/submitted footage played in the order desired by the audience or merged into games [24, 25, 26, 27]. Instead, we render the documentary footage in a graphics environment inside bubbles, like "memory bubbles", based on a personal story (including the concept of the "memory bubbles" themselves) which can be called out by their color via voice or a key/click.

We rendered the documentary in two instances, one is based on OpenGL [15] and the newly redone one is based on XNA and Kinect.

The OpenGL version features the AVI documentary footage, photographs, or tidgets, and the bubbles and GLUI and mouse-based interaction.

The XNA+Kinect version has more advanced bubble effects, speech recognition, and Kinect interaction on top of the keyboard-based version. This later version has various perspectives being rendered for projection onto various surfaces in the performative blackbox environment. Two of the "fancy" soap bubbles can be hand-controlled. The other bubbles are split into media-containing ones (that can be called out by their color) and soap bubbles. The content is our own production including the short documentary film. The bubbles float in 3D space and collide with each other.

1.3.3.3 Illimitable Performative Space Installation

Illimitable performance space installation, while bounded in a particular environment, such as a theatrical black box, removes the limits of participation of local and remote audience. In the first prototype installation it involves real-time animation with audience participation as well as effects of fancy bubble animation along with background live feed video projected on the floor (from the ceiling). With Kinect we work with color and depth streams for fascinating motion tracking effects of the

main performer—a dancer wearing traditional Chinese dance long sleeves. The bubbles move along with the dancer movements to different directions. Other cameras and possibly Kinect sensors observe the audience who are also projected on different displays, walls, etc. The dancer and audience video is preserved from one session into a new bubble and another and is "memorized" and documented and played back along with the subsequent performances. This way each performance is contributed to and enables multiple dancers projections at the same time. A similar dance or audience participation can be submitted as well via the Internet and played back on the Internet creating a global show.

1.4 Organization

This chapter starts with the introduction and overview of my cross disciplinary background, the overall multidisciplinary research overview, and proposed research questions in Section 1.3. Chapter 2 describes all the necessary background and related work. Specifically, we briefly review the related work and literature on various topics of interactive cinema, documentary, and our position about it as well as new media and interaction research in TV, cinema, and documentary in general. Chapter 3 details the methodology of real-time softbody object simulation applied to a jellyfish modeling and animation. Chapter 4 presents an interactive documentary approach based on the application of some very common computer graphics tools and techniques and the personal linear *I Still Remember* documentary [1] told non-linearly inside "memory bubbles" as *Tangible Memories* with advanced user interaction features. Chapter 5 introduces the design and implementation of an illimitable performative space and installation. Chapter 6 concludes with the contributions, advantages, and limitations of our approach and an outline of the near and longer term future work. Appendix A details some operational instructions and user manuals on the developed systems.

Fair use of imagery. The author Song declares the use of screen captures and the related imagery produced by others in this academic work are for the illustrative reasons under the fair use rationale (similar to Wikipedia's [28]) where such work may still be under copyright and the author does not make any claims of ownership or authorship of the said images. The screen captures and the like are attributed in the text and are necessary to illustrate the points presented in this academic research work. A certain number of images from the cited documentary films are already in the public domain or have compatible documentation licensing.

Chapter 2

Background and Related Work

In this chapter, some historical aspects of the topics related to the present research are reviewed to give deeper context into the foundations of this dissertation's work. These related works involve computer animation and advanced rendering (Section 2.1), interactive media and documentary film production (Section 2.2), and performance arts and theatre production (Section 2.3). The related technology used in or considered for this research is described in Section 2.4.

More specifically, the multidisciplinary background described in this chapter is of relevance and the source of inspiration for the chapters that follow:

- many of the the computer graphics, animation, and rendering techniques described further are employed primarily in Chapter 3 (*Toward Realtime Jellyfish Simulation in Computer Graphics*) as well as throughout the other main contribution chapters (Chapter 4 and Chapter 5)

- the interactive media and documentary film production background serves as a source of inspiration and analysis of the novelty needed for non-linear storytelling and advanced interaction to produce the prototype of the interactive installation documentary *Tangible Memories* detailed in Chapter 4 (*Tangible Memories in Interactive Documentary*)

- the performance arts and theatre production provides the theoretical and practical foundation to the discovery and design of the illimitable performative space concept using the new technology for traditional performance art forms detailed subsequently in Chapter 5 (*Illimitable Space in Responsive Theatre*)

- the technological background describes the affordable technologies today that aid with the prototyping for and experimenting with the above main three aspects and these technologies are used throughout one more of mentioned chapters

- the topics discussed in this chapter continue to be relevant not only in the current contributions summarized in Chapter 6, but form basis for the vast open-ended future work possibilities summarized there as well to complete the missing pieces and serve as a guide for direction or source of inspiration and improvement over for the purposes of the modern evolution of the present art forms

2.1 Computer Animation and Advanced Rendering

Graphics are drawings and visual representations which can be traced back from 40,000 to 10,000 BC or even earlier [29, 30][1]. "Animation is a graphic representation of drawings to show movement within those drawings."[2] Computer graphics and animation are the drawings and sequences of drawings generated by computer, which has had a profound impact on television, movies and the video game industry. It has significant value for medical research and media arts achievement as well with relative long history since 1960s [31].

In this section we review only some of the most prominent computer graphics

[1] `http://en.wikipedia.org/wiki/Graphics`
[2] `http://en.wikipedia.org/wiki/History_of_animation`

techniques for advanced rendering and animation in 3D games and beyond instead of traditional graphics and animation. The analysis is based on the various literature, tutorials, and practice reviews, some of which are [32, 33, 34, 35, 36, 37] as well as other works cited throughout the text [22].

2.1.1 Traditional and Asian Animation Influence on Computer Animation

We briefly review traditional animation, some of the early animated film in both Western (e.g., Figure 9) and Asian (e.g., Figure 10) countries and their traditional techniques. Moreover, several computer animated films have also been included in the discussion, particularly, having Asian artists and their contributions assessed.

(a) *Snow White* (1937) Poster (b) *Princess Iron Fan* (1941) Poster

Figure 9: Early Animated Feature Films

Animation has a long history. The first feature animation film was *Snow White*

(Figure 2.9(a)) made in the US in 1937. And the second one was *Princess Iron Fan* (Figure 2.9(b)), made in 1941 in China. Chinese animation played a very important role in Asian animation, especially influenced Japanese animation film. It was interesting to know that the first Japanese color animation feature film, *The Tale of the White Serpent*, was even based on the Chinese folk story.

(a) *Uproar In Heaven* (1961,1964) Poster (b) *Astro Boy* (1963) Poster

Figure 10: Early Asian Color Animations

Traditional 2D animation was drawn by hand, such as Chinese water-ink animation. Other animation art forms use puppets, paper cutting, and folding-paper. Some excellent animation works in the 20th century were *Uproar In Heaven* (Figure 2.10(a), in China, 1960s), Asian first feature animation in color, based on Chinese the earliest chapters of the classic story *Journey to the West*; *Astro Boy* (Figure 2.10(b)), Japanese manga series broadcast on TV program (in Japan, 1963); *Hong Gildong*, the earliest Korean animation movie in which the main character was from an old

Korean novel (in Korean, 1967); Hong Kong's first puppet animation film of theatrical animation, *Prince of the Big Tree* (in Hong Kong, 1948); and Malaysian animated cartoons, *Hikayat Sang Kancil* (in Malaysian, 1978) was produced for television [30].

After CG technology appeared and since CG animation techniques stemmed originally from the traditional animation, many cartoonists turned to animators in order to accommodate themselves in the new digital era producing first CG-animated films (e.g., Figure 11). After the first CGI feature-length animation film, *The Toy Story* [38] (Figure 2.11(a), in the US, 1995), was created, ten years later, first Asian 3D-CG film, *Thru the Moebius Strip* (Figure 2.11(b)) was made (in China, 2005), (whose cost was about 20 million in US Dollars), and has finally been released.

(a) *Top Story* (1995) Poster (b) *Thru the Moebius Strip* (2005) Poster

Figure 11: Early CG Feature Animation Film

In general, there are 2D and 3D CG animation types. Looking at the Chinese market as an example (e.g., Figure 12), there are two categories of animation: conventional animation (usually well-financed projects by government or production companies) and "webtoons" (mostly produced by individuals who post them on the web).

Chinese TV series and animation film box office winner, *Pleasant Goat and Big Big Wolf-The Super Snail Adventure* (in China, 2009, Figure 2.12(a)) was criticized however by the experts because it aimed at audience aged from 6 to 8. In contrast to the Western CGI film usually attracts a wider range of audience and could portray very serious topics in films.

Similarly to the roles of CG in North America, in Asia, CG animation has been widely used for theatre performance, film, advertisement, special effects in post production, education, medical training, military rehabilitation, and video games.

Some examples includes an augmented reality installation, *Citizens Comfort* (in Singapore, 2008) [39]; motorized installation, *Cloud* (Tai Wan Interactive Technology Laboratory) [40]; mixed reality installation, *The Swimming Fish* (Beijing, Tsinghua University) [41] (where an interactive virtual reality installation translates the visitor's hand gestures to generate a swimming fish as if he or she reenacts the magical touch of the painter Ma Liang—a Chinese legend). Moreover, some new developing CG companies, such as *E-GO Computer Graphics*[42], focus on high-end digital animation in film, television, theater, and large scale outdoor projections (e.g., Figure 2.12(b)).

(a) *Pleasant Goat and Big Big Wolf-The Super Snail Adventure* (2007) Poster

(b) *Tsinghua University 100 Anniversary Celebration Building Projection* (2011)

Figure 12: CG Applications in China

The burst of CG and innovation in animation in Asia can be witnessed by creation of the major branch of the SIGGRAPH conferences—SIGGRAPH Asia in 2008 and continuing to today that went through Hong Kong, Singapore, Japan, Korea to additionally accommodate ever-growing dissemination of works in computer graphics that include not only technical papers, but also art papers, exhibitions, animation festivals, apps, and the like, featuring numerous artist from Asia and abroad [43, 44, 45, 46, 47, 48, 49, 50]. There has also been a boom in video games (mostly from Japan (Sony) on PlayStation platforms) and other vendors.

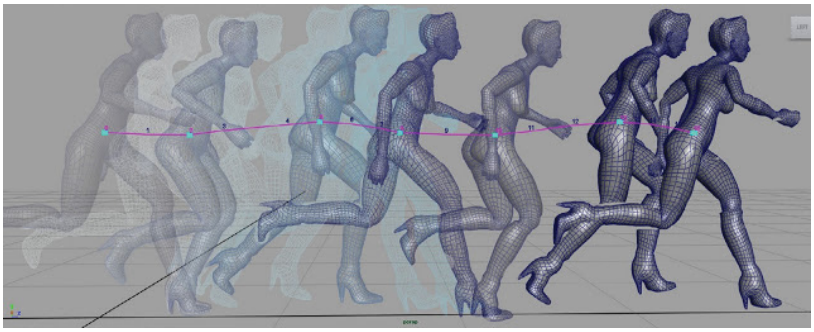
2.1.2 Computer Animation

Computer animation is a sub-field of computer graphics, which uses both 2D and 3D computer graphics [51] to create moving images. The currently accepted rate of computer generated animation is 30 frames-per-second (FPS) or 24 FPS for film. The animation can be divided into two main categories: offline and real-time. Both off-line and real-time methods apply acceleration methods, algorithms, and approximations in order to meet the efficiency requirements as much as possible while trying to retain some realism/believability elements (e.g., Figure 13).

2.1.2.1 Offline Animation

Offline animation (also called "precomputed animation"), constructs a list of fixed animated scenes without responding to users' actions in real time. The animator or programmer has exact control of the animation. The change of the scene database and scene viewing happen independently. For example, offline animation generates a film and plays it back later. It is used mainly to create high-quality lifelike scenes for film industry.

The performance of the rendering is only the second priority compared to its realism. Most special effects in today's movies and entire animation films are created by computer graphics. In order to achieve high-quality effects, the graphics hardware has become more programmable. Some good examples of offline animations are *Final*



(a) *Offline Animation from Maya Software* [52]

(b) *Real-time Animation from Angry Bird Game*

Figure 13: Offline and Real-time Animation Example Screenshots

Fantasy (Figure 2.14(a)), *Toy Story* [38], *Finding Nemo*, *Cars* Pixar movies.

Some of the techniques for modeling and animation of the offline scenes are tool-based modeling, data-driven modeling, procedural modeling, behavioral animation, and dynamic physical based modeling, which we discuss next.

Tool-Based Modeling. We have a variety of techniques to model the animated scene, such as using software modeling tools to directly specify 3D models and their transformations. One of the traditional offline animation techniques is *keyframing* (e.g., Figure 2.13(a)), which depends on an artist to generate key frames and the

in-between frames are drawn automatically by a computer. Linear interpolation creates the in-between frames at equal intervals along straight lines, whereas nonlinear interpolation can create equally spaced in-between frames along curved paths. Another example for tool-based modeling is that the artist directly controls how a given skeleton specified by joint angles moves and the computer solves for the angles that achieve the motion. In general, tool-based modeling is tedious and inflexible.

Data-Driven Modeling. This technique measures the real world, and then uses that data to synthesize or alter models. *Motion capture* is an example of data-driven modeling, which captures the motions of the actor and aligns motion data with a CG character. Other examples include laser scanning and eye tracking techniques. Moreover, the recently developed Wii, Kinect, and haptic devices produce a lot of sensor data that can be used in real-time data-driven modeling.

Procedural Modeling. During procedural modeling, one writes programs to automatically generate models and their transformations, for example, to apply in special effects, such as explosions, water, and flocking behavior by specifying simple physical rules of motion [53].

Behavioral Animation. This technique provides realism to computer-generated animation, which is defined by describing an actor's behavior or goal [54]. It gives the characters autonomous intelligent behavior and is related to AI. For example, an actor's behavior defines how the actor interacts with other actors and the environment. It is particularly useful for crowd animation.

Dynamic Physical Based Modeling. Such modeling offers a concise, but rich and flexible way of defining the behavior of animated objects and characters, based on the physics laws to determine or guide their motion. Particle systems are common examples here, which represent objects such as fire, smoke, etc., as a collection of individual particles [55, 56, 57]. Moreover, dynamic physical based modeling also

applies to the rigid body animation, deformable objects animation [58, 59], cloth simulation, and fluid simulation.

2.1.2.2 Real-Time Animation and Simulation

Unlike offline animation, real-time animation allows changes of the scene database and scene viewing to happen at the same time. Representative real-time animation techniques are *machinima*[3,4] [60, 61, 62], digital puppetry, and motion capture, with which users could create a performance in real-time and capture it in a virtual world. Real-time simulation is mostly physical based, usually imitating some real physical phenomena, such as fluid, deformable objects, depth of field simulation, and others, with which users can also interact in the virtual scenes in real time by altering the simulation data. Real-time animation and simulation appear together in particular in interactive applications, where it is important in those disciplines, such as video games (e.g., Figure 2.13(b)), flight, car, or surgery simulators. The goal for real-time animation and simulation is to maximize realism at the minimum cost. Real-time animation and simulation represent a very wide range of research topics, and we only discuss a few of the methods that are relevant to this dissertation work. They are character animation, motion capture, softbody simulation, depth of field simulation, and collision detection.

Character Animation. Compared to traditionally animated characters, which would either be hand-drawn or in 2D images, sprites, the demand for more realistic real-time character animation has increased considerably in the recent past due to the advances in computer hardware and software. At the time, many of the off-line animation techniques, which only were listed as possible future developments after several years, like skeletal animation, deformation of 3D skin meshes, are now established methods for real-time 3D character animation. Some of the techniques are used in character animation are hierarchic articulated objects, animation blending,

[3] `http://en.wikipedia.org/wiki/Machinima`
[4] `http://en.wikipedia.org/wiki/Machinima:_Virtual_Filmmaking`

and skeletal animation.

In hierarchic articulated objects, different body parts are separately stored in a hierarchy joined at pivot points. The transformation is recursively traversed from the root down to its children within the hierarchic data structure. Animation data can be generated using inverse kinematics (IK) and applied to 3D model in real-time. Animation blending uses more than one object, which represent same character showing in different poses. The objects, which contain the same number of vertices are blended or just switched in order to achieve the animation effects. The original Quake game [63] used this method, such as vertex blending for character animation [64]. Skeletal animation was designed to simplify the animation process for dealing with articulated objects in order to provide more realism of characters. It also uses a hierarchic structure of joints as skeleton, which drives a rigid or soft skin mesh. The skeleton is animated, and the animation applied to the skin.

Motion Capture. Motion capture records the movement of objects and human beings and then maps the collected data to computer generated objects and characters. This technique is often used in film animation and computer games. The MoCap systems and software were quite expensive in recent decades. They typically place markers or sensors on the skin or clothing near actors' joints or even on the face for facial animation. However, these days, some inexpensive MoCap hardware and software systems for entertainment purposes emerged, such as Kinect (see Section 2.4.1.1) and iπ have been developed, which are quite powerful, accurate, and easy to use [65].

Softbody Simulation. Softbody simulation also known as deformable object simulation, uses physical based simulation to mimic non-rigid bodies, such as human and animal soft body parts and tissue, and other non-living soft objects, such as cloths, gels, liquids, and gas [66]. It has been increasingly used for character animation, computer games, and surgical training due to the improvement of the quality and efficiency of the new generation of computer hardware. However, there are still a

lot of aspects that can be explored in the real-time softbody deformation simulation visualization area because it has emerged as a challenge in computer graphics, and, therefore, was not fully exhausted yet [14, 22]. We proposed in the past the Softbody Simulation System framework to be released open-source [14, 22]. Additionally, a large profile deformable object library called *Vega* [59] has also been released in the open-source and exhibited at SIGGRAPH 2012 showing continued interested in and usefulness of the subject. We explore further on this topic in Chapter 3.

Collision Detection. Collision Detection (CD) [67] is a very efficient way to increase the level of realism when animated objects collide on the scene (otherwise they would pass through each other) [68]. It is practically impossible to make realistic games, movie production tools (e.g., *Toy Story* [38, 35]) without collision detection, because we would get "quantum effects" all the time [32]. In general, there are two major parts in most CD algorithms: *collision detection* itself, which is usually a geometric intersection problem; *collision response* [69], which is the actual (approximated) physics of determining the unknown forces (or impulses) of the collision [35, 70].

The following are the major algorithmic techniques to realize CD surveyed in some detail in [22]: *ray tracing*, which is relatively simple to implement, not very accurate, relatively fast, and sufficient for most cases; and *bounding volume hierarchies* (BVHs), which are more complicated, somewhat more accurate, and slower. Using BVHs one can compute exact results; *efficient CD for several hundred objects*, which are dedicated solutions [32]. Other optimization structures such as Octree, BSP (binary separating planes), OBB tree (oriented bounding boxes, which is a popular instance of BVHs), KD tree, K-dop tree, uniform grid, and hashing also exist and were explored and exploited to various degrees by CG practitioners [35, 32].

CD is a very important part of games or any type of physical-based simulation involving impact situations. There are several different algorithms and frameworks to choose from, which were mentioned earlier, so they should be selected based on the situation being implemented. One way to approach such an implementation, is

to provide a collision detection framework to allow a common API for the algorithms and possibly run-time parameters adjustment of them, e.g. [71]. In this work we do not attempt to make such a general framework, but merely expressing the idea in a proof-of-concept (PoC) implementation [22].

2.1.3 Advanced Rendering

Computer graphics rendering is a sampling and filtering process [72]. Rendering research and development has been largely motivated by finding efficient algorithms to simulate the most realistic scenes. In a nutshell, it is a process of generating an image from a model by any specialized hardware, such as a graphics card. The model, which is a description of three-dimensional objects in a strictly defined language or data structures, typically contains geometry, viewpoint, texture, lighting, and shading information.

Rendering, originally taking place entirely in the CPU, has been used in architecture, video games, film or TV special effects, and design visualization [72]. The advantages of rendering everything with CPU is that it is not restricted to the specific capabilities of graphics hardware providing a form of programmable "card-independence". The obvious shortcomings for the CPU-based rendering are not taking the advantage of the throughput and speed the specific graphics hardware is optimized to provide and share the processing load [32].

Rendering rates of display monitors can accommodate refresh rates of up to 100 frames per second (FPS). However, beyond 70–80 frames, the human visual system cannot perceive the changes in moving images. Movement is perceived by the human visual system when images are refreshed at 16 frames per second or higher (called *persistence of vision*). Smooth motion cannot be perceived at much lower speeds. For animation, one typically plays the images at 30 frames per second.

Real-time rendering and animation (see Section 2.1.3.2) usually denotes scene rendering computations at 15 frames per second or higher. For 3D animation, it is possible to pre-compute some images (not necessarily in real-time) and play back

at the desired frame-rates [32]. *Offline rendering* (see Section 2.1.3.1) is more used to create realistic images and scenes for movies; real-time rendering is frequently used to interactively render a scene, such as in 3D games [73, 74, 33]. From the rendering aspect, ***the movie and game, the only difference is in the number of samples and the quality of filtering*** [32] *and* the level of **interaction**.

Some examples of the advanced and classical rendering include shading and rending characters' skin [75], hair [76] (e.g., Figure 2.14(a)), jellyfish [77], and a whole documentary [78].

(a) *Offline Rendering from Final Fantasy Film*

(b) *Real-time Rendering from Final Fantasy Video Game*

Figure 14: Final Fantasy Example Screenshots

2.1.3.1 Offline Rendering

To render a static scene is to determine the color to be assigned to every pixel of an image depicting the scene (usually 3D) as viewed by the viewer [72, 79]. *Image-based Rendering* (IBR), which has images as the primary data used for rendering, most commonly is photorealistic depiction for providing a realistic viewing experience. Most IBR is done in offline rendering, which can tolerate one frame of animation taking even hours to render [72, 79]. This is acceptable in film production where real-time response is not needed.

2.1.3.2 Realtime Rendering

In practice, real-time applications often omit rendering correct soft shadows, depth of field, small-scale surface detail, and realistic surface materials in order to achieve the speed efficiency for user interaction [80, 33]. The performance of real-time rendering is extremely important because it has to allow user interact with the computer-generated graphics in real-time. Real-time rendering tries to achieve at least three goals: it allows more frames per second, generates higher resolution images, and more realistic objects.

Unlike the offline *image-based rendering* (which in some cases could be used for illustration or artistic purposes); it is known as *Non-photorealistic Rendering* (NPR). NPR, also called *stylistic rendering*, has a wide range of goals, such as creating an image similar to technical illustrations and only the relevant details would be displayed [32, 33]. NPR has always been related to computer games, in which each frame of animation is rendered in real-time (e.g., Figure 2.14(b)). In order to achieve $\approx$ 25 FPS, we could only compromise the quality of images somewhat to render dynamic images (change in scenes), such as view position and view angle changes, object movements, object deformations (in shape or other properties), and changes in environment, such as lighting [33, 35].

2.1.3.3 General Tools and Techniques

2.1.3.3.1 Ray Tracing. Ray tracing produces a very high degree of photorealism by tracing the path of light. The derived algorithms from ray tracing are beam tracing, cone tracing, path tracing, and metropolis light transport [33, 35].

Ray tracing was traditionally an offline rendering technique until relatively recently when with the GPU support real-time and distributed rendering became possible including using the CUDA framework[5,6], URay [81] and the like.

The technique is particularly good as in one algorithm it covers shadows, reflections, and translucency at the same time. The classical ray tracing does not do soft shadows but is adjustable with modifications to make them possible in combination with other techniques [33, 35].

2.1.3.3.2 Global Illumination. Global illumination has been used more in offline rendering and it generates the higher level of realism based on ray tracing, radiosity, and other techniques. It uses a group of algorithms in 3D computer graphics to demonstrates how realistic lights are reflected by surfaces in order to increase the overall perception of and realism [82, 79]. The direct illumination refers to the light directly coming from a light source; indirect illumination describes the reflected light rays from other surfaces. Some examples of global illumination are reflection, refraction, color bleeding, and hard and soft shadows. The algorithms used in global illumination are diffuse inter-reflection, radiosity, ambient occlusion, photon mapping, and image based lighting [82, 79].

2.1.3.3.3 Level of Detail (LOD). The basic idea of LOD is it use simpler versions of an object when it makes less and less of a contribution to the rendered image. In computer graphics, accounting for level of detail involves decreasing the complexity of a 3D object representation as it moves away from the viewer or according other metrics such as object importance, eye-space speed or position [83, 84]. Different

[5] `http://en.wikipedia.org/wiki/CUDA`

[6] `https://developer.nvidia.com/cuda-toolkit`

levels of detail are used at different distances from the viewer. There is not much visual difference, but rendering can be significantly more efficient. We could use area of projection of bounding volumes (BVs) to select appropriate LOD [84]. When the object is far away, we replace it with a quad of some color; when the object is really far away, we do not render it (called *detail culling*) [83, 84].

Although most of the time LOD is applied to geometry detail only, the basic concept can be generalized. Recently, LOD techniques included also shader management to keep control of pixel complexity [85, 86]. A form of level of detail management has been applied to textures for years, under the name of mipmapping [87], also providing higher rendering quality [33].

The LOD selection is to make a choice for which object to render or to blend. A metric called the benefit function, is evaluated for the current viewpoint and the location of the object. The metric may be based on the distance from the viewpoint to the object, or the projected area of the bounding volume of the object [83, 87].

Subdivision. Subdivision surfaces are powerful in defining smooth, continuous, crackless surfaces [32, 83]. They also provide the infinite LODs. The staring mesh is called the *control mesh*. The first phase called the *refinement phase*, which creates new vertices and reconnects to create new, smaller triangles. The second phase is called the *smoothing phase*, which computes new positions for some or all vertices in the mesh.

DLOD is the simplest type of LOD algorithm, which is based on subdividing the space in a finite amount of regions, each with a certain level of detail [83]. The result is discrete amount of detail levels. There's no way to support a smooth transition between LOD levels at this level. Moreover, DLOD could provide various models to represent the same object [83].

2.1.3.3.4 GPU. Hardware-based rendering on the graphics card's graphics processing unit GPU adds a significant computational power to the CPU-based rendering by offloading a good amount of computation and rending in parallel to the CPU and

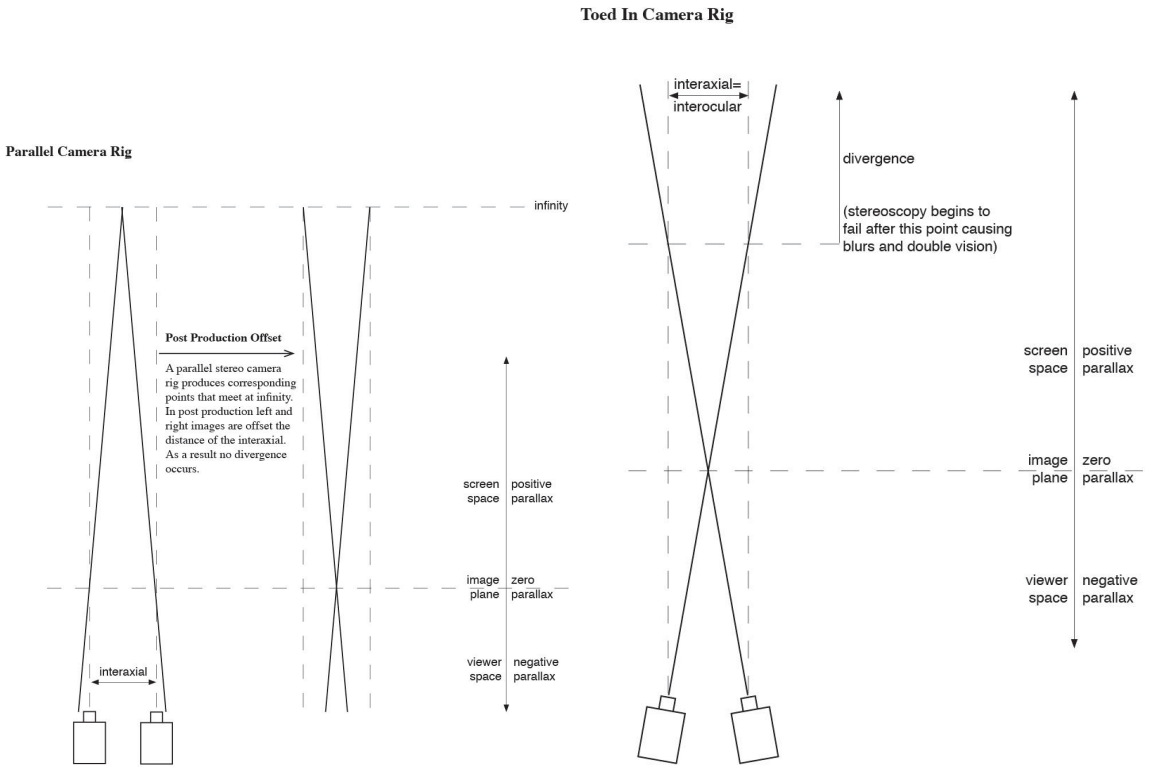
multiple computational units on the card. The GPUs in most graphics pipelines primarily work with the geometry (vertices) and the pixel fragments after rasterization. The shader programs that provide access to both of these such points in the pipeline have standards defined for them [88, 89]. The programs one writes are the shaders that run on the GPU and can do more than traditional graphics API provide faster. The shading programming languages used for the GPU programming typically are either cross-vendor assembly [88, 89][7] or higher-level shading languages, such as GLSL [36, 90] or HLSL [33][8].

The concrete sections that describe the use of the GPU shading (using the shading languages mentioned) provide some more implementation details in Section 3.4.2.1.2, Section 3.4.2.4.3, Section 4.2.2.1.2, Section 4.2.2.2.2, and Section 5.2.3.1.1.

2.1.3.3.5 Stereoscopic Rendering. The basic stereoscopic rendering has to do with rendering the scene twice at slightly different angles and recombining the two images back into a single image such as it is seen as a real three-dimensional scene through red-blue glasses. This is achieved by rendering the scene offset by two stereo synthetic cameras. The distance between these stereo cameras, called the *interaxial*, is adjustable and the cameras may be set to remain parallel as in Figure 2.15(a) or to *toe in* (angling towards a center of interest) as shown in Figure 2.15(b). Supplemental attributes need to be added to deliver predictable stereoscopic effects and to facilitate stereoscopic post-production. A greater understanding of stereoscopy and the notion of parallax (illustrated in Figure 16) key to building such camera rigs [2, 7].

Film Production. Stereoscopy is increasingly used in the animated stereo film production, e.g., in the films like *Folding* [91] and *Journey To The Center Of The Earth* [92] and the ones that inspired the work on the stereoscopic plug-in at the time [7] and many others. Section 2.2 describes some of this as well.

[7] `http://en.wikipedia.org/wiki/ARB_(GPU_assembly_language)`
[8] `http://en.wikipedia.org/wiki/HLSL`



(a) Parallel Stereo Viewing (b) Toed-In Stereo Viewing

Figure 15: Parallel vs. Toed-In Stereo Viewing [2]

Stereoscopy in VR and Medical Research. Simulation of stereoscopic softbody objects in haptic environments [14, 22, 19] can help surgeons to train for real an virtual (tele) surgeries on virtual responsive subjects. Stereoscopic virtual reality (VR) environments for medical and rehabilitation research are other applications where it has potential for use as a cheaper alternative to expensive head-mounted display units. For more details in VR in medical research please refer to Section 2.5.2.

2.1.4 Additional Related Work

We review a number of related work items considered in the design and implementation of this work as well as future plans and considerations. This subsection primarily

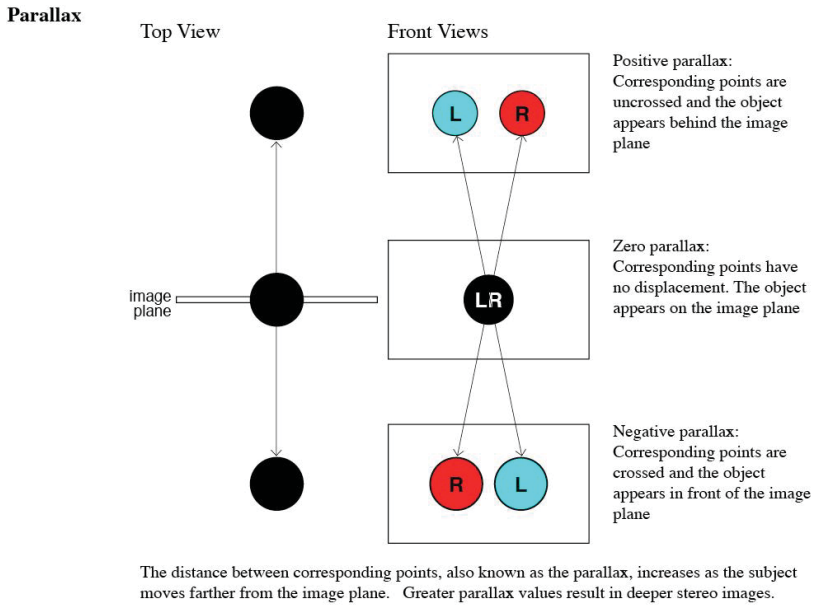


Figure 16: The Concept of Parallax [2]

covers the related work on the modeling, animation, and rendering of a jellyfish (Section 2.1.4.1) and OpenGL slides in Section 2.1.4.2. The rest of the related work is cited and described throughout the rest of the section and where appropriate throughout the thesis.

2.1.4.1 Jellyfish Modeling, Rendering, and Animation

The inspiration and aesthetics related work along with modeling, rendering, and animation of the jellyfish project can be found in [77, 93, 94]. An advanced interactive jellyfish game implemented in the Unity engine is presented in [95]. The main issue with all these works, which look realistic and nice is that most of the animation is precomputed offline/with keyframes and is modeled in tools such as Blender or Maya [96, 9] and then played in a scene or a game. We'd like to make a similar type of animation but in real-time based on physics [97, 98] with the aim at a real-time tangible interaction or manipulation of the jellyfish in a game or an interactive installation with haptics and motion tracking [11].

2.1.4.2 OpenGL Slides Framework (OGLSF)

OGLSF gives ability to make OpenGL "slides", navigate between them using various controls, and allow for common bulleted textual widgets — the *tidgets*. It also allows to override the control handling from the main idle loop down to each individual (current) slide (a scene). All slides together compose a concrete instance of `Presentation`, which is a collection of slides that uses the Builder pattern to sequence the slides. Each slide is a derivative of the generic `Slide` class and represents a scene with the default keyboard controls for the tidgets and navigation. It is understood that the tidgets can be enabled and disabled to allow the main animation to run unobstructed [99]. Each scene on the slide is modeled using traditional procedural modeling techniques [53] and is set as a developer or artist desires. It can include models and rendering of any primitives, complex scenes, texturing, lighting, GPU-based shading, and others, as needed and is fit by the presenter [99]. The main program delegates its handling of the GLUT callback controls for keyboard, mouse, and idle all the way down to the presentation object that handles it and passes it down to each current slide [99]. Our Softbody Simulation System was integrated into such a collection of OpenGL slides for educational and demonstration purposes.

2.2 Documentary Film and Interactive Media

We briefly review the topic on the role of computer graphics in the production of documentaries, which was quite often ignored in favor of other topics [100]. Typically, except for some rare occasions (the number of which is now currently growing in this emerging trend), documentary producers and computer scientists and or digital artists who do computer graphics are relatively far apart in their domains and infrequently intercommunicate to have a joint production; yet it happens, and perhaps more so in the present and the future.

We attempt to classify the documentaries on the amount and techniques of computer graphics used for documentaries. We come up with the initial categories such

as "plain" (no graphics), "mixed" (or "hybrid"), "all-out"—nearly 100% of the documentary consisting of computer-generated imagery. Computer graphics can be used to enhance the scenery, fill in the gaps in the missing story-line pieces, or animate between scenes. It can incorporate stereoscopic effects for higher viewer impression as well as interactivity aspects. It can also be used simply in old archived image and film restoration.

We analyze the impact of the computer graphics on the present-day documentaries in the two major sub-classes and project the future of the documentary films and TV production, as well as potential cognitive pattern-recognition-based interactivity with the documentary scenery might look like in the not-very-distant future as the professor's responsive hologram in the motion picture *I, Robot* [101]. Of course, the visual effects and animation is only one aspect or role of the vast computer graphics topic. Obviously, it counts for more than traditional animation for documentary film production.

2.2.1 Related Work

Computer graphics (CG) today is nearly everywhere, including documentaries. First, a very comprehensive review of CG itself is in the online set of lecture notes [29] describing the history and evolution of computer graphics hardware, software, algorithms, and techniques and their use for animation, including all kinds of CG standards, prolific conferences, artists, production companies, etc. (see also Section 2.1). CG is a key enabling point of new media that influences the modern day productions [100].

Perspective and cultural overview of new media forms from theoretical and philosophical standpoints, and in particularly the interaction aspects of the new media are described in [102]. Manovich's work [103] was deemed to be arguably as the "most systematic and rigorous theory of new media" along with its reliance on the old media from historical point of view. This work covers specific themes of interest including the fusion of cinematography and computers as well as the notion of virtual reality

slowly merging with the real [103].

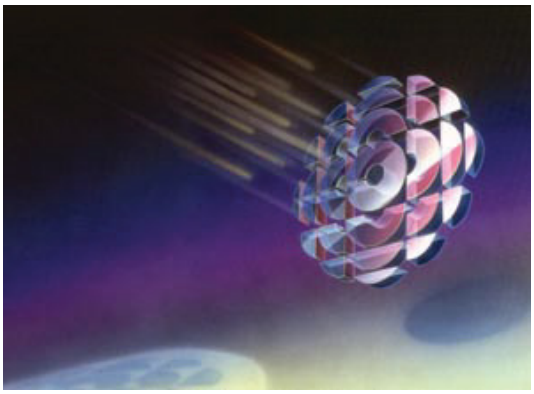
Computer graphics has been widely used in physics, biology, video game industry, education, and military matters. It has also been used in advertising, music video, motion picture, and television such as international network promos for CBC, e.g. shown in Figure 2.17(a) produced in 1985, and in the feature film *Jurassic Park* (1993) shown in Figure 2.17(b) [29, 104]. In the past, computer graphics seemed only to be associated with big budget productions; therefore, productions of the most of documentary films done in low budget had rarely applied these new techniques in their work. However, in recent years, more and more documentary films started to introduce computer generated 2D and 3D images and CG elements in a partially or fully animated film. Some other documentary films have used stereoscopic techniques, such as *Hubble* and *Born to be Wild* have been very successful in IMAX 3D theatres.

No matter whether a hand-drawn animated documentary or computer-generated animated documentary, they have certain aspects in common—they are "broadcast" to the audience. There is no doubt that animation is taking on a new hybrid-like style for documentary film. However, some questions arise. Can animation bring more realism to the documentary films? Is it necessary for the artists to apply this non-traditional medium to documentary film production? What is the motivation of the artists? How will animation further impact in the documentary film production process in the future [105]?

Moreover, the interactivity between human and computer generated images is a particular special characteristic related to computer technology. Thus, there is "interactive documentary" — a brand newly emerged genre, which has various forms, such as 3D and web-based, where the audience can participate in one way or another to as what is happening in the documentary.

2.2.2 Traditional Animated Documentary

The animated documentary is a genre of film, which combines the genres of animation and documentary. Non-fiction documentary animation, which deals with non-fiction



(a) Canadian Broadcasting Promo

(b) Jurassic Park's Raptor

Figure 17: Some Example Graphics for Production

material can utilize documentary audio interviews, or it can be an interpretation and re-creation of factual events [106]. Some audiences have reacted negatively because they found out they had the feeling of "forced empathy" and the animated documentary was "propaganda". They more associate the cartoons to children and could not accept this to documentary which expresses truth. However, animated documentary is more obvious and transparent for audience to distinct the reconstructed scenes and bring them the extraordinary emotion than live-action. Thus, to me, animated documentary films have more truth-value.

2.2.2.1 Animation, Realism, and the Truth-Value

"Truth-value" and "film realism" are two often opposing topics debated in academic discussion since 1990s [107]. Some filmmakers try to enhance the realism of a film because some of the scenes in it are not "real". Some other filmmakers say that there is no perfect truth in documentary film because as soon as the film has been edited, some of the truth has been hidden and is a subject to the opinion of the filmmaker. For some hybrid documentary films, some scenes have to be reconstructed. Do the reconstructed scenes still have the truth value?

Compared to traditional non-animated documentary film, the scene in an animated documentary does not "exist" at all. It is neither a live-action camera footage nor a photograph, but it is created by artists based on the real world objects or based completely on their imagination. How much truth-value will remain in an animated documentary film after reconstruction? Does it exist there at all? We always talk about how to make a film more realistic and increase its realism in order to make the "faked" scene look more believable. Obviously, for such a film, the truth-value is not comparable with the one that contains real footage and photographs.

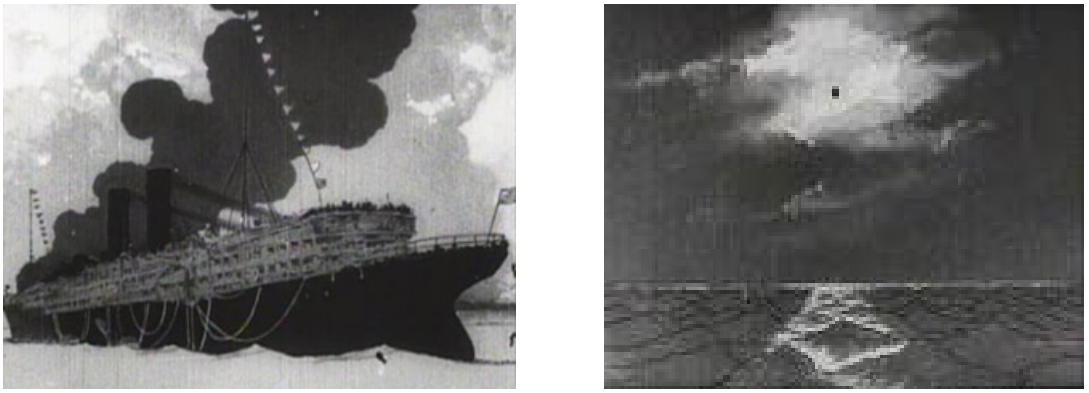
As shown, the biggest challenge for most of the documentary filmmakers is to capture truth. Moreover, it is a difficult task for them to make a film when there is a shortage of footage. There are various solutions to fill in the gaps, such as scene reconstruction, photographs, texts according to historical events, and the needs from artists and the makers. Some artists decide to use the photos to represent the past, such as American filmmaker Ken Burns who uses the "photographing of live-action material" [108]. Some use either traditional animation or computer graphics, or both to reconstruct the scenes because they feel it is more vivid and can make their scenes closer to the truth compared to photos. Some explain the necessity to use animation in documentary to show the footage that one can not capture in the real life, such as imagination, dream, hallucination, and memory. A typical example such as in documentary *American Teen* (2008) [109, 110], animation was used for "teen dream" and "teen imagination".

> *"Realism is not what animation is best at, instead, freer invention, fantasy, and exaggerating reality are its privilege. The realer-than-real environment follows the concept of* hyper realism*, which offers a completely artificial environment as a representation of the real."* [111]

2.2.2.2 Traditional Animation in Documentaries

The use of animation in documentary production is not new. The traditional animated documentary can be traced back all the way to 1918 by Winsor McCay in

his 12-minute-long film *The Sinking of the Lusitania* (1918) [112], the first animated documentary. It used animation to portray the 1915 "sinking of RMS Lusitania after it was struck by two torpedoes fired from a German U-boat" [113]. Quite obviously, there was no live-action footage recorded when the event occurred. The animation used in this silent documentary film for things such as the underwater fish swimming and the explosion and sinking of Lusitania, some of which shown in Figure 18. The explaining texts were throughout the film between the animated scenes. This is very traditional story telling in silent film. The only difference is the animated scene replaced the live-action footage.



(a) Example 1 (b) Example 2

Figure 18: *The Sinking of the Lusitania* (1918) Example Screenshots

Animation has been used in other educational and social guidance documentary films, such as *The Einstein Theory of Relativity* (1923). The purpose to use animation in earlier times is because of the need to be able to illustrate abstract concepts in mainly live-action examples of these genres [114, 115]. In traditional animation, inbetweening, cell animation [116], and rotoscoping [117] have been later introduced by Disney in order to be efficient when working with many single-frame images. Disney's film exaggerate reality in order to create a greater impression of realism is often called "ultra-realism". However, these old techniques required artists to illustrate thousands of pictures to be filmed when close to the end of the production process.

2.2.3 Computer Animated Documentary

With the appearance and rapid development of computer hardware and software, traditional animation techniques mentioned above, have been slowly replaced by computer generated graphics and animation. Since *Walking with Dinosaurs* (1999), a six-part documentary series produced by the BBC in 1999 [118], the style of a traditional documentary was simulated. More and more documentary films these days include visual effects and CGI (Computer Generated Imagery) as a norm. Comparing to *Walking with Dinosaurs*, still the most expensive documentary series per minute ever made, production costs have come down in recent years with the rapid development of computer hardware and software. Thus, even for documentary films with a low budget, directors could consider introducing CG animations into their film nowadays. This new genre of documentary films usually reconstructs the historical and informational footage, which is not available. Moreover, computer generated graphics and special effects can illustrate very serious and heavy topics with a humorous connotation in order to attract the audience's attention.

> *"If you have several visual effects shots that work together to tell a story, using them separately earlier on in the documentary can increase their impact. ... Earlier in the documentary, each shot will work well in sections of the film detailing the different aspects of the threat, and then when the entire animation is brought together for the climax of the documentary, the effect will be even greater."* [119].

Computer animation has a lot of similarities when compared to the traditional animation. However, it is more advanced than the traditional one and more and more adopted by the current film industry because of its efficiency, accuracy, lower cost, and more advanced rendering effects, which can enhance the realism of a film more than the traditional animation could. Even though computer graphics has been playing a significant role in formulating new aesthetic grounds for both fiction and non fiction films, will this kind of film abandon the most important concept, the "truth-value",

which is precious in documentary film? How this new type of documentary film will affect the older audience and new generation audience? How much could they believe CGI documentaries? Through the discussion about this and other topics, we will reach to a conclusion in that regard.

2.2.3.1 CGI and Realism of Documentary

According to the recent work by Landesman [120], the documentary film overall in the past years has been experiencing a notion of formal change from traditional "observation and omniscient narration" in terms of being less strict in the need to objectively portray the material. More and more the documentary film has been embracing the paradigm switch to performance rather than recording of an observation, some subjective rather than objective aspects and even fiction; equally no longer requiring to be as certain and complete as possible in the argumentation in the film instead of just showing dry factual knowledge [120]. CGI techniques are here to help with the emerging trends of documentary film concepts and production Landesman described.

More and more documentary films use computer animation and special effects in their production. As mentioned earlier, some spectators found out that animated documentary is more obvious and transparent for audience to distinguish the reconstructed scenes and bring them the extraordinary emotion than the live-action. Moreover, it could sometimes reach an audience that might not watch live-action documentary and is easier to incorporate difficult, delicate, and controversial topics [121, 12]. There is no doubt that animation is taking on a new hybrid-like style for documentary film [12]. However,

- Can animation bring more realism to the documentary films?
- Is it necessary for the artists to apply this non-traditional medium to documentary film production?
- What is the inspiration that lead artists apply this new layer to documentary film?

- How does the audience respond?

Some of the notions and beginning of such documentary style appear in different documentary film early and later-day works [122, 121, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133] as conveniently compiled by Lee and Nitoslawska [134].

Rowley talks about the quest of traditional animated film (hand-drawn style) for a particular kind of realism in order to succeed in feature film making, such as "visual realism", "aural realism", "realism of motion", "narrative and character realism", and "social realism" [107]. We borrow some of these terms for completely different perspective of analysis in "computer animated documentary". We not only explain their concepts, but also look at computer graphics role on how enhance theses realism, and which realism types are applicable for our study.

Visual Realism. It evaluates the extent to which the animated environment and characters are understood by the audience compared to the ones from the actual physical world [107]. Dimensionality and the level of detail (LOD) are two main aspects of visual realism. Dimensionality refers to the extent that an illusion of depth is created, whereas LOD describes the extent to which the background depicts complex particularities of the environment. 3D modeling software, e.g. Maya [9] or Blender [96], can not only provide the advance modeling tools for shaping the objects, but also supply the advanced rendering techniques for artists to build the very realistic environment with vivid textures and lighting. Moreover, even an individual without any drawing skills, can use most of the current 3D modeling software to model the objects, characters, and landscapes. The visual realism type is the most applicable from the CG point of view in our study as it depicting complex particularities of the environment where it impacts and impresses the audience most.

Realism of Motion. It contrasts the extent to the characters moves and motion in artificial environment and physical world and laws of physics. Traditional animation relies on persistence of vision and refers to a series motion illusion resulting from

the display of static images in rapid-shown succession [107]. Artists have to use not only their drawing skills and intuition, but also possess some knowledge of physics to make the objects behave as if they are in the real world or close to it. The motion of the virtual objects will not convince audiences if no natural laws of physics are applied [107]. Moreover, drawing the virtual objects moving from one frame to another frame is an inefficient way without functionality provided by software. However, one of the computer techniques, motion capture is very efficient and accurate to describe virtual objects motion. It attaches sensors on actors bodies and records the data for their movements and apply these data to a computer generated characters. This technique increases the realism of motion dramatically. Additionally, some physics engines combined with computer graphics rendering techniques, e.g., softbody simulation [5, 14, 22], contribute a lot in this realism type by tweaking physical simulation parameters of the laws, such as gravity, material properties, inertia, one can produce interesting visual motion outcomes to make a point in a film.

Narrative and Character Realism. This aspect attempts to make audience believe the fictitious events and characters of the animated film actually exist. For example, in order to portray the animated character vividly, artists use the squash-and-stretch method exaggerated for soft parts of the character in traditional animation techniques [107]. However, it is a very time consuming procedure. Today's computer graphics software often provides a group of functionalities and a library, such as hair, skin, clothes animation, skeleton animation in order to simplify artists' work and achieve more realistic results.

Social and Psychological Realism. Social Realism makes audience believe that the event is taking place in the fictitious animated world is as complex and diverse as the real world. This concept applies both on traditional animation and computer animation. In order to achieve social realism, artists not only rely on other visual, character, motion realism, but also count on the writing of the documentary production. Psychological Realism was the first time brought up by Chris Landreth, the

director of animated short documentary *Ryan* [135]. It does not consider the physical based motion as the priority, nor use some techniques, such as rotoscoping and motion capture. Instead, he uses CGI animation in his work with added elements, such as an original, personal, and hand animated three-dimensional world. Using psychological realism technique puts surrealist styling into the animated documentary, so that peoples' psychological traumas are represented by twisted, surreal characters and their deformable faces. Films intend to use psychological realism to show the emotions of the characters in a way never seen before.

> *"There is no pre-existing reality, no pro-filmic event captured in its occurrence, an animated film exists only when it is projected. With no any existence in the world of actuality, the animated film like the partially dramatized documentary, rely on a kind of artistic re-enactment, depending, in part, on imaginative rendering as a compensation for the camera's non-presence at the event."* [114]

2.2.3.2 Types of Visual Effects and Animation in Documentaries

Christian Darkin categorized the types of computer graphics and the corresponding techniques used in documentary film as following in several categories [119] that we recite below to complement our study.

Explanatory or Explanation Graphics. According to Darkin, using *explanation (explanatory) graphics* is a quite acceptable means to provide explanatory notes, ideas, as well as information when the available footage cannot portray it sufficiently well. The CG-animated explanatory supplements can be very exciting and creative; 2D or 3D; text, cartoon, moving characters—it is up to the director, animator, artists, and their creativity, resources available, and the corresponding needs of what to portray. Such as CG type can arguably be fitting with any documentary style [119].

Animation for Color Shots. This is a general CG mechanism that is applicable and relevant to many types of documentary film production, especially if at some point there is more of the narration material than footage. Darkin also gives good examples of such color shots, such as a 20-second 3D animation where a virtual camera rushes through a bloodstream of a patient and blood cells and others "fly" past can easily be used in many medical-related documentaries and be "on topic" and not boring. Similarly, in the crime-related documentary, one can animate a "fly-through" a CG-generated building where the crime happened while narration is running and prior to when the real footage begins [119].

Visual Effects Reconstructions. As opposed to the explanation graphics, visual effects constructions are required in the absence of footage for the most prominent and necessary events to portray in the documentary film that could not have been possibly shot, physically inaccessible (the scale of cellular biology or the Universe), or too far in the past, e.g. the assault on Baghdad, dinosaurs, an assassination, the Big Bang, or if needed to show the fat molecules getting from a burger to one's thighs, the 3D animated CG reconstruction can greatly help to portray such events in a documentary film [119].

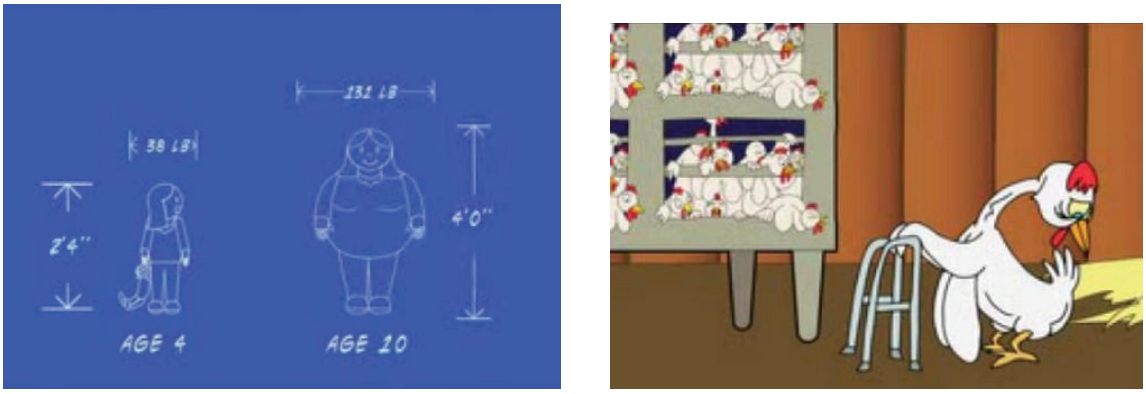
Text and Title Animation. This type of graphics is commonly employed to bring up documentary titles, scene announcements, some short on-screen paragraphs or questions raised by the maker, and subtitles. An introduction to the film, its topics, and ideas are good candidates for this type of animation in order to set the tone the documentary film is to proceed with. Some of the Motion Graphics techniques mentioned earlier can be very well applied to text [6] as well here as the CG techniques are typically common across the board, except the text animation is optional and can be just a still [119].

2.2.3.3 CG Role in Documentary Production

We will have a closer look at some of the following documentary films followed by an analysis of the impact of the computer graphics featuring computer graphics techniques to a various degree that feature CG-based animation:

***Super Size Me* (2004)** is a personal experimental documentary with some computer animation sequence. This is a small budget documentary film, produced by an independent filmmaker Morgan Spurlock. In this film, he did a test on himself that he had to only eat McDonald's three meals a day for 30 days [136]. The film is very fast-paced and full of sense of humor. It is a "hybrid" animated documentary, which contains partially animated graphics and scenes according to our classification of the animated documentary film stated earlier. It combines computer generated graphs, charts, and animation, with the tune in humor, which is the best way to get an audience interested in topics like that.

The simple graphics captured from the film and shown in Figure 2.19(a), is 100 times more visually vivid for audience to understand the information about how fast food would cause a 10 year old to a 14 year old girl to gain so much weight. Another example of animation used in this film shown in Figure 2.19(b) exaggerates how McDonald's make the chicken nuggets from "unusually large breast chicken".



(a) Example 1 (b) Example 2

Figure 19: *Super Size Me* (2004) Example Screenshots

When Morgan Spurlock was asked of the fact he used "a lot of video games, a lot of computer graphics and cell animation" and if it is the way to make his points "more accessible to the mainstream", he replied,

> *"Definitely. Absolutely. One of my beliefs as a filmmaker is that if you can make somebody laugh, you can make them listen. With laughter, you can get somebody's guard down, you can open them up to listening to you. They don't feel like they're being preached to or talked down to. I think it helps, it makes really hard to understand information a little more accessible and palatable. And at the end of the day, it makes a movie a little more fun. It doesn't feel so heavy handed."*

***Little Voices* (2003)** is a hybrid documentary and a computer animation film. The film's director, Jairo Eduardo Carrillo, made a number of interviews of displaced children in Colombia's capital, Bogota, during the Colombian Civil War [137]. The core theme of the film runs through the real stories told by the real children in their own voices, but the stories themselves were illustrated initially by the children's drawings and paintings of the scenes they were describing. Then Carrillo took those 2D drawings of characters, scenery, etc., made by the children and turned them into the animated 3D CG models. A combination of the children's art, computer animation, virtual and augmented reality techniques [138, 139, 140, 141] together with the children's voices create an impressive environment and an art piece for the audience that not only engaging, but also preserving the charm, integrity, and energy of the original art work of the children in their 3D animated counterparts [137]. Following the creation of this hybrid documentary film, Carrillo further created an educational game for displaced children, with the same title *Little Voices* while he was staying during his residency at the Banff New Media Institute (BNMI). In Figure 20 are examples of a drawing, then a 3D-redone scene, and the photograph of two children participants.

> *"There is a certain poignancy in the Little Voices project, that can be*

(a) Example 1

(b) Example 2

(c) Example 3

Figure 20: *Little Voices* (2003) Example Screenshots

> *found in the intimate self-portraits drawn by the young people of Bogota recontextualized into a large-scale animated world that is both compelling and deeply disturbing,"* says BNMI director Susan Kennard. *"Carrillo draws our attention to this juxtaposition through a representation of place that is both real and unreal."* [142]

***Born Under Fire* (2008)** is another computer animated documentary by Carrillo that follows the similar style as the *Little Voices*, but this time, it is based on the interviews with and drawings by the new generation of children who were 8 to 13 years old at the time of the interview, and have grown up in midst of violence and chaos in Colombia [143].

***Chicago 10* (2007)** is a unique and unconventional documentary uses motion-capture animation to portray the "Chicago Conspiracy Trial". This is known as a very good and commended example where the visual effects reconstruction type of graphics used for animating the missing footage of the court room scenes of the Chicago Conspiracy Trial proceedings all the way back in 1968 where the animation is nicely blended with some available footage archives back from 1968 in order to accent the development of the story and its emotion more sharply [144]. Some example screenshots from the animation are in Figure 21.

(a) Example 1

(b) Example 2

Figure 21: *Chicago 10* (2007) Example Screenshots

***Ryan* (2005)** won an Oscar at the 77th Annual Academy Awards for Best Animated Short Film. It is a prominent example of an "all-out" graphics documentary. In this documentary, based on a period of life of a Canadian animator Ryan Larkin, the audience perceives the voice of Ryan and the surrounding people, but 3D CGI characters visualizing Ryan and the others appear a bit strange, twisted, see-through, sometimes broken and disembodied, which are humorous or disturbing at times [145, 135] (as, e.g., shown in Figure 22). While the rendering of the CGI scenes in *Ryan* is non-photorealistic, the 3D characters and the virtual environment are very detailed and make an impression of being very realistic despite the fact that this documentary was created not with the use of rotoscoping or motion capture techniques presented earlier, but rather by using a 3D modeling software [9] with some extra modifications and plug-ins to enable the non-photorealistic rendering and mixed perspective and non-linear projection [78]. Scholars classify this as an autobiographical documentary, but non-traditional, because the whole film, even the interviews were turned 100% into 3D computer graphics scenes instead of being filmed by a live-action camera.

Director of this film, Chris Landreth, says that after he learned Ryan Larkin's story:

"There's a lot wrapped up in that, as far as a great story to tell, and I

(a) Example 1

(b) Example 2

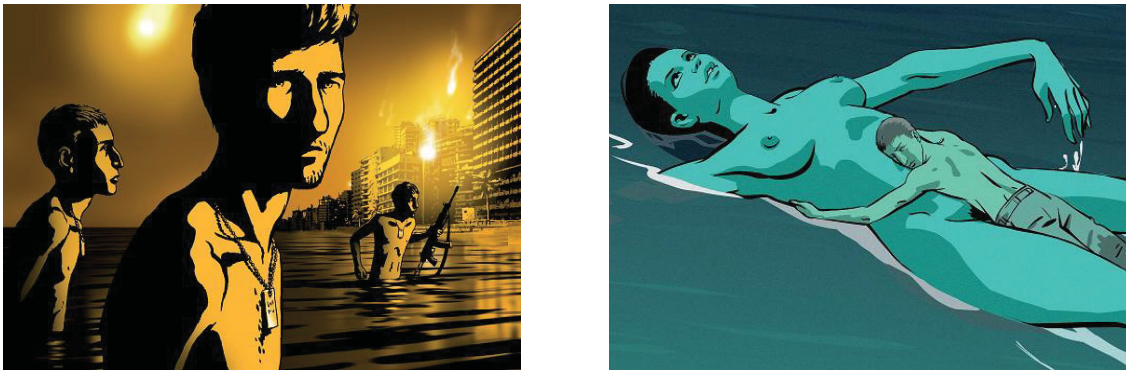
Figure 22: *Ryan* (2005) Example Screenshots

> *wanted to tell that in a way that was as powerful as possible. Making an animation out of a documentary was the best way, in my opinion, to do that.*" [145]

***Waltz with Bashir* (2008)** [146] is a CGI-animated documentary film, which was advertised as being the first feature-length CG animated documentary except the short part in the ending that was featuring the real documented results of the Sabra and Shatila found in an archived footage from the news at the time. The film uses the animation to portray the memory of Ari Folman (the director of this film), in the first Lebanon War twenty years after the war.

In this documentary Israeli director Ari Folman attempts to reconstruct his missing memories from his time as a soldier in the 1982 Lebanon War by using the animation. First, each his drawing was sliced into hundreds of pieces that were moved in relation to one another in order to create the movement. Then, the film was preliminary shot in a sound studio as a 90-minute video, which was then transferred to a storyboard. From there about 2300 original illustrations were drawn with respect to the the storyboard [147]. All those illustrations, which together eventually formed the actual film scenes, were composed using the aforementioned Flash animation, classic

animation, and other 3D technologies [147]—in reality a combination of Adobe Flash scenes and classic animation techniques. Folman by using the freedoms that animation provides to take the file into scenes that could not have been possible to shoot in the traditional way. It also impacts the audience's preconceptions of cartoons always belonging to the realm of narrative filmmaking and attempting to emphasize where the line between the fiction and reality lies [146, 148]. Overall, it took four years for the director to complete the film, which is "a combination of Flash animation, classic animation, and 3D" [148]. Two example screenshots from the documentary are in Figure 23.



(a) Example 1 (b) Example 2

Figure 23: *Waltz with Bashir* (2008) Example Screenshots

This opens up the tools and techniques used in the production process to complement the missing footage. Despite the documentary being mostly animated, the Folman's story was told undistracted and still impacting the audience. On the team the animator was Yoni Goodman, who produced the majority of the types of styles of vivid and stunning animation, sometimes hand-drawn, he did not obscure the main point of the story [149].

> *"For a few years, I had the basic idea for the film in my mind but I was not happy at all to do it in real life video. How would that have looked like? A middle aged man being interviewed against a black background, telling stories that happened 25 years ago, without any archival footage*

to support them. That would have been SO BORING! Then I figured out it could be done only in animation with fantastic drawings. War is so surreal, and memory is so tricky that I thought I'd better go all along the memory journey with the help of very fine illustrators. The animation, unique with its dark hues representing the overall feel of the film, uses a unique style invented by Yoni Goodman at the Bridgit Folman Film Gang studio in Israel." [148]

Some other talented documentary filmmakers and artists who creatively applied computer technology, theatrical elements into documentary film making, such as Robert Legage and his work, *The Image Mill* (see, e.g., Figure 24); Dennis Del Favero with his *iCinema* installation.

(a) Example 1

(b) Example 2

Figure 24: *The Image Mill* (2008) Example Screenshots

2.2.4 Emergence of Interactive Documentaries

Everything has been affected by arising new technologies, including newspaper, surveys, applications, communications, and so on, change from paper format to mobile or computer web-based. The same has been happening to the interactive media, TV, and documentary production. As we explained in Section 2.2.2 and Section 2.2.3, there are several different types of animation and computer graphics (CG) techniques used in the traditional and modern documentary production, such as computer-generated 2D and 3D images (CGI), computer animation.

Moreover, there are even "manipulation" of the data generated by the computer, which is also called computer-human interaction [150, 151, 152]. Interactivity is another important element of the emerging documentary film production, in which other broadcast media cannot compete with. The interactivity of the new genre of documentary film enables the audience to make the decision on what will be going on in the film. They could participate on what should be included in the documentary film and in which order. Today, mass media can be easily used for documentary production with video clips shot by virtually anyone, uploaded to Youtube or similar services. Some documentary-specific online projects allow user control and interactive documentary making from either pre-compiled scenes or video clips or user-generated video content on a particular topic (e.g., weddings, city streets, etc.).

2.2.4.1 Interactive TV

Interactive TV was arguably the first to introduce the interaction aspects into viewers' experience (outside of computers themselves), followed by the interactivity being introduced into cinema and specifically documentaries. The United Kingdom is particularly worthy of detailed study because it is the most well-developed interactive television market in the world [153]. A good classical example of interactive TV was when before Christmas 2000, Sky One announced the launch of the UKs first interactive TV program, *Harrods Christmas Special* [154], which was in itself an

entertainment for audience and additionally offered them interactive shopping experience. It was the pioneering TV of shopping service, which arguably made the broadcasting history in the world. The viewers shop from the world's most famous store while they watch the program of how Harrods has celebrated Christmas from its beginning in 1849 to the present day with their TV hand set. As [155] put it, "the recent advances in the STB (Set-Top Box) technology have introduced real-time video capturing and rich multimedia at consumers' homes." Digital STBs, like TiVo store television content, while the user controls the television flow with an on-screen user interface [156, 155]. Moreover, television content can be augmented with rich computer generated content, like animated characters and Internet information sources. Consumers are starting to have the need for a multimedia experience that seamlessly integrated diverse sources of information and entertainment content [157, 158]. Even before the emergence and widespread adoption of the Internet and the Web, researchers were suggesting an integrated computer-television product [159]. However, immature technology, not enough consumer demand, and the success of Web have been postponing the convergence between the computer and the television [156, 155]. Interactive TV is a hot research topic and a subject of some conferences [159], such as *Euro ITV* and others where the interactive TV also naturally invades the Internet is quite common in people's households and Set-Top Boxes. The interactive TV is relevant as one medium of interaction with interactive TV documentary productions as opposed to web-based or installation-based productions.

2.2.4.2 Interactive Cinema

Interactive Cinema and *Interactive Movie* are not very new terms, which produce new storytelling experiences for audience by combining traditional linear storytelling with interactive digital media. They can be considered as a generalization of interactive TV. Interactive cinema is the evolutionary approach to traditional movie, which gives the audience an active role in the showing movies. Compared to traditional movie, the interactive cinema allows viewers to interrupt the movie from time to time, and to

choose among different possibilities of how the story goes on. *Kinoautomat* (1967) was the world's first interactive movie [160] shown at Expo '67 in Montreal. The audience can change the plot by voting, by pushing buttons at nine points during the film the action stops. A moderator appears on stage to ask the audience to choose between two scenes; following an audience vote, the chosen scene is played. Another example of the interactive movie is *I'm Your Man* (1992), which allows audience to decide which direction the plot should move forward. The film has a story-decision-tree where a choice must be made from a list of options and then eventually it reaches one of the leaf concluding points. A complete research overview on the evolution of cinema in general, modern digital art (to 2003), and specifically including the interaction aspects are well described in the book *Future Cinema* [161].

In recent years, with the overbearing tide of video games, interactive cinema is normally taken to be blended with 3D computer games, which gives viewer a strong amount of control in the characters' decisions, typically of the adventure or quest type (e.g., reincarnation of *The Lord of the Rings: The Return of the King* with the scenes of Peter Jackson's 2003 movie with the rolled into the role-playing game). A good example of such a technique is *Policenauts* written and directed by Hideo Kojima. In this work, a cinematic adventure game with a hard science fiction storyline, allows viewers or players to interact with the game through a point-and-click interface. Video game based interactive cinema, evolving from the mathematical models and procedural programming, becomes alive when some of its parameters are controlled by a viewer. These works generally focus on continuous story playout environments, story authoring systems, and scenarios for interaction [162]. The real-time challenges of the aspect are the most interesting to solve in a less expensive way than traditional approaches.

2.2.4.3 Interactive Documentary

Accompanied with the development of digital media and computer-based technologies, new forms of documentary are challenging the traditional ones everyday. Except

the CGI documentary we discussed in previous section, beside interactive cinema, interactive documentary is another new media type directly related to interactive computer graphics [163], which had been widely used in video game programming. Only since the 21st century, interactive computer graphics has been slowly introduced to documentary production. The related projects are mostly web-based to enable user to not only view but also participate in the making of documentary film. Some pioneers of "Interactive Online Documentary" [164] such as Australian Film Commission (AFC) [165] and Australian Broadcast Corporation (ABC) [166] collaborated with local Australian filmmakers and digital media artists to work on these new documentary projects. Interactive documentary, as a brand new field in new digital media arts, is the revolutionary approach to traditional documentary, which gives the audience an active role in the showing documentary. The notion of interactive documentary has roots in traditional documentary story telling and narrative, and web-based computer-based techniques with user-interaction, interactive TV, and computer games. Some research work has been done in this new field:

- Perspective and cultural overview of new media forms from theoretical and philosophical standpoints [167, 168], and in particularly the interaction aspects of the new media, are described in Rieser and Zapp's book [102].

- Bill Nichols is the only documentary theorist who uses documentary mode as a conceptual scheme to distinguish various styles of documentary film. The interactive mode in documentary theory appears in the introduction of the *four modes* in documentary theory, such as ex-positional, observational, *interactive*, and self-reflexive, as detailed in his work [169].

- Barfield discusses what interactive documentaries could be back in 2003 and the problem of narrative vs. interaction and interactive story telling with 4 main structures [170].

- Tarrant published a very relevant description of technical and non-technical sides of making an interactive video archive by a brother of a person with the

degrading Usher syndrome (who can't hear or see any longer) who in turn made home-shot family audio/video footage spanning across 20 years [171].

- Another relatively recent article details the production process of the documentary *A Golden Age* in England as an interactive configurable documentary (*interactive narrative*) and the use of the "ShapeShifting Media" technology with the technical implementation details and the Narrative Structure Language (NSL) [25].

- Then there is also a similar shift in education and drama portrayal. A recent book [172] describes drama teaching using computer games with the intent to make a memorable learning experience in a simulated environment including the documentation of the research and practice of the approach [173].

2.2.4.3.1 Systems and Tools. Current existing interactive documentary production system and technology:

Diamond Road Online (DRO). DRO is an experimental interactive documentary system with user interface and recommendation systems to present the documentary stories (in a keyword indexed database) of diamond trade allowing semantic links between clips to make a continuous story off those clips [174].

Interactive Drama Engine (IDE). In the journal article [175] the authors implement an IDE based on theoretical foundations of narratives and drama as well as practicality and interactivity of 3D first-person fiction/adventure/etc. games where participants can deeply affect the storyline unlike in traditional games nor documentaries.

Korsakow. An open-source software system, Korsakow [26] in Java is used to build one own's documentaries in tree-like structure. Florian Thalhofer, the Korsakow

inventor says, “linear storytelling, a special behavior, the story has been told same way every time...”

2.2.4.3.2 Prominent Examples. More generally, interactive web-based documentaries are a rapidly emerging medium. Now, we have a more detailed review of the artists’ new approach to documentary filmmaking and the associated technologies found in their works here:

***The Unexplained* (1996)** produced by the company FlogTower could arguably be considered as the first interactive documentary because in its manual of *The Unexplained*, it says, “FlagTower has named this concept the Interactive Documentary, a title which reflects the televisual appeal of our style of production”.

***Jerome B. Wiesner* (2004)** The project *Jerome B. Wiesner (JBW) (1915–1994): A Random Walk through the 20th Century*, developed in 1996 by Golrianna Davenport’s MIT Media Lab on interactive cinema, is part of the “evolving documentary” genre [176]. This “hyper-portrait” introduces the audience to a remarkable man whose life centered on science, government, education and issues of cultural humanism [176]. In this hyper portrait (which runs on the World Wide Web), audience is invited to explore the 20th century through an extensible collection of stories about and recollections by the central figure. Audience who knew JBW are also invited to share a memorable story with the growing society of audience [176].

***Traveling in Zagori* (2004)** Another similar project developed by the MIT Media Lab is *Traveling in Zagori*. Through visual and textual snapshots of the landscape, architecture and people, the audience is invited to construct a story about a local legend while discovering aspects of Zagori’s extraordinary history and legacy [176].

***Man With a Movie Camera: The Global Remake* (2007)** is the best example to represent the concept of interactivity of documentary [177]. It is inspired

by Vertov's *Man With A Movie Camera* made in 1929. The original documentary film records the progression of one full day synthesizing footage shot in Moscow, Riga, and Kiev. It begins with titles that declare it "an experiment in the cinematic communication of visible events without the aid of intertitles, without the aid of a scenario, without the aid of theater." [178] Vertov's footage was shot in three different cities, the industrial landscape of the 20's. What images translate the world today? The current project is a participatory video shot by people around the world who are invited to record images interpreting the original script of Vertov's film and upload them to the website. Anyone can upload footage and contribute as part of a worldwide montage. The artist Perry Bard says,

> *"Vertov's 1929 film is a great point of departure for the Internet because it has so many dimensions from the documentary to the performative to the effects along with its use of an archive which translates to a database and it's a film within a film... going global was obvious and the rhythm is very contemporary, there's no shot in the film longer than twenty seconds. It seemed like a perfect vehicle for global input and in keeping with Vertov's intentions as a filmmaker."* [178]

***Out My Window* (2010)** is the most recent item and outstanding example of a web-based and database-driven interactive documentary by the National Film Board of Canada's project *Out My Window* designed for digital storytelling [179]. This project explores our urban planet through through highrise windows. Later, this 360° web-based project is transformed into the *Highrise StorySpace* installation, which extends the original documentary into the larger than life-size physical environment. The installation takes the audience on a journey into the center of a "spatialized cinematic experience". In Figure 25 are two example illustrations from this work.

***Ceci N'est Pas Embres* (2012)** is a Korsakow film, created by the ARC team (Adventures in Research/Creation), co-directed by Matt Soar. This project is a

(a) Example 1

(b) Example 2

Figure 25: *Out My Window* (2010) Example Screenshots

database-driven, diary-style to tell a story about a family from Quebec moving to southwestern France. ARC is a research group co-directed by Professors Monika Kin Gagnon and Matt Soar at Concordia University, Department of Communication Studies. The audience could interact the system which include personal testimony, animated scenes, and soundtracks [179].

***Sea Monsters* and *U2* (2008).** Another 3D aspect—stereoscopy—that comes with the new stereoscopic movies documenting potential life of animals millenia ago, or the animals now in the see or simply documenting a concert in 3D, such as done by IMAX for their animations and stereo-enhanced motion pictures such as *Sea Monsters* (2008) and *U2* (2008) concert (and the previously mentioned *Hubble*). The audience

in this case, while swimmingly passive and watching, interacts with the show through the stereoscopic illusion their eyes receive as a part of the show enhancing the perception and impression of the show. One can argue it is one-way interactive, but nonetheless should not be neglected. Needless to say, the stereoscopic shows can also be made interactive, and even more so in the future with the techniques presented throughout this and other chapters.

2.3 Performance Arts and Theatre Production

Theatre is not only an art form to give the actors an opportunity to present a live performance in front of a passive audience within a specific space, but also a structure of such space itself.

The ancient Greek drama as the earliest history in theatre, could be traced back to 550–220 BC[9]. There were some scenic techniques that had been used in Greek theatre as early as the fourth century BC, such as, e.g., a *mechane* was used to lift flying actors from or onto the stage; trap doors or openings in the ground to bring actors onto the stage; *Pinakes*, pictures hang or built on the stage to create scenery; a *Mask* was also a significant element in ancient Greek theatre.

Western theatre, which is derived from ancient Greek drama and developed and expanded under Roman theatre, has been a very important art form not only for entertainment, but also had a significant impact on their countries's cultures.

Asian theatre also has a long and complex history, for instance, Chinese Shang theatre has more than 2500 years history. There were some traditional techniques used in Asian theatre. Masks, which have also been widely used in Asian theatre performance, especially in today's Japanese Noh performance, are still one of those very important performance elements [180]. Other technology used in conventional theatre stagecraft such as puppetry, sound, theatrical fog are also still quite popular

[9] `http://en.wikipedia.org/wiki/Theatre_of_ancient_Greece`

in modern theatre production. Today Chinese theatre is not only limited to traditional Chinese opera, but also refers to modern Chinese drama. For example, in Beijing Opera (one of the traditional Chinese opera), facial makeup and costumes are extremely important to distinguish historical characters, their roles and social ranks in the scene.

Theatre focuses on live performers enacting in front of the audience within the same space. However, it also involves other contributions from a playwright, costume designer, makeup artist, or set and props designer–all introducing some kind of "technology", albeit often primitive, into a theatre setting.

Subsequently, we review the theatre background from the classic point of view in Section 2.3.1, Asian theatrical influence in Section 2.3.2, and the current digital theatre further in Section 2.3.3.

2.3.1 Classic

In the 20^{th} century, there have been different voices about the relationship between theatre and conventional non-computer technology: how to use technology in performances, and how much technology should be applied in theatre productions.

Some important literature and theories include such as those of Artaud's *Theatre of Cruelty* [181], Grotowski's *Poor Theatre* [182], Brook's *Holy Theatre* [183], and Bertolt Brecht's *Epic Theatre* [184]. And some of these theatre practitioners were against the use of technology in theatre production, most notably Grotowski (who is considered the father of contemporary theatre). He declared in his book *Towards a Poor Theatre* [182] that theatre should not, because it could not, compete against the overwhelming spectacle of film and should instead focus on the very root of the act of theatre: actors co-creating the event of theatre with its spectators. I support his position, because to me, the performance of actors, and the moments of their communication to audience are the most precious elements in a theatre performance. However, at that time he could not have imagined what we have today: computers, the Internet, cellphones, e-newspaper, iDevices, and other different types of new

media technology, which all have a potentiality to influence theatre performance noticeably. If technology could only enhance actors' performance and broaden audience imagination, rather than overwhelming audience with unrelated digital stuff, why cling to those theories?

On the other side, an artist such as Artaud, was very interested in movements, sounds and encouraged that theatre needs to "recapture from cinema, music-hall, the circus and life itself, those things that always belong to it" and indicated that "our sensibility has reached the point where we surely need theatre that wakes us up heart and nerves" [181]. His theatre of cruelty whose sentiments could still easily apply in today's cyber theatre space. Artaud's theory to me, seems holy, an awesome power that is both a creator and destroyer. Whereas Grotowski's reference to theatre generally applies to the dedication of the actor, in giving himself as a gift over performance which is transcendent in a much more human-sized way. Moreover, he insisted that there was no point in trying to compete with film but that theatre should rather convert back to its roots, "If it [the stage] cannot be richer than the cinema, then let it be poor." [185]

Peter Brook's recent research on theatre says, "Holy Theatre for short, but it could be called The Theatre of the Invisible-Made-Visible: the notion that the stage is a place where the invisible can appear has a deep hold on our thoughts." [183]

The German artist Brecht, the most significant artist in Germany (Bavaria at the time), was very interested in Chinese and Japanese traditional theatre. His epic theatre theory, in particular has been influenced by the Chinese theatre, also employed technology to theatre production. Brecht expected the audience to be aware that they were always watching a play and always be rational so that could provoke their self-reflection, "It is most important that one of the main features of the ordinary theatre should be excluded from [epic theatre]: the engendering of illusion" [184]. Moreover, he applied some practices of using bright lighting, loud sounds, images and texts to interrupt audience in order to remind them that "the play is a representation of reality and not reality itself." [184]

At the beginning, one may get the impression that they raised up different voices as to whether technology and theatre performance could coexist; however, to me, their positions are not in conflict at all. The acting and performance are always the most important element in theatre production; the role of technology is to widen, enrich, and enhance the experience of live performance instead of obstruct, replace, or destroy the art of performance.

2.3.2 Asian Influence

There exist several theatrical forms in Asia, such as the earliest Sanskrit drama (8^{th} Century BCE) in India; Beijing Opera in China; traditional Kabuki puppet theatre and Noh drama in Japan. The story *A Treatise on Theatre* is the most complete work and evidence of Sanskrit drama. It addresses "acting, dance, music, dramatic construction, architecture, costuming, make-up, props, the organization of companies, the audience, competitions, and offers a mythological account of the origin of theatre" [186]. There are four main types of traditional theater in Japan: *noh*, *kyogen*, *kabuki*, and *bunraku*. Each of these forms of theater performance is very distinct and unique from the others. In Noh theatre (e.g., Figure 26), most of the characters in these plays are concealed by masks, and men play both the male and female roles; however, in Kyogen theater, which is performed between Noh's acts intermission, actors do not wear masks. Bunraku theater uses puppets. The puppets about three to four feet tall are controlled by puppeteers who dress completely in black.

Shang theatre as early as 1500 BC could be considered the earliest Chinese theatre, which often involved music, clowning and acrobatic displays. During the Han Dynasty, shadow puppetry first emerged as a recognized form of theatre in China. The rods used to control puppets were attached perpendicularly to the puppets' heads so that puppeteers were not seen by the audience when the shadow was created. In the Tang Dynasty, Emperor Xuanzong formed an acting school, the Pear Garden, to produce a form of drama that was primarily musical. In the Sung and the Yuan Dynasty, there were many popular plays involving acrobatics and music with a four- or

Figure 26: Noh Drama Stage

five-act structure. Yuan drama spread across China and diversified into numerous regional forms, the best known of which is Beijing Opera (e.g., Figure 27), which is still popular today. It combines music, vocal performance, mime, dance and acrobatics into one entertaining show.

Figure 27: Beijing Opera *San Cha Kou* Example

Facial Makeup. Facial makeup has obtained the reputation as "painting of the heart and soul", which enable the audience to get a glimpse of the characters' inner

world through their symbolic facial makeup. It utilizes the color of red, purple, black, white, etc., with each color representing a unique character stereotype: red symbolizes utter devotion and loyalty; black represents faithfulness and integrity; white implies craft.

Costumes. Costumes could distinguish the rank of the character being played, for example, Emperors are in yellow robes, and high-ranking characters wear brilliant colors. The gowns for high-level ranking female characters usually have water sleeves, and the characters of no rank wear simple clothing without embroidery [187]. Actors use the long flowing sleeves to facilitate emotive gestures (about hundreds of gesticulations), such as sadness and shyness are expressed by one hand pulling another water sleeve to cover the face; raising and putting up two persons' water sleeves to embrace each other.

Modern Chinese Drama started to develop in 1907 in Shanghai. Unlike traditional Chinese opera, it realistically reflected the changes in the lives of Chinese before and after the founding of New China. A number of Chinese playwrights have realistically portrayed the lives of common folks greatly affected by Western playwrights such as Shakespeare, Chekhov, and Moliere. The most memorable Chinese plays include *Teahouse* (Figure 28), *The Thunderstorm*, and *The Family*. Younger generations of playwrights have tried to develop a more modern style. The most innovative theatre director, Meng Jinghui, who employs a lot of theatrical techniques, such as electronic music, acoustics, lighting, and novel stage settings in his plays in addition to the new ideas in order to impress the audience both mentally and visually. However, Modern Chinese drama is still quite ignored in the West in favor of traditional theatre.

2.3.3 Current Theatre and its Technology in Digital Era

Digital theatre, *cyberspace theatre*, *multimedia performance*, *interactive theatre*, *virtual theatre*, and more possible terms associated with today's theatre, have all appeared based on the new technology and digital media innovation. We are in the new digital

Figure 28: Modern Chinese Drama *Teahouse* Example Screenshot

era, performers, audiences, and space—are they still all inter-related as conventional theatre proposed? Is it necessarily that actors and audience in today's live performance are within the same physical space? What if performers are not real human beings? What is the new definition of THEATRE now or for future?

In Laurel's very early work in 1991, *Computers as Theatre* [188], she applied her knowledge of theatre to Computer-User interface design. She says, "In many ways, the role of the graphic designer in human-computer interaction is parallel to the role of the theatrical scene designer." Both create representations of objects and environments that provide a context for action. She thinks of the computer, not as a tool, but as a medium. Moreover, she examines Aristotle's six elements of theatre, "action, character, thought, language, melody or pattern, and spectacle or enactment" in his *Poetics* [189] versus that of HCI. She proposes the reader various experiments addressing touch, smell and taste and notes the possible site-specific interactive plays and performance art model.

Another very extreme artist in this field is an Australian performance artist, known as *Stelarc*, whose research is in *Cyborg Theatre*. Cyborg theatre uses cybernetics as a method, and establishes the relationship between a human being and technology [190].

Stelarc's work, *Stomach Sculpture* uses nanotechnology, "to insert an art work into the body." As described by Stelarc, the viewer not only would observe inside their body, but are also an "artist" himself. His intention is to use cyber-systems to break the barrier of skin in order to extend his body's performance. He is both an artist and his body, even his organs, is an art work at the same time. Stelarc believes, "New technologies tend to generate new perceptions and paradigms of the world, and in turn, allow us to take further steps." [191]

There is a significant amount of innovative multimedia artists, performers eager to explore the possibilities that multimedia could bring to theatre production. There are conventional techniques for theatre productions, such as stage lighting, sound effects, set design, costumes, and special effects, such as fog and explosions, still have been widely used for most of the theatre production companies. Likewise, digital technology helps to revolutionize the design of these theatrical techniques in order to make the theatre the true sense of LIVE performance.

Some of the multimedia artists who apply digital technology to theatre productions, such as Chris Salter with his real-time audio, image, gesture in responsive space [192], Natasha Tsakos's live 3D animated show [193], Marc Hollogne's cinema-theatre, in which one cannot decide between a movie and a play [194], Silhouettes Dance Group's shadow performance combined with photographs [195], and a Taiwanese artist Stan Lai, who applied many creative staging innovations, such as bringing a live camera on stage and not only present the scene of the stage, but also broadcast another dimensional image directly from various cameras. We catch a glimpse of their projects and exploited technological advances:

- Salter's artistic creativity focuses on "dynamic and temporal processes over static objects and representations". For instance, in his work, *SCHWELLE II*, a live dance theatre performance, a solo dancer experienced a traumatic transformation from death to rebirth. During the live performance period, the dancer wears several wireless acceleration sensors, the input of which dynamically affected the audio and visual performance output based on the sensor data

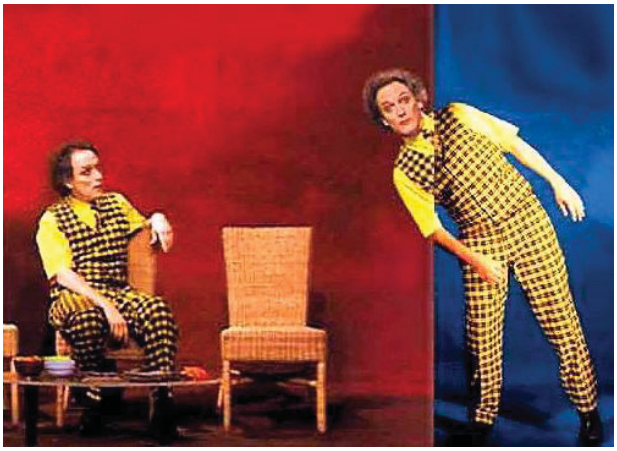
obtained from the performer.

- Natasha Tsakos' *Up Wake* is a live performance with only one actor and a bare stage lit with perfectly synchronized computer graphics imagery, digital sound effects to demonstrate a cartoon character's dream and wake throughout his day. The projection techniques of CG animation and graphics have been widely used in her work, which is a very important solution for environment/space transformation and time vicissitude [193]. In Figure 29 is the example illustration from her work.

Figure 29: *Natasha Tsakos Performance* Example Screenshot

- Most Marc Hollogne's works include a cinema screen on the theatre stage, projected with filmed sequences. Everything happened on the stage, the live actors' performance and broadcasting stories on the screen, have to be extremely well-synchronized as if the story continually happens within the same physical spacetime. The setup requires a great deal of precision on dialogue, gestures, and motions, for example, when the moment actors walk in and and out from the side of the screen to behind while the scenarios about the actors are continuing smoothly on the screen, one could not tell what is real in the film and what is not real on the stage. In Figure 30 are two example illustrations from

his work.



(a) Example 1

(b) Example 2

Figure 30: *Marc Hollogne Performance* Example Screenshots

- Denis Marleau, a director, production designer and creator of stage installations, is a major figure in Quebec theatre. His work is distinguished by innovative use of audio and video technology. A unique example is the play showed in 2002, *Les Aveugles*, in which twelve mannequin faces were animated by prerecorded video without any actor on stage. In about summer of 2011, Denis Marleau and Stéphanie Jasmin collaborated with Jean Paul Gaultier's fashion show exhibition in Montreal Fine Arts Museum. They placed 30 mannequins throughout the galleries and animated these faces through an ingenious projection system with prerecorded videos [196]. In Figure 31 are two example illustrations from

his work.

(a) Example 1

(b) Example 2

Figure 31: *Denis Marleau Performance* Example Screenshots

- The inspiring performance from The Silhouettes dance group makes the shape of the pictures with dancers' bodies and then projects the photographs mapped along to appear on the stage. This very traditional technique which has been widely used in shadow theatre as early as the 18th century, now is innovatively applied to The Silhouettes's performance combined with new digital technology. In the performance *Believe*, a number of figures were created on the stage by actors' shadows: White House, a ballet dancer, swimmer, and an astronaut on the Moon; following that, the real photographs of the figures projected on the canvas blending with the shadows. In Figure 32 are two example illustrations

from this dance group's performance.

(a) Example 1

(b) Example 2

Figure 32: *The Silhouettes Dance Group Performance* Example Screenshots

- Stan Lai is the most celebrated Chinese language playwright and director for not only his memorable works, but also creativity of new genres and staging innovations. In his recent work, *Watch TV with Me*, a seven-act drama incorporates several stories involving both television and everyday people in mainland China since the 1980s. Lai used the most advanced sound and lighting technology to create a time and space change with development of television. The montage effect, the interactive theatrical performances between live performance and virtual characters on the same stage, and the multimedia concept leave audience with the burdens of memory, history, longing, love and appreciation

of the power of theater itself.

To summarize, the technology itself has no vitality without artists' creativity, without the connection to the audience, without the context of society and humanity. Recent computer technology developments have improved the theatre performance, have extended theatre performance and expanded it to more dimensions. It has closed in on the gap among physical world, human body, and digital world. I would conclude that "Poor Theatre" is never actually poor, it is the richest medium.

2.4 Technology

In this section we review the relevant technological aspects pertinent to our work in terms of visualization interaction terminology, hardware, and software. We cover the recent interaction hardware in Section 2.4.1, viewing and projection aspects in Section 2.4.2, the related application programming interfaces (APIs) and libraries in Section 2.4.3, followed by the credits in Section 2.4.4 to the open and shared source communities for providing programming examples to individual features, components, or effects that we got inspired from, relied on, re-used, improved and built up like supporting puzzle pieces to the development of the prototype installation work in the chapters that follow.

2.4.1 Devices

Advanced interaction devices emerged recently and became accessible and affordable to many increasing the landscape of human-computer interaction in a variety of disciplines. The devices have to do either with some kind of motion, gesture, posture, etc., tracking or haptic force feedback allowing for the sense of touch of the virtual in the real [197]. Any such tracking and force feedback garnered a number of research contributions such as [198, 199, 200] and many others. We subsequently briefly review two such devices that we use in our research and creation: Microsoft Kinect (see Section 2.4.1.1) and Novint Falcon (see Section 2.4.1.2).

2.4.1.1 Microsoft Kinect

Microsoft Kinect (see Figure 33) is a modern and affordable powerful sensor device with multiple camera and audio inputs that allow exploring new ways of media interaction accessible to a large audience base and provides for nearly limitless ways of creativity for artists, a truly formidable device. It goes beyond devices like Wii in such that the motion tracking does not require the users holding other devices by providing robust skeletal tracking with visible and depth data; it also can act as an audio source.

Combined with its API and SDK, besides its common use for motion tracking in gaming, it was used for stereoscopy, security tracking, and fashion projection on people, (see the Wikipedia page [201]).

Since the introduction of the device by Microsoft for XBOX360, a number of APIs and drivers came into existence because Microsoft at the time hasn't released its official SDK yet, being in Beta testing and only for Microsoft OS platforms. (Microsoft have released their own 1.0 in February 2012 along with a somewhat more expensive Kinect for Windows with the developers beta version from last summer and now it is in 1.5.x series.) Open source versions from OpenKinect and OpenNI by PrimeSense were also available earlier to make the device accessible from other operating systems. For example, OpenKinect's open-source library/API/driver are available for Windows , Linux, and Mac OS X (uses GLUT, etc.) [202].

Figure 33: Kinect Device

The most recent Microsoft SDK supports primarily C++ and C# (and VB) and provides skeletal tracking and viewing [203] and many other recent code samples and the toolking from [204] to deal with depth, audio, color data streams. In Section 4.2.2.1.3 we elaborated on the actual API used in a part of this work from [203]

and primarily in C#. The SDK provides Kinect Studio for depth debugging [205]. Additionally, their documentation provides Human Interface Guidelines and design principles for skeleton tracking, depth processing, speechs processing and the like [206].

2.4.1.2 Novint Falcon

The inexpensive haptic Falcon device (see Figure 34) is used in this research as another means of tangible interaction. It has means of integration with OpenGL/GLUT and others. Its main goals include [3, 207]:

- Give the user a sense of touch by applying forces, vibrations, or motions to the user
- Used for medical research, art design and video games
- Novint's Falcon, the first inexpensive 3D touch device
- Three-dimensional degree of force feedback allow the haptic simulation of objects, textures, recoil, momentum, and the physical presence of objects in games

Figure 34: Falcon Haptic Device [3]

Falcon comes with the SDK, drivers, examples, and integration with various Windows platforms via their new community tool F-Gen and its SDK [208, 209, 210].

2.4.2 Viewing and Projection

Viewing and projection techniques have to do with various technologies capabilities that they provide to the users for eventual rendering and projection for performance, interaction, or other purposes.

2.4.2.1 Depth

The notion of depth is important in a number of aspects. It is used in stereo viewing and image generation, as well as 3D gesture/motion/posture recognition via either two cameras observing the same target within interaxial distance and/or with the infrared mapping. Devices such as Kinect provide *depth frames* alongside the color image readings under a specified resolution and consisting of 16-bit values of the depth sensor readings within its field of view. These frames can later be used for green screening, skeleton tracking, and other applications. Since depth frames have no natural color assigned to them, to be visualized, the depth pixels (*dexels*[10]) have to be mapped somehow to the visual domain pixels first. A depth mask can help, e.g., with the green screen effect.

2.4.2.2 Green Screen

The notion of green screen[11] more formally known as chroma key screen, is the idea of having a color screen present behind a host or a performer, a color that is unlikely to be present on the host or a performer, can be replaced during production with static or moving imagery as if it were a true background originally when viewed or projected. The notion of depth allows us with the devices such as Kinect to implement such a green screen like effect dynamically in real-time [203] and use it in a real-time installation and performance. Here the depth frames would serve in part as the "green screen".

[10] `http://en.wikipedia.org/wiki/Dexel`
[11] `http://en.wikipedia.org/wiki/Green_screen`

2.4.2.3 Skeleton Tracking

Not strictly viewing and projection but in the context of Kinect it helps with green screen abstraction, depth, and motion tracking making in the viewing of avatars, skeletons, or green-screened video feed possible in real-time. Skeletons can also be visualized, of course. Tracking skeleton and joints allows for interaction of the virtual character as well as filter out non-player pixel data to help with the reduction of the depth/color data/mask processing. Kinect allows about 2 skeletons but 6 depth-frame "players" to be tracked that way.

2.4.2.4 Stereoscopic

Stereoscopic viewing and project has to do here either with traditional anaglyph (red-blue) glasses or a 180° projection screen (see Section 2.5.2) as opposed to a 3D graphics rendered on a 2D monitor screen or a traditional projection.

2.4.3 APIs

We briefly review various APIs used in this work.

2.4.3.1 OpenGL

OpenGL[12][211, 212] is an industry standard API and a library for cross-platform graphical processing, rendering, animation, GPU programming, visualization, game development, and the like (starting with version 1.0 in 1992 and the most recent is 4.3 in 2012, including support for OpenGL ES 3.0 for mobile devices running iOS and Android). It is used on desktops and mobile devices by various vendors and has bindings to many programming languages. It implements a programmable graphical processing pipeline enabling various CG algorithms and techniques. The pipeline works to transform initial 3D space virtual world objects (their geometry, lighting and texture information, etc.) to rasterized pixels rendered into the final image on

[12] `http://en.wikipedia.org/wiki/OpenGL`

the computer screen. There are a number of books, tutorials, rendering engines, and publications that deal with OpenGL [211, 212, 213, 32, 31, 214, 215, 216]. There are a number of common extensions to OpenGL that work together with such as GLUT [217], GLUI [218], and others.

2.4.3.2 Graphics Processing Unit (GPU) and Shaders

Since around the year 2000, CG has placed more emphasis on real-time photo-realism. Gaming and 3D cinema are the driving forces to such development of 3D acceleration hardware performance. GPUs are seen as co-processors of the main CPU and general-purpose computing applications are being developed to use GPUs (so-called *General Purpose GPU Computing*—GPGPU). With the increasing power of mainstream programmable GPUs made the SGI type of graphics less pertinent. NVIDIA and ATI are presently dominating the consumer graphics cards with GPUs even in commodity hardware. They are also major contributors to the Vertex and Fragment extension programs' specifications alongside NVIDIA and Microsoft [88, 89, 215]. With the GPUs a lot of new features have been appearing very fast such as programmable pipelines, floating-point support, and hardware occlusion support [73, 74, 219, 35]. Especially, in 2002 the video game industry throughput surpassed the film industry.

2.4.3.3 XNA and HLSL

XNA [13] is a Microsoft framework and toolset originally designed for XBOX and simplified game development. Later it was generalized to Windows and Windows Phone with the latest version being 4.0 [220, 221, 222, 223, 224]. It runs on top of a lower-level API of Direct X (a Microsoft proprietary equivalent of OpenGL) and uses primarily C# bindings to all the related libraries for the manged code requirements by the XBOX and Phone that are easier to program with C# than C++. We adopt XNA because a large number of examples with easier programmability and rapid prototyping for originally came out for Kinect in C#. XNA also has a lot of internal

[13] `http://en.wikipedia.org/wiki/Microsoft_XNA`

support for media content built-in and shader support via Microsoft's equivalent of GLSL described earlier—HLSL—the High-Level Shader Language [14] that provides us with various GPU programming effects we use in the installation.

2.4.3.4 Kinect SDK

Kinect SDK provides C# (and since version 1.5, C++ and VB) bindings to its API [203, 204] nestled under the `Microsoft.Kinect` [225], which provides all the classes, device status, access to color, depth, skeleton, and audio streams from the Kinect as well as access to its tilting motor that XNA applications can then process and use for advanced mode of interaction and artistic performance.

2.4.3.5 BASS Audio Library

BASS for audio processing for visualization is a fairly popular tool [226, 227] with bindings to several languages and frameworks, including a .NET version for C# and XNA with the visualization–`TalkShowHostXNA` [228]. It loads an audio file and uses signal processing techniques to extract spectral characteristics of a tune, such as beats and their frequencies so the animation can run to the beat of the music.

2.4.4 Code Samples

The majority of the prototyping work in this thesis relies on the great examples posted online of single or multiple effects or the use of APIs under permissive or open-source licenses. The author Song expresses her gratitude to the creators of the examples and providing their source code to learn from and re-use enabling rapid prototyping of the artistic work presented in this thesis. Specifically, we credit the following works that inspired, or got re-used and enhanced during this thesis, mostly in chronological order. These works are also cited inline where appropriate. You could find them in the reference here [229, 217, 230, 231, 203, 232, 233, 234, 235, 236, 228, 207].

[14] `http://en.wikipedia.org/wiki/HLSL`

2.5 Virtual and Augmented Reality for Medical Research and Beyond

This section reviews an example of the combined multidisciplinary research experience and collaboration [10, 7] that is of relevance and has given rise to some ideas and inspiration in the follow up work. It is a brief research experience report as to work with, experiment, and develop computer graphics environments for a real-world VR application.

My first encounter with VR was when doing a research survey job in mixed reality, which refers to the merging of real and virtual worlds to produce new environments [237, 138, 139, 141]. Then, I got the opportunity to work as an operator of the Virtual Reality CAREN/Motek equipment at the *Pain, Mind and Movement* research laboratories at the Constance-Lethbridge Rehabilitation Center, directed by Simmonds during 2008—2010, and to use virtual tools to better understand and manage pain and movement difficulties. The research work for students included assisting them with Virtual Reality Research. The major equipment included a virtual reality suite interfaced with a self-paced instrumented treadmill on top of a motion platform with a motion capture system. It was inspiring to see Dr. Simmonds design experiments and use the state of the art VR equipment in innovative research to disentangle the mental, physical, and social components of pain and its impact [10]. Dr. Simmonds's clinical research greatly benefited the clinical community in the rehabilitation center and elsewhere.

The domain of the research in Computer Graphics is of direct relevance to, and plays a very important role in, virtual and augmented reality [141] research, allowing users to interact with the computer-generated virtual environment at elevated sensory levels. Associated gaming, motion capture, stereoscopic effects etc., used in the VR lab are also the very related and relevant subjects in Computer Graphics and the new interactive media.

2.5.1 Overview

We have done a *Pain and Performance in Virtual Reality Environments: A Pilot Feasibility Study* [10] alongside with the prototype development that included the CAREN VR system, the split-belt instrumented self-paced treadmill, the large panoramic screen being used in rehabilitation research to test how experimental pain threshold differed across VR environments with the overall purpose of decreasing pain and improving movement.

Of relevance to this work we extrapolate from some of our experience in and propose further possibilities of 3D (stereo and non-stereo) computer graphics techniques on perception of pain and rehabilitation in a virtual reality (VR) system setting and transposing this into a preliminary discussion of going beyond the medical research into the realm of interactive documentary production.

2.5.2 Related Work

We subsequently outline the research and research work done in various degrees of relevance to the virtual reality, computer graphics, medical and cinematic documentary research. A list of the core sources or works considered includes the below with the rest cited [238, 239, 240, 241].

VR and Medical Research. In this body of work, researchers did scientific and clinical studies with an expensive head-mounted display for stereoscopic effects to demonstrate the impact of virtual reality environments on pain reduction; more specifically that manipulation of optic flow speed in such a display was consistently and continuously influencing the speed and locomotion of individuals and that it can improve walking. They then interfaced a VR environment with a walking treadmill, which further expanded to link the perception of lower body chronic pain while walking to psychological perception rather than physical condition and suggested the VR trials for pain reduction do work and are recommended. Currently such VR applications are used by Dr. Simmonds and her colleagues to reduce the perception of pain

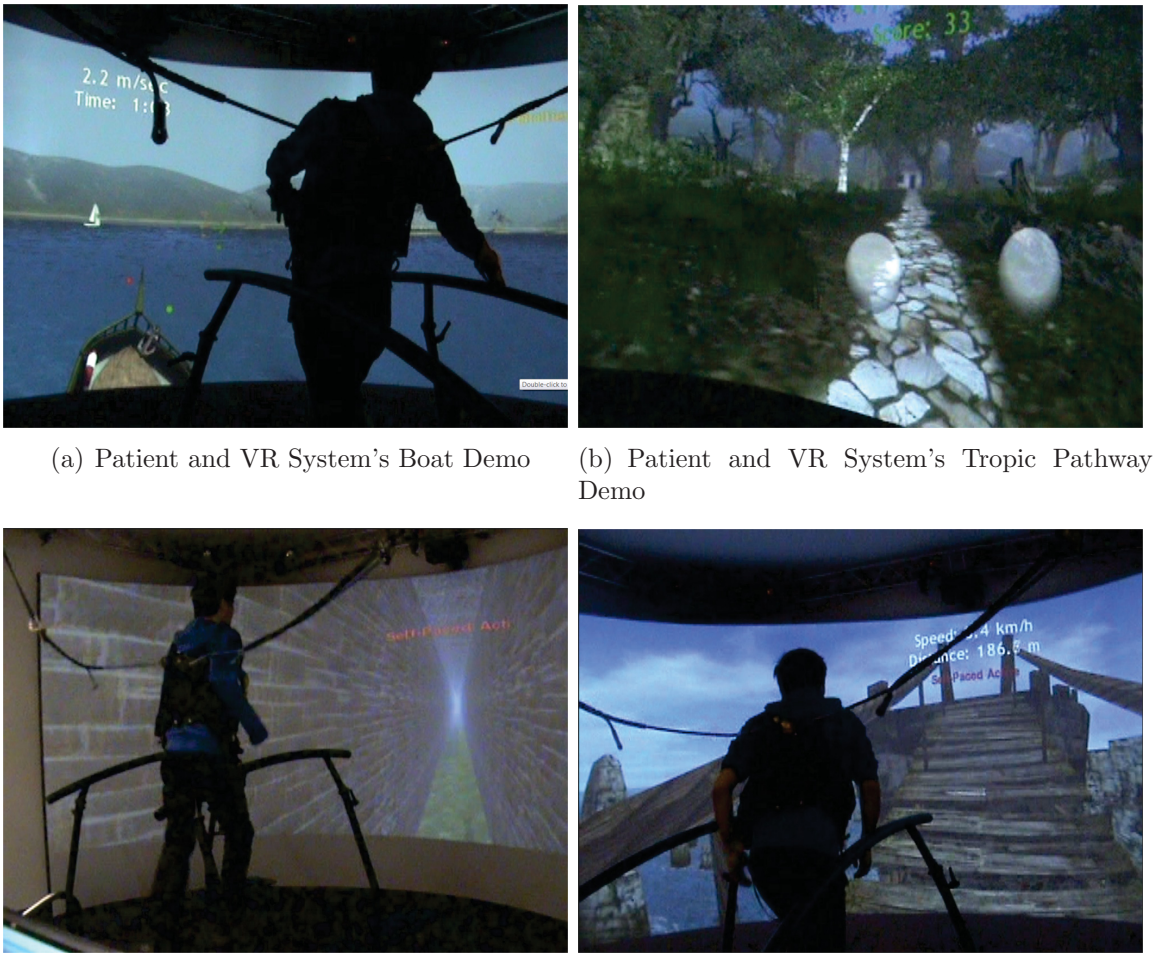
and improve active movement [242, 243, 244, 245, 246, 247, 248, 249, 250, 251].

The VR-treadmill interface used at the time, while offered a 180° projection screen, was not stereoscopic [12, 7, 252]. Stereoscopic virtual reality (VR) environments for medical and rehabilitation research are some of the applications where our own `stereo3d` plug-in [7] could be used. It has a potential for such a use as a cheaper alternative to the expensive head-mounted display units. It is feasible that provision of a more affordable stereoscopic solution would be effective and have clinical and research utility by requiring overall less resources (hardware, software, etc.) to setup and use. This would certainly increase its availability to many in-clinical and research environments [12, 7, 252].

Implementation. We successfully implement a VR virtual walk, *Endless Road*, including a successful implementation of the self-paced functionality with other applications [12].

CAREN and Applications. CAREN is the core system behind the hardware and software setup from Motek used in this research. It gathers readings from the cameras, renders 3D graphics, has an API to build VR applications against, and so on. Our experience involves managing and operating the system, its licensing, logging, wireless setup, maintenance of the manual and other resources, training of, collaboration with and demonstration to various people and organizations and more importantly patient examination with some initial pilot results [10]. We made a simple application and test wireless controller with CAREN were troubleshooting bugs in several applications, such as Maze, random target; testing self-paced modules, and its integration with Vizard3D [253]. We also acquired Tracking Tools 2.0 beta 4 to be able to have the handle on the tracking data as well as configure VR computer with multiple monitors for optimized utilization and viewing and troubleshooting. CAREN software itself is based on the open-source 3D rendering engine OGRE3D [254] and is written in C++ [12].

There are several applications that came with CAREN for the Motek VR system



(a) Patient and VR System's Boat Demo

(b) Patient and VR System's Tropic Pathway Demo

(c) Patient and VR System's Endless Road Demo

(d) Patient and VR System's Bridge Demo

Figure 35: Various CAREN Demo Examples

for the medical research and one developed by us. Example screenshots of a patient navigating a virtual boat by leaning on the treadmill is in Figure 2.35(a), walking a tropical pathway "fighting off" incoming flying objects in Figure 2.35(b), our own "Endless Road" in Figure 2.35(c), and a high bridge walking is in Figure 2.35(d). Most applications require the patient on the treadmill walking and leaning as well as arm motions with markers attached to perform the research and capture the data [12].

Applications used as a demo, education activities for a University class, the possible project in Virtual hospital, and the patient testing. A pilot study was done

to collect a baseline for the healthy individuals prior to moving to the patients with chronic pain [10, 12].

2.5.3 Summary

While covering a wide spectrum of activities, tools, and techniques, by working at the Dr. Simmonds's VR lab for pain and movement research, expanded documentary production, VR and graphics techniques for simulation, animation, modeling, and stereoscopy gave an unique insight on the immediate future projects to undertake and to advance the research [12].

In some relevance to the conducted research there are recent works [10, 7, 8, 99] that we plan on investigating further and find a possibility of collaboration [12].

Finally, a similarly set up VR lab resources can also be used not only for medical research, but also for documentary film and art extending the usefulness and utilization of the expensive equipment and software in other dimensions. Further, in Section 6.5.4.4 we describe some such plans of the audience immersion and interaction with and feedback from the documentary film and computer graphics art [12].

Chapter 3

Toward Realtime Jellyfish Simulation in Computer Graphics

My research in Computer Graphics (CG) concentrates around physical based softbody simulation because it is the continuation of the research and development originated in my master's studies [5]. This work has also resulted in offshoot projects/applications that use the softbody simulation library, specifically modeling and animation of a synthetic jellyfish in 2D and 3D space.

The Softbody Simulation System project itself has been further redesigned and developed from my master's studies [5] in three primary sub-directions: (a) optimizing its software simulation framework, (b) generalizing its design and requirements into similar types of systems to further the extensibility and usability, (c) and then augmenting the design to employ advanced rendering technology and algorithms, integrate the softbody objects with haptic devices, and make use of it for more artistic applications, such as the real-time softbody interactive jellyfish prototype (the culminating case study of this chapter).

3.1 Overview

The Softbody Simulation System and its framework, just like requirements engineering, are a moving target, so the contribution we come up with here simply cannot cover all of CG-visualized interactive simulation systems' needs, but we provide a basic foundation instead with the goal of building upon it in our future work as well as that of other researchers in the same field [23].

This is also an academic project that had not been rigorously specified and designed prior to its initial realization due to its relatively small scope at the beginning. As the scope of the requirements grew over time with subsequent iterations of the framework, the need to "back-port" the requirements that arose and to plan for future extensions as well [23].

We first begin by iteratively synthesizing the guidelines for the functional and non-functional requirements and design for interactive computer graphics physical based simulation systems through a detailed case study of the original Softbody Simulation System [5, 14, 255, 23]. We then attempt to generalize our findings in order to compile a set of common requirement guidelines for the software architects, so that they already have a reference list of artifacts to refer to when specifying and designing similar new computer graphics systems [23].

We further review the Softbody Simulation System, the framework it was engineered into, and how such a framework evolved over time until the present day with a constant influx of new requirements that had to be back-ported into the system and the framework, sometimes requiring non-trivial redesign and re-implementation [23].

Furthermore, to validate the look and feel, physical properties and realism in physical-based elastic softbody simulation visualization requires a comprehensive interface to allow "tweaking" various simulation parameters at run-time while the simulation is running, instead of editing, re-compiling and re-starting the simulation program's source code every time a single parameter is changed. The typical values in our simulation that change are various forces applied to the body's particles, such

as four different types of spring forces with elasticity, damping, and gas pressure and even user-interaction with the object by dragging it in a direction with a mouse as well as collision response forces, spring stiffness, and so on. Since the simulation is real-time, another version of level-of-detail (LOD) [83, 75] adjustments includes the number of particles, springs, subdivisions, at the geometry level. At the simulation level the variations include the complexity and sensitivity of the physical-based simulation algorithms and the time step that they can bear: the finer the granularity of the algorithm, the more computation time is required, but higher accuracy of the simulation at the time step is achieved. The problem here is how to visually study these properties, validate them conveniently through either expert-mode or less-than-expert-mode user interface included with the simulation program, in real-time [255].

Thus, we additionally propose an iteration of the GLUI-based interface [218] to our real-time softbody simulation visualization in OpenGL [163, 211, 212] that allows "tweaking" of the simulation parameters. We introduce its visual design, as well as some details of the software design and the mapping between the GLUI components and the internal state of the simulation system we are working with. We propose the current interface in its current iteration of the design and improvement [255].

This work expands even further with the integration of the OpenGL Slides Framework (OGLSF [99], see Section 2.1.4.2) to make presentations with real-time animated graphics where each slide is a scene with tidgets–and physical based animation of elastic two-, three-layer softbody objects.

Finally, we forge the softbody objects into a jellyfish character, which is interactive and preserves all the properties of the softbody objects, but is augmented with breathing and self-swimming animation. It inherits all the advanced rendering and animation techniques, the UI developed for the Softbody Simulation System objects when it was redesigned, generalized, and modularized further as a library and an API in itself.

3.2 Organization

All the related CG animation and rendering background and related work have been described in Section 2.1. More specifically, in Section 2.1.4.2 we discuss the background and the related work and the properties of the OGLSF [99]. Additionally, we have reviewed some of the previous related work done in jellyfish modeling and animation earlier in Section 2.1.4.1.

We then describe a brief methodology and layout in Section 3.3. What follows further in this chapter, is the description of the requirements and design iterations of the system followed by the summary of derived requirements and guidelines [23], all the related design and implementation aspects of the Softbody Simulation System in Section 3.4. Even more specifically here at the case study at hand, we describe our own modeling approach as a transition from a static generated model to a dynamic physical based model of the jellyfish in 2D and 3D along with the modeling of the tentacles and the surrounding environment in Section 3.4.2.3. The animation and implementation aspects are discussed in Section 3.4.2.5.6 and Section 3.4.

Most the sections are illustrated with the actual resulting screenshots from the OpenGL softbody system presentation slides referenced where appropriate [99].

Then we conclude describing our achievement, the limitation of the approach [11], and highlight possible future directions in the overall summary in Section 6.4.

3.3 Methodology

This section discusses the methodology overview suitable for the exploration of the proposed system with the step-wise process conceptualized in a general manner on the focus to achieve individual sub-goals including the steps that have already been done with the overall goal in mind (see Section 3.3.1).

3.3.1 Goal

We describe the overall design goals and describe a subset of them realized in this work with the plans to have a complete installation exhibited in various galleries after the thesis completion and polishing the installation work.

First of all, in the general case the intention is to optimize the current framework of the Softbody Simulation System [5, 13] and make it more usable and adaptable to other human-computer interface libraries, bound to APIs for various devices. This goal is largely met as shown throughout Section 3.4.

Then we plan for the framework to be able to work with a joystick-like or a sensor glove-like haptic devices to interact and deform a softbody object in a 3D OpenGL environment that provides force feedback to the user restricting their movements based on the elasticity and shape of the modeled virtual objects acting as "inverse sensors". We succeed partially with this goal with the Novint Falcon haptic device.

The other goals are faster processing and response times of the simulation by applying common "speed-up techniques", especially to the subdivided 3D object with the higher LOD algorithms (e.g., RK4's finer computations are more CPU-intensive than Euler's). Thus, we in general need a framework and algorithm implementations for faster real-time rendering, intersection testing and collision detection, and ray tracing and global illumination (see Section 2.1.3). Another goal is to have more realism in the material properties, shadows, lighting effects, and so on. Thus, the list below is a short list of techniques that can help us with our goals. We did get to try and implement some of them to some degree or provide a ground work to enable them in the future. The starred entries (*) have been at least partially realized or streamlined the framework to allow for easier extensions:

1. Object representations (GPU programming)
2. (*) Vertex and fragment shading (GPU programming)
3. (*) LOD

4. (*) Collision detection

5. Game engine architecture

6. Shadows of the softbody objects

7. Real-time animation of softbody parts attached to an existing skeleton

3.3.2 Proof of Concept

The proof-of-concept covers many aspects present throughout the design and implementation process in Section 3.4 in a number of screenshots taken and referred to in the chapter. The current project on the softbody framework is to model a 3D character, such as the jellyfish [11, 256] (see Section 3.4.2.5). The audience is able to interact with the 3D creature in real-time. After an associated haptic or MoCap system calibration, audience could drive the jellyfish by moving their body and perform various actions. This application also could be used for digital theatre performance with dynamic (input-sensitive) sound and real-time simulated computer graphics, such as the little mermaid dances with various sea creatures. This would represent a more extensive version than the earlier collaboration prototype (see Section 6.4.3.8) [257, 256] and follows the conceptual design outlined in Figure 37.

3.4 Design and Implementation

As previously mentioned, the design- and implementation-related work centers around deformable softbody objects simulation via physical based methods [58] applied to a certain type of softbody objects using the corresponding simulation framework [14]. Its core is OpenGL [211, 212] along with the CUGL [258] library providing a set of convenient extensions to OpenGL [11]. Thus we detail the conceptual design (Section 3.4.1) and implementation of this work with the results (Section 3.4.2) throughout this section.

3.4.1 Conceptual Design

The conceptual design centers around the operation of the Softbody Simulation System. The two-layered elastic object consists of inner and outer elastic mass-spring surfaces and compressible internal pressure. The three-layered elastic object adds a center particle inside the inner layer. The density of the inner layer can be set differently from the density of the outer layer; the motion of the inner layer can be opposed to the motion of the outer layer. These special features, which cannot be achieved by a single layered object, result in improved imitation of a soft body, such as tissue's liquid non-uniform deformation. The inertial behavior of the elastic object is well illustrated in environments with gravity and collisions with walls, ceiling, and floor. The collision detection is defined by elastic collision penalty method and the motion of the object is guided by integrating Ordinary Differential Equations. Users can interact with the modeled objects, deform them, and observe the response to their action in real-time and we provide an extensible framework and its implementation for comparative studies of different physical-based modeling and integration algorithm implementations [13]. The artistic rendering of such objects is the specific application of jellyfish.

The conceptual component design of the system in Figure 36 is based on the derived software methodology we detail further, where re-usable components for the modeling, rendering, animation, and interaction are defined. The overall system pipeline follows the process outlined in the conceptual design in Figure 37, where the users (participants) take on the center stage to interact with the system providing some input, which then proceeds to the Softbody Simulation System to do the required input processing, simulation, and graphical processing prior to providing sensory feedback back to the users. Regardless the actual modeling, all components follow the same operational simulation sequence illustrated in Figure 38 [13].

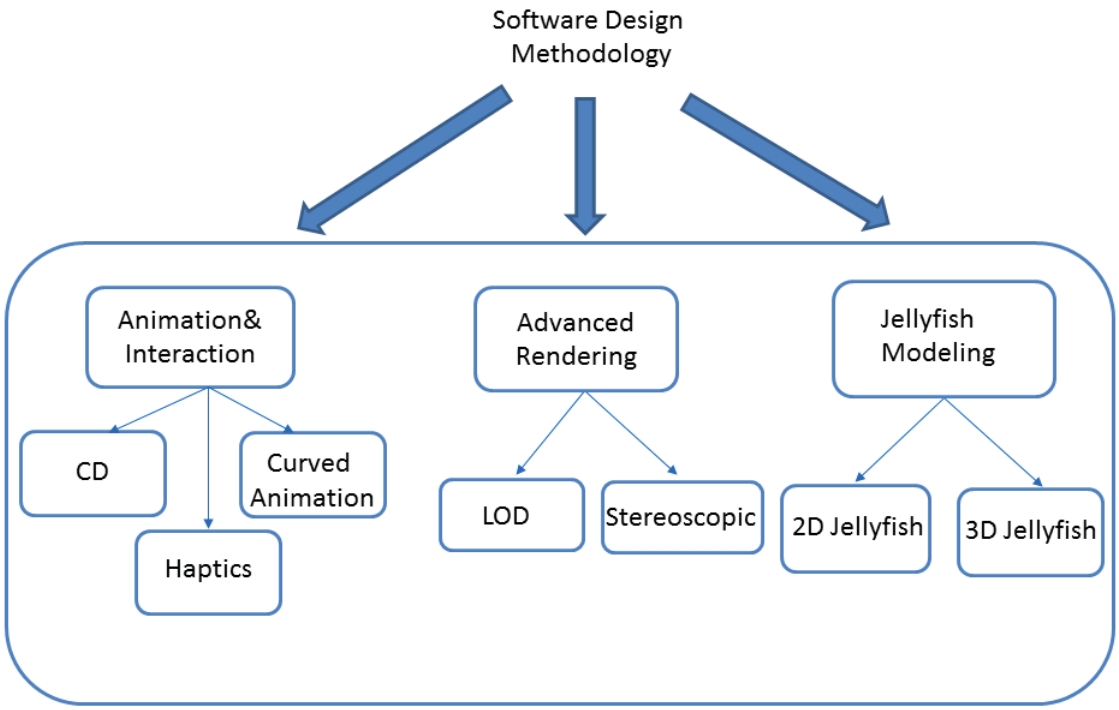


Figure 36: Conceptual Design of a Softbody Simulation Installation Component Breakdown

3.4.2 Implementation

The implementation covers the use of a few OpenGL, GLUT, GLUI, CUGL and GLSL API calls (Section 3.4.2.1) to support this installation and its interaction mechanism. The founding core of the jellyfish installation is the Softbody Simulation System and its underlying framework (Section 3.4.2.2). The framework and system underwent significant (re)design updates to make it more re-usable and extensible to support various applications, a case in point here being the jellyfish, as well as others, and the ways of interacting, shading, control, collision detection, etc., that made the implementation aspects more streamlined and simply easier from the time of its original realization [5]. Thus in this contribution we detail the process of such transformation of the system along with the generalization of its functional and non-functional requirements along the way in Section 3.4.2.2. We then detail the modeling details of various aspects in Section 3.4.2.3, such as the introduction of the center particle

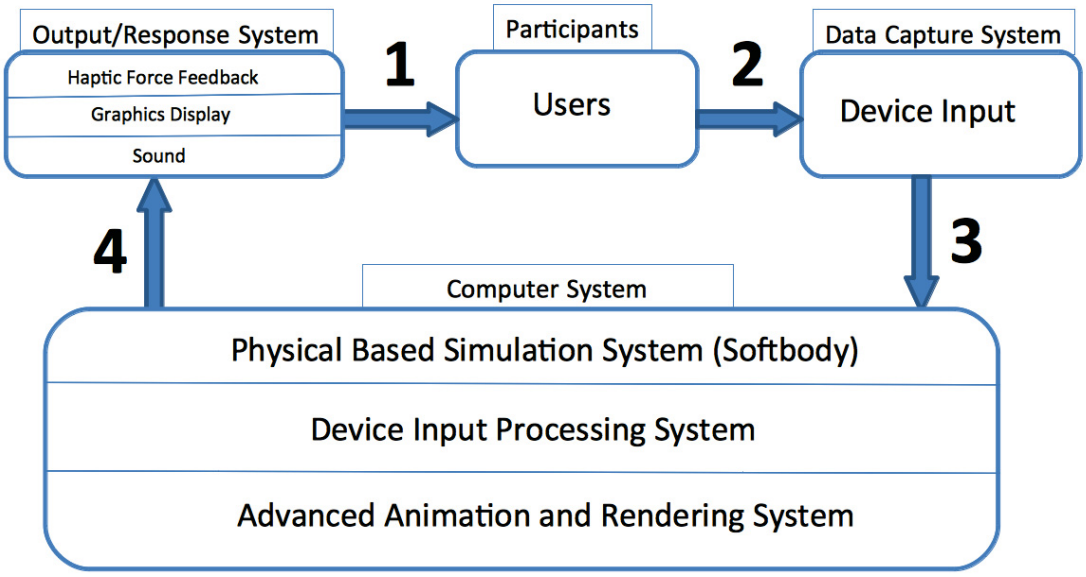


Figure 37: Conceptual Design of a Softbody Simulation Installation

"layer", Feynman algorithm, GLUI LOD controls, drag-based animation, curve-based animation, and finally the jellyfish itself constructed (Section 3.4.2.5) as a real-time softbody object and the screenshots of the PoC's runs in various rendering modes and effects. We then review the animation, interactions, lighting, and projection issues.

3.4.2.1 APIs

We briefly review the APIs used in this installation from OpenGL, GLUT, and GLUI in Section 3.4.2.1.1, GLSL in Section 3.4.2.1.2, CUGL in Section 3.4.2.1.3, and Falcon SDK API in Section 3.4.2.1.4.

3.4.2.1.1 OpenGL, GLUT, and GLUI. The detailed overview on OpenGL and related libraries is in Section 2.4.3.1. We use a number of GLUI-based functions [218] in our `InitializeGUI()` to connect GLUT callbacks (commonly present in many OpenGL applications, cf. Section 4.2.2.1.1) to the GLUI elements via

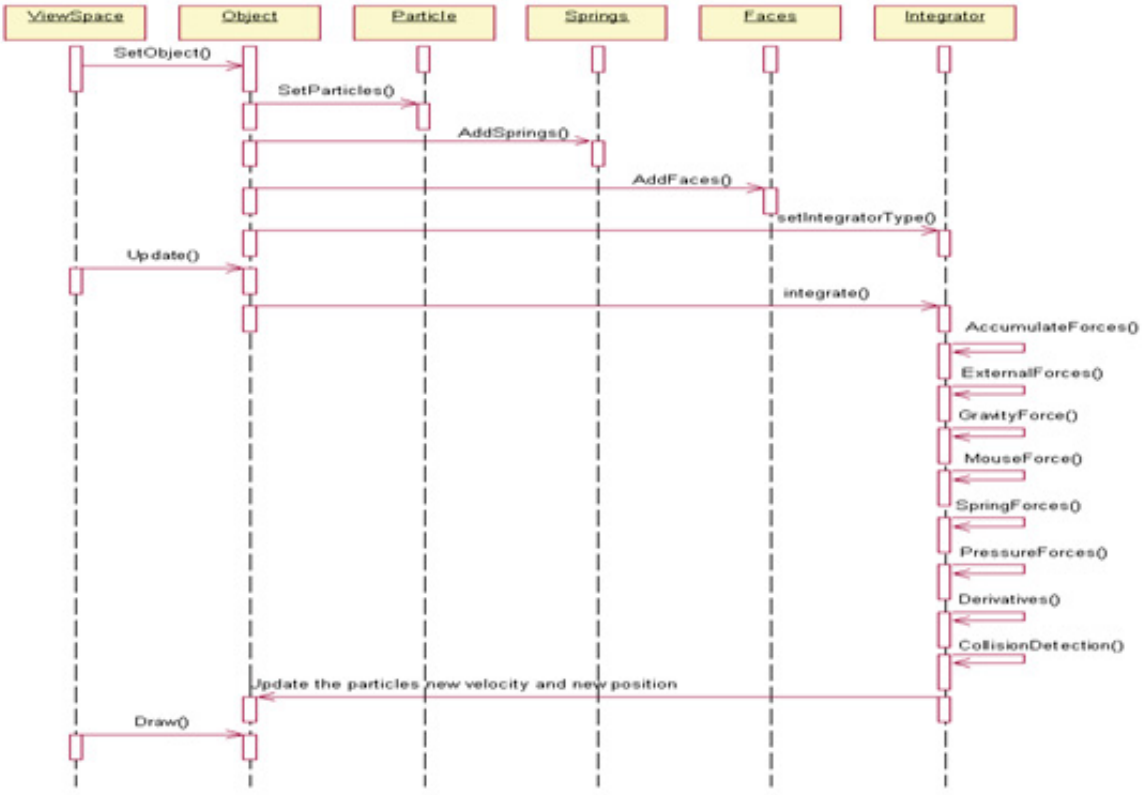


Figure 38: Softbody System Simulation Sequence Diagram

`GLUI_Master.set_glut*()`. We then compose the UI itself via rendition of various widgets via `GLUI_StaticText`, `GLUI_Separator`, `GLUI_Rollout`, `GLUI_Spinner`, `GLUI_Checkbox`, `GLUI_Panel`,`GLUI_RadioGroup`, `GLUI_Button`, and others.

3.4.2.1.2 Shaders and GLSL. For the detailed background overview please see Section 2.4.3.2. The GPU shaders framework implementation is described in Section 3.4.2.4.3.

The general principle is having to load and compile a source code shader file and dynamically link it from within OpenGL. Prior to that the graphics card is queried to see which extensions it supports via `glGetString(GL_EXTENSIONS)`. Then we look for the extensions called `GL_ARB_vertex_program` [88] and `GL_ARB_fragment_program` [89]. Once detected we load the shader code into the buffer, compile, and link it by registering functions to do so [36, 259, 260] to setup functions such as `glGenProgramsARB`, `glDeleteProgramsARB`, `glBindProgramARB`, `glProgramStringARB`, and

`glProgramEnvParameter4fARB`.

The below sequence of operations required in OpenGL 1.4 and 1.5 to install a shader for GLSL, for example, as shown in Algorithm 1.

```
1  Create an empty shader: glCreateShaderObjectARB();
2  if successful then
3  |  Provide source code: glShaderSourceARB();
4  |  if successful then
5  |  |  Compile: glCompileShaderARB();
6  |  |  if successful then
7  |  |  |  Create object code: glCreateProgramObjectARB();
8  |  |  |  if successful then
9  |  |  |  |  Attach the shader object to the program:
   |  |  |  |  glAttachObjectARB();
10 |  |  |  |  if successful then
11 |  |  |  |  |  Link all the shader objects into a program:
   |  |  |  |  |  glLinkProgramARB();
12 |  |  |  |  |  if successful then
13 |  |  |  |  |  |  Install the executable program as a part of current
   |  |  |  |  |  |  OpenGL's state: glUseProgramObjectARB();
14 |  |  |  |  |  end
15 |  |  |  |  end
16 |  |  |  end
17 |  |  end
18 |  end
19 end
```

Algorithm 1: Installation of a Shader ARB Extension

3.4.2.1.3 CUGL. The synthetic camera object in this work is implemented using the API of `cugl::Quaternion`s, which relies on the Concordia University Graphics Library (CUGL)'s API [258]. The (x, y, z) axes are drawn at the scene's origin using the API `cugl::axes()` call primarily for debugging purposes.

3.4.2.1.4 Novint Falcon. Please refer to the detailed overview Section 2.4.1.2. Here we detail some specific API details that are in use as inferred from the Novint manuals [207, 208].

We use the application's `HapticsClass` sampled from the Falcon SDK [208] to

contains all the synchronization and computation methods while deferring the actual softbody simulation to our Softbody Simulation System's objects by retaining a reference on the object in question.

The continuous Novint servo callback function from it called `ContactCB()` keeps polling (`HDL_SERVOOP_CONTINUE`) the Falcon for its buttons' status and position, and then calls our own computation method `softbodyContact()` that interacts with the softbody object "touch" and "response" processing and putting a mass pressure on to the device's handle perceived by the interacting user.

We also have an on-demand (`HDL_SERVOOP_EXIT`) synchronization callback function `GetStateCB()` that uses non-blocking data synchronization with Falcon instead of continuous tracking.

The data in both cases constitute handle button states, position (x, y, z) of the sensor's handle in its local coordinates, and feedback forces (f_x, f_y, f_z) transfer from and to the device. `vecMultMatrix()` helps us to translate between the local device and world coordinates. The corresponding Novint's SDK's calls `hdlToolPosition()`, `hdlToolButton()`, `hdlSetToolForce()` help us with that communication.

3.4.2.2 Softbody Simulation Software Framework

A two-layer physical based softbody deformation framework and the implementing system were designed and developed throughout a series of works [5, 14, 255] and simulating in using OpenGL [211] in real-time.

The original Softbody Simulation System evolved in a number of ways that were not originally specified or planned for in the requirements or were under-specified as "nice-to-have" high-level system descriptions. A lot of integration work was planned or ongoing with other frameworks and systems that also alter the requirements. In this section we list the evolution that the system has undergone or is currently undergoing through that ideally would need to be really anticipated from the start (stated as requirements) and designed for subsequently. Nonetheless, some earlier requirements and subsequent design decisions have helped to ease up certain tasks and refine the

process through a number of smaller iterations.

The system we are working with was first conceived and implemented in my thesis studies [5, 14], originally did not have much user-interactive interface to LOD and other details except the mouse drag and some function keys. For every parameter change, the simulation had to be edited and recompiled. The simulation program and its framework is written in C++ for 1D-, 2D-, and 3D- real-time physically-based two-layer elastic softbody simulation system. Originally, there was no provision to alter the simulation state at run-time or startup via configuration parameters or GUI control, so we made a contribution in improving the framework by adding the notion of state, LOD hierarchy, and its link to the GLUI [255].

As our experiments continued we made the use of the softbody objects in other projects to further validate the softbody's framework usability on the design and the source code level to apply the studied techniques in the softbody itself and then to use that in another project.

The projects wishing to use the softbody objects have to be able to set the include directory to where the softbody header files are. The minimum required to `include` is `ObjectXD` class where X is 1, 2, 3, or 2-in-3. Then the project has to make an instance of the desired dimensionality classes and insert the corresponding drawing and frame step update handling into the appropriate callbacks of OpenGL, such as `Display()` and `Idle()`.

Such projects have two choices—either add the code of the Softbody Simulation System to their directory structure and build system or use it as a library to link against. Currently, we provide only a static library compilation `libsoftbody.lib` under Windows 7, using the Visual Studio 2010 platform and project files. In the same solution set we use the said `libsoftbody.lib` in our Bezier-curve-based animation where a softbody object is dragged along the curve [22].

3.4.2.2.1 Early Design Decisions. Some early design decisions, in the reverse order, have put some requirements in place that allow the system to sustain the

change and extension to some degree. As shown in the previous section the model was broken down into objects of various dimensionality and that can have some code and functionality reused through inheritance. The same applies to the hierarchy of integration algorithms and the underlying data structures [5, 14, 13]. GUI at the time has been planned for the prototype system, but it was not specified of how it will look like, which widgets it would have and where it is to be placed within the simulation window, and their respective functionality [23].

3.4.2.2.2 Evolution. Further evolution of Softbody Simulation System is outlined below in a set of milestones. The evolution began from its first version presented in [5].

- The emphasized aspects of the level-of-detail (LOD) "knobs-and-switches" have been realized and more formally addressed [23]. The hierarchy of the LOD has also been specified, from primitive geometry to the higher-level algorithms [255] (see Section 3.4.2.4.1).

- Then, the initial GUI based on GLUI [218] has been built to allow the run-time management of the complexity and number of the LOD parameters [255], and this is where the first actual GUI design has first appeared and materialized [23] (see Section 3.4.2.4.1).

- The architecture produced allowed dynamic integration algorithm selection at run-time, e.g., Euler vs. midpoint vs. Runge-Kutta 4 (RK4) vs. Feynman. It has necessitated a plug-in like architecture to allow any number of such algorithms to be available for the real-time simulation and rendering as a part of the physical based integration [23].

  In particular, the Feynman integrator has been developed (see Section 3.4.2.3.3) significantly later in the time frame from the other three. The architecture required only minor adjustments to accommodate the new integrator, thus validating the approach [23].

- The next set of requirements for the redesign and additional design came from the need to integrate the vertex and fragment shader processing within the system—written in either shader assembly or GLSL languages, so the framework of the system had to be extended to allow parametrized generic loading, compilation, linking, and enabling/disabling of the shaders at run-time [23] (see Section 3.4.2.4.3).

- One of the items of the work involving the Softbody Simulation System is enhancing its interactivity with reactive and responsive controls that include haptic devices to provide force feedback. Applications based this can be used beyond gaming, e.g., for specialist training (medical), or even interactive cinema [19, 100, 23]. (See Section 3.4.2.1.3 and Section 3.4.2.5.6).

- The original Softbody Simulation System has implemented only a single algorithm [5] for collision detection (CD), the penalty method; however, now there is a requirement to be able to use other collision detection algorithms, so the implementation and design were updated to reflect this new requirement to allow a collection of CD algorithms, just like the collection of integration algorithms there is, with a proper API, abstraction, and so on [23].

- Some library requirements were not explicitly specified either because they were assumed "obvious" but their lack results the requirement specification being incomplete when we widen the scope of applicability of the system. The libraries include OpenGL [211] along with the drivers for GLSL support, GLUT, GLUI [218] or even less obvious CUGL [258]. This include the HDL library for Novint haptic devices, such as Falcon [208]. Other plans include the SDL [261], Direct X and HD support, which were not planned for. For the latter we can build up on the experience from OGRE3D [254] that has solved that issue [23].

- The system compiles and runs on the Microsoft Windows 7 platform, but with growth it is natural to accommodate other platforms, such as Mac OS X, Linux

and mobile such as iOS and Android. The Softbody Simulation System is written in C++, and the requirement is to be source-code portable. It is not a very major effort to make the system portable as OpenGL itself is portable, but there are build system efforts and code changes to make it work reliably [23]. Some of the core softbody applications, including jellyfish have already been ported to Mac OS X.

- Other related projects include the integration of stereoscopic rendering of the softbody objects as well as teaching some computer graphics and simulation aspects and techniques and integration into the corresponding OpenGL presentation slides framework [23].

- Subsequent requirements cover interoperability with different game and rendering engines and systems (potentially distributed), extended simulation systems, and the use of the system as a library [23].

All these past, new, or recent requirements put constraints on and necessitate making re-design decisions of the system and its components since they were not adequately planned for in the first place. (Not to mention the testing aspect and quality assurance, which we leave for another endeavor project [23].)

3.4.2.2.3 Requirements Synthesis Methodology. From the discussions in the previous sections and the reverse engineering of the design and implementation to some extent we synthesize the previously unstated (or sometimes rather implicitly stated) requirements, both functional and non-functional. This list is rather complementary and is not replacing existing typical software engineering requirements for information systems [262, 263]. The list also contains sometimes a checklist of guidelines rather than just the requirements [23].

1 `ViewSpace` initializes the virtual world and provides the user an interactive environment in the form of an interface to allow the user to drag the objects, or choose the parameters. For example, the user can choose the object type, one-dimensional, two-dimensional, or three-dimensional. The user can choose the integrator type, Euler, Midpoint, Feynman, or Runge Kutta 4. The user can set up the springs' stiffness, damping variable, and the pressure;
2 `SetObject()` creates an elastic object based on the interface variable set from Step 1;
3 `SetParticles()` sets up the particles' position and their other initial properties, such as mass and velocity;
4 `AddSprings()` connects particles with springs according to their index;
`// This step is omitted if the object is one-dimensional`
5 `AddFaces()` connects the springs with faces based on proper index;
6 `SetIntegratorType()` tells the Controller, which integrator users select through the interface;
7 `Update()` updates the integrator's time step;
8 **begin**
9 `Integrate()` contains two major functions, `AccumulateForces()` and `Derivatives()`. It is based on all the object geometric information modeled and all the forces information accumulated, to integrate over the time step to get new object position and orientation;
10 `AccumulateForces()` sums up the forces accumulated on each particle;
11 **begin**
12 `GravityForce()` accumulates gravity force based on the particles' masses;
13 `MouseForce()` is the external force from the interface when user interacts with the object. It will be added or subtracted from the particles depending on the force's direction;
14 `SpringForce()` accumulates internal forces of the particles connected by springs;
`// For one-dimensional object, this state is omitted`
15 `PressureForce()` accumulates the internal pressure acted on the particles;
16 `CollisionForce()` checks if the object is out of boundaries after the integration state. If the new position is outside of the boundary, then it will be corrected and reset on the edge of the boundary. Moreover, the new collision force will be added to the object;
17 **end**
18 `Derivatives()` does the real derivative computation of acceleration and velocity in order to get new velocity and position of elastic objects based on the integrator type defined by users;
19 **end**
20 `Draw()` displays the object with new position, velocity, and deformed shape;

Algorithm 2: General Softbody Simulation Algorithm

Functional Requirements.

- Most graphics systems adopt a version of the MVC architecture, so we need to plan for user I/O, multiple displays, and the data structure modeling.
- User I/O GUI and other input devices should plan for an LOD GUI as well as potentially multiple algorithms of handling it. This is also prevalent among typical PC games through their game settings allowing them to run tuned on a wide range of hardware.
- Implement statistics gathering for various real-time performance metrics not only for simulation, but also for rendering.
- Plan to allow for multiple algorithms selection at run-time for comparative studies on any aspect of visual realism to run-time performance and memory usage.
- Plan for support and the corresponding GUI for the vertex and fragment shaders written in either cross-vendor GPU assembly, GLSL, and HLSL.
- Plan for lighting and texture-mapping techniques. This is tricky in particular with some softbody objects, so the data structures should be planned ahead of time to support normals, and so on.
- Provide ability for softbody-like objects being able to be "attached" to "hard-body" (rigid body) objects, e.g. to simulate muscles on a skeleton and provide points of attachment of one to the other.
- Allow for modeling and alteration of the Archimedean-based graphs and different types of them than just an octahedron [264] and just the number of iterations as a run-time LOD parameter.
- Interactivity through haptic devices [197] with the softbody feedback (e.g., for surgeon training or interactive cinema [19]). Additional interactivity through sensor devices like Kinect.

- Allow the state dump and reload functionality in order to display each particle and spring state (all the force contributions, velocity, and the position) at any given point in time in a text or XML file for further import into a relational database or an Excel spreadsheet for plotting and number analysis, perhaps by external tools. Reloading would enable to reproduce a simulation from some point in time, a kind of a replay, if some interesting properties or problems are found. This is useful for debugging as well.

- Plan for multiple rendering backends, potentially distributed or multithreaded such as OGRE3D [254], URay [81], and others.

- Allow for stereoscopic effects.

Non-Functional Requirements.

- *Usability* may be of importance to researchers to test different physical based phenomena at real-time [23].

- *Portability* of the source code (at minimum) and plan for deployment and building under different platforms and build systems, e.g., Linux with Makefiles [265] and autoconf, or Mac OS X with XCode or also Makefiles, and mobile device in order to cover larger user and researcher bases [23].

- *API hooks* should be always provided for plug-in architectures for all algorithms where there could be more than one instance, such as integration, subdivision, collision detection. This will simplify the integration effort with other projects and other programming languages [23].

- Libraries, frameworks, APIs, such as OpenGL, GLSL, GLUT, GLUI, CUGL, Direct X [23].

- Export and generation of the system as a library itself, or a collection of libraries and APIs for use in other applications [23].

- Constrain the exported API and globals in the system's own namespace to avoid clashes with external applications during linking [23].

- Plan for the academic value of teaching and learning computer graphics [266] and physical based simulations, by structuring the code, comments, and documentation per consistent naming and coding conventions and the API. This is especially valuable for open-source and academic projects [23].

- Allow for extension of the main algorithm, e.g., of Algorithm 2, by subclassed applications so it is less rigid [23].

3.4.2.3 Augmented Softbody Modeling

3.4.2.3.1 Center Particle – the Third Layer. The core of the model design in the softbody system is that it has two interconnected layers of tissue using the spring-mass system and is tested against various integration algorithms (e.g., Euler, midpoint, and RK4) as well as a hierarchy of various level-of-detail (LOD) parameters as summarized earlier. The simulation is built upon one to three softbody objects of 1D, 2D, or 3D which are simulated and interacted within a limited "boxed" environment. To make the softbody simulation as a tissue for example on an animated game character's skeleton or any other type of animation, the softbody needs to be "attached" somehow to the rigid body that it is moving along with. The softbody attachment aspect in the softbody simulation framework was missing, thus we contribute a first version of attachment of softbody objects to hard surfaces for "ride-along" animation to achieve higher realism.

We present an enhanced real-time two-layer softbody shape simulation system implemented using C++ and OpenGL with a third, single center point layer for attachment of the softbody objects onto hard surfaces and drag to simulate e.g. soft skin tissues. We present the user interaction and the corresponding GLUI interface to the softbody objects for comparative purposes. We also add the Feynman algorithm implementation to complement the original earlier framework for comparative

studies with classical Euler, mid-point, and RK4 algorithms. Alongside our softbody attachment contribution we add and discuss the Feynman algorithm implementation and its properties as compared to the existing implementations of Euler, midpoint, and RK4. We also compare the generation of the modeling the 3D objects spheres by subdivision or sphere mapping when using the attachment points and the Feynman algorithm. We attempt to make the simple extension that does not violate framework constraints as a proof-of-concept and animate it along a Bezier curve.

3.4.2.3.2 Curve-based Trajectory. Moving a softbody along a Bezier curve is the first simplest example of advanced animation using the softbody object outside its own simulation application. We were able to successfully link the softbody features and use them through their API exported as header files of the available softbody objects of various dimensions [22].

The animation is real-time and while the softbodies are being dragged along the curve, they perform their designated default physical based dynamic deformation. We simulate the mouse drag along the Bezier curve by the attachment center point and the `Drag` object. The controls are detailed in Appendix A.1. These key handling is implemented in the callback function `KeyboardKey()` and `FunctionKey()` in the `controllerCallbacks.cpp` file.

The system also has mouse-based and GLUI [218] controls. The user can adjust the curve by adding more control points. The control points can be created either through the "Add New Point" button in the GUI from a list of preset points or by middle button (or left and right together of only two buttons) mouse click within the rendering view (see `Mouse()` in `controllerCallbacks.cpp`). The control points are properly connected by a dashed line in the order they are referenced. The implementation of it is located in the function `void GUI::guiContorollerCallback()` in the `gui.cpp` and `globals.cpp` files. An example screenshot is in Figure 39.

In this image, this simulation displays the cubic Bezier curves for the control points. Four control points are required to specify the curve. The implementation is

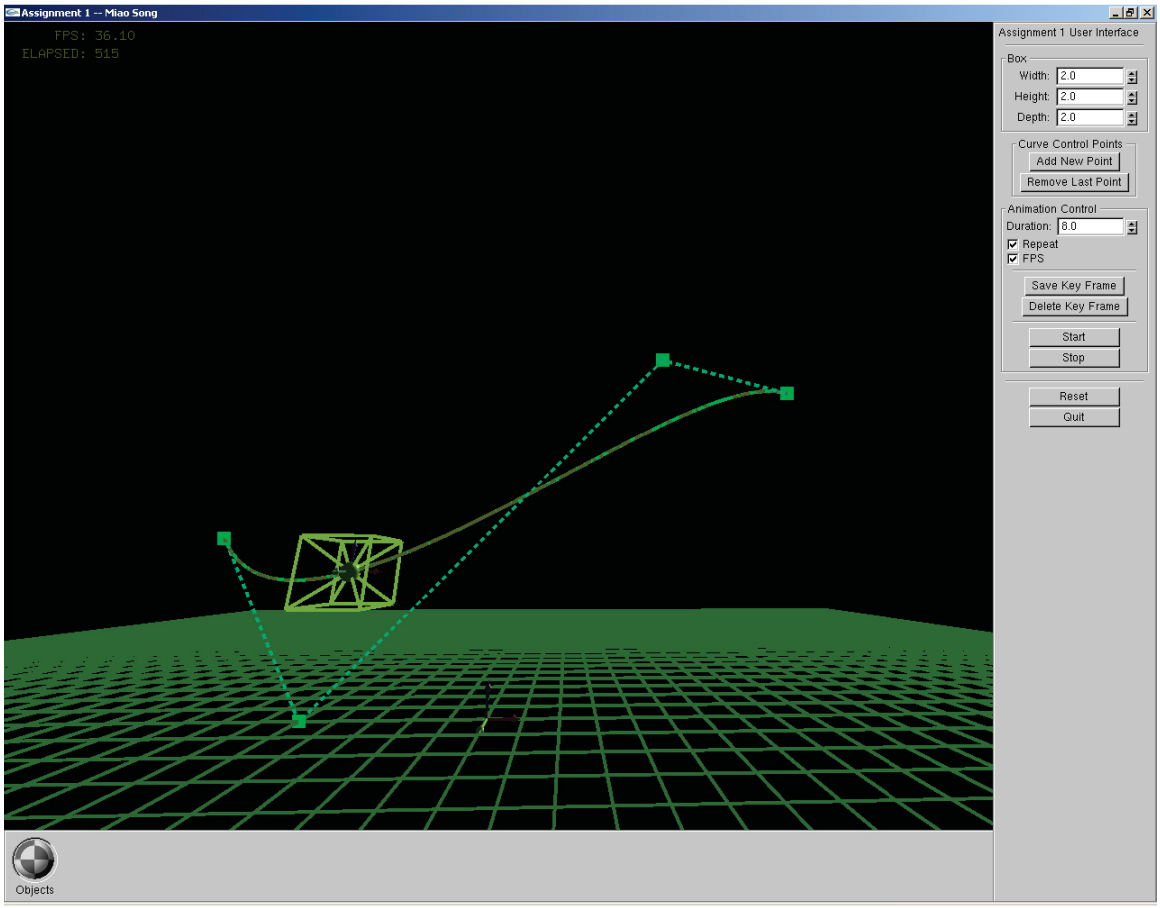


Figure 39: General Bezier-Curve Based Animation Screenshot

in `Drawcurve()` function in the `drawTools.cpp` file. The Bezier curve (or each curve segment) is only drawn when there are enough (4) points to have it done.

The user is able to specify the time the motion should take from start to finish. There is a GLUI control for that. At the implementation level, the time is not in true seconds, but allows to speed up or slow down the motion. It also depends on the frame rate of a particular environment, OS, hardware, etc. The implementation is in `void GUI::guiContorollerCallback()` function in the `gui.cpp` and `globals.cpp` files.

Another GUI rotation control at the bottom-left of the GLUT window allows the user to input rotations into the box object by manipulating an arcball control. The control displays as a checkerboard-textured sphere, which the user manipulates

directly. The rotation control can be constrained to horizontal-only movement by holding the CTRL key, or to vertical rotation by holding the ALT key. Rotation controls can optionally keep spinning once the user releases the mouse button. The local coordinate axes have been drawn with CUGL's API. The implementation is in `void GUI::guiContorollerCallback()` function in the `gui.cpp`.

The user also has a control of the camera that can look around the animated scene using quaternions and the camera as well as axes from the Concordia University Graphics Library (CUGL) [258].

The softbody's center point is following the curve in the animation while the the physical based properties still in effect in the area of the softbody. The end goal of this is to have the body follow the curve motion and react accordingly. When different stiffness softbody objects placed on the skeleton or otherwise character, the idea as they move with the character attached by the attachment point, the animation takes places. This can simulate the soft body parts of a running or walking human as well as when humans breathe. The curve based animation is a testbed for a periodical walk to see how the body behaves. We have not implemented yet or attached any actual softbody to any skeleton or character yet so this is deferred to the future work.

3.4.2.3.3 Feynman Algorithm. We implement for comparative purposes the Feynman algorithm within a C++-based framework for the softbody simulation. To facilitate the comparison, we design the timing measurements to be on the same hardware against that of Euler integrator in the softbody framework by varying different algorithm parameters. Due to a relatively large number of such variations we implemented a GLUI-based user-interface to allow for much more finer control over the simulation process at real-time, which was lacking completely in the previous versions of the framework. The Feynman algorithm [267] is still quite efficient comparatively to Euler yet more accurate due to the step and a half stepping. Our framework does not yet implement comprehensive statistic measurements so we cannot provide actual numerical results, just visual observations. We show our current results based on the

enhanced framework.

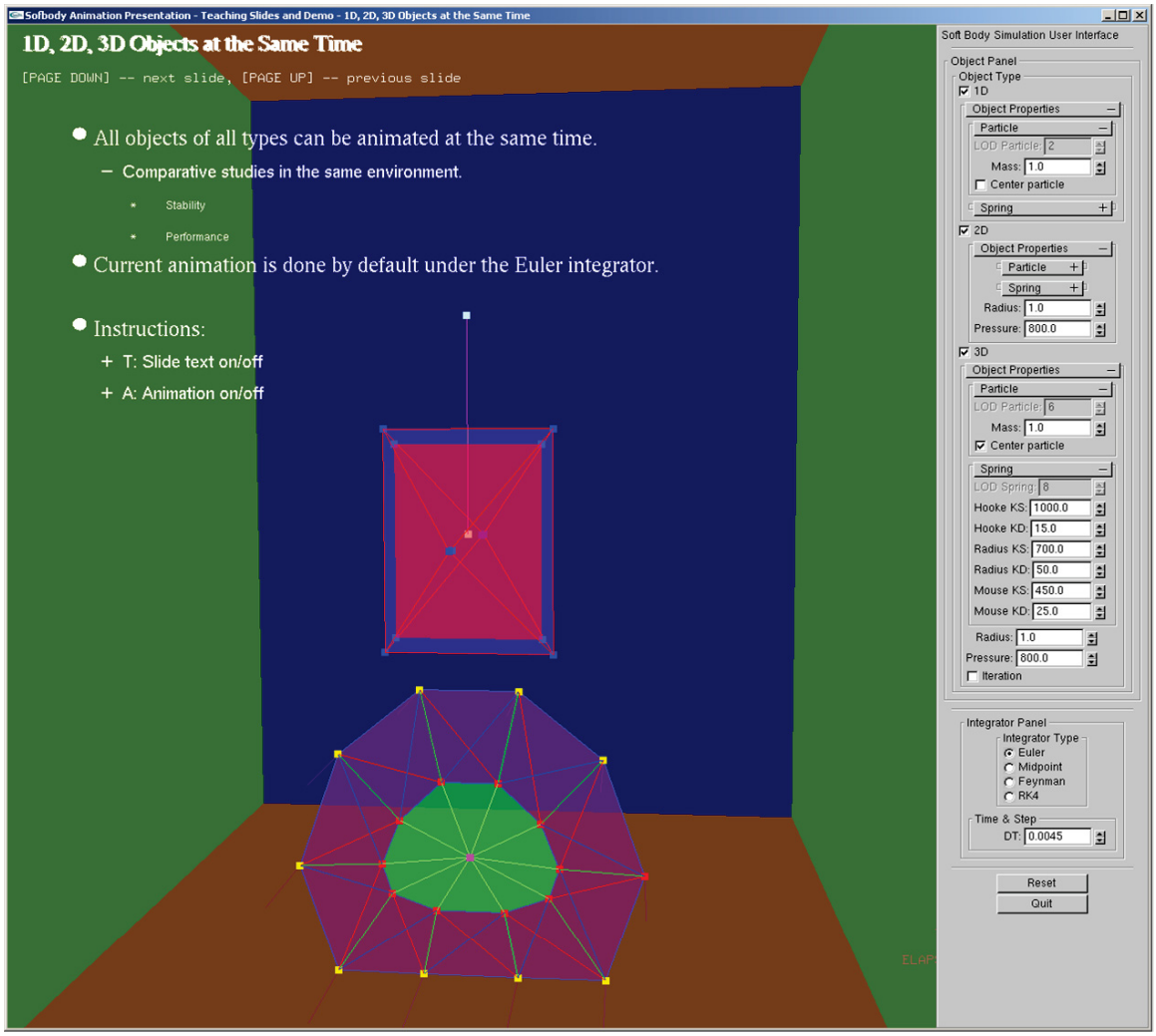
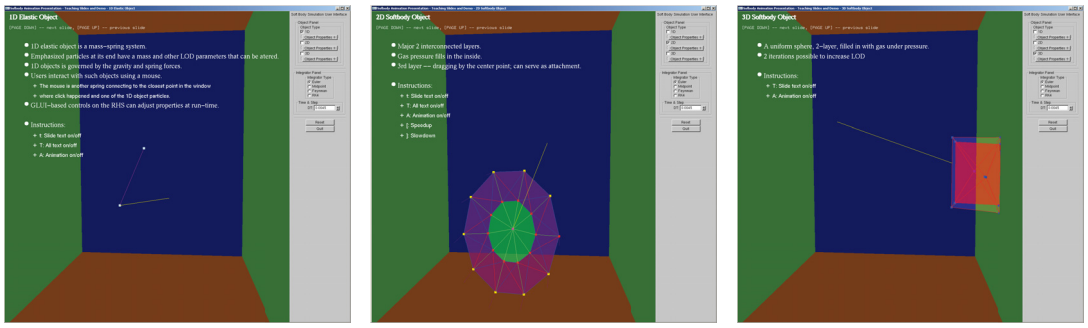


Figure 40: Three Types of Softbody Objects Simulated Together on a Single Slide Scene

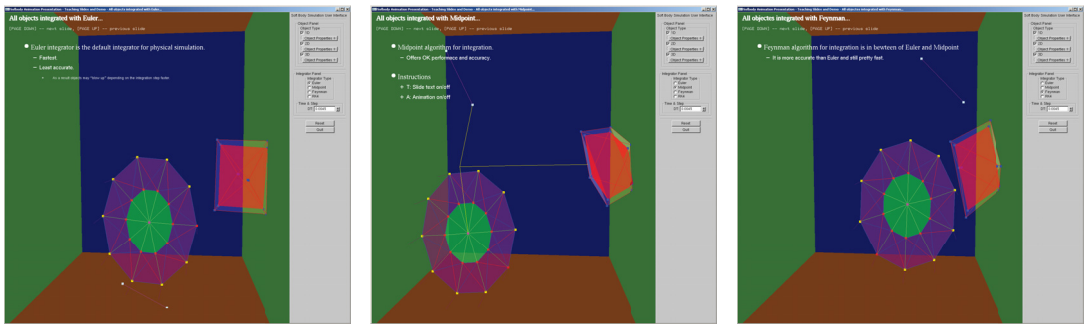
3.4.2.4 Advanced Softbody Rendering

Following the description in Section 2.1.3, we apply some of the techniques in here. Specifically, and LOD-related processing and its user interface in Section 3.4.2.4.1, GPU shading in Section 3.4.2.4.3.



(a) 1D Elastic Object Simulation Slide (b) 2D Elastic Object Simulation Slide (c) 3D Elastic Object Simulation Slide

Figure 41: Simulation of Single 1D, 2D, and 3D Softbody Elastic Objects Slides



(a) Simulation with Euler Integrator Slide (b) Simulation with Midpoint Integrator Slide (c) Simulation with Feynman Integrator Slide

Figure 42: Simulation of all Softbody Object Types with Various Integrators Slides

3.4.2.4.1 LOD Interaction Interface. The most important contribution of this work, aside from greatly improving the usability of the simulation system by scientists, is capture and comprehension of a hierarchy of the LOD parameters, traditional and non-traditional. For example, allowing arbitrary number of integration algorithms constitute a run-time LOD sequence, which would not normally be considered as LOD parameters, so we introduce the higher-level LOD components, such as algorithms, in addition to the more intuitive LOD parameters like the number of geometric primitives there are on the scene and so on [255].

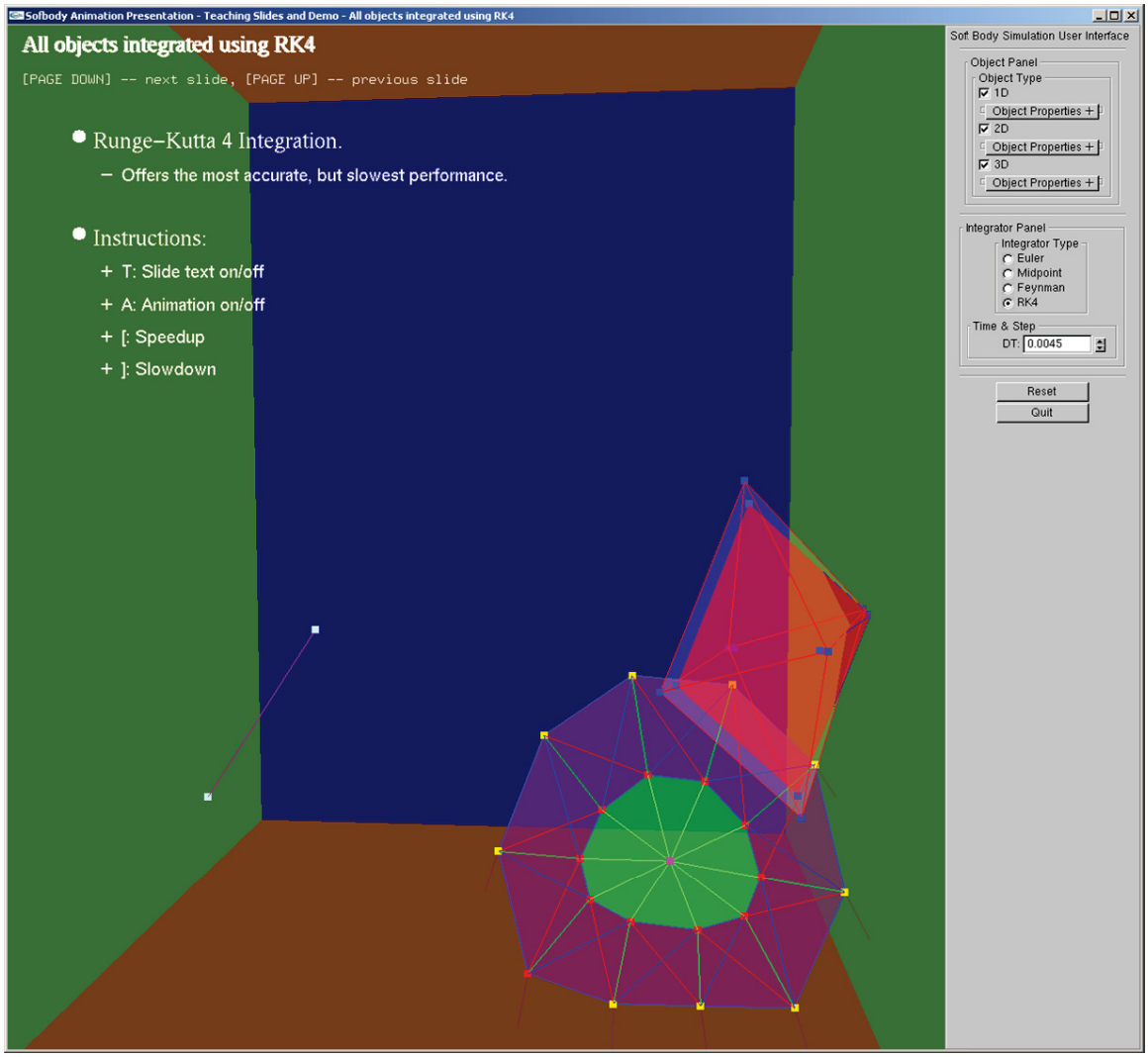


Figure 43: Simulation with Runge-Kutta 4 Integrator Slide

We summarize an interactive GLUI-based interface to the real-time softbody simulation using OpenGL. This interactivity focuses not only the user being able to drag 1D-, 2D-, and 3D-deformable elastic objects selectively or all at once, but also being able to change at run-time various "knobs" and "switches" of the LOD of the objects on the scene as well as their physically-based modeling parameters (see, e.g., Figure 41, Figure 42, and Figure 43). We discuss the properties of such an interface in its current iteration, advantages and disadvantages, and the main contribution of this work [255].

In Section 2.1.3.3.3 we mentioned a number of LOD possibilities. We implement

some of them to a degree of the LOD levels in our softbody simulation system. Most of these details are summarized in the corresponding publication [255] where we outline the LOD hierarchy and its relationship to the corresponding GLUI elements. The hierarchy of the LOD parameters varies from the traditional geometry of the softbody objects, to the physical simulation parameters, and all the way to the integration algorithms used, where we consider the algorithm complexity and accuracy to be an LOD aspect along the traditional notions for the level of detail. As a part of the future work we plan to extend and deepen further the important notion of LOD according to the surveyed techniques for the research and optimization purposes for larger scale simulations [22].

The software design is centered around the notion of state. The state is captured by a collection of variables of different data types to hold the values of the simulation parameters. The whole architecture follows the Model-View-Controller (MVC) [268] design pattern, where the Model is the state of the geometry and LOD parameters, View is the simulation window and the GUI into the model details, and the controller that translates users actions onto the GUI into the model changes, particularly the LOD state [255].

The system state has to encode a variety of parameters mentioned earlier that are exposed to the interactive user at run-time. Our initial iteration of the visual design of the LOD interactivity interface is summarized in Figure 40. The LOD components are on the right-hand-side (in their initial state, not expanded). And the main simulation window is on the left (interactivity with that window constitutes for now just the mouse drag and functional keys). Following the top-down approach we bring in more details of configurable simulation parameters [255].

3.4.2.4.2 Adjustable Parameters. The adjustable LOD parameters can be categorized as dimensionality, geometry, integration algorithms, force coefficients, and particle mass [255].

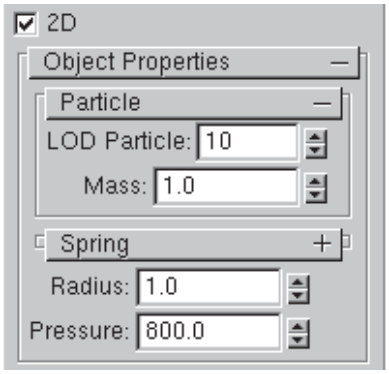
Dimensionality. Switching the 1-, 2-, and 3-dimensional objects present in the simulation is simply done as checkboxes as shown in Figure 3.46(a). The "Object Properties" buttons expand the higher-level representation into the finer details of the objects of each dimension type [255].

The LOD on dimensionality is 1D, 2D, and 3D. All three types of objects can be rendered on the same scene at the same time. This is encoded by the instances of state variables `object1D` of type `Object1D`, `object2D` of type `Object2D`, and `object3D` of type `Object3D` as well as their corresponding Boolean flags reflecting the status of whether top render them or not. This is done to illustrate the object behavior under the same simulation environment and how the objects of different dimensionality respond to the same simulation settings [255].

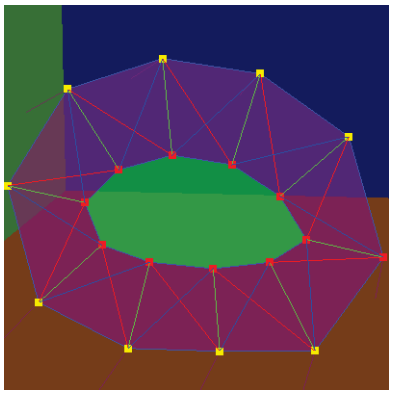
Geometry. In 1D there are really no geometry changes. In 2D, the geometry means number of springs comprising the inner and outer layers of a 2D softbody. Increasing the number of springs automatically increases the number of the particles at the ends of those springs, and brings a better stability to the object, but naturally degrades in real-time performance. In Figure 3.44(a), Figure 3.44(b), Figure 3.44(c), and Figure 3.44(d) is an example of the 2D object being increased from 10 to 18 springs, from the UI and rendering of the object perspectives [255].

The geometry of increasing or decreasing of the number of springs in 2D, as in Figure 3.44(a), Figure 3.44(b), Figure 3.44(c), and Figure 3.44(d) is not the same approach used for the 3D objects in the original framework. The 3D object is constructed from an octahedron by an iteration of a subdivision procedure, as selectively shown in Figure 3.45(a) Figure 3.45(b), Figure 3.45(c), and Figure 3.45(d). The iteration increases dramatically the number of geometrical primitives and their interconnectivity and the corresponding data structures, so the efficiency of the real-time simulation degrades exponentially with an iteration as well as the detailed time step, which can make the users wait on slower hardware. Using the LOD GLUI components we designed, a researcher can fine-tune the optimal simulation parameters for

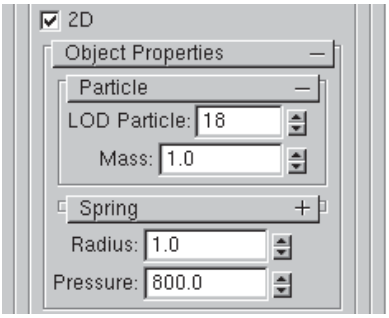
the system where the simulation is running. Since it is a combination of multiple parameters, the complexity between editing and seeing the result is nearly immediate compared to the recompilation effort required earlier [255].



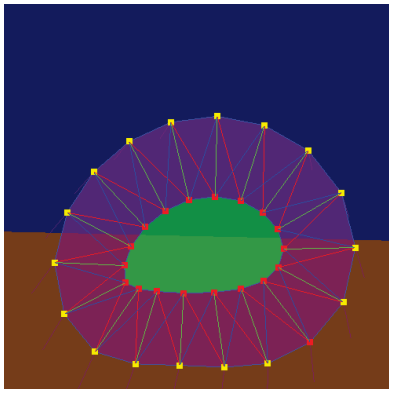
(a) LOD 2D UI with 10 Springs



(b) LOD 2D Object with 10 Springs



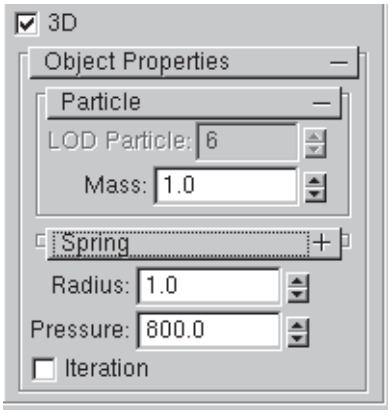
(c) LOD 2D UI with 18 Springs



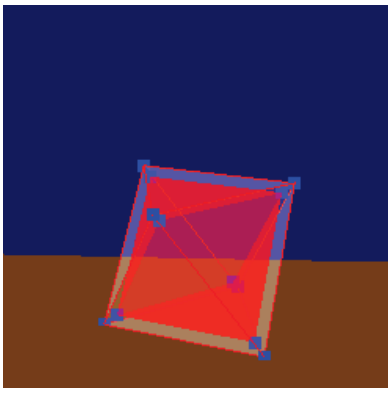
(d) LOD 2D Object with 18 Springs

Figure 44: Some Examples of Various 2D LOD Settings

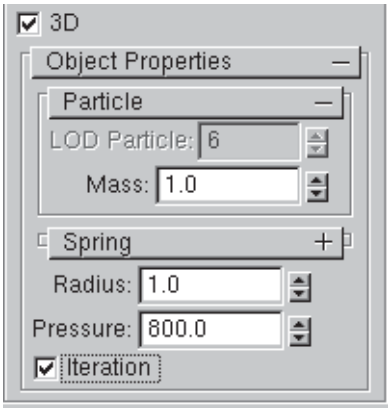
The 2D and 3D objects have an additional LOD parameter related to their geometry complexity. In 2D the geometry LOD specifies the number of particles the inner and outer layers have and by extension the number of structural, radial, and shear springs they are connected by. The 3D geometry LOD is based on the way the sphere is built through a number of iterations and the default set of points upon which the triangular subdivision is performed on the initial octahedron. The LOD here is encoded by the `iterations` state variable [255].



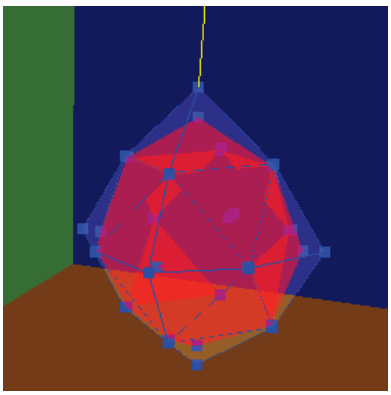
(a) LOD 3D UI without Iteration



(b) LOD 3D Octahedron



(c) LOD 3D UI with Iteration Once

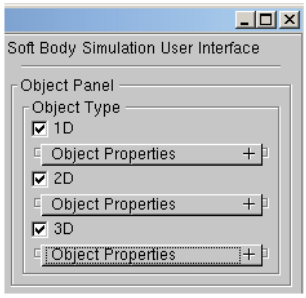


(d) LOD 3D Object with Iteration Once

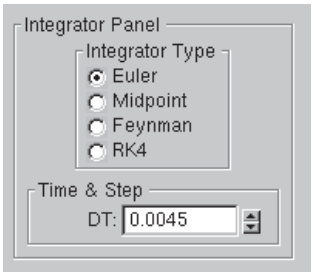
Figure 45: Some Examples of Various 3D LOD Settings

Integration Algorithm. Currently available integrator types are in Figure 3.46(b), from the fastest, but least accurate and stable (Euler) to the most accurate and stable (RK4) [5]. The Softbody Simulation Framework allows addition of an arbitrary number of implementations of various integrators that can be compared in real-time with each others by switching from one to another while the simulation is running [255].

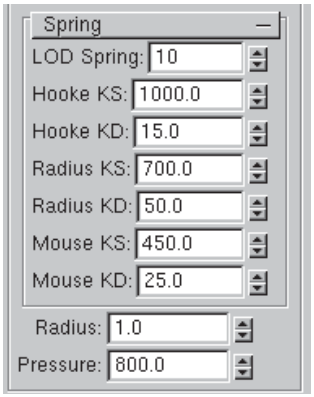
The algorithmic LOD includes the selection of a physical interpolation and approximation algorithms for the run-time integration of the velocities and acceleration exhibited by the particles based on physical laws, such as Newton's, and the different types of integrators: Euler, mid-point, Feynman, and RK4. These are the



(a) Object Type Dimensionality



(b) Integrator Type LOD UI



(c) Force Coefficients LOD UI

Figure 46: Some Examples of Various LOD Parameters

four presently implemented integrators in the Integrators Framework of the system, so the corresponding state variable is `integratorType` that allows selecting any of the available integrators and witness the effects of such selection at run-time. The LOD aspect here goes about the simplicity and run-time performance of an algorithm implementation versus accuracy of approximation of the physical motion, with the least accurate (but fastest) being the Euler's integrator, and the most accurate (but slowest) being the RK4 [255].

Force Coefficients. In Figure 3.46(c), we provide in the current form a way to adjust the coefficients used in force accumulation calculations as applied to each particle. KS and KD represent elasticity and damping factors of different types of springs. The default Hooke parameter refers to the structural springs comprising the perimeter layers of the object, and then the radius springs, as well as the spring created by a mouse drag of a nearest point on the object and the mouse pointer [255].

There are a number of forces that are taken into account in the simulation process. Each of them corresponds to at least one state variable supporting the simulation and the interface. The coefficients correspond to Hooke's law forces on springs, gravity force, drag force, gas pressure (for 2D and 3D enclosed objects), and the collision

response force. The spring forces (structural on each layer, radial, shear, and mouse drag) typically have the elasticity spring coefficient, KS, and the damping force coefficient, KD. The gravity force by default is $m \cdot g$, where both the mass m of a particle and g can be varied as LOD parameters [255].

Particle Mass. Each object's properties have sub-properties (e.g., velocity, position, etc.) of comprising it particles, including mass, as exemplified in Figure 3.44(a), Figure 3.44(c), Figure 3.45(a), and Figure 3.45(c). We argue mass is an LOD parameter as each layer's particles of the softbody object can get different masses (but uniform across the layer), or even different masses for individual particles. The latter case however does not scale well in the present UI design, and has to be realized differently [255]. More specifically, particle mass is another LOD parameter that affects the simulation and can be tweaked through the interface. Internally, the simulation allows every particle to have its own mass (or at least two different layers in the softbody can have two distinct particle weights), but it is impractical to provide a GUI to set up each particle with such a weight, however, in most simulations we are dealing with the particles of uniform mass, so we can allow resetting the mass of all particles with a single knob. Thus, the default particle mass LOD parameter is mapped to the `mass` state variable [255].

3.4.2.4.3 GPU Shaders. We created a small subframework for GPU shaders to allow to work with the softbody simulation system. The shaders themselves, the shader vertex and fragment source text files come from various open source vendors from their examples and tutorials [269, 216, 270, 271, 90, 272]. We implemented an abstract ability to load and enable the shading of softbody objects from either assembly shaders or GLSL shaders. Our abstraction layer makes the softbody system dependent only our own API, and we provide the adapters to the code written by the other open source developers. The class declarations are found in the `include/GPU` directory and called `Shader` and `ShaderLoader` and their implementation is in `GPU`. There are their concrete implementation classes such as `RGSMShaderAdapter` and

`CWCShaderAdapter` to map our functions to the loading and initialization functions of the concrete shader loaders for the assembly (former) and GLSL (latter) respectively. We tested some example shader programs for their interesting material effects on the softbody objects and the scene that worked. Thus we succeeded to run the softbody system either with the GLSL shaders to assembly vertex and fragment shaders, but could not fully experiment with this feature. However, it opened the doors for further experiments and research and writing softbody-specific shaders in the future work, including shader-based texture mapping, stereoscopy, normal computation, and others to speed things up. The screenshots in Figure 47 and Figure 48 are simple shading examples applied to softbody system as a proof-of-concept.

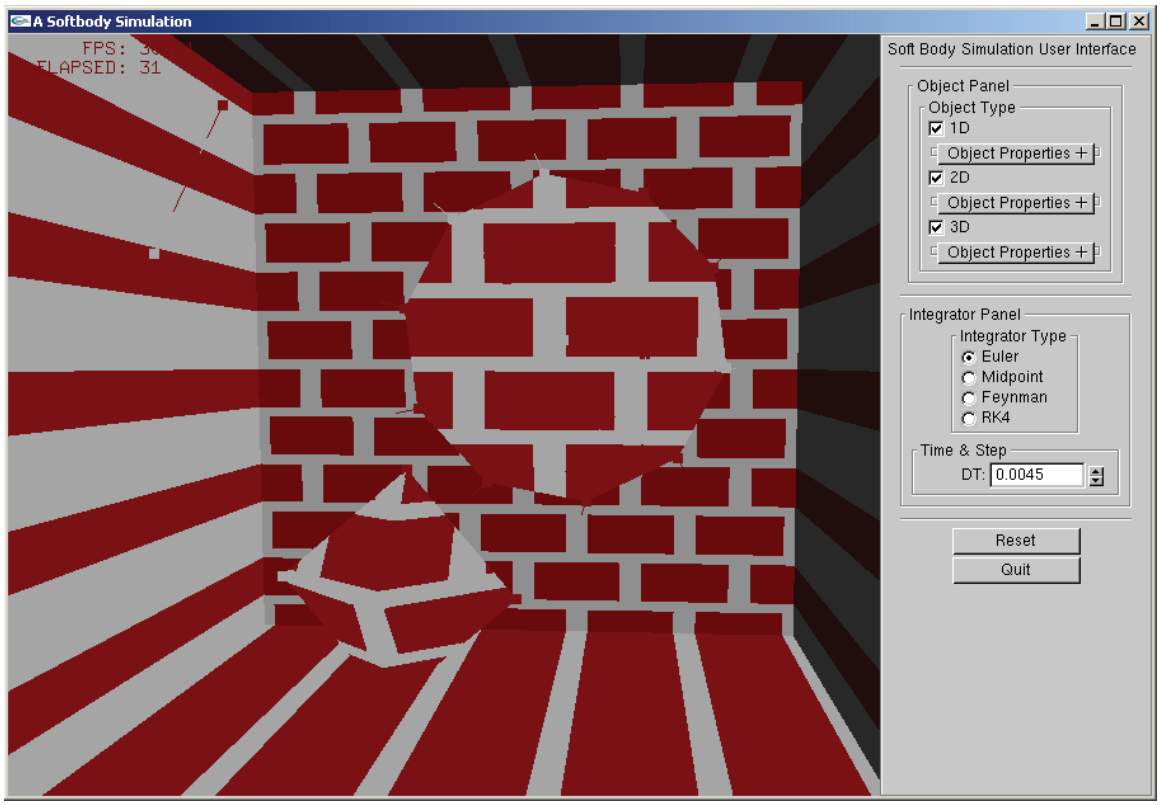


Figure 47: Simple Brick Assembly Shader Applied to the Softbody Simulation System

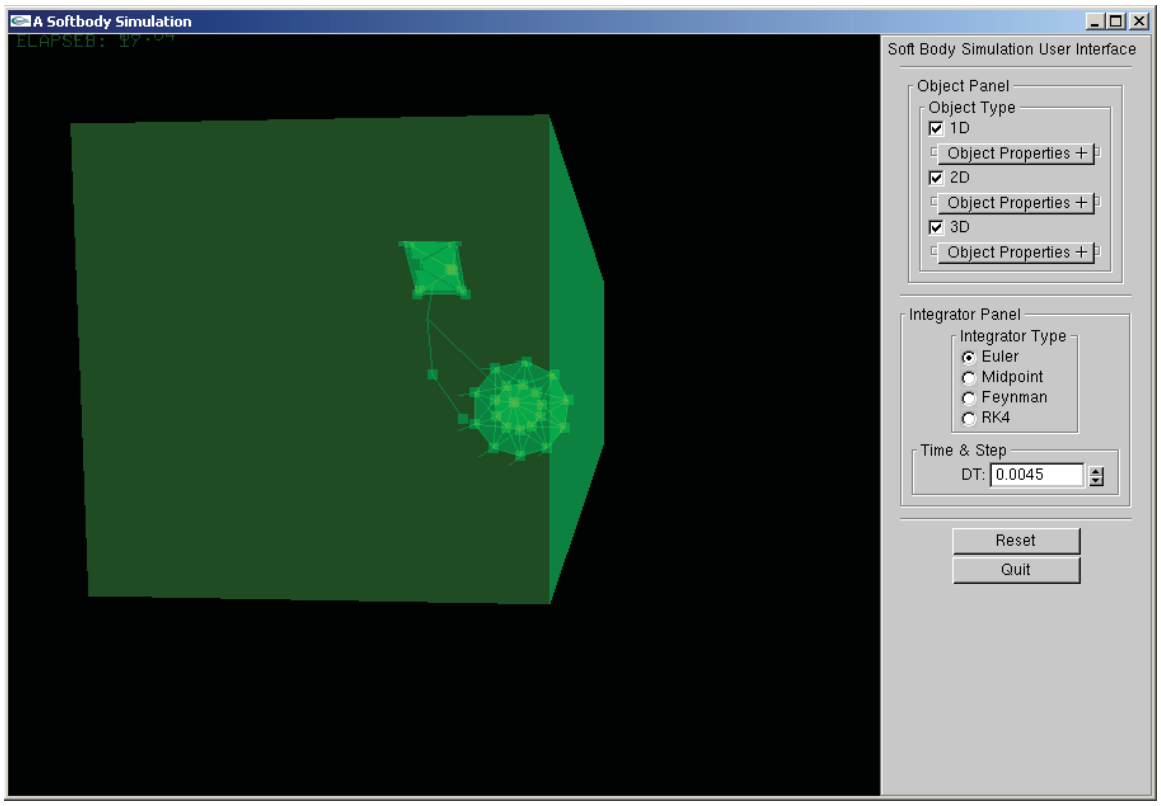


Figure 48: Simple GLSL Shader Applied to the Softbody Simulation System

3.4.2.5 Jellyfish Application of the Softbody Simulation System

We overview our approach to implement a realtime simulation of a jellyfish using our physical based Softbody Simulation System founded in laws of physics and spring-mass systems, visualized in OpenGL [11]. We outline our progress so far and the difficulties encountered and ways we solve some and propose to solve some other difficulties in this work.

We describe details of modeling and implementation of an interactive work of the initial prototype of jellyfish using OpenGL and the Softbody Simulation System in Section 3.4.2.5.1. We review the animation and interaction aspects in Section 3.4.2.5.6 and lighting in Section 3.4.2.5.7.

3.4.2.5.1 Jellyfish Softbody Modeling. The modeling for the jellyfish soft-body simulation includes environment (sea world), 2D jellyfish (see Figure 49 and

Figure 50), and 3D jellyfish. There is no 1D jellyfish.

The jellyfish objects inherit from the Softbody Simulation System `Object2D` and `Object3D` classes, whose geometry was altered fit into the approximation of the jellyfish shape in terms of particles and the corresponding connecting springs.

The tentacles framework is designed to provide 2D and 3D tentacles of various configurations, animation, and LOD (Section 3.4.2.5.3).

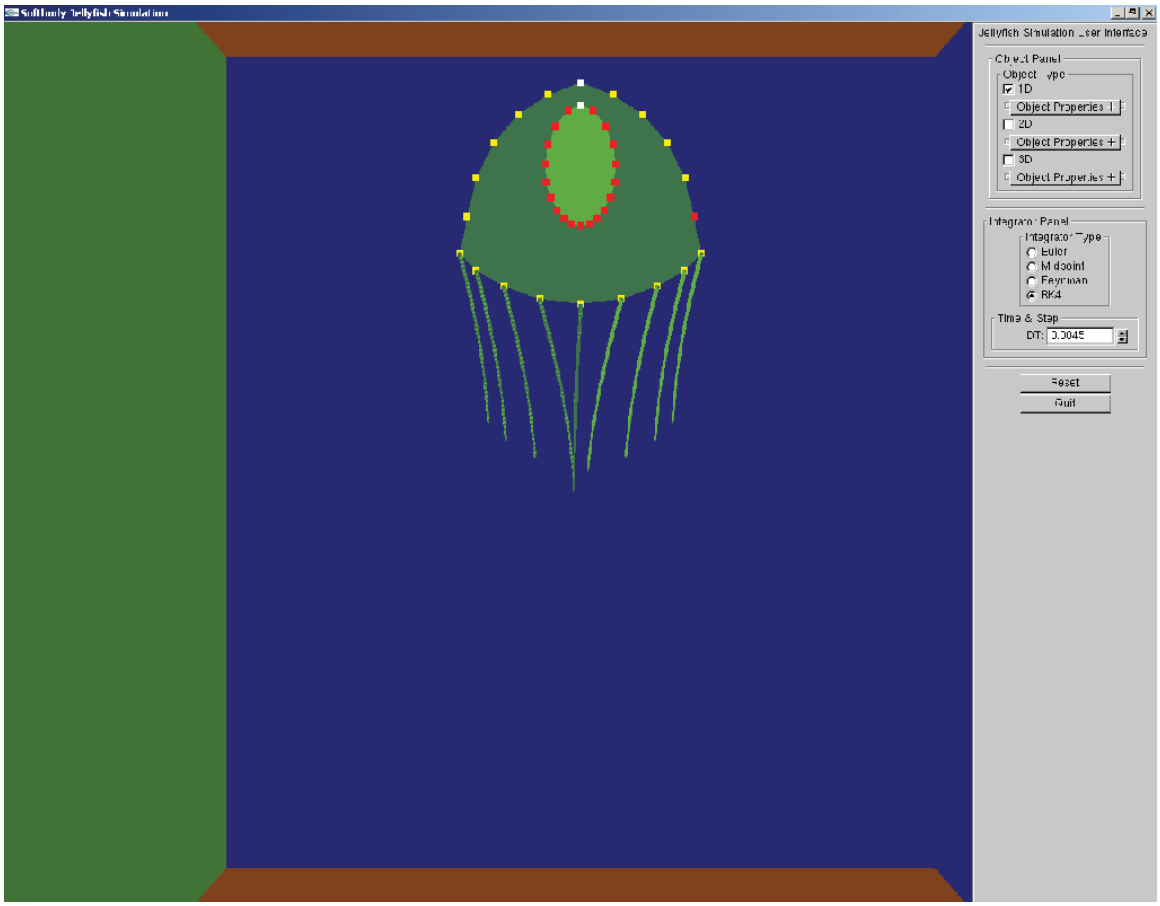
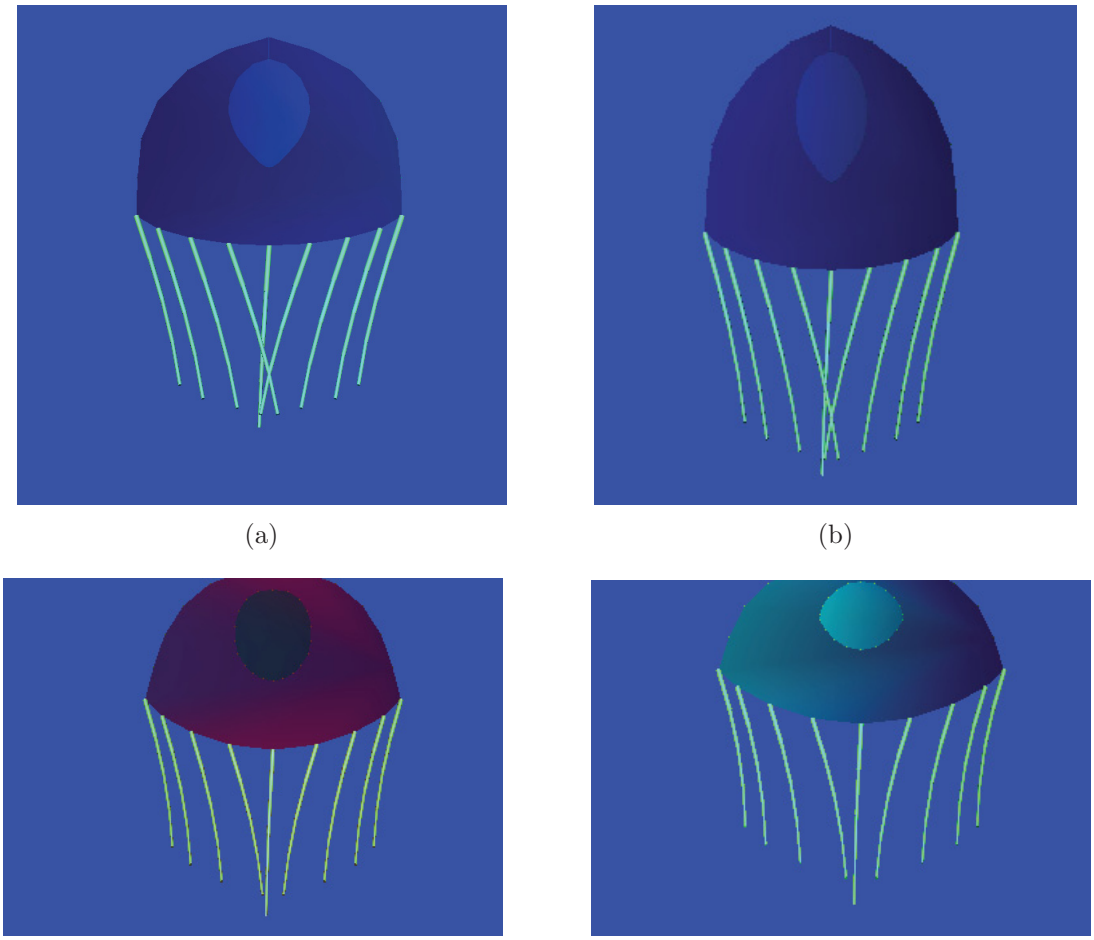


Figure 49: 2D jellyfish

3.4.2.5.2 2D. The 2D jellyfish (Figure 49) is a structural modification of a two-layered 2D softbody object with its spring-mass system for its bell. The user retains the capability to "drag" it using the mouse, or it "drags itself" with the frontal spring to simulate the real-time swimming realistically and let the physical laws in softbody do the rest. The tentacles (Section 3.4.2.5.3) are cylindrical joints individually animated kinematically while attached to the bottom particles of the bell [11].



(a) (b)

(c) (d)

Figure 50: 2D Jellyfish Lit Up with Lights

The dragging is done via the newly designed `Drag` object as a part of the redesign and refactoring. In the nutshell, it simulates the user pulling onto one particular of the particles of the softbody object periodically and that causes the whole system to animate.

The pressure value on the enclosed jellyfish is also set so that it helps the animation of movement and "breathing" combined with the drag.

3.4.2.5.3 Tentacles. True tentacles are rather complex to model and render in real-time for the animation purposes as they would rely on complex modeling and

animation inspired from the biomechanical modeling and description of jellyfish locomotion [97, 98] and fluid dynamics. Instead we build upon simplified models that are attached to and depended on the jellyfish bell softbody design or have an independent own proper motion. We also define `tentacles` as a mini-framework, and provide incrementally different implementations of it, from simpler and less accurate, but fastest, to more realistic and detailed, suitable for PoC demonstrations and adaptive LOD depending on the hardware available where the program is run [11].

We produced an implementation based on inverse kinematics (done in 2D at the present) with the tentacles as cylindrical joints individually animated kinematically as `Linkage` [273, 274, 258] while attached to the bottom particles of the bell [11]. The other implementations in the work involve the use of the softbody elastic springs, or the latter two mixed to some degree with the hair/thread simulation from another project [11].

3.4.2.5.4 Environment. An undersea project [275] would serve as an ultimate environment for this interactive simulation; which is currently abstracted by a partially translucent "aquarium box". Presently, this box is waterless environment with gravity [11]. The undersea scene will include some seaweed, fish, dolphin, and other sea creatures, as, e.g., modeled in *Spectacle* [276].

For a blackbox installation, the back wall or all walls, ceiling and floor in the box will have an AVI movie played back the author shot at a Vancouver zoo of the real jellyfish in the background. The worlds framework design is detailed further in the next section.

3.4.2.5.5 The Worlds Framework. The system has been extended to allow switching between different scenery worlds in the simulation relatively easier (before it was nearly impossible). The default `ViewSpace`, which was an open box became a framework of worlds. It's default implementation got split out into the `worlds::Box` while defining the initial API that could be used to implement other worlds. The second world being added to the system using this new API is the previously mentioned

underwater sea world conceived by Song *et al.* from an earlier project [275], which is a more natural aesthetic environment for a jellyfish than a testing box [11].

At the moment, switching between the worlds is done in the main application code by assigning one view space instance or another via 'W'. A GLUI option for this is planned in one of the several revisions [11].

3.4.2.5.6 Animation and Interaction. The animation in this project [11] is primarily physical based and is principally done through the Softbody Simulation System and its framework detailed earlier through the chapter and the previous work [14]. The jellyfish is composed of the softbody bell and tentacles with the bell having several attachment particles arranged in a circular fashion. The user "drags" the jellyfish by those points. If no interaction is taking place; and self-dragging swimming is not strong enough, the jellyfish "falls gently down" on the floor and gets "squashed" there by "bouncing off" two or three times (since there is no physical water environment present yet making the jellyfish being effectively let go and drop on the floor) just as a plain softbody object would. As before, the interaction UI on the right hand side allows selection of different integration algorithms, time step and level of detail to alter the animation in real-time to select most optimal parameters for the simulation. Haptic and motion-capture based interaction are partially implemented as well [11].

Falcon Haptics Softbody Interface. Here we detail our realization-specific information about the interaction aspects using a Novint Falcon haptic device. For the background information on Falcon haptics see Section 2.4.1.2 and Section 3.4.2.1.4.

The link between the `HapticsClass` [208] and the Softbody Simulation System is via `Object2D`. `Object2DJellyfish` and `Object3DJellyfish` are various degree descendants of `Object2D` and all can be passed on via `HapticsClass::init()` at the initialization time.

The subsequent interaction links the button on Falcon to the `Drag` spring, same as the mouse, and the virtual haptic "cursor", visualized as a sphere in the `BoxWorld`. When the cursor and the jellyfish collide (sphere-softbody-particle intersection test)

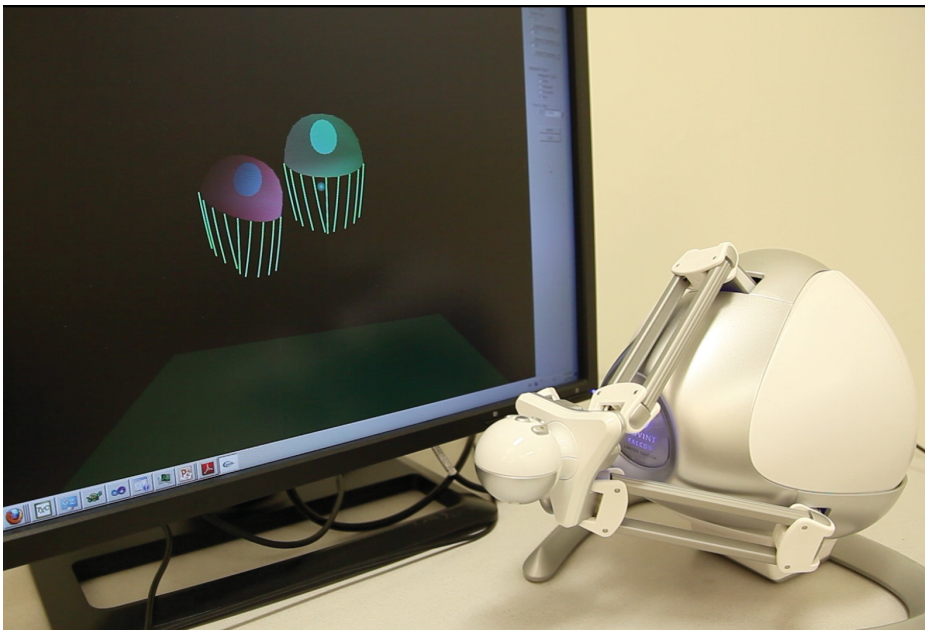


Figure 51: Jellyfish Connected with Haptics Device

the spring-mass forces trigger the servo motors within the device to partially block it and create a drag force pulling or pushing onto the jellyfish where the resistance can be felt. In other way, the servo cursor moves along with the jellyfish and when it's at the bottom, the interfacing user can feel the mass of it. In Figure 51 is the illustration of two jellyfish, one of which is interfaced with Falcon by mapping the particle forces to the servo motors and observing the synchronized motion effect mapped exaggerated "breathing" felt through the haptics.

3.4.2.5.7 Lighting. Currently, three moving spotlights light up the scene as if it's undersea highlighting portions of the projections on the blackbox walls simulating sun blinking and glints.

Chapter 4

Tangible Memories in Interactive Documentary

Linear documentary story telling has its own limitations and every documented story becomes an event of the static past after being first released. What makes an interactive installation documentary piece different from a traditional documentary is the aspect of interactivity itself and the audience being able to compose their preferred pathways and storylines selectively, bidirectionally, with the ability to go into details of interesting points to them unlike a linear passive progression. In the advanced way, the interaction can be also immersive and bidirectional with haptics and virtual reality techniques.

Why is it important to introduce interactive media technology to traditional documentary film making? The research could be a long journey as witnessed by some of the items in Section 2.2, but we take a small step at a time.

> *"Mostly, the memories are like little pictures in my head, like float around in the bubbles... So, when I want to see them, I search all those bubbles... Then I go into one, I can remember them..."*

That is how memories are for the little girl in the award-winning short documentary film *I Still Remember*.

We would like to extend this idea to a real touchable interactive installation. The prototype is augments the short film with interaction and projection in the form of the *memory bubbles*.

In this chapter we further review the methodology used in Section 4.1, the artistic and software design and implementation in Section 4.2, and finally the concluding summary in Section 6.5.

4.1 Methodology

In general, it is difficult to make a traditional documentary film interactive. How the computer graphics techniques could possibly be applied to and combined with cinema and interactive media to me is also a very exciting topic [19, 100]. I explore this topic in detail further on by stating general goal, conceptual design, and my proof-of-concept realization of it.

4.1.1 Goal

First of all, the film elements in interactive documentary should be controlled by real-time conditions as well as algorithmically controlled behaviors. Then, enhanced with haptic devices and stereoscopic effects, the feeling of the interaction will consume the audience to be an integral part of the storyline in the movie, which is a lot much more than a game. And the perceptual experiences can be different each time the movie is played. Moreover, I wish I could cross over another barrier in traditional documentary film: that the cinema screen is flat, a piece of cloth. The documentary film could have depth, which would be consciously or unconsciously associated with audience behaviors.

4.1.2 Proof of Concept

In the interactive documentary installation, the *Tangible Memories*, I first made the *I Still Remember* documentary's [15] floating memory bubbles interactive with the

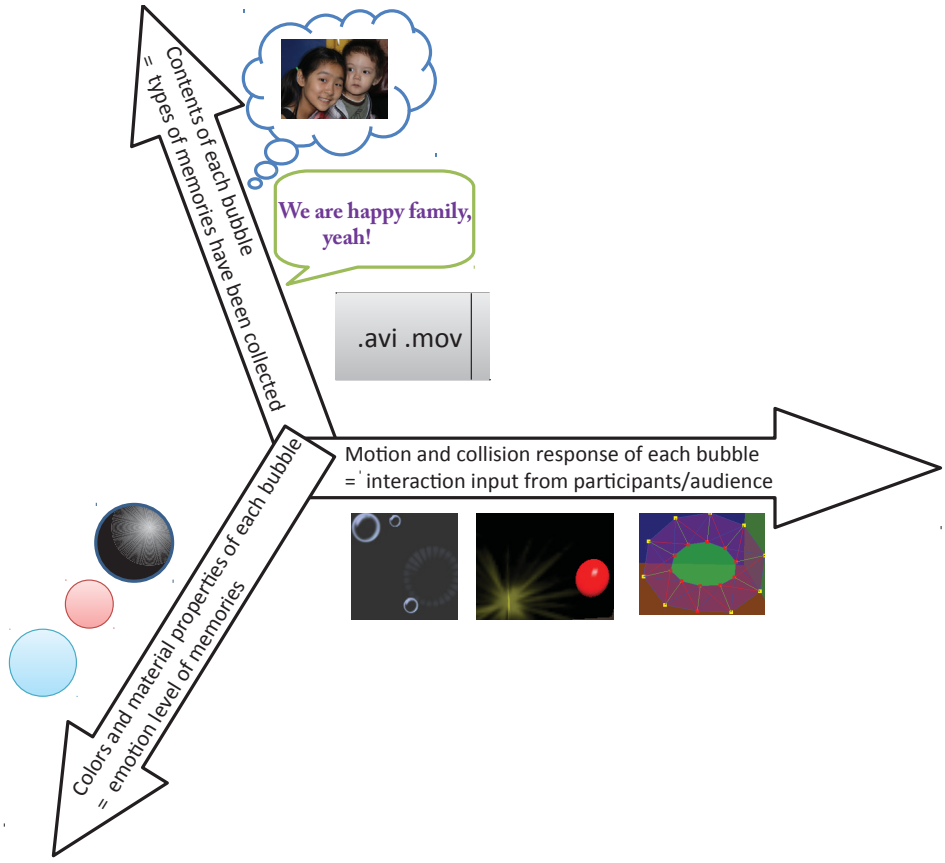


Figure 52: Overall Interactive Documentary Process

audience's participation using a near day-to-day OpenGL.

By porting some of the of the *Tangible Memories* system with later enhancements using an inexpensive MoCap system, such as Kinect (with XNA), we could achieve the real interaction for documentary perception experience.

The virtual memory bubbles are projected in a physical environment, such as walls and ceilings with multi-projectors. Audience will not sit and passively watch the documentary film, instead, they need to use their body movement or voice to advance the dynamic scenarios of the story or stories.

The resulting "artistic high-level algorithm"'s pseudo-description, with the process and basic design, are conceptually illustrated in Figure 52, Figure 53, and in Algorithm 3 respectively.

There are two runnable proof-of-concept programs we have developed in OpenGL

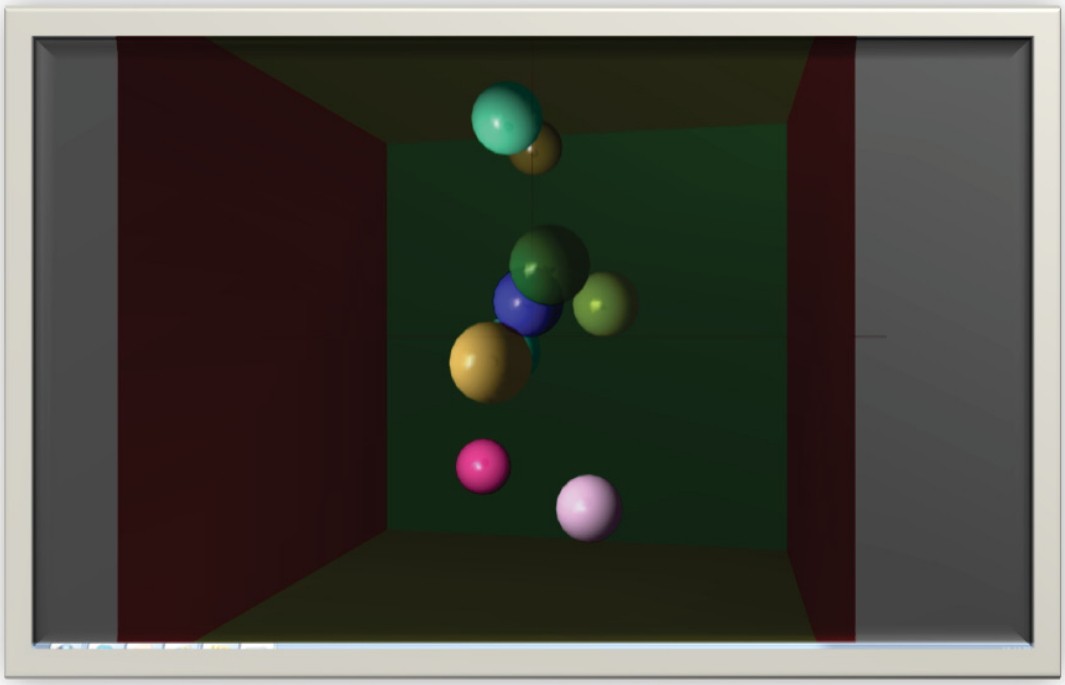


Figure 53: Simple OpenGL Bubbles Prototype Sample

and XNA to demonstrate how the 3D bubbles float randomly in the space. Each of them contains a piece of memory, and will project the bubbles on the environment surfaces, such as walls, ceiling, or floor.

4.2 Design and Implementation

The design and implementation of the interactive documentary work focus on several key aspects: modeling, animation, and interaction. These aspects are supported by specific technologies used. In Section 4.2.1 we begin the conceptual overview of the work as it was being designed. Then the realization details are presented in Section 4.2.2.

1 The media represent memories could be home videos, photos, audio, and animated text (e.g. as in [6]);
2 Model each bubble object in the first iteration as a translucent sphere with a flat polygon inside, procedurally;
3 **begin**
4 Each bubble is defined by the property of the contained media and the outer color. Of course, the bubbles also have a radius r and a position (x, y, z);
5 Each bubble has a unique color, which represents the emotion level of memories, such as lightheartedness and sadness or rainbow-like colors or ease of calling out;
6 The media polygon is a 2-sided quad, onto which a texture photograph can be mapped, or a text item with or without animation, or more generally a video clip;
`// A more advanced bubble modeling under consideration is the utilization of the softbody objects instead of plain spheres to add some realism and tangibility`
7 **end**
8 The audience is allowed to interact with the bubbles, and the bubbles' motion and collision response are to be based on the behavior and gestures of the audience;
`// The effects of bubbles could be soap bubble bursting, softbody dynamics, and rigid body sparkling collision`

Algorithm 3: "Artistic High-Level Algorithm" for *Tangible Memories*

4.2.1 Conceptual Design

As mentioned, the *Tangible Memories* prototype is the augmented *I Still Remember* short film with interaction and projection of the "memory bubbles" in order to visualize how memories are represented by the ten-year-old girl's recollection (see Section 4.1.2, specifically Figure 52 and Figure 53) including additional footage not originally present in the linear *I Still Remember* and the ability to expand it arbitrarily further in the future.

The system's conceptual design is presented in Figure 54, which is an evolved version of the design from Chapter 3 (following the same core principles). The participants, the interactive audience, are central to interact with the installation providing

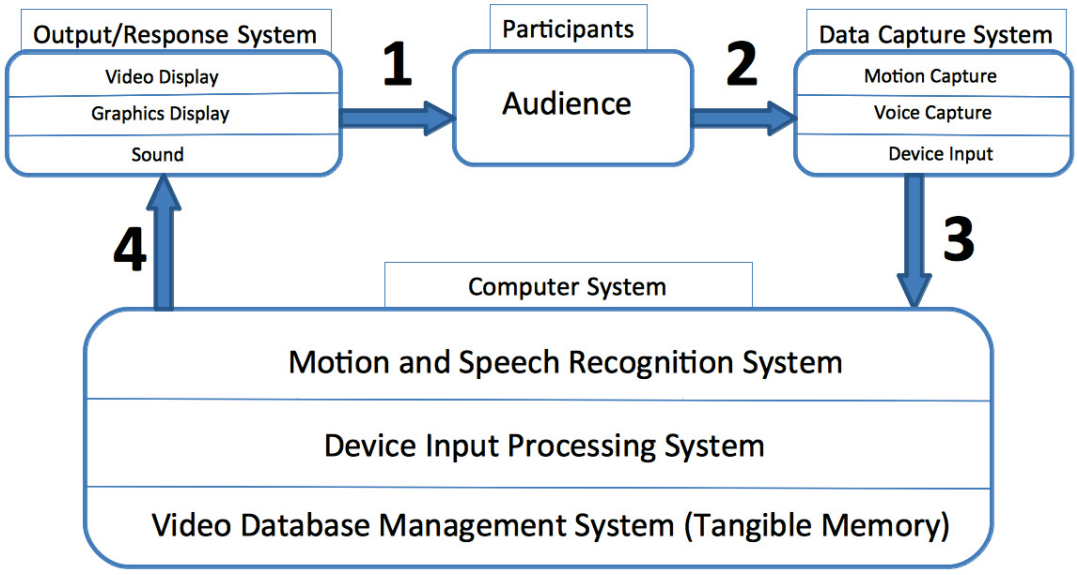


Figure 54: Conceptual Design of an Interactive Documentary Installation

motion, voice, and other captured data as an input to the computer system that processes the supplied input and produces a real-time graphical and audio feedback to the interacting audience.

4.2.2 Implementation

There are two implementations: the first is in OpenGL [15] (primarily with mouse and keyboard interaction only; and haptics planned) and the newer one is in XNA (Kinect, mouse, voice, and keyboard interaction and higher quality footage). We further detail the realization details of the two versions beginning with the API in Section 4.2.2.1, modeling in Section 4.2.2.2, animation in Section 4.2.2.3, content (Section 4.2.2.4), the interaction aspects (Section 4.2.2.5), and projection in Section 4.2.2.6.

4.2.2.1 APIs

We describe some specific APIs that are dominant in the realization of the interactive documentary proof-of-concept work.

4.2.2.1.1 OpenGL and GLUT. The general OpenGL and GLUT concepts are described in Section 2.4.3.1. A GLUT window is created with registered callbacks for idle animation, mouse and keyboard handling along side with a GLUI [218] interface for more control and based on earlier successful GLUT examples [217].

With GLUT, we set up our handling for the `Mouse()` and `Motion()` callbacks with `glutMouseFunc()` and `glutMotionFunc()` for mouse-based tracking and clicking for interaction purposes to estimate a 2D click in the GLUT window to the nearest bubble in the 3D perspective environment. Our `Keyboard()` and `SpecialKeys()` callbacks registered with `glutKeyboardFunc()` and `glutSpecialFunc()` respectively enable the keyboard-based interaction (see Section 4.2.2.5.1. The rendering and timer-based updates callbacks `Display()` and `Update()` are registered with the GLUT's functions `glutDisplayFunc()` and `glutTimerFunc()`.

4.2.2.1.2 XNA and HLSL. The XNA framework provides a collection of relatively easy to use APIs for fast game development in C# (and other Microsoft .NET languages) for various Microsoft platforms. We use XNA and .NET APIs for speech processing in a relationship with Kinect mentioned in Section 4.2.2.1.3 as well as to manage the various media content. In particular, loading the media such as photographic images and video playback within an interactive scene and the corresponding keyboard and mouse controls.

The specific API is used for video playback control and getting movie frames as textures. This is achieved by the provided `VideoPlayer` and `Video` classes. The `Microsoft.Xna.Framework.Content` framework has a variety of importers for media types (e.g. `.wmv`, `.png`) that pre-compile the media into the usable format by the rest of the XNA components, such as `VideoPlayer` that streams the frames as textures into the running program and allows for the user control (pausing, resuming, stopping, etc.).

`SpriteBatch`, `Texture2D`, `SpriteFont`, and the likes are helpful and commonly used classes in XNA to manage the collection and representation of 2D graphics to be

rendered in the pipeline in a certain order. The `Model` class generally represents a 3D mesh. The `Effect` class allows for shader-based effects and the likes to be applied to the models. There are various supporting data types and structures, such as `Vector2`, `Vector3`, `Rectangle` etc., that are commonly used by the various XNA components.

The `Game` API provides methods for initializing and loading/unloading the media content, provides a control loop, make idle and otherwise updates and drawing, a somewhat similar to OpenGL/GLUT pipeline and callbacks.

For the video playback we used the XNA `VideoPlayback` 4.0 sample of how to play frames and control the `VideoPlayer` and `Video` stream from Microsoft under their Permissive License (Ms-PL).

4.2.2.1.3 Kinect and Speech SDK. We rely on a number Kinect API functions to make a rapid proof-of-concept demonstration of the interactive *I Still Remember* memory bubbles remake as *Tangible Memories* [277] from OpenGL primarily because Kinect API [203] originally offered built-in skeleton tracking and associated speech components and a number of code samples (see Section 2.4.4) of how to use those functions. The API also offers the integration with the XNA framework. (The more recent Kinect SDK starting from 1.5 includes C++ samples now as well among other new features.)

We use the `Microsoft.Kinect` framework [225] to obtain references to class instances like `KinectSensor`, and its API to access the skeleton stream frames (the `KinectSensor.SkeletonStream` class), from which `Skeleton`, `Joint`, and `JointType` data are received, along side with the `ColorStream` and `ColorImageFrame` as an input from the normal camera.

`KinectAudioSource`, `KinectSensor.AudioSource`, combined with the API from `Microsoft.Speech.Recognition` framework and its Speech SDK (a part of the .NET Framework 4) [278] to allow for the language-dependent speech dictionary recognition to offer voice based command interaction with the system.

`KinectStatus` provides information about the sensor state and availability.

We used an example implementation as a source of reference and inspiration for Kinect skeleton tracking, including the hand joints, and rendering the overall input video stream. The Kinect API reference, examples, speech recognition come from the Microsoft Kinect for Windows SDK Sample Browser from [231, 203].

The other details of the use of the Kinect API in Section 4.2.2.5.2.

4.2.2.2 Modeling

We describe the modeling for both OpenGL and XNA versions. This includes the memory bubbles themselves, media texture polygons, and the simple environment they float in.

4.2.2.2.1 Modeling in OpenGL and GLUT. Modeling is done as per the related algorithm steps in Algorithm 3, all of which are done using OpenGL. As stated, each bubble is defined by the property of the contained media (`screenMedium`) and the outer color (r, g, b). Naturally, the bubbles also have a radius r, a position (x, y, z), and velocity $\overrightarrow{vel} = (vel_x, vel_y, vel_z)$ [15].

The bubbles float in a confined rectangular space, a world "box", designed for multi-projecting on different surfaces in an installation environment of a blackbox or space alike.

In Figure 55 is a software component model for the OpenGL installation detailing the system design. The `MediaBubble` class is a representation of a memory bubble with a media of one of three types of image, video, or text. It contains the properties described above as well as animation-related properties for the media display or playback as to where to approach to in the virtual space, flash up to indicate selection by the user, or return back to the bubble pool. It contains the `screenMedium` instance reference to a particular media item contained within: `Medium`.

The `Medium` is a generic media class that provides a common API for media items to be rendered within media bubbles. It provides for getting and setting the actual media as well as `Draw()` and `Update()` calls for rendering and animation to

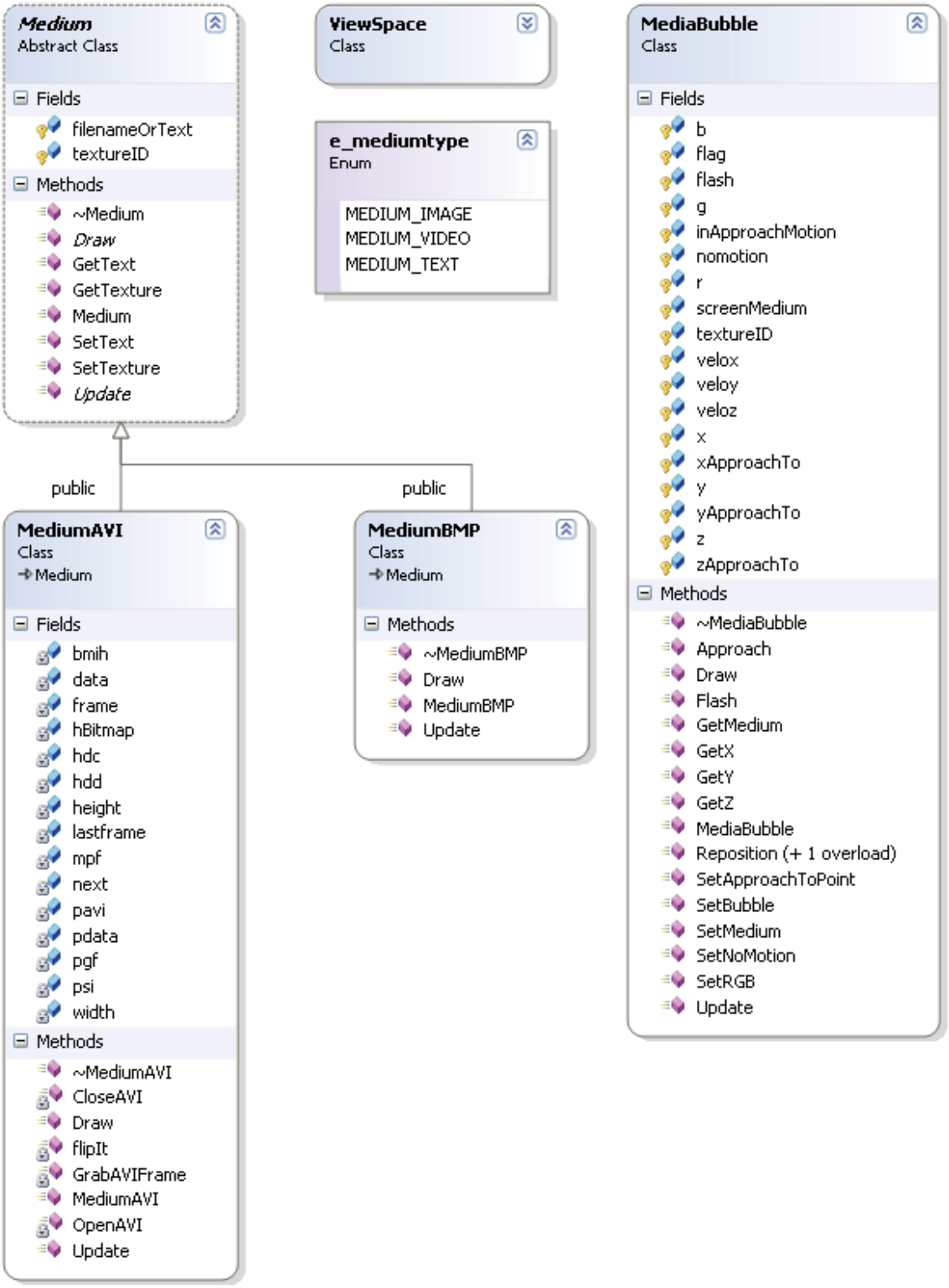


Figure 55: Class Diagram Model of the Software Side of the OpenGL Installation

be obligatorily overridden by concrete media class implementations.

The `MediumBMP` and `MediumAVI` classes are such implementations in that they concretely implement the media-specific rendering functionality. The former delegates the work to a `BMPLoader` to load and store a single texture in the `.bmp` format, whereas the latter wraps and invokes the AVI loader and playback example [229] for playback of the AVI-formatted video content. Each media bubble can contain any of these.

`ViewSpace` provides the updatable world model/environment for modeling of the documentary display on screen, debugging in a color box, or a projection in the blackbox. A light source is also modeled within `ViewSpace` to make the bubbles shiny.

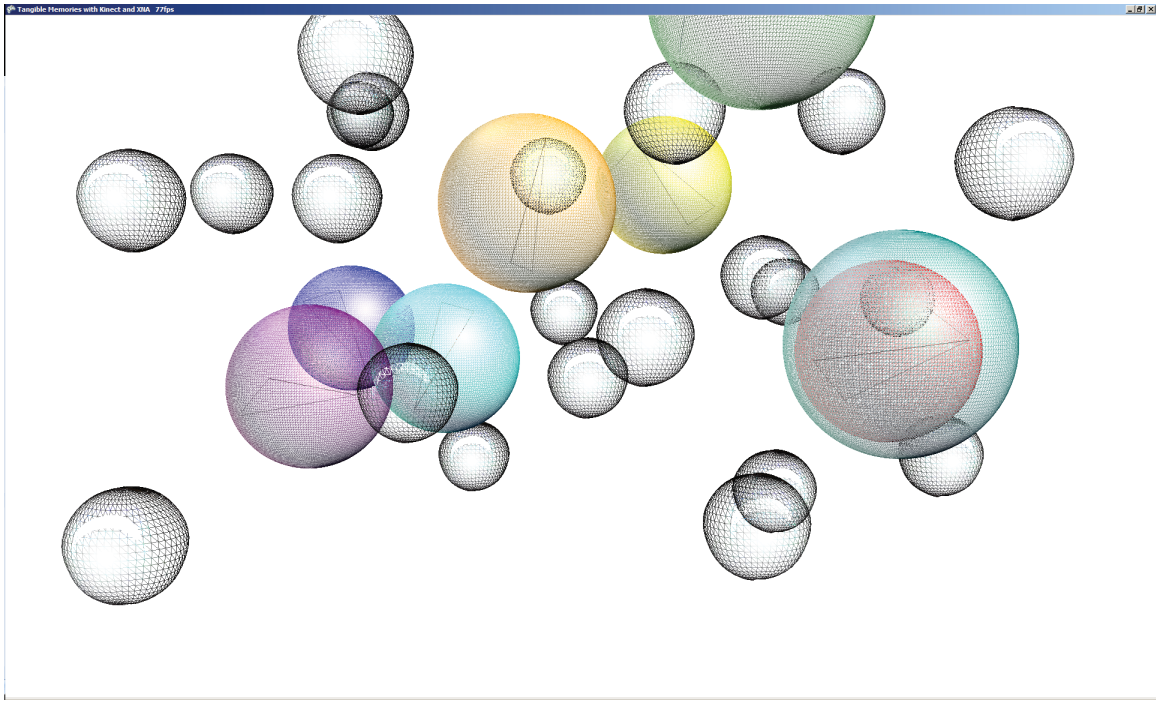


Figure 56: Modeling of the Fancy and Media Memory Bubbles in XNA

4.2.2.2.2 Modeling in XNA and HLSL. The classical performance measuring sample bubbles [230] are computationally modeled sphere meshes with a programmatically specified LOD. Fancy soap animated bubbles [236] have their spherical mesh stored as a `sphere.X` model coupled with the `Bubble.fx` HLSL shader file that does all the texture mapping, pixel/vertex shading, and normal alterations. The wireframe

mesh of the two types of bubbles is illustrated in Figure 56.

The fancy bubble model has an environment map of using the `RenderTargetCube` XNA API to selectively map either a skybox, or video faces (real-time or pre-recorded) and other animations. The ground floor (if shown) as well as the projected media inside the rainbow color bubbles are a simple flat square modeled as two rectangles from `Ground.x` [230] (see Figure 68 on page 155, for example).

4.2.2.3 Animation

Some of the animation aspects in this documentary are quite simple and through that simplicity the main message is unobscured. However, there are some fancier animation features added to illustrate various attractive capabilities.

4.2.2.3.1 OpenGL. The animation in the OpenGL version is simple bouncing off the walls media bubbles in a box world (which is removable for projection purposes). Once a bubble is called upon, it changes its pseudo random bouncing trajectory towards the viewer to occupy the majority of the visible space. Then it is sent back when the user is done watching.

More specifically, the special animation exists when a bubble is selected (`selected` is `true`) by whatever means (mouse click or keyboard): it flashes, then it changes its trajectory to approach the near plane, towards the default viewing position of the synthetic camera ($(x_{ApproachTo}, y_{ApproachTo}, z_{ApproachTo})$ in Figure 55) such that the media in the bubble comes into full view and fills up the screen. When the selected bubble is pushed back (by the same means), it returns to the pool of the bouncing bubbles.

The AVI playback [229] animation of the video media is done by swapping textures, the movie frames, on the media polygons. The video media is constantly played in a loop.

The animation sequence is driven via the `Update()` GLUT callback set through

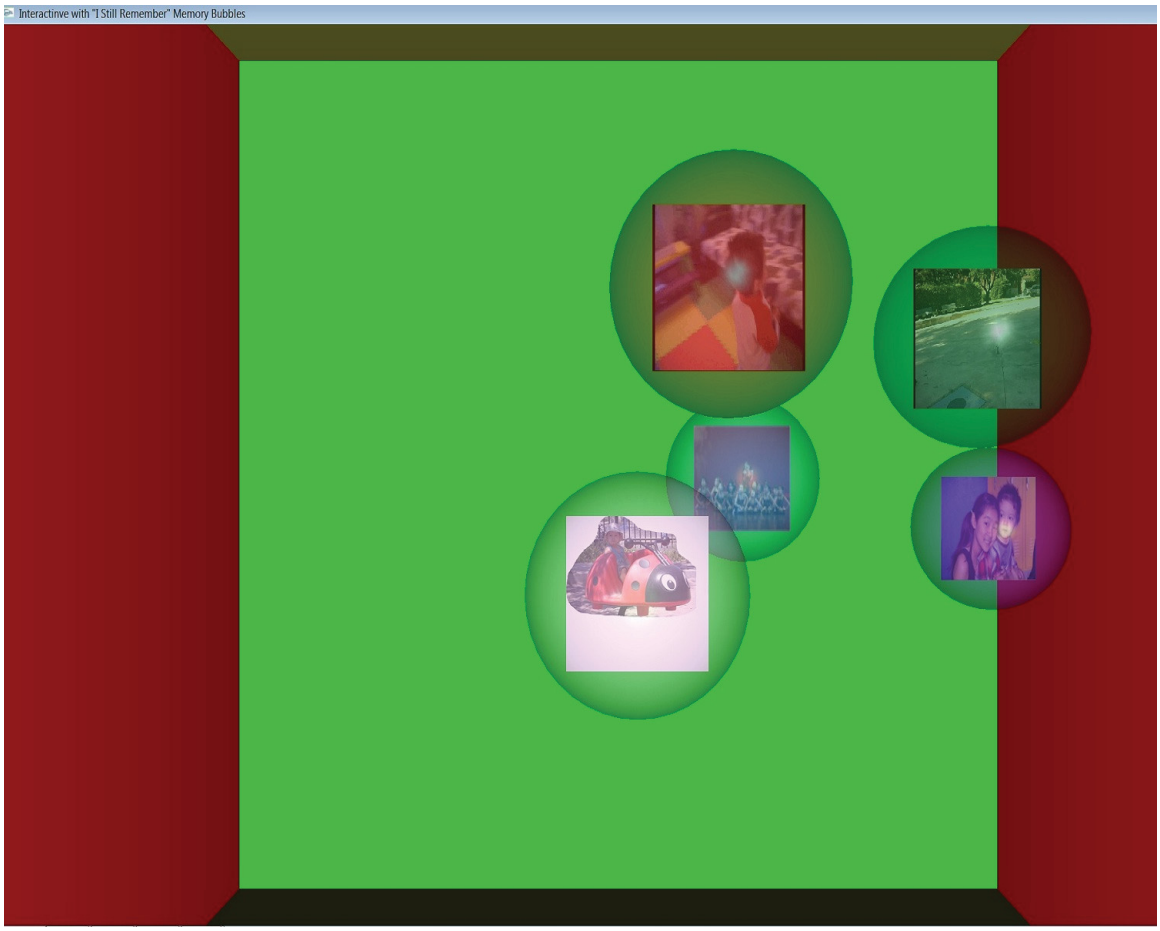


Figure 57: Bubbles Float with the Screen Media Inside

`glutTimerFunc()` programmed to update every 20ms. This allows for balanced consistent animation speeds across different hardware.

4.2.2.3.2 XNA. The basic core animation of the memory bubbles in the XNA version is very similar to that of OpenGL. Additional optional collision detection is performed among the bouncing bubbles themselves [230]. A large number of bubbles are possible, but the zoom-in playback animation is done only for the rainbow-like color bubbles. The media playback is animated only when it is called upon and is done similarly as in the OpenGL version by texture-mapping the video frames one at a time.

Additionally, the fancy bubbles that mingle among the rainbow color ones are animated as floating soap bubbles adapted from [236]. That animation can be altered

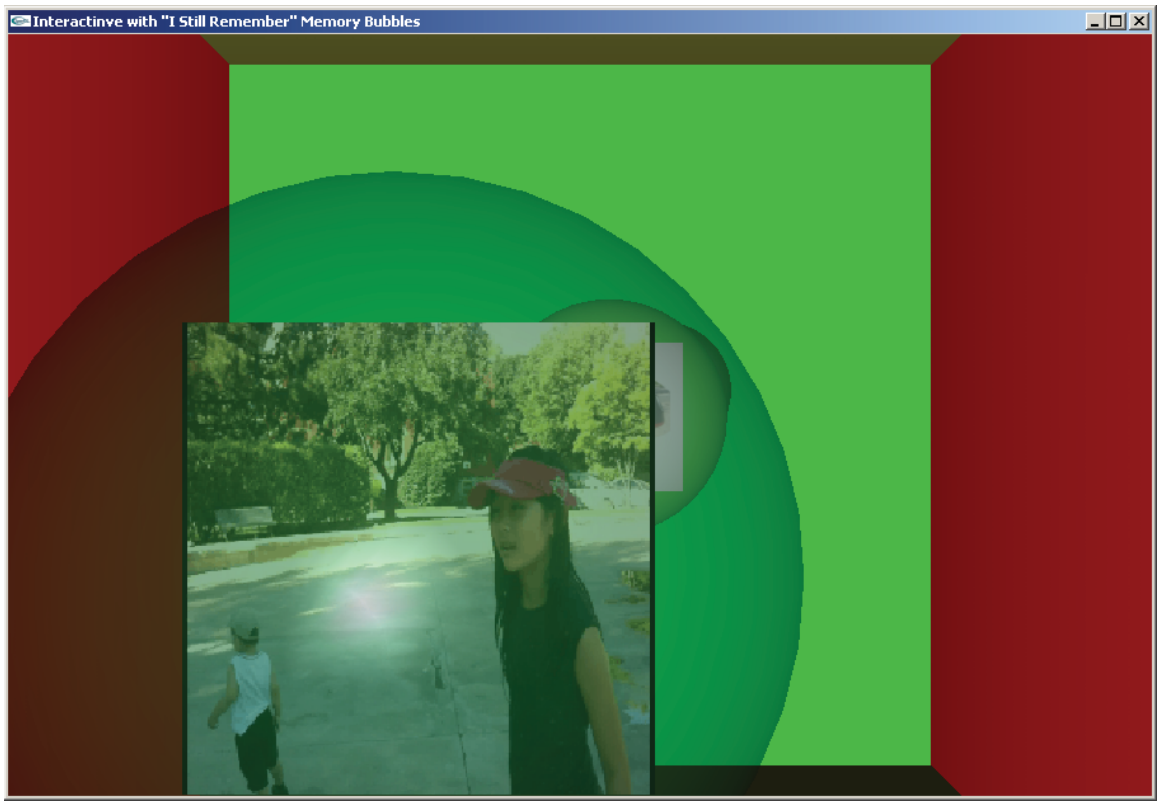


Figure 58: A Close Up Example of a Selected Bubble with a Video Content Playing

by playing various video feed, real-time, or pre-recorded textured on the bubble as it moves and shakes (this feature is explored more in detail later in this chapter and subsequently in Chapter 5). The waving and shaking effect can also be altered by the beat of a music tune being played back along with the use of the BASS.NET [226] library. This wavy `Effect` is primarily implemented in the HLSL vertex and pixel shaders [236] that alter the bubble's mesh and perturb normals and at the same time re-applying textures (environment, skybox, or otherwise) to the new coordinates. These fancy soap bubbles were augmented to collide like the classical media bubbles do with all the other bubbles in their environment.

The media plane inside the media bubbles is animated via simple matrix transformations to rotate the ground plane around x, y, and z axes by the "padded" `wave` value—this also allows for the music-based animation to rotate them along with a beat of a tune if a tune playback is enabled.

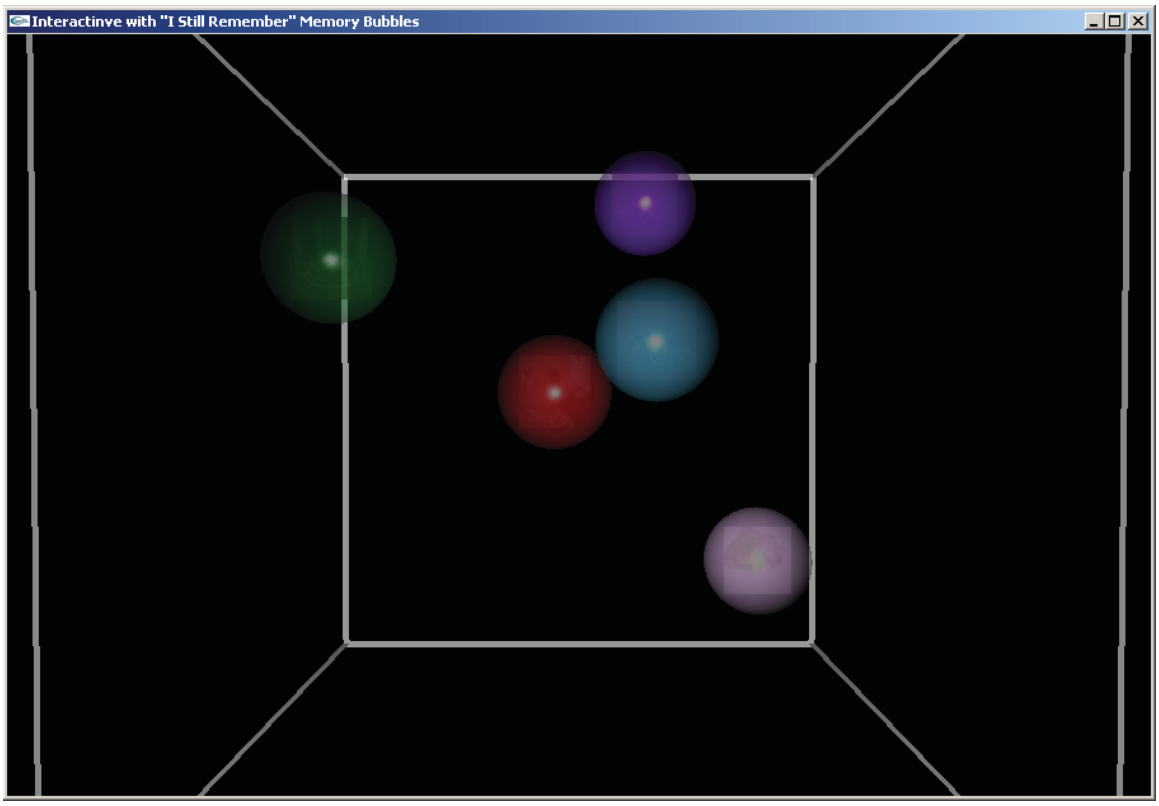


Figure 59: OpenGL *Tangible Memories* in the Black Background for Blackbox Projection

Hand-based animation is also there to partially rotate two fancy bubbles that can also be "held" with the left and right hands via Kinect skeleton tracking. This aspect, however, goes beyond *Tangible Memories* is more about an interactive performance described in Chapter 5.

The animation sequence is driven from the XNA `Update(GameTime)` callback of the `Game`-derived class (see Algorithm 4) that takes into the account elapsed time so the animation could be adjusted to the frame rate of the slower or faster hardware.

4.2.2.4 Media Content

The media bubbles are generically designed to contain the main three types of visual media—a photograph/picture, text, and video. All three can also be animated as desired and augmented with sound (especially the video content). There are some minor differences at present in the two installations produced, but we list the content

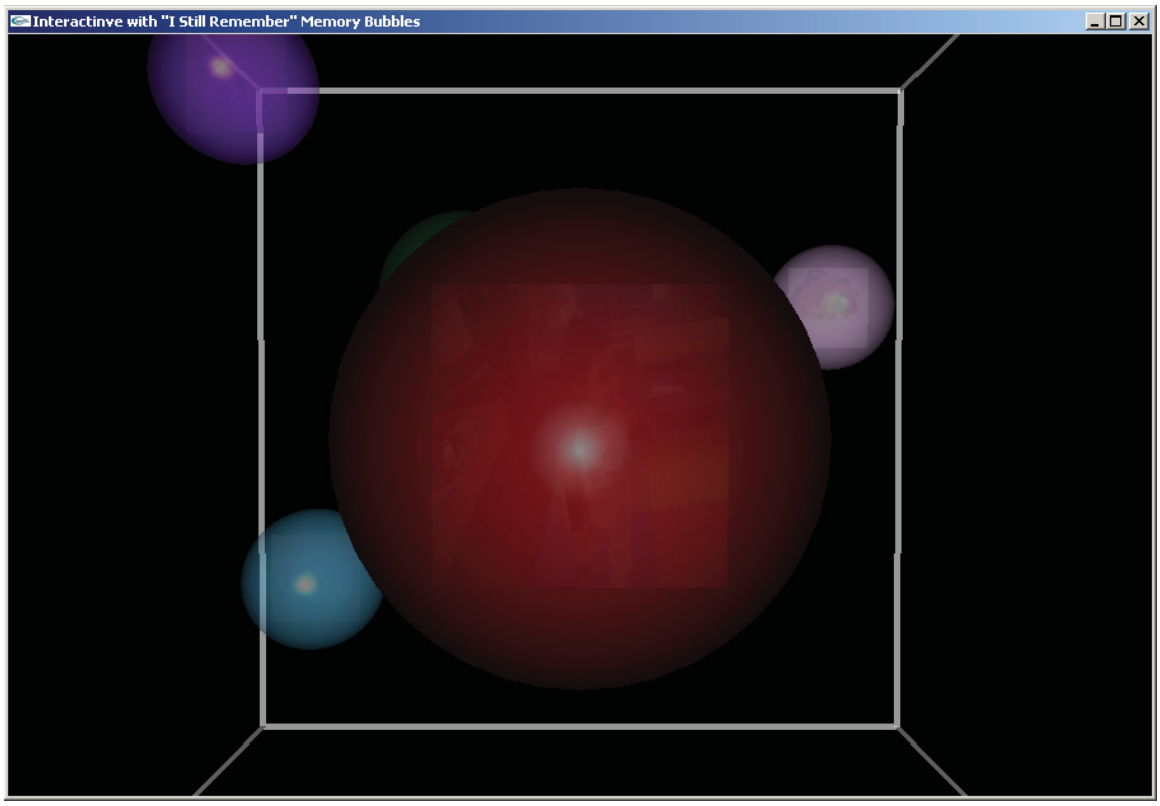


Figure 60: OpenGL *Tangible Memories* in the Black Background with the Red Bubble Called Out

contained in the bubbles. All the content is of our own production and the audience is invited to run and interact with the installations for themselves and explore each bubble.

4.2.2.4.1 OpenGL Installation. Below is the OpenGL's installation bubble contents as it was captured at some point in time. The bubble colors are described, but are obviously the best perceived in the electronic form or a color printout than a black-and-white printout. The media contents in this version are picked randomly shot by home digital camera for prototyping purposes, such as:

1. The green-shaded bubble has a short video of the kids playing in Montreal Westmount park's fountain in 2009.

2. The red-shaded bubble has a another short video of my son in 2009 wandering at home.

3. The light blue contains a photograph of my daughter with her dance school at a Chinese New Year performance in 2006.

4. The white bubble has a photograph of my son on the ladybug buggy in a park.

5. The purple bubble has a photograph of the kids at home.

4.2.2.4.2 XNA Installation. The media contents in this installation are more of professional making and complete. Most of the footage is shot with a professional HD camera.

1. The orange bubble contains the award-winning 13-minute *I Still Remember* with the interview of my daughter shot in the Hexagram blackbox; the work that prompted this whole project of the interactive documentary presented in this chapter [1, 15].

2. The violet/purple bubble has the footage of the ceremony of *I Still Remember* receiving the Best Short Documentary Award at the 1st BJIFF (Beijing International Film Festival) at the closing ceremony.

3. The yellow bubble is the collage of VWIFF (Vancouver Women in Films Festival), at which *I Still Remember* also was officially screened.

4. The blue bubble has the interview by the BTV (Beijing Television) *View China* Program about how *I Still Remember* was made.

5. The red bubble contains a short mini-documentary *St-Valentine*, in which children were preparing the Valentine candies and cards for their classmates and teachers. They are also the real scenes and life behind *I Still Remember*. *St-Valentine* was edited by my daughter.

6. The light blue / cyan bubble contains a short 3-minute documentary my daughter did about her little brother—*My Little Brother* that got selected as one of

the top 100 international films and was screened at the Young Cuts 2012 film festival taking place at Concordia in October 2012.

7. The green bubble features the *Spectacle* animation I did in Maya [276].

8. The teal bubble, which is not a part of the 7 rainbow-like color bubbles, has a real-time video feed from the Kinect's color stream camera itself.

4.2.2.5 Interaction

We describe the implementation of the interaction and the user control of the work for both versions (OpenGL in Section 4.2.2.5.1 and XNA/Kinect in Section 4.2.2.5.2) and their particularities.

Similarly to the followup chapter's recorded clips and the discussion on the generalization of the proposed system Section 5.2.3.6, we record the interaction example Clip 4—voice control of the bubbles (with a person calling out the bubbles, etc.) along with the narration about multidisciplinary the research recorded in the white-box production room.

4.2.2.5.1 OpenGL Version. The present level of interactivity (shown, e.g., in Figure 57) by the audience is via simple mouse clicks in the 2D coordinates, which are then translated to the nearest bubble, which is then "selected" and zooms in slowly onto the viewer with its content gradually taking over the full screen. In the bubble the corresponding media content (photograph, text, video clip) is rendered (see Figure 58). Clicking the zoomed-in content makes it recede back to the bubble pool of memories [15]. A similar effect can be achieved by the assigned keys '1'–'5' for quick and precise testing and showcasing. 'W' can also switch the box environment from the color (Figure 57 and Figure 58) to black box for projecting (Figure 59 and Figure 60).

More advanced methods of interaction in the works are described in Section 6.5.4.

Figure 61: Interaction in Debug Mode with Speech and Keyboard On

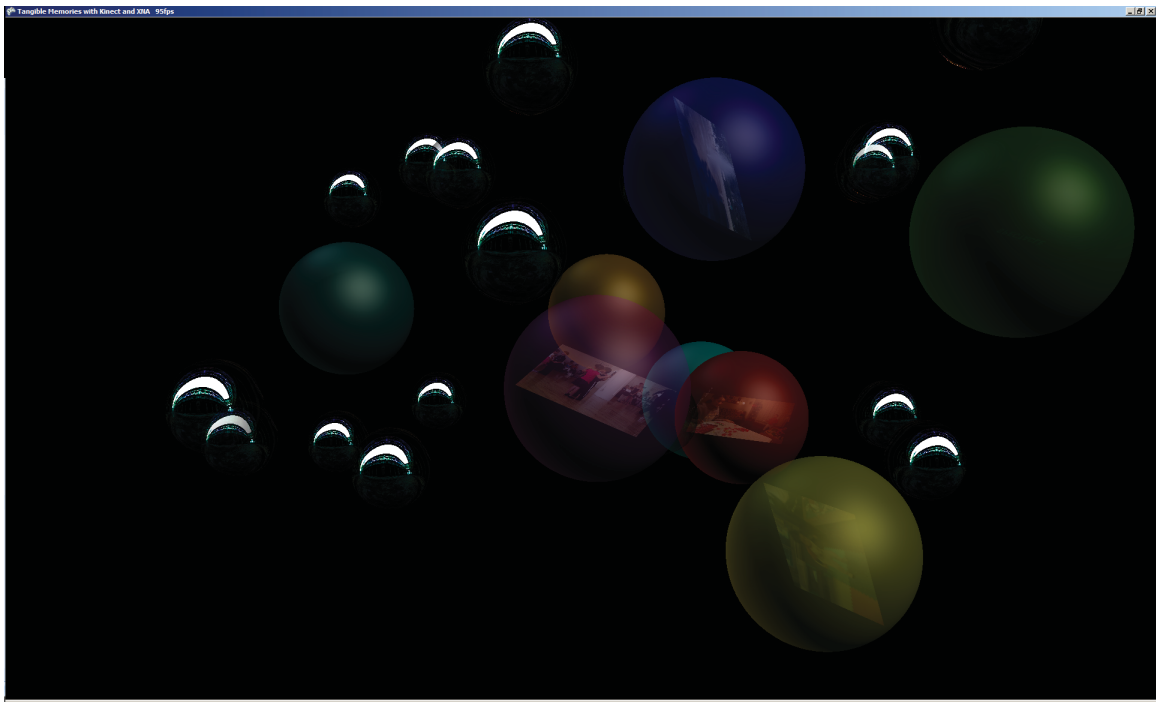


Figure 62: Fully Shaded XNA Bubbles on Black Background

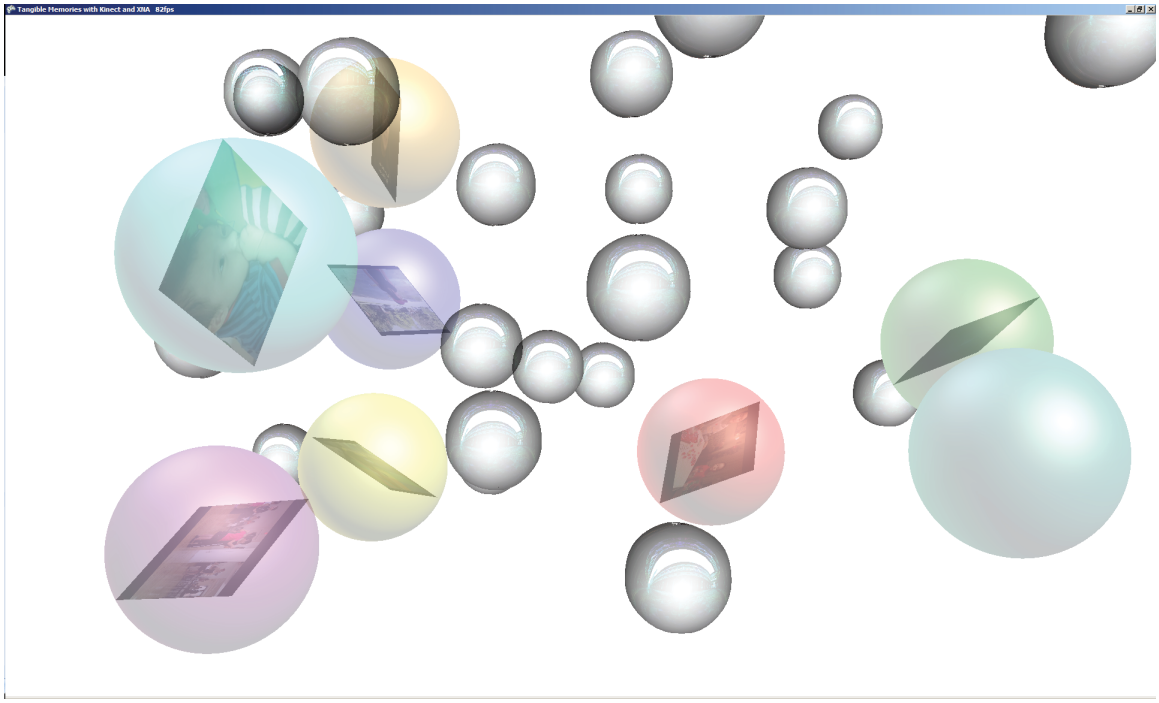


Figure 63: Fully Shaded XNA Bubbles on White Background

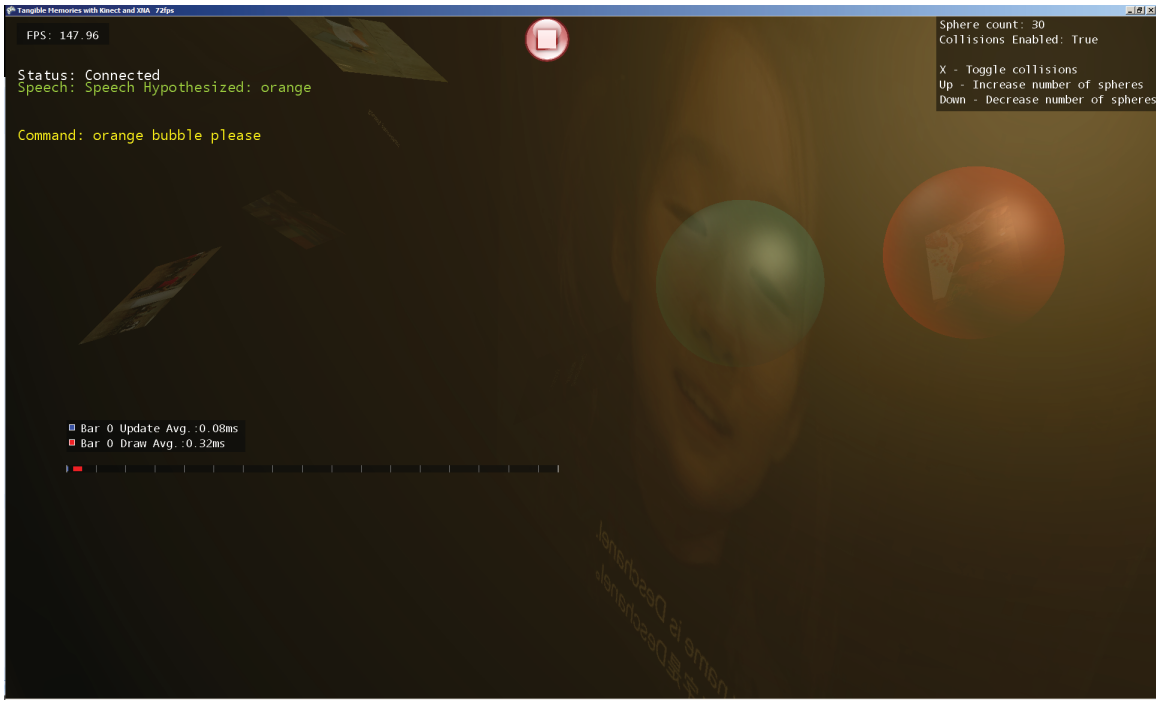


Figure 64: Calling out an Orange Bubble by Voice Containing the *I Still Remember* Documentary Playing (Debug Mode, Black Background)

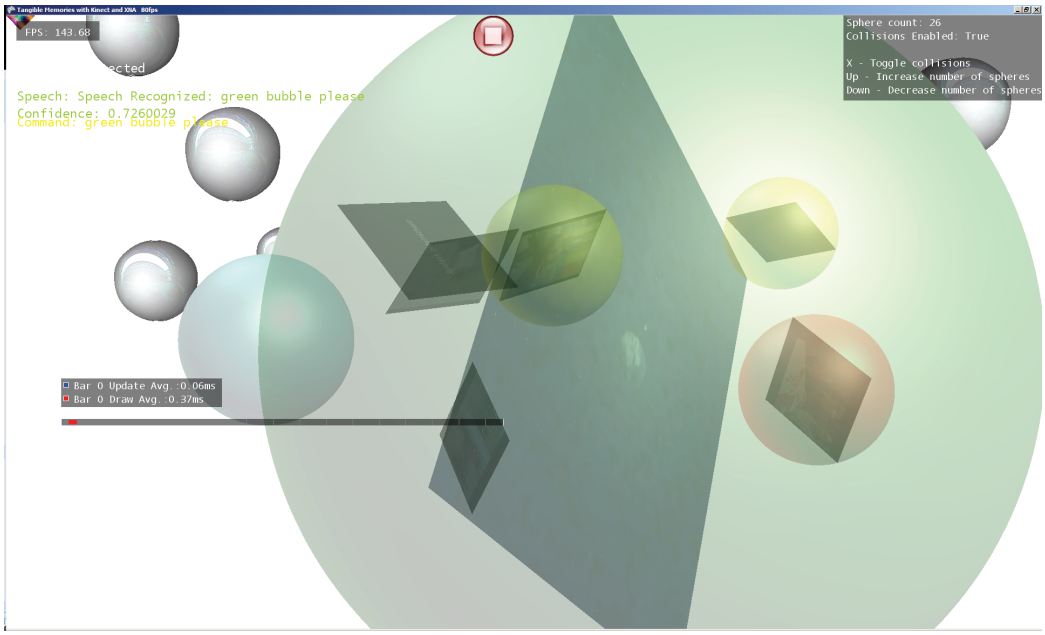


Figure 65: Calling out a Green Bubble by Voice Containing the *Spectacle* Animation Playing (Debug Mode, White Background)

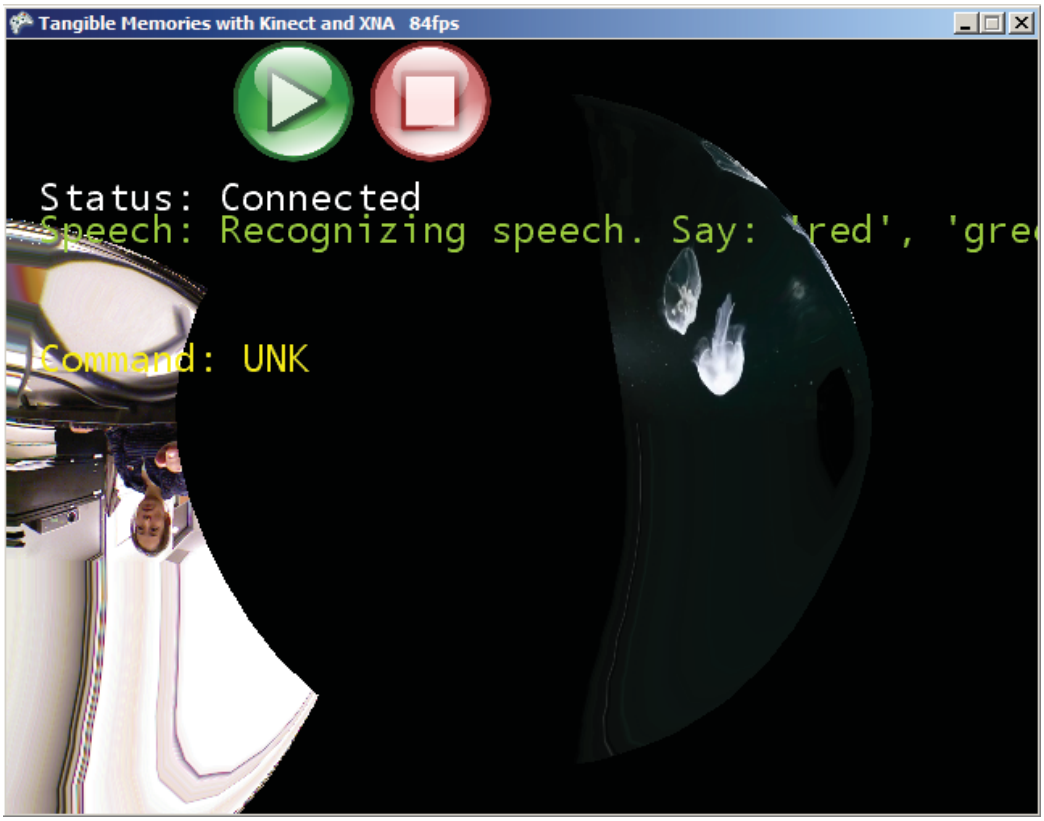


Figure 66: Two Hand-Controlled Fancy Bubbles With Read and Recorded Video Feed in Debug Mode

```
1  Program.Main();
2  begin
3      Game.Run();
4      begin
5          Initialize();
6          begin
7              Register Kinect observer/listener method of status change to be KinectSensor.KinectSensors.StatusChanged =
               KinectSensors_StatusChanged();
8              DiscoverKinectSensor();
9              begin
10                 foreach KinectSensor sensor ∈ KinectSensor.KinectSensors do if sensor.Status == KinectStatus.Connected
                   then
                       // Found one, set our sensor to this
11                     kinectSensor = sensor;
12                     break;
13                 end
                   // Init the found and connected device
14                 if kinectSensor.Status == KinectStatus.Connected then
15                     InitializeKinect();
16                     begin
                           // Color stream
17                         kinectSensor.ColorStream.Enable(ColorImageFormat.RgbResolution640x480Fps30);
18                         kinectSensor.ColorFrameReady = kinectSensor_ColorFrameReady;
                           // Skeleton Stream
19                         kinectSensor.SkeletonStream.Enable ( new TransformSmoothParameters()  Smoothing =
                           0.5f, Correction = 0.5f, Prediction = 0.5f, JitterRadius = 0.05f, MaxDeviationRadius =
                           0.04f  );
20                         kinectSensor.SkeletonFrameReady = kinectSensor_SkeletonFrameReady;
21                         kinectSensor.Start();
22                         Speech.Program.StartKinectAudioProcessing(kinectSensor);
23                     end
24                 end
25             end
26         end
           // Loads media files:  images, videos, models, effects
27         LoadContent();
           // Updates within a loop like idle
28         Update(GameTime);
29         begin
30             Process player states to stop/resume;
31             Process call back keys;
32             Update skeleton/bubble capture condition;
               // The positional skeletal tracking and voice streams are updated by the registered Kinect callbacks in the
                  InitializeKinect() method where the variables for joints of the left and right hands are tracked.
                  Likewise, asynchronous audio input source from Kinect is tracked by a speech recognized initialized there
                  and a global variables set with the states that are used here.
33             Update video playback condition of a captured bubble (by gesture of a left hand in predefined spots or voice
               commands “stop” or “play”);
34             Update free-floating memory bubbles position;
35         end
           // Renders content
36         Draw(GameTime);
37         begin
38             Update free-floating memory bubbles positions;
               // The playback is stopped when bubbles freely float for speed
39             Acquire and render video frames as textures for close-up video playback of a captured bubble;
40             Update and render left capture hand for play and stop buttons;
41         end
           // Unloads/disposes of contend upon exiting the loop
42         UnloadContent();
43     end
44 end
```

Algorithm 4: Kinect/XNA Interaction Algorithm

4.2.2.5.2 Kinect Skeleton and Audio Control. In Algorithm 4 is the complete overview of the algorithm how Kinect- and XNA-based interaction are performed (that goes beyond the interactive documentary concept and into the interactive performance, a topic of Chapter 5). The details of the keyboard, gesture, and speech

based specific commands for interaction are listed in Section A.2.

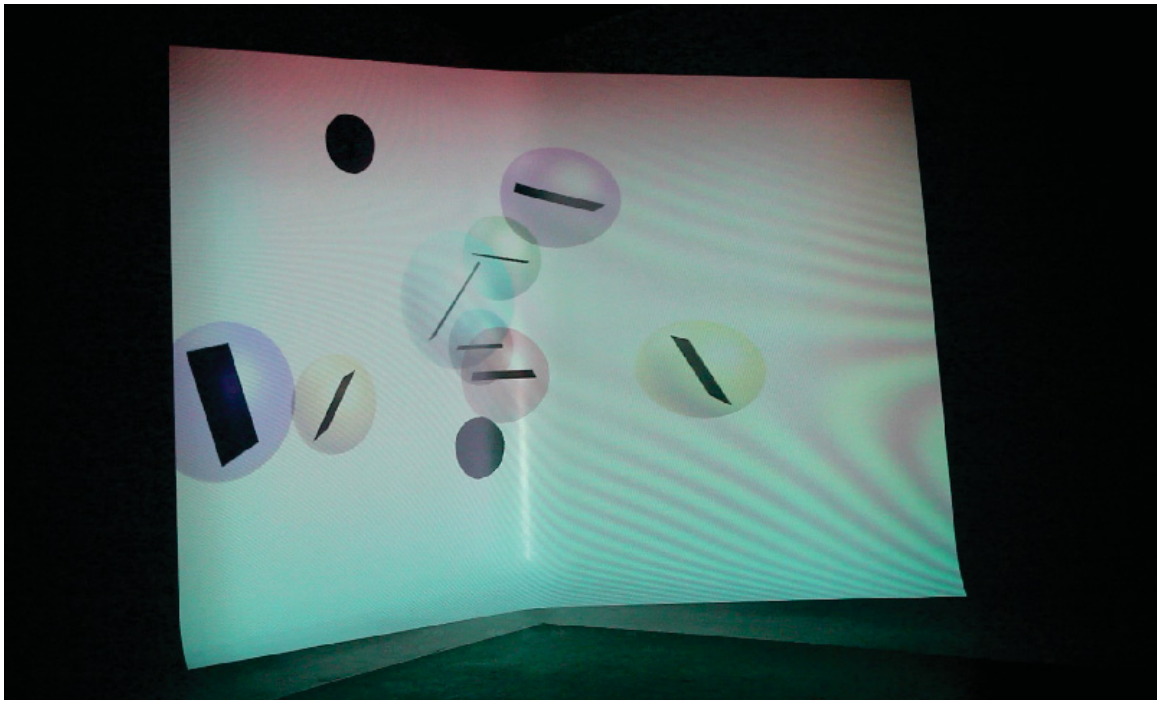


Figure 67: Projection of *Tangible Memories* in the Production Room Corner

In Section 4.2.2.1.3 some audio and Kinect APIs that we use are mentioned. Specifically, we capture the audio stream and run `Microsoft.Speech`'s recognizer for the speech-based interaction. We define a dictionary of words and phrases (see Section A.2.3) using the English grammar instance (which we plan to try with other languages).

Skeleton tracking provided by the Kinect API allows tracking joints, and, in our case, specifically left and right hands to hold on to or bounce off the bubbles in this installation instance.

In Figure 61, Figure 62, Figure 64, and Figure 66 are some screenshot results of the prototype XNA-based installation with Kinect and speech recognition.

The bubbles can be called out by their color or by keys (see the Algorithm 5 and Chapter A.2 for more specific details on that aspect).

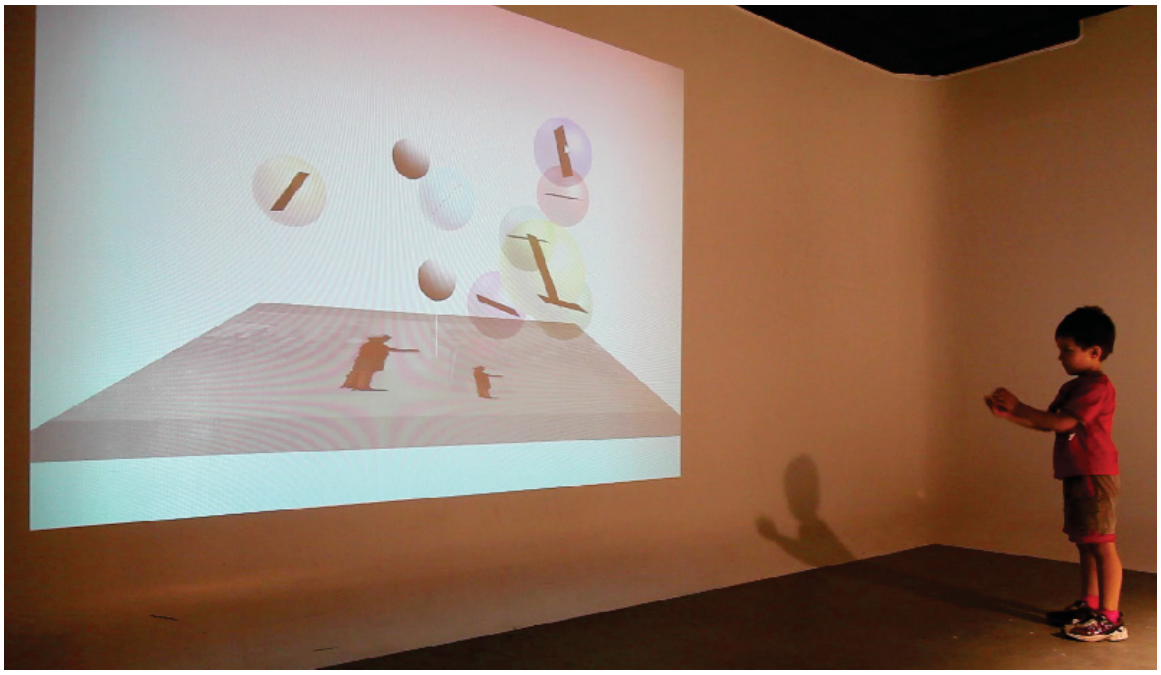


Figure 68: Audience Interacts with the Projection of *Tangible Memories* in the Production Room

4.2.2.6 Projection

The installation is projected in a blackbox or an enclosed production room alike with a full-screen mode on with either black or white background. On the low-end the inexpensive projectors are used with standard VGA-based 1024×768 projected image. At the higher end the HD projection for large spaces can be used.

In Figure 67 and Figure 68 are examples of a medium scale projected interactive installation of the XNA *Tangible Memories* inside a production "whitebox" room with a white background and the ground floor rendered with green screen capture techniques and extraction of audience interacting with the bubbles.

```
1  begin
       // By default
2      Music off;
3      Enable speech by default (bug: still need to press 'T' twice);
       // Start as normal with bubbles
4      begin
5          Show full screen (press 'F' and then double-click the title bar);
6          More or less fancy bubbles;
7          begin
8              By keys "↑"/"↓";
9              By speech saying "more bubbles", "less bubbles";
10         end
11         "PgDown"/"PgUp" zoom in/out for the best fitting projection;
12         Call out the media bubbles;
13         begin
               // See the color list in the keys list
14             By voice using the bubble color;
15             By key;
16             begin
17                 "F4" – simulate "red" command for testing;
18                 "F5" – simulate "orange" command for testing;
19                 "F6" – simulate "yellow" command for testing;
20                 "F7" – simulate "green" command for testing;
21                 "F8" – simulate "cyan", "light blue" command for testing;
22                 "F9" – simulate "blue" command for testing;
23                 "F10" – simulate "violet", "purple" command for testing;
24             end
25         end
26     end
27 end
```

Algorithm 5: Configuration and Setup Interaction

Chapter 5

Illimitable Space in Responsive Theatre

Theatre is not only the nexus where computer technology and artistic creation meet, but also the most powerful artistic form in itself. Live performance includes all the possible elements, lighting, fashion, video, music, storytelling, painting, sculpture, computer graphics and animation, mobile media, whatever term one could think about: all could be integrated into theatre performance. However, I would still use Aristotle's six essential elements of drama [189, 279], *Spectacle*, *Character*, *Plot*, *Diction*, *Melody*, and *Thought*, to guide this theatre work innovated with the new technology and thoughts. We map these elements to the contributions in this work in Section 6.6.2. In fact, it could be the backbone carrier of different media types, which binds science and arts elements.

In Section 5.1 we further describe the methodology behind our specific approach, then we delve into the details of the realization of the installation in Section 5.2, and we summarize our findings in Section 6.6.

5.1 Methodology

The core ideas behind the interactive performance include a Kinect-driven, real-time CG animation with the audience and participants interacting with the system (and potentially each other).

There are four primary examples in this work described in Section 5.1.2. However, the possibilities with the system prototype are not limited at all to just these specific four elements and forms and it can be reconfigured for a large number of interactive performance scenarios.

Thus, we state the goal of the methodology in Section 5.1.1, followed by the illustration of the proof-of-concept realization of this goal in Section 5.1.2.

5.1.1 Goal

First of all, traditional theatre is limited within three walls, a ceiling, and a floor. I intend to break this standard rule to free actors and audience in a dynamic illimitable space. Second, the conventional relationship between the audience and actors, spectating vs. performing, needs to be altered; instead, audience will also actively participate the *Plot* and be a part of the performance.

5.1.2 Proof of Concept

The presented proof of concept (PoC) is illustrated in the follow up stills of a test performance production in a Hexagram Concordia video production room. The illustrations, in Figure 70, Figure 5.71(a) Figure 5.72(a), Figure 73, and Figure 74, show the perception of depth, performance, and real-time visualization of sound (in this example, we visualize the music background). This is the first proof-of-concept realization of the installation that has vast possibilities to be used in many performance art aspects, some of which we will explore in the future directions and work in Section 6.6.3.

The first is the dancer with the three graphical figures of her performing and

animated at the same time: the depth shadow, greenscreen background replacement and real-color image, and a "stick-man" skeleton (see Figure 77 and Figure 78). All are founding core techniques that can be augmented and wrapped with various extra features available in today's technologies such as "avateering" [203] (let's call it *digital puppetry*), depth and skeleton based interaction with the virtual elements, and art. This example [280][1] combines some of the elements from the subsequent examples that were incrementally built up.

A real-time green screen mapped audience is added to a pre-recorded video footage (see Figure 79 and Figure 80), in the effect to multiply the audience, with various sizes, distances, depths, and potentially projected onto different scene elements [281][2].

Water-sleeve Chinese dance with music visualization blended together is the third example of the real-time visualization of the rendered depth in rainbow-like colors, (e.g. shown in Figure 70, Figure 73, Figure 74, and related). This is the classical example of working with the depth stream and real-time music visualization [282][3].

Real-time interaction and video feed can be projected onto a multitude of virtual and real surfaces. In the fourth example, we re-use the augmented fancy soap bubble design [236] from Chapter 4, to project the depth stream onto it while the bubble's wave animation and rotation is affected by the beat of the music played as well as waving hands (gestures) to rotate it. A 6-plane render target projection is mapped onto the bubble's sphere; each face can contain a real-time video feed, a stored video, a static texture, an animation, music visualization, green screen effects, or all of the above blended based on the desired configuration of the installation artist. See Figure 75 and Figure 76 for examples [283][4].

[1] `https://vimeo.com/49682696`
[2] `https://vimeo.com/50069419`
[3] `https://vimeo.com/49399617`
[4] `https://vimeo.com/51329588`

5.2 Design and Implementation

The design and realization of this installation centers around the notion of special data streams provided by the Kinect sensor and its drivers [203, 204]. These specifically include in our case the color stream, depth stream, skeleton stream, and audio stream (source). The streams are provided by the sensor's driver to the installation application in real time with the corresponding API (see Section 5.2.3.1).

We combine color and depth streams to form a single texture frame by frame, which is used in a green-screen like processing regardless the background, or rainbow-color the depth layers that are truly representing the perception of depth (see, e.g., Figure 70).

Another important aspect of the installation is the sound (see Section 5.2.3.4). A melody playback via the BASS library [226] is translated into the visual representation and blended into the depth/color images. It also can drive the animation of a fancy soap bubble presented earlier in Chapter 4 for its music-driven wave pattern.

The skeleton stream gives us a reference point of the performer presence in the scene. These data are important for the gesture-based interaction aspects (see Section 5.2.3.6).

We begin the review of the conceptual design in Section 5.2.1 and the associated challenges in Section 5.2.2. We then present a detailed overview of the technical production and realization of the work in Section 5.2.3.

5.2.1 Conceptual Design

The logical pipeline of the performance installation is depicted in Figure 69, which combines computer graphics, documentary montage, interactive media, and theatre performance into one artistic piece, which transitions through four primary states:

- State 1: In a makeshift initial theatrical space, such as blackbox, where lighting, projection, sound props, and related facilities are properly set up for actors' performance, audience's participation, and other user group's interaction devices.

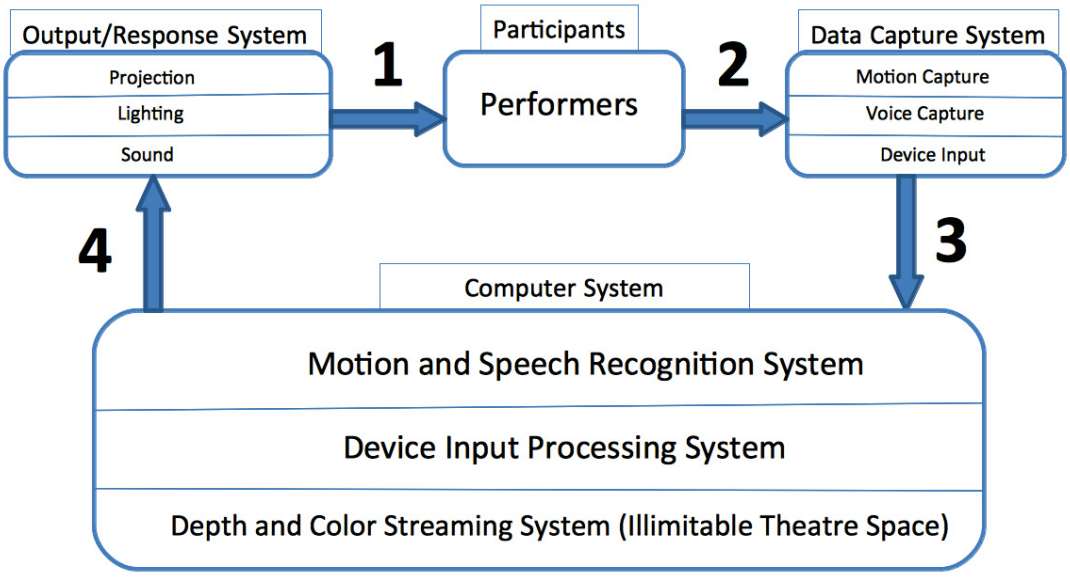


Figure 69: Conceptual Design of an Interactive Performance Installation

- State 2: Participants, including actors, audience, and other users could freely move in the physical space. They could move their body in different postures with different gestures, talk or sing, and could play musical instruments, or touch haptic-enabled devices. The state of human being's performance, such as motion of actors' body movements, audience's voices and the forces of users applied through the haptic devices, are captured through the integrated Data Capture System to generate a response.

- State 3: The captured data from the theatrical space are transferred to the PACS (Performance Assistant Computer System) to be analyzed. The computer system mainly contains three big components: recognition, computing, and media output. The motion recognition system, voice recognition system, and device processing system are filtering, parsing, and sorting motion, audio, and device/sensor input streams captured into system by various capture devices. The computer system then computes and simulates physical based computer graphics and dynamic audio based on the mathematical and physical based

algorithms, and fetch proper video footage from a video database management system following some querying algorithm.

- State 4: The outputting system then reconfigures the theatrical technologies afterward based on the computation results and state updates. The outputting system then re-renders the newly computed computer generated imagery onto projectors, plays generated audio signals through speakers, and even controls dynamic lighting (interacted with originally as a result of participants' actions).

In fact, the presented abstract system is applicable and can be generalized to incorporate all the presented concrete simulation and animated systems so far: the Softbody Simulation (see Chapter 3), Tangible Memories (see Chapter 4), and the Illimitable Space performance presented in this chapter.

Figure 70: Depth-Based Dance Performance Capture and Visualization

5.2.2 Challenges

The core posed challenges presented in this work have the same constraint:

- Realtime sound/music
- Realtime graphics/video effects
- Realtime CG objects physical based simulation/digital puppet theatre
- Live performance

That is being live and real-time–i.e. happening *now* and at the same time as opposed to a pre-recorded post-production piece. The hardware and software technology we use today allow us, inexpensively, to meet this hard constraint.

5.2.3 Implementation and Production

The implementation, realization, and production of the interactive performance involves the use of the technological innovations in terms of software and hardware presented earlier as well as traditional dance, and documentary art forms. This corresponds to the software APIs used in Section 5.2.3.1, modeling details in Section 5.2.3.2, projection aspects described in Section 5.2.3.3, details on the lighting setup and audio environment are described in Section 5.2.3.4, the live performance aspects weaved into the installation as well as some preliminary proof-of-concept stills are rendered in Section 5.2.3.5, then how the performance interaction and the real-time animation with it are detailed in Section 5.2.3.6, and, then, we review how that maps back to the most prominent theatre elements in Section 6.6.2.

5.2.3.1 APIs

We rely on a number of the APIs introduced and detailed earlier in part or in detail in Section 2.4.1.1 and Section 4.2 as the technologies used are integrated together in the same application, so they can be used for different installation types, including the theatrical performance or interactive documentary, but the emphasis shifts between different aspects of the implementation can be configured for the desired installation type and environment.

5.2.3.1.1 XNA. A lot of the XNA API described earlier in Section 4.2.2.1.2 is directly applicable in this installation as well as the two share a lot of elements in common and the illimitable space installation is in fact a generalization of all the approaches presented here.

The additional API is used more in this installation is the `RenderTargetCube` from `Microsoft.Xna.Framework.Graphics` for the dynamically updated sphere mapping onto the fancy soap bubble model with the desired media (instead of the default static `TextureCube`). Each face in the cube is an updatable `Texture2D` allowing us to project onto the bubble any kind of video playback, animations, visualizations, etc.

A `Model` instance is loaded with `Ground.x` for the primary ground projection ("ground" here denotes a flat surface that is at the "bottom" of the virtual environment in the installation by default, but technically can be projected anywhere, e.g. such as walls, ceiling, or media bubble's content). The `Effect` instance is loaded with `Bubble.fx` [236] as before for the HLSL fancy soap bubble shaders (a single pass collection of the vertex and pixel shader to map the glow texture to the cube map onto the bubbly sphere while deforming its mesh).

The `RenderTarget2D` is used for blended textures used in the sound visualization projected onto the ground. There are targets each having `DepthFormat.Depth24` color bits and render the sound frequency and wave coefficients as rotating amplitude curves and spectrum bands blended into the same `SpriteBatch` with the alpha channel enabled for transparency and without erasing the previous content for the curves [228] (see Section 5.2.3.1.3 for more details). We subsequently blend this visualization with the depth, skeleton, and green-screen color images to form a final image, such as, e.g., in Figure 77 and Figure 78.

5.2.3.1.2 Kinect. A lot of the Kinect SDK API described in Section 4.2.2.1.3 is also applicable here. Additional elements in this installation include the more extensive use of the `Microsoft.Kinect.KinectSensor`'s `DepthStream`, `SkeletonStream`,

and AudioStream APIs. They help to realize the methodology and conceptual design described in this chapter. The algorithm presented earlier in Listing 4 is applicable here as well with the extensions mentioned: just extra paths that are configured to invoke the processing of the three additional streams, mapping their content to color and visualization, and finally rendering them.

Most of the relevant API comes from the Microsoft.Kinect package.

First, we work with the visible spectrum camera and its color stream. We capture each ColorImageFrame with the default format of ColorImageFormat set to RgbResolution640x480Fps30, which is a reasonable compromise. Then we maintain a ColorImagePoint array of color coordinates for maintaining the depth-to-color mapping for the depth visualization. The KinectSensor's API provides a helper method to that effect of KinectSensor.MapDepthFrameToColorFrame(). The majority of this color stream API invocation comes from the callback handler called kinectSensor_ColorFrameReady(), which in addition to what was described does the greenscreen processing/abstraction by capturing the depth data of the object in motion in the Kinect's field of view, that becomes tracked and the corresponding set of indices are mapped into the color frame's image and the rest of the pixels are set to transparent as in the green screen example [204].

Similarly to the color stream frames, each DepthImageFrame is acquired by our callback handler kinectSensor_DepthFrameReady() at the DepthImageFormat of Resolution320x240Fps30. Each depth value is equally-spaced in the spectrum beween the nearest and farthest values to map to the near-rainbow color layers with the red begin the farthest and violet the nearest layers (see e.g. Figure 73 and Figure 74).

The KinectSensor.AudioSource serves as the audio stream input, typically however enabled when the speech recognition is on (see Section 4.2.2.1.3). The audio stream processing from an .mp3 file for visual music feedback is described in Section 5.2.3.1.3.

We also receive in the callback handler kinectSensor_SkeletonFrameReady() SkeletonFrames with the Skeleton data that allow us to track Skeleton.Joints.

These are used in SkeletonPoints together with the corresponding method called KinectSensor.MapSkeletonPointToColor() [231, 203] both to visualize the skeleton as an element of debugging or performance (e.g., see Figure 77 and Figure 78) and to use this information to interact with the virtual objects on the scene, e.g., bouncing things off (Skeleton.Joints gives us a lot of elements of a tracked skeleton encoded in JointType, e.g. JointType.Head, JointType.HandRight, JointType.FootLeft, JointType.AnkleRight, JointType.HipCenter, and many others) or rotating the objects by hand-waving (see, e.g., Figure 75 and Figure 75).

5.2.3.1.3 BASS.NET. The .NET wrapper [226] of BASS [227] offers API to tap into the audio stream's properties including the beat via frequency characteristics. This information is used as an input to drive the animation of the music visualization on the projection plane as well as the fancy soap bubble rotation and wavy effect (e.g. Figure 5.75(a) and Figure 5.75(b)). (Additionally, in the *Tangible Memories* case the bubbles' random floating motion can be affected by the music being played back at the time.)

We have integrated the music processing referenced from TalkShowHostXNA [228] that relies on BASS. The library is loaded, setup, and configured to playback .mp3 files. More specifically, from the Un4seen.Bass package BassNet is loaded and registered (BassNet.Registration() and the Bass class is the one that provides the functions needed to playback .mp3 file and access its properties during the playback.

Bass.BASS_Init() initializes the library to sample at 44.1kHz. Upon success, the library loads the specified file and creates a stream out of it and returns a handle to it using the Bass.BASS_StreamCreateFile() call. At the same time it starts the playback with Bass.BASS_ChannelPlay().

The Bass.BASS_ChannelGetData() call acquires the spectral characteristics of the sound being played (usually a song, and more specifically for the Chines water sleeve dance performance it is called *Song* in Pinyin ("pipa" is "lute")). The fltRotationWaves1 scalar variable is constantly updated in TalkShowHostXNA via

the fast Fourier transform (FFT) [284] analysis to compute the 2048 frequency coefficients and their peaks during the playback. We use the value of that variable to update our `wave` variable that controls the fancy soap bubble wavy effect as well as its rotation with the music beat.

The channel spectra is used to draw the spectral bars at the bottom-left and upper-right mirror corner roughly equally distributed over the spectral coefficients. At the same time the two wave patterns are drawn using different colors, remembering the previous state and translating it sideways so it gradually fades away from the same FFT data combining certain frequencies per pixels and approximating the lines between the two adjacent point pixels.

Finally, the clean up BASS API includes functions `Bass.BASS_ChannelStop()`, `Bass.BASS_StreamFree()`, `Bass.BASS_Stop()`, and `Bass.BASS_Free()` to stop the playback and free resources.

5.2.3.2 Modeling

The modeling in this case is often very dynamic and is dependent on the data streams from Kinect. There are also static elements modeled in the installation. Otherwise, the modeling is very simple in many aspects making it perform robustly and responsively in real-time.

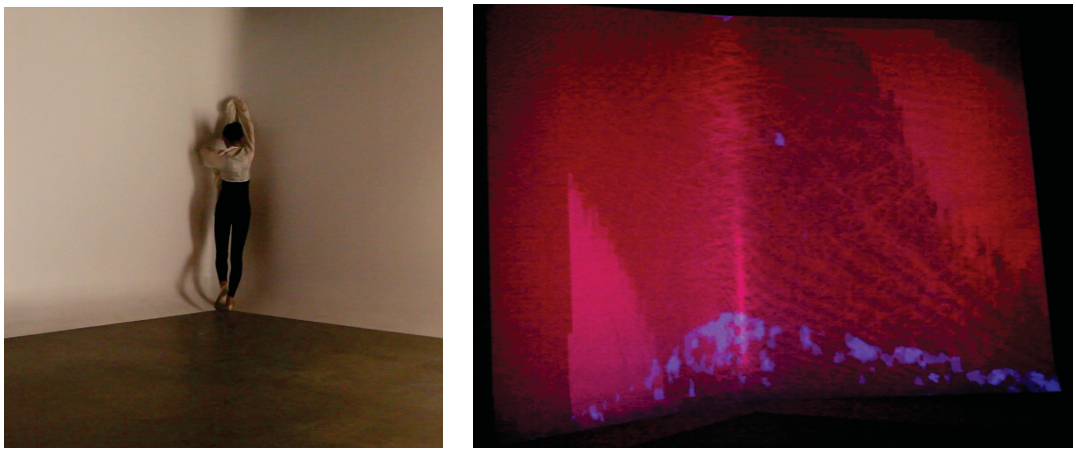
5.2.3.2.1 Environment. The virtual environment gives the audience realistic experience. This is to be done by either using video footage or rendering synthetic 3D models to build the virtual environment for the surrounding walls. We also create a virtual skybox on the ceilings and computer generated textures on the floor.

5.2.3.2.2 Media Rendering. Most of the media is rendered into a rectangular model of a polygon consisting of two triangles or a render cube consisting of 6 of such polygons. They have texture objects associated with them to which the dynamic texture data are written. The alpha transparency for certain `Color` pixels is used to

blend multiple textures rendered onto the same polygon. The soap bubble model and the polygon model are the same as in Section 4.2.2.2.2.

5.2.3.3 Projection

A similar setup as in Section 4.2.2.6 can be and is used in here in the basic installation setup and requirements. The examples of the PoC projection are in Figure 71, Figure 72, Figure 75, Figure 76, Figure 77, and Figure 78. The future augmentation of the projection possibilities are further described in Section 6.6.3.5.



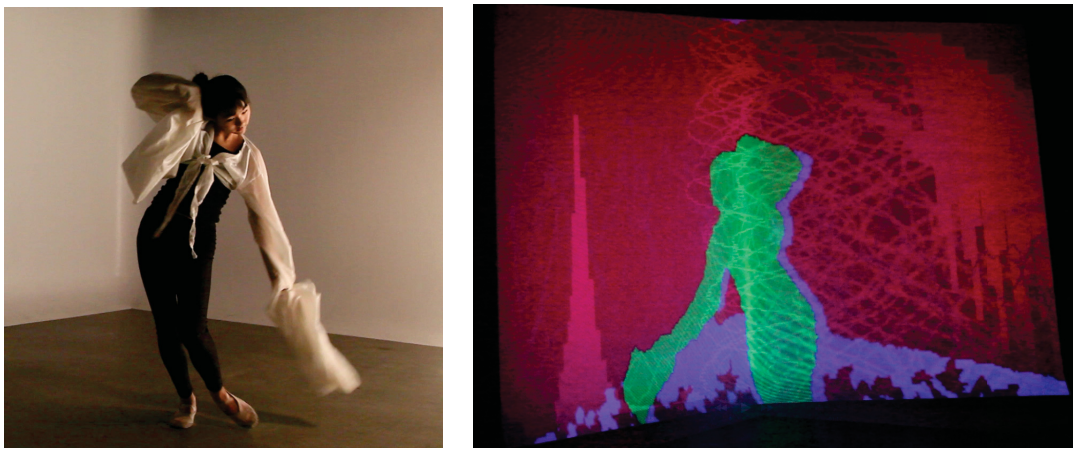
(a) The Dancer is Ready to Begin Her Water Sleeve Dance

(b) Depth View of the Same Scene; the Dancer is Not Detectable and Will Emerge Into the Scene. Music Visualization is Playing On Top Of the Depth Image.

Figure 71: Beginning of the Water Sleeve Dance by a Dancer (Real and Depth)

5.2.3.4 Lighting and Audio

Lighting is static for now (but is planned to be dynamic per Section 6.6.3.4 using audio and motions as an input). It is setup in such a way using production room spotlights as to highlight the performance area, but not directly at the performer in case of smaller spaces, but toward the edge, such that the performer and audience are not blinded and the colors captured by Kinect for the color stream are not oversaturated. The area has to be lit just enough for visibility and good quality video capture. There are almost no spotlights directed to a nearby area in the production environment so as to keep it dark enough for the projection to be the most crisp. Combinations exist with



(a) The Dancer is in the Middle of a Figure in Water Sleeve Dance

(b) Depth View of the Same Scene; the Dancer is Detectable and Rendered with Different Depth Color Layers. Music Visualization is Playing On Top Of the Depth Image.

Figure 72: A Figure in the Water Sleeve Dance Performance by a Dancer (Real and Depth)

almost no lighting as well as some lighting cast into the projection and performance areas when the two are overlapping or merged. Some of these are exemplified in the stills in this chapter. In the general case the Kinect's color stream should not be capturing projected performance except the specific case where it is desired for additional add-on effects of "infinite depth".

The sound in the installation configuration that does affect the visualization or the animation of the projected fancy bubble comes from usually an `.mp3` file read and played back via the BASS.NET library. The frequency patterns of the beat are read and fed to the animation and visualization variables. This mode of operation is primarily for a dance-like performance.

There is a known issue where the sound playback may cause problems when the speech recognition is enabled at the same time. That is when either a video playback with the sound on or a melody have speech in them resembling the dictionary of words we used primarily in the interactive documentary configuration (to call out bubbles, etc., see Appendix A.2.3) are detected, the system may accidentally begin a playback of that media content; therefore, the speech recognition is usually disabled

in this installation when the audio playback is on.

5.2.3.5 Live Performance

A live performance could be a play, a concert, or a dance show performed in front of audience. The fascination of live performance is that audience could direct interact and connect with actors, and it is the best way to engage audience with the performance and give actors emotional response. It is the thrilling magic moment co-created by actors and audience.

Live performance in this installation, has even more advanced meaning of co-creation and interaction between actors and audience.

The main disadvantages of the great recent works that involve performance and technology, e.g. by Hollogne, Tsakos, and Puma [193, 194, 285, 286] is that, for the scenarios involving technology, the actors must rehearse, practice, and synchronize their activities with the digital and cinematic pre-recorded and CG video replay, which is usually a pre-filmed or an offline animated piece that is well timed with the dialog and actions of the on-stage actors. In my opinion, this staticness unnecessarily constraints the actors and denies them extra dynamism and creativity and instead of focusing on and immersing entirely into the *Character* the role they are playing, they should also get distracted to recall at which time and place to synchronize with the new media. **The proposed work here is to remove such constraints and limitations and let the actors fully embrace their performance while the technology dynamically would adjust and respond to the actors' actions and *Diction* (speech). Thus, the actors are liberated from the necessity of synchrony with technology.**

Let's take long sleeve dance performance for example: a ghost character could be visible/hidden at different distances to the audience on the stage without the need to rehearse every time for accurate timing and positioning (see, e.g., Figure 71 and Figure 72).

In the fancy soap bubble example (Figure 75 and Figure 76), the audience could be

augmented to use their gestures to alter (perhaps just slightly) the *Plot* of story from a pre-scripted version possibly influencing the actors who may change their decision in the performance based on the audience choices and inputs (a traditional puppet show for children in Sherbrooke's cultural festival in 2012 has done something similar and in some cases the puppets asked for children's advice and acted upon it in some parts of the *Plot*).

In the virtual audience example (see Figure 79 and Figure 80), the audience may be captured to be projected onto the stage, and actors are part of the virtual world as well in real-time, unlike Hollogne's.

There is more depth and connection of the augmented performance between the real, physical world and the virtual world. Using a skeleton stream delivered in real-time, four dancers, actors could drive in real-time a team of digital puppets (avatars) instead of one-to-one performance.

Figure 73: Depth-Based Dance Performance Capture and Visualization of Long Sleeves 1

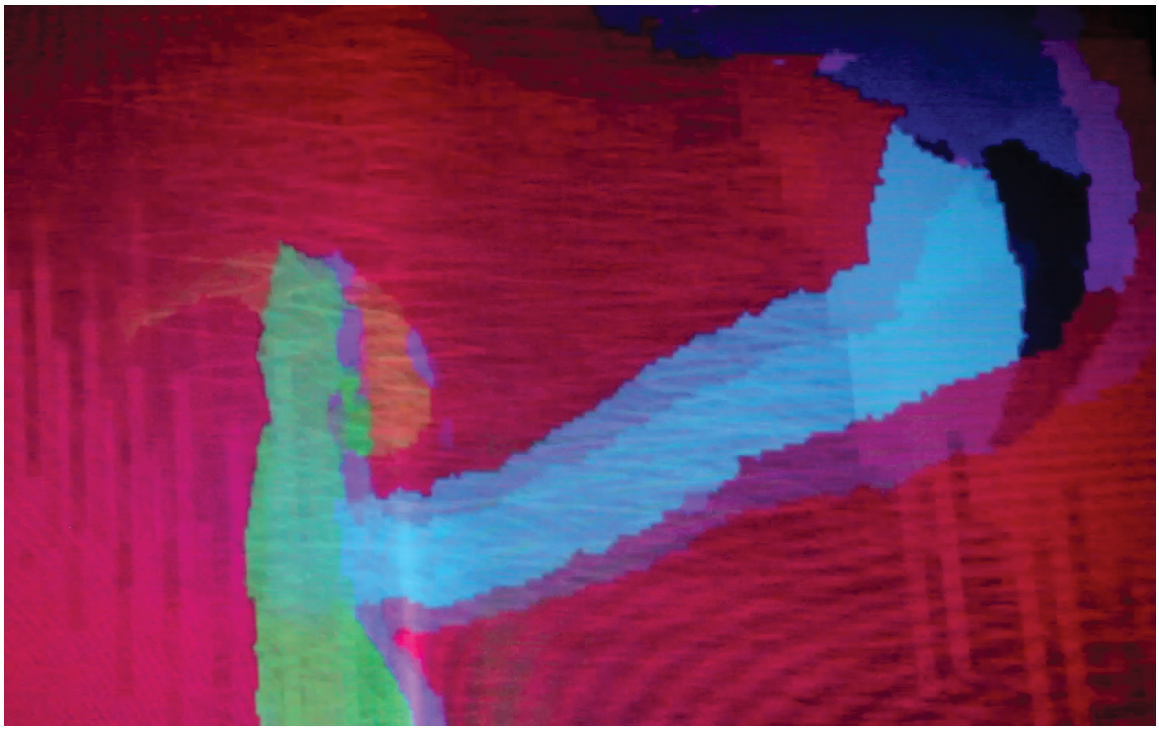
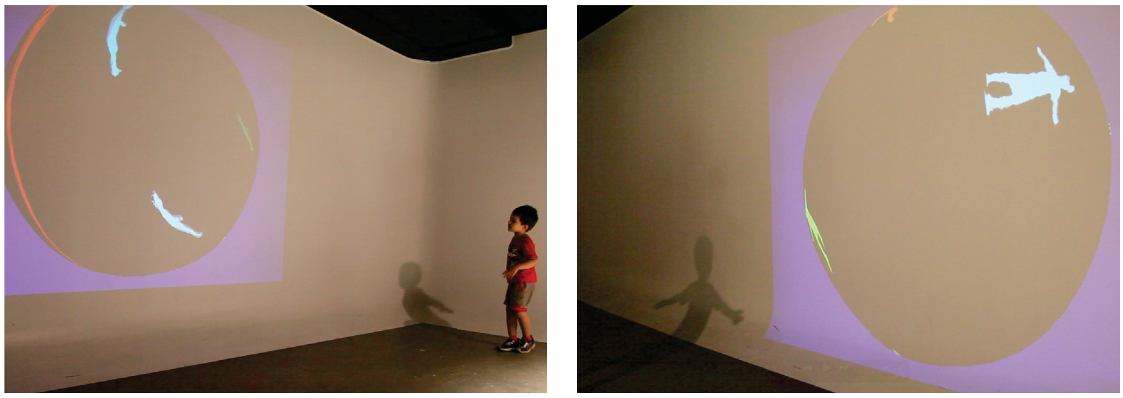


Figure 74: Depth-Based Dance Performance Capture and Visualization of Long Sleeves 2

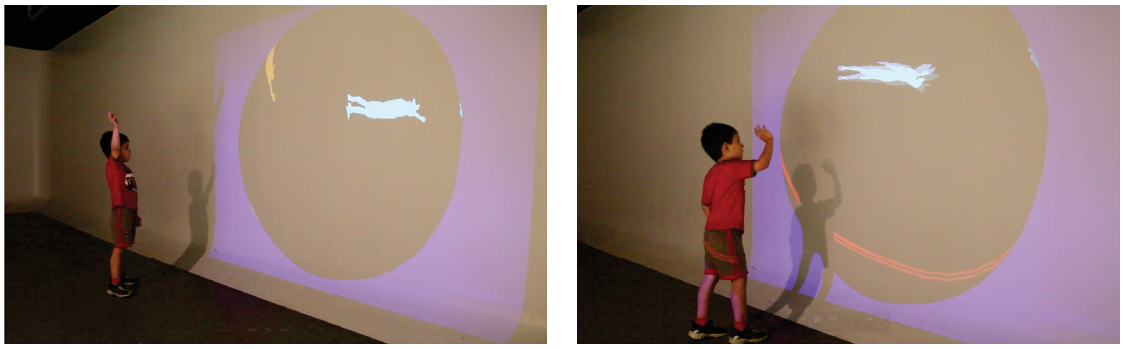


(a) (b)

Figure 75: Fancy Soap Bubble With Depth Green Screened Audience Images Mapped Onto Its Surface Animated by a Melody

5.2.3.6 Animation and Interaction

Devices, such as Kinect, camera, and microphones could capture the participants' acting data so that they could interact with the CG graphics and their animations. When there are no interruptions from people, the CG graphics, lighting, and audio

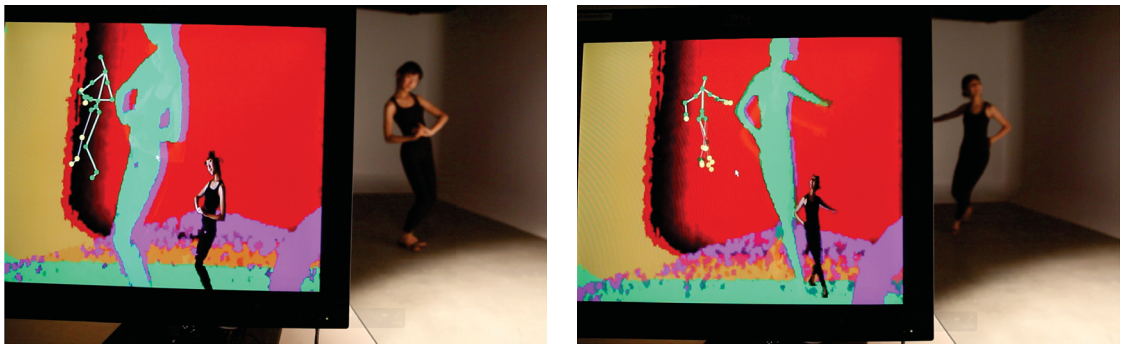


(a) (b)

Figure 76: Fancy Soap Bubble With Depth Green Screened Audience Images Mapped Onto Its Surface Animated by Hand-Waving

(a) (b)

Figure 77: Four Dancers: Real, Depth, Greenscreened, and Skeleton



(a) (b)

Figure 78: Four Dancers Again: Real, Depth, Greenscreened, and Skeleton

have their own initial default states and perform based on the default scenarios.

As mentioned earlier the interactive performance is the nexus of many things that come together all related to animation and its dynamic real-time alteration in the

(a) (b)

Figure 79: Recorded Real Event Performance with Real-Time Additional Virtual Audience

(a) (b)

Figure 80: Recorded Real Event Performance with Real-Time Additional Virtual Audience 2

computer graphics world, such as music control of a big fancy soap bubble and small bubbles, green screen and ground projection, motion-tracked skeleton, gestures with a gentle swipe of the fancy soap bubble.

In Listing 6 is the interaction sequence setup for this installation to get the desired configuration with or without a soundtrack. There it's primarily a keyboard-based interaction to set things up. After that, the details related to depth, audio, and skeleton streams come into play.

The proof of concept illustration of the presented interaction is depicted in the following clips, some of which were previously mentioned in Section 5.1.2:

- Clip 1 — fancy soap bubble "dancing" with music and audience "green-screen"

color approach to observe and swipe. It encompasses the HLSL shader deforming the soap bubble, while the textures onto it are dynamically mapped from various streams. The waving pattern is fed to the shader from the melody playback into the `wave` variable deforming the soap bubble model. It is also effected by the hand gestures gently swiping rotating the bubble (see, e.g., Figure 75).

- Clip 2 — long sleeve dance with the depth camera and music visualization on the ground floor [282]. This is becoming the classical performance piece that can be combined with all the others. The graphical imagery is affected by the data from the Kinect's depth stream and the audio stream from the melody playback via BASS. See Figure 73 and Figure 74 as the examples of the projected stills from the clip.

- Clip 3 — a recorded children dance performance video with one of them a real-time audience multiplied in the space and greenscreen-projected on to the video feed [281] (see Figure 79 and Figure 80).

---

1 **begin**
2 Toggle white/black background for white walls or blackbox with 'W';
Enable ground projection with 'G';
3 Switch to the projected mode, 'P';
4 Take off the fancy projected bubble, '3';
5 Revert to the frontal view with 'P';
6 Enable main melody playback with 'M';
7 Set back to the projection mode, 'P';
8 Adjust the zoom as necessary via the field-of-view angle λ with 'PgUp', 'PgDown';
`// The installation support is running now; time to act and perform.`
9 **end**

Algorithm 6: High-Level Interaction Algorithm for *Illimitable Space*

Chapter 6

Conclusion

In this chapter, I gather together all the thoughts on the achievements, contributions, their advantages and limitations at the time of this writing and the rich possibilities and future directions in the follow up work, academically and commercially. The chapter reviews the rationale and findings of this research (Section 6.1), multi-, inter-, and trans-disciplinary research realities (Section 6.2), consolidated contributions overview (Section 6.3), followed by the three summaries of main themes of this thesis (in Section 6.4, Section 6.5, and Section 6.6 respectively).

6.1 Rationale and Findings

The proposed research, scoped as **Computer-Assisted Interactive Documentary and Performance Arts in Illimitable Space**, covers in the reverse order arriving at:

1. theatre arts and the performance,
2. theoretical and practical experience in film/video production,
3. extensive 3D computer graphics knowledge and programming skills, and
4. experience in the new media field.

Any of the above mentioned research directions is a very wide field of research and creation. I have also discovered that the deeper research I do, the more questions I encounter (and am eager to find answers to). I really need to force myself sometimes to keep the focus on several findings and constrain myself to thinking rationally, which is often quite conflicting with my artistic creativity.

The research findings, positions, and installations argued throughout the thesis indicate the following statements are *true*:

- Virtual reality, computer generated images, and new digital media technology will not destroy the live performance of traditional theatre and the aesthetics of documentary film, instead, the technology will enhance and improve their representation.

- The boundaries among different art forms, such as film, theatre, computational arts, tend to be blurrier and blurrier because people are more interested in innovative combinations of various art forms.

In short, my position is:

Technology is essential; art is eternal.

6.2 Multi-, Inter-, and Transdisciplinary Research Realities

After many years of knowledge, practices and leadership accumulated in various projects, I feel I have at last arrived at a very productive stage on emergence of traditional art forms combined with the rapidly developing technology.

Moreover, after reading some of the literature, such as Artaud's *Theatre of Cruelty* [181], Grotowski's *Poor Theatre* [182], and Brook's *Holy Theatre* [183], I am astonished that their ideas are still quite relevant to today's modern theatre and its

performance, some of them could even continue leading the direction in today's new media and technology era.

Originally, I came to study computer science from a pure artistic background. After ten years of extremely intensive training in science and computers, at some period of time, I partially lost connection to my creative abilities as an artist. Now, I am able to regain it all back and to combine the power of both the artistic and scientific backgrounds. The whole process is such a painful experience! More than this, only considering the artistic background, there are significant differences between a theatre actor to a documentary filmmaker and a director, and to a media artist; let alone scientists.

The unique experience and this project are a worthy undertaking given a good amount of time and life to analyze and document, with a group of academic and artistic professionals and experts to supervise and contribute. Hopefully, the thesis itself will be a good contribution to the new students and researchers who would proceed a multidisciplinary or interdisciplinary research.

6.3 Contributions Overview

Here is a consolidated short summary of the work accomplished as a part of or alongside the research and creation, in terms of publications of the related work, design and implementation, exhibitions, installations, performances, and other showings.

While working on the research and analysis detailed in this thesis, as well as the corresponding creation, design and implementation we produced or are in the process of realization of the following works:

- Miao Song and Peter Grogono. Real-time modeling and physical-based animation of a jellyfish from softbody in OpenGL. In Bipin C. Desai, Sudhir P. Mudur, and Emil I. Vassev, editors, *Proceedings of the Fifth International C* Conference on Computer Science and Software Engineering (C3S2E'12)*, pages 123–124, New York, NY, USA, June 2012. ACM. Poster

- Miao Song, Peter Grogono, Jason Lewis, and Maureen J. Simmonds. A poor woman's interactive remake of the "I Still Remember" documentary with OpenGL. In Junia Anacleto, Sidney Fels, Nicholas Graham, Bill Kapralos, Magy Seif El-Nasr, and Kevin Stanley, editors, *Proceedings of ICEC 2011*, number 6972 in LNCS, pages 362–366. Springer, October 2011

- Michael Fortin, Miao Song, David Gauthier, Sha Xin Wei, and Peter Grogono. Jellyfish simulation. Poster at the 2nd Grand Conference, May 2011

- Miao Song. Tangible Memories: Multi-dimensional interactive documentary presentation, April 2011. [Concordia University Humanities Doctoral Student Annual Conference]; `http://dislocationsconference.wordpress.com/schedule/`

- Miao Song (Director). *I Still Remember*. 1st BJIFF "See the world through films" Contest for Best Documentary Short, 2011. [Documentary film]; 13 minutes; Best Documentary Award. `http://www.bjiff.com/en/bjiffnews/n214618230.shtml`

- Miao Song and Peter Rist. Water ink animation film – a career of animation, complex for water-ink. Concordia University, 2011. An Essay Translation from book: "Between Looking up and Stooping: the Memory of Three Generation Female Chinese Film Photographers"

- Miao Song. Interdisciplinary research presentation related to research-creation media arts. Concordia University Hexagram, December 2010

- Miao Song, Maureen J. Simmonds, and Peter Grogono. Innovative medical applications and beyond of 3D techniques in a responsive virtual reality lab: Experience report. Unpublished, December 2010

- Miao Song and Peter Grogono. Deriving software engineering requirements specification for computer graphics simulation systems through a case study. In *Proceedings of the 3rd International Conference on Information Sciences and Interaction Sciences (ICIS2010)*, pages 285–291. IEEE Computer Society, June 2010

- Miao Song (Director). *I Still Remember*. Showcased in HTMlles 2010, 2010. [Documentary film]; 9 minutes; `http://www.htmlles.net/2010/projects/emergence/`

- Miao Song. *Interactive Elastic Two-Layer Soft Body Simulation with OpenGL*. Lambert Academic Publishing, June 2010. ISBN: 978-3-8383-4137-8

- Miao Song, Peter Grogono, and Maureen J. Simmonds. Towards innovative application of computer graphics techniques in responsive virtual and augmented realities. Poster at the 1s ECSGA Colloquium, April 2010

- Miao Song and Peter Grogono. Soft body simulation: Physical based animation with OpenGL. Poster at the 1s ECSGA Colloquium, April 2010

- Song Wang, Miao Song, Zhang Ling, and Maureen J. Simmonds. Pain and performance in virtual reality environments: A pilot feasibility study. Poster at the 3rd Pain, Mind and Movement Symposium, August 2010

- Serguei A. Mokhov, Miao Song, and Ching Y. Suen. Writer identification using inexpensive signal processing techniques. In Tarek Sobh and Khaled Elleithy, editors, *Innovations in Computing Sciences and Software Engineering; Proceedings of CISSE'09*, pages 437–441. Springer, December 2009. ISBN: 978-90-481-9111-6, online at: `http://arxiv.org/abs/0912.5502`

- Miao Song and Peter Grogono. Application of advanced rendering and animation techniques for 3D games to softbody modeling and animation. In *Proceedings of C3S2E'09*, pages 89–100, Montreal, Quebec, Canada, May 2009. ACM

- Miao Song and Peter Grogono. Are haptics-enabled interactive and tangible cinema, documentaries, 3D games, and specialist training applications our future? In *Proceedings of GRAPP'09*, pages 393–398. INSTICC, February 2009

- Serguei A. Mokhov and Miao Song. OpenGL project presentation slides interface and a case study. In *Proceedings of GRAPP'09*, pages 409–412, Lisboa, Portugal, February 2009. INSTICC

- Miao Song, Serguei A. Mokhov, Alison R. Loader, and Maureen J. Simmonds. A stereoscopic OpenGL-based interactive plug-in framework for Maya and beyond. In *Proceedings of VRCAI'09*, pages 363–368, New York, NY, USA, 2009. ACM

- Miao Song, Serguei A. Mokhov, and Peter Grogono. Designing an interactive OpenGL slide-based presentation of the softbody simulation system for teaching and learning of computer graphics techniques. In *Proceedings of C3S2E'09*, pages 131–136, New York, NY, USA, May 2009. ACM

- Miao Song, Serguei A. Mokhov, and Peter Grogono. Teaching physical based animation via OpenGL slides. In Tarek Sobh and Khaled Elleithy, editors, *Innovations in Computing Sciences and Software Engineering; Proceedings of CISSE'09*, pages 483–488. Springer, December 2009. ISBN: 978-90-481-9111-6

- Miao Song. The role of computer graphics in documentary film production. [online], April 2009. `http://arxiv.org/abs/1101.0663`

- Miao Song. The role of computer graphics in documentary film. Presentation at FSAC (Film Studies Association of Canada) 2009, Carleton University, Ottawa, May 2009. `http://www.filmstudies.ca/FSAC_conference2009.pdf`

- Miao Song. Feynman algorithm implementation for comparison with Euler in a uniform elastic two-layer 2D and 3D object dynamic deformation framework in OpenGL with GUI. [online], 2009. `http://arxiv.org/abs/0906.3074`

- Miao Song. ABRA. CLSP Research Concordia Prom Video, 2009

- Alison R. Loader, Serguei A. Mokhov, and Miao Song. Open Stereoscopic 3D Plugin Collection. SourceForge.net, 2008–2012. `http://sf.net/projects/stereo3d`, last viewed November 2010

- Miao Song and Peter Grogono. An LOD control interface for an OpenGL-based softbody simulation framework. In Tarek Sobh, editor, *Innovations and Advances in Computer Sciences and Engineering, Proceedings of CISSE'08*, pages 539–543. Springer Netherlands, December 2008. Published in 2010

- Serguei A. Mokhov and Miao Song. An OpenGL-based interface to 3D PowerPoint-like presentations of OpenGL projects. In *Advanced Techniques in Computing Sciences and Software Engineering, Proceedings of CISSE'08*, pages 533–538. Springer Netherlands, December 2008. Published in 2010

- Miao Song and Peter Grogono. A framework for dynamic deformation of uniform elastic two-layer 2D and 3D objects in OpenGL. In *Proceedings of C3S2E'08*, pages 145–158. ACM, May 2008

- Miao Song. Unraveling Her Story: Miao. Theatre Production, 2007. Directed by Emily Burkes-Nossiter, Co-directed by Chia-Wen, Scriptwriting and acting by Lois Jones, Lucy Lu, Joy Ruben, Dorothy Singer, Miao Song, Talia Weisz, and Mimi Zhou (in [294])

- Miao Song. Dynamic deformation of uniform elastic two-layer objects. Master's thesis, Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada, August 2007. ISBN: 978-0-4943-4780-5, `http://arxiv.org/abs/0907.4364`

- Miao Song and Yolanda Ye. Interactive underwater sea world simulation in OpenGL. Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada, 2003. A computer graphics project

- Miao Song. Spectacle. Animation Film, 2003. 1 minute 15 seconds

- Miao Song. Tangram. Interactive Flash Games, 2003

6.4 Softbody, Jellyfish, and CG Summary

Through the study of the particular case of the Softbody Simulation System and its evolution, we explicitly set a collection of requirements for similar interactive computer graphics simulation systems. We hope that this set of requirements can be used and expanded on by researchers in the field and will mature to be a reference standard for similar systems that will be created and revised accordingly [23].

We have encountered some integration difficulties due to the frameworks' original design and implementation considerations discussed [23, 99].

Physical based softbody simulation is still a very hot topic in computer graphics [59], especially, with realtime animated feature, such as jellyfish, to my knowledge

based on own recent survey, there have not been an algorithm or an application published yet. Moreover, we integrate a number of technologies for the interactive user-controlled, or, rather, user-assisted control of a swimming jellyfish [256].

We further discuss the limitations of the proposed approach as well as the future (and ongoing) work on this and the related projects.

6.4.1 Softbody, Jellyfish, and CG Contributions

To a various degree of completeness in the implementation, the contributions include the following:

- We have modeled a third layer in the softbody objects consisting of a single center particle connected by the radius springs to the inner layer particles for for 2D and 3D softbody objects. The particle serves conveniently as the single attachment point, and, is, therefore, easy to use instead of attaching multiple points of the second layer. It also adds stability to the softbody objects at the smaller time steps in Euler and Feynman algorithms making them more usable. This setup provided the means of attachment of the softbody objects on "hardbody" ones or animate along the curve [22].

- The Softbody Simulation System LOD aspect got more prominent and visible with the GLUI interface, especially for the LOD hierarchy [255]. The simulation interface allows adjustment of the LOD parameters, including the whole algorithms at run-time as well as co-existence of all dimensional objects in one simulation scene [255].

  To enable the LOD parameters and the interface to them we extended the original Softbody Simulation Framework with the notion of state as well as greatly enhanced the interactivity of the simulation system by exposing the state to the GLUI interface at run-time. The users and researchers working in the field with the framework can easily observe the effects of a vast variety of LOD parameters on the physical-based softbody simulation visualization at

real-time without the need of altering the source code and recompiling prior each simulation increasing the usability of the system.

We identified a scale problem with some of the LOD parameters when mapping to a GUI: for example, spreading the mass over each particle or layer of particles is unmanageable and has to be specified by other means when needed [255].

- Currently we have a complete 2D jellyfish, to which we add haptic interaction support and for 3D a single-layer Euler-integrated 7 slices of 12 particles animated and interacted with in real-time for the jellyfish's bell without tentacles [11].

- GPU-based shading—we implemented the first draft version of a general vendor-independent API for shader use within the softbody system and provided two implementations of that API: one that loads GLSL vertex and fragment shaders and the other that loads the cross-vendor assembly language for shaders [22].

- We implemented the Feynman algorithm alongside the existing Euler, midpoint, and RK4, to further validate the integration framework design and to compare [22].

- We implemented a rudimentary Bezier-curve animation as a separate standalone application and we produced the softbody system as a library to be used later in this application and render the softbody objects along the curve to demonstrate the ability to use the softbody framework outside its own main simulation application [22].

- We abstracted the default penalty-based collision (see Section 6.4.3.5) detection implementation to allow for replaceable other collision detection and response algorithms integrated in the future for comparative studies [22].

- We completed the first proof-of-concept integration of the Softbody Simulation System and OGLSF frameworks. We made a number of slides in a OpenGL-based softbody presentation typically found in lab/tutorial like presentations,

which are to be extended to a full lecture-type set of slides [99]. Power-point slides in OpenGL is a way to present for demo and teaching how the softbody objects are modeled and rendered [99].

- We have presented the initial iteration of the interface for real-time stereoscopic modeling within Maya as a MEL and C++ plug-in, with the Maya-independent OpenGL core that can be used in other 3D modeling, animation, game engine, and medical VR tools [7]. The stereoscopic OpenGL-based rendering plug-in [8] originally for Maya, and then beyond, is another advanced rendering technique this work was designed to include in order to render the softbody and jellyfish simulations in stereo in the future.

6.4.2 Limitations

There are some assumptions and limitations to the realization described here.

- This work assumes the CG topics presented are renderable at real-time, like the Softbody Simulation System [99]. While it is of course possible to render the animation images offline first (possibly of photorealistic quality), and then replay them back, or use keyframe animation, but this is not what the challenge is at. However, combining real-time simulation and offline-made imagery can be used to enhance the overall perception of the scene and, e.g., to playback video footage as dynamic texturing of polygons, alongside the real-time animation that transforms the geometry, and this is what we do in our installation since one can playback AVI (standard and HD) movies in OpenGL.

- May be hardware dependent for real-time processing, though today's commodity hardware should generally be good enough.

- We also do not have Linux support for the moment, only Mac OS X and Windows platforms [7].

6.4.3 Future Directions

For the most part the future work will focus on the addressing and resolving some of the limitations, unfinished items or experiments mentioned in the earlier sections. Some specific items are mentioned further in point form and details.

- – Our goal for the 2D jellyfish is to render it automatically with an artistic appeal by texture mapping it accordingly
  – Integrate with a real-time game and MoCap systems (see Section 6.4.3.6)
  – Harness the GPU power to speed up the simulation and rendering processes
  – Public haptics [296] installation with a projector screen [11]

- Allow softbody modeling and alteration as the Archimedean-based graphs and different types of them than an octahedron [264], not just the number of subdivision iterations as a run-time LOD parameter [255].

- Provide animation state tracing, saving, and replay with the intermediate values for analysis [255]. Specifically, allow the state dump and reload functionality in order to display each particle and spring state (all the force contributions, velocity, and the position) at any given point in time in a text or XML file for further import into a relational database or an Excel spreadsheet for plotting and number analysis (perhaps by external tools). Reloading of the state would enable to reproduce a simulation from some point in time, a kind of a replay, if some interesting properties or problems are found. This is useful for debugging as well [255].

- Continue with improvements to usability and functionality of the user interface for scientific experiments [255].

- Port the source code fully to Linux and Mac OS X. Currently it only compiles properly under Windows 7 under Visual Studio 2010 and basic softbody and curve-based animation also in Mac OS X.

- Release our code and documentation as open-source implementation (see Section 6.4.3.3) either a part of the Concordia University Graphics Library [258] and/or as part of a Maya [9] plug-in and as a CGEMS [266] teaching modules.

- Allow advanced interactive UI controls of the scenes and slides by using haptics devices [19] with the force feedback, head-mounted displays and healthcare virtual reality systems [7] (see Section 6.4.3.6).

- Showcase various softbody shading techniques and shaders via the OpenGL slides. We already implemented the first draft version of a vendor-independent API for shader use within the softbody system and provided two implementations of that API—one that loads GLSL vertex and fragment shaders and the other that loads the cross-vendor assembly language for shaders.

- Demonstrate attachment of softbody objects to skeletons for character animation (see Section 6.4.3.4).

- The application of softbody deformation in computer graphics has significant value for medical research and media arts production. The resulting models from the simulated models, will help to create responsive/feedback environments that will map the virtual feedback to physical through an haptic interface.

- Complete/make experiments with the game and rendering engines [219, 254] (see Section 6.4.3.1 and Section 6.4.3.2).

- Investigate the replacement of vertex normal computation as well as other vertex properties in relation to animation and physics on the GPU to speed up performance on multiple iterations.

- In the longer term future, attempt to map the graphical simulated softbody jellyfish to electrical pulses to control physical synthetic jellyfish like the one made by Nawroth *et al.* [98, 297, 298].

6.4.3.1 Fly3D Game Engine

Supplying a game character and other attributes with the the softbody objects was one of the targets to test the softbody object properties with the textbook's [74] Fly3D game engine [219].

6.4.3.2 OGRE3D

The Object-oriented Graphics Rendering Engine, or simply OGRE3D [254], is another popular open source engine that was considered for experiments of integration and testing of the softbody objects in. We did not reach the ability to experiment with this engine yet, but its possibilities look very promising [22]. In particular it overlays its API over either OpenGL or Direct X, so an application has a chance to be deployed and run on Microsoft XBOX, for example.

6.4.3.3 Open Source Release

We plan to refine our set of requirements and design decision guidelines for interactive computer graphics physical-based simulation systems, such as the Softbody Simulation System, further during our ongoing work. Specifically, our experiments and derived metrics will be based on the system's integration with the CUGL [258], the OpenGL presentation slides framework [290] as well as the stereoscopic effects of another framework that generalizes the handling of stereoscopy [7]. Furthermore, the haptic devices' sensory input as well as integration of the softbody library `libsoftbody`, which contains the complete implementation of the current simulation framework into games and other more realistic environments, such as virtual reality systems and head-mounted displays for further studies in various domains ranging from cinema production to medical research in pain management. We plan to release the different builds and iterations mentioned above as a part of the open-source initiative incrementally.

6.4.3.4 Character Animation

The functionality development of elastic simulation modeling for 3D software design and implementation has emerged as a new challenge in computer graphics. One of the existing software with the elastic modeling functionality is Maya [9], which provides shape deformation, especially facial animation, for a group of objects. It is more convenient than traditional frame animation. However, the elastic object movement is not attached to skeleton animation. Furthermore, this elastic simulation is not done in real time. A possible future work that can be done based on the elastic simulation is to define a skeleton system and to map the body mesh onto it. The different parts of the body can be defined as with different degrees of deformability based on the elasticity. For example, the mesh is less elastic on the arms, legs; the mesh is more elastic on the fatty or softer areas, like belly, breasts, etc. The weight of the elastic property of the muscles can be mapped and dynamically set according to the skeleton's joints. The system can then be integrated into the advanced animation software as a plug-in. The skeleton could a model or a motion-captured from the Kinect SDK [203] and its C++ API.

6.4.3.5 Collision Detection

CD between soft objects is a complex phenomenon. In our current system, we are using the penalty methods [67], which do not generate the contact surface between the interacting objects. This method uses the amount of inter-penetration for computing a force which pushes the objects apart instead. Even though the result is fair enough based on estimation, in reality, the contact surfaces should be generated rather than local inter-penetrations. Especially, if we want to use computer animation to imitate organ surgery and help surgeon practice as if interact with real objects, the penalty method is no longer appropriate. There must be a more accurate algorithm to define the collision between rigid body and soft body, or soft body and soft body. Our software should be able to describe other soft body deformation, such as fractures, possibly benefiting from freshly released Vega [59]. An approximation of the softbody

object collisions can be done by forming a few temporary invisible springs between the nearest particles of the would be colliding objects with the distance below certain threshold and let the spring forces do work while the distance is below the threshold.

On the other, artistic side, advanced CD is important for all kinds of artistic animation. Thus, the future work would be to "import" the tricks from traditional animation into computer animation:

- Give characters a pseudo personality
- Stretch and squeeze is used to highlight dynamic action such as deceleration due to collisions
- Shape distortion (the collision can be thought of as having two phases: compression and restitution)
  - In the compression phase, kinetic energy of motion changes into deformation energy in the solids
  - If the collision is perfectly inelastic ($e = 0$), then all of the energy is lost and there will be no relative motion along the collision normal after the collision
  - If the collision is perfectly elastic ($e = 1$), then all of the deformation energy will be turned back into kinetic energy in the restitution phase and the velocity along the normal will be the opposite of what it was before the collision

While we have not managed to implement new collision detection algorithms to compare with the existing one, the existing algorithm was abstracted to create a more flexible subframework to support multiple collision detectors, i.e., one can switch between the implementations of collision detection at run-time or start time. The framework provides a general `CollisionDetector` class and its concrete implementation for the penalty method. Any new algorithm would have to subclass `CollisionDetector` in order to participate in the framework [22].

6.4.3.6 Virtual Reality

The possible future work on the VR side would consist of expanding the system and installation onto the advanced VR equipment and various integration works to make the whole experience more immersive and tangible and further turn it into a part of the interactive documentary project. Some equipment I worked with was quite expensive and I hope to be able to regain access to such equipment to produce a great installation [12].

The major way of integration is via OpenGL and the related software technologies based on OGRE3D and Vizard3D with C++ and Python interfaces to the major CG parts of the CAREN/Motek/Sensics system and the HMD display respectively. Thus, it would be nice to see the HMD VR Vizard3D [253] integration with softbody to completion. In addition to that, I would like to complete the integration of the Falcon device in the same VR context to enhance the feedback response to the audience. Finally, relying on the previous and ongoing achievements, I would like to provide a stereoscopic softbody interface to the system employing the open-source stereoscopic plug-in [12].

If I am lucky to have access to the equipment and will have completed the integration, I am considering creating or adapting an immersive 3D VR documentary installation based on this [12].

6.4.3.7 SDL

We plan to explore the use of the Simple Direct media Library (SDL) [261] that provides an open cross-platform media API that works uniformly with OpenGL and many interaction peripherals and devices. It is a lot more comprehensive than GLUT, and allows for better video playback.

6.4.3.8 Softbody Jellyfish with Kinect

Finally, it is very natural, interesting, and logical to produce an interactive installation based on the recent technological advancements to enable a human being skeleton to

drive a 3D computer generated character with the possibilities to proceed to the game and cinema scenes to enrich them with the interactive features. An initial PoC with our jellyfish, Kinect, and fluid simulation was partially achieved in a collaboration with Fortin [257, 256] where the jellyfish was swimming partially in motion-blur fluid with limited Kinect connectivity [256].

Another version of own production is in the works with more features and interaction scenarios and integration with the works such as interactive documentary and illimitable space installations discussed in the earlier chapters.

We describe several details of modeling and implementation of an interactive work of the earlier jellyfish control HSC system using OpenGL, the Softbody and the Fluid Simulation System, Kinect, and Jitter [256].

In this PoC prototype, we integrated a number of technologies for the interactive user-assisted control of a swimming jellyfish. In the foundation of this interactive installation are a number of hardware and software platforms. The hardware included the same Kinect platform from Microsoft as in Chapter 4 and Chapter 5 and the Jitter/Max/MSP setup. The real-time fluid flow and the described physical based softbody simulation frameworks were designed and developed in-house [256]. The PoC featured real-time physical based simulation of 3 or 4 jellyfish swimming in a fluid-simulated environment. Jellyfish followed the fluid waves and could exert a small force back to fluid through its inner compression. The audience not only could disturb the animation of fluid by their motion blobs captured by Kinect device, but also could somewhat drive a jellyfish. The unattended jellyfish swim about randomly [256].

The future work planned for jellyfish is to make it into an advanced real-time interactive game akin to *Blush* [95].

6.5 Interactive Documentary Summary

We summarize our findings, achievements in Section 6.5.1, as well as the advantages (Section 6.5.2) and limitations (Section 6.5.3) of our approach and offer future directions (Section 6.5.4).

6.5.1 Interactive Documentary Contributions

When I described the interactive documentary system I have implemented, I almost forgot the contents of those documentary films featured there. From the international winning short film *I Still Remember* [299], which I have made almost entirely by myself, including shooting, lighting, editing, subtitling, with my daughter's involvement to be my main character who truly shared her feelings and memories with the audience. I think it might not be a coincidence that the film got several awards and got screened at several film festivals, but the artistic creativity and skills accumulated from my previous studies in arts and working experience in TV/film productions have helped here. I am pleased that my artistic talents come back to me finally and I have been successfully qualified by peers as an artist again after many years of training in computer science and software engineering. The art piece has a social value, which draws on the society's attendance and increases the public awareness and garners sympathetic response to some children who have similar experience as my daughter, the main character. There are also more memory bubbles related to this original story, such as the TV interview to the filmmaker about this film, film festival screening scenes, the continuing life of the main character's life after the film was made, the impact to the character and filmmaker after the film gain lots of successful experience (see Section 4.2.2.4). Making a linear documentary film is a life-long personal project.

Now, look at the newly invented concept of interactive documentary design and system. The tangible memory bubbles concept came from the content of the original documentary itself of a little girl describing her own memories as floating bubbles she could "pick" and "see inside" if she wanted to remember something in particular.

There is originality and a real need derived from a static documentary film by its character's description to her memories instead of trying to make some fancy ideas to show off the interactive media computer technologies.

At this moment, interactive documentary or cinema mainly is referred to web-based works instead of installations [24]. Moreover, the interaction is limited to mouse drag-click. The footage of cinematic materials is static, pre-filmed materials instead of being able to dynamically displayed upon interaction or even record in place. My interactive documentary system fills the insufficiently covered gaps between the static media. At first, I presented the initial "poor woman" way of approaching an interactive documentary with a widely available OpenGL library and a set of online openly available resources to play clips, texts, and images from a passive storyline documentary into making it interactive. That working interactive prototype featured five bubbles with three videos and two images (as illustrated in sample screenshots in Figure 57 and in Figure 58). Subsequently, in the XNA/Kinect version, the audience could use their body movement, gesture, and speech to interact with the system. Meanwhile, the system changes the method of existing interactive documentary, by making audience live participation also a part of the documentary project itself, audience become part of the interactive documentary project. The interactive documentary non-linear story telling system is an eternal project which will never become a PAST tense.

6.5.2 Advantages

We outline the advantages of both OpenGL and XNA installations first.

6.5.2.1 OpenGL Installation

Advantages of the OpenGL version include the fact that OpenGL itself is an open standard and runs on many OS platforms, desktop or mobile [213]. This, and the author's familiarity with it, initially prompted the development of the first OpenGL *Tangible Memories* version. Additional examples [207, 208] were readily available of

haptic-based interaction in OpenGL that were planned for this installation.

6.5.2.2 XNA Installation

The advantages of the XNA based installation include the easier development effort required as opposed to the OpenGL- or Direct X-level to come up with runnable prototypes and as well as the recent influx of many open- and shared source examples for it; including Kinect examples [203] and speech processing libraries. The author's growing familiarity with the subject has also been influenced by preparing for and instructing laboratory sections for the related computer graphics and animation courses.

6.5.3 Limitations

It should be noted that the interactive documentary approach described here is more suited for home-use by a family or a few individuals rather than large audiences in traditional cinema theatres as the interactivity aspect in movies does not scale very well nor does it make much sense to have many people to interact with it unless to create chaos (and document it). The home and small audience aspect, the technologies studied and developed in this work should be affordable by regular home users, small education groups in kindergartens, schools, colleges, universities, and other educational institutions. Current limitations that plague interactivity in the documentaries, specifically a profound difficulty to for multiple people to interact with the same documentary piece instance at the same time is problematic (e.g., one would need to support multi-input in a form of multiple mice, keyboards, haptic devices, cameras or motion tracking sensors from more than one individual from the same audience in the same space-time.) We further summarize platform-specific limitation with some hints for possible solutions.

6.5.3.1 OpenGL Installation

OpenGL-based programming is generally more tedious as it is lower-level than some other libraries or APIs like XNA and it therefore takes longer to prototype a piece. However, open-source rendering engines, such as OGRE3D [254] could be of help here.

At this point we limit the interaction by the audience to use only mouse clicks or preset keys in a GLUT window to bring a bubble of choice or to move a synthetic camera in the environment with a keyboard. Speech, haptic, and Kinect based interaction did not make the cut at this time of this thesis but the work on those has already started. Moreover, the bubbles float in 3D space, but the basic mouse interaction is 2D and clicks have to be translated to the bubbles nearest in the z dimension, which is confusing sometimes when x and y are near for two or more bubbles, and the user wants one bubble, but gets another one instead to view.

In OpenGL, we used only the AVI format playback (however more are offered via SDL [261], Jitter/Max/MSP [300, 301], or PureData [302], but in this thesis they are not explored). Kinect with OpenKinect [202], or Jitter or PureData can also be done as the latter have patches for its support [303].

Despite the limitations mentioned, the author plans to continue exploring the OpenGL version with other technologies as time and resources permit.

6.5.3.2 XNA Installation

The main disadvantage of the XNA-based solution is that it is by default Microsoft-only for the use primarily with Microsoft devices. However, XNA could also at least partially run on Mono [304] and MonoXNA [305] (which uses OpenGL calls to render things in Linux and Mac OS X for example) in C#.

Additionally, XNA video playback is only for `.wmv` files in a specific constrained resolution settings, where the video has to be converted first before the XNA based content resources can compile them into the internal `.xnb` representation like they do for all the media content before it can be used by the application.

6.5.4 Future Directions

Some of the future directions have to do with expanding and exhibiting the installations world-wide, not only as installations or desktop applications, but also as web and mobile applications. In the meantime, the other aspects would address some of the limitations presented earlier (Section 6.5.3).

One aspect is the interview recording for the documentary can be done on the spot and the content produced stored in one of the memory bubbles while the installation is progressing at the same time (with the care taken to temporarily stifle the speech recognition pipeline to avoid the interview's spoken words interference). Such submissions or users' interactions with the installation documenting it can be the part of the ever growing footage available at the installation and be different at each location where the installation is deployed, including web.

6.5.4.1 OpenGL Installation

A more advanced bubble modeling is under consideration in utilization of the softbody objects (see Chapter 3) instead of plain spheres to add realism and tangibility. Bubbles can either modeled as softbody objects, or, e.g., as buckyballs acting as animated containers [306, 307] with proper spring stiffness.

The amount of bubbles and their content is determined presently via a preset set of media in the code constants, but is planned to be more dynamic.

For the video (clip-based) bubbles on zoom-in the sound is to be activated. The sound may also optionally be activated on textual and photo bubbles if it is present in a form of a narration.

Finally, more advanced interactivity is planned with haptics and camera-marker motion capture, a Wii or C++-based Kinect API implementation code with the techniques already discussed.

6.5.4.2 XNA Installation

An audience could query the contents of the bubbles by some kinds of keywords, or create some brand new bubbles with realtime footage, such as dynamic recording of the additional footage of the users interacting with the bubbles and retaining it in the bubbles beyond the eight currently active; perhaps in fancy bubbles. The audience will even be able to interact with the scene and its objects in order to push forward the story scenarios via gestures.

6.5.4.3 Rendering and Projection

An option is planned for the projection to be stereoscopic. Another projection option is the 180-degree screen VR system that we are considering like CAREN as explained in the next section.

6.5.4.4 VR and Documentary Production

Applying new media with VR equipment into documentary production definitely will be a unique new model of representation in Cinema in the new digital era. Again, typically, except for some rare occasions, documentary producers and computer scientists and/or digital artists who work in computer graphics are relatively far apart in their domains and rarely intercommunicate to have a joint production; yet it happens, and perhaps more so in the present and the future. There were attempts to fuse interaction, drama, narration, and computers in the past such as the works [175, 172, 174, 25, 170], but not on such a scale as a complete VR system [12].

6.5.4.5 Film Production and Studies

Today, CG and animation are more advanced than traditional animation according to the interactive, physical based (mimicking live-action camera), provide more effective visual results, and impose less burden on the financial/budget concerns [15, 100].

In the past, the autobiographic documentary film type was often constrained to

for filmmakers themselves only as the ones possessing the knowledge, resources, and equipment to do so. With the wider accessibility of digital content and high computing power these days, autobiographic documentaries are now accessible to virtually any household [15, 100].

Furthermore, the notion of script writing in terms of screenplay writing—i.e., the writing for the film screen or television screen for the new genre of documentary films is an emerging, new research area requiring more work [100, 15].

Moreover, teaching in the research and application of "CGI for documentary films" is also an important topic that is going to be inevitably introduced into the core of Film Studies for scholarly research and film production industry [100, 15].

Finally, the "rich woman"'s augmented and expanded interactive documentary installation [15] immersive and tangible experience will include the following items:

- Research on and address the limitations mentioned earlier
- Feature haptic connectivity
- Softbody/bucky bubbles
- Black box and VR projection installation
- Stereoscopy
- Database-driven dynamic footage acquisition, selection, and display

6.6 Illimitable Space Summary

We merge virtual and real performance augmenting the real performance with CG imagery driven in real-time by the actions of the performers. The CG images can be augmented in real-time with extra elements, clothing, extremities, etc., and driven by the real actors in the real-time and interact with the virtual beings. There are also a number of larger future directions to extend this work expressed further in Section 6.6.3. We begin, however, with the description of the contributions in this

direction in Section 6.6.1 and the correlation of the work we presented earlier in Chapter 5 to the theatre elements in Section 6.6.2.

6.6.1 Illimitable Space Contributions

Just as new media was shown to build on old media and include interaction aspects into TV, cinema, and documentaries, a somewhat similar approach is happening to theatre and performance arts. The process begins with a historical record of Grotowski's experiments in "poor theatre", both theory and practice [182]. It is followed, among other works, by Giannachi in her book [190] where she reviews interfacing between digital arts and theatre production with virtual reality. And the most recent book by Salter about the influence of the technology on the artistic performance, including historical overview of the performance experimentation in theatre among several other fields, including computational and responsive environments [192].

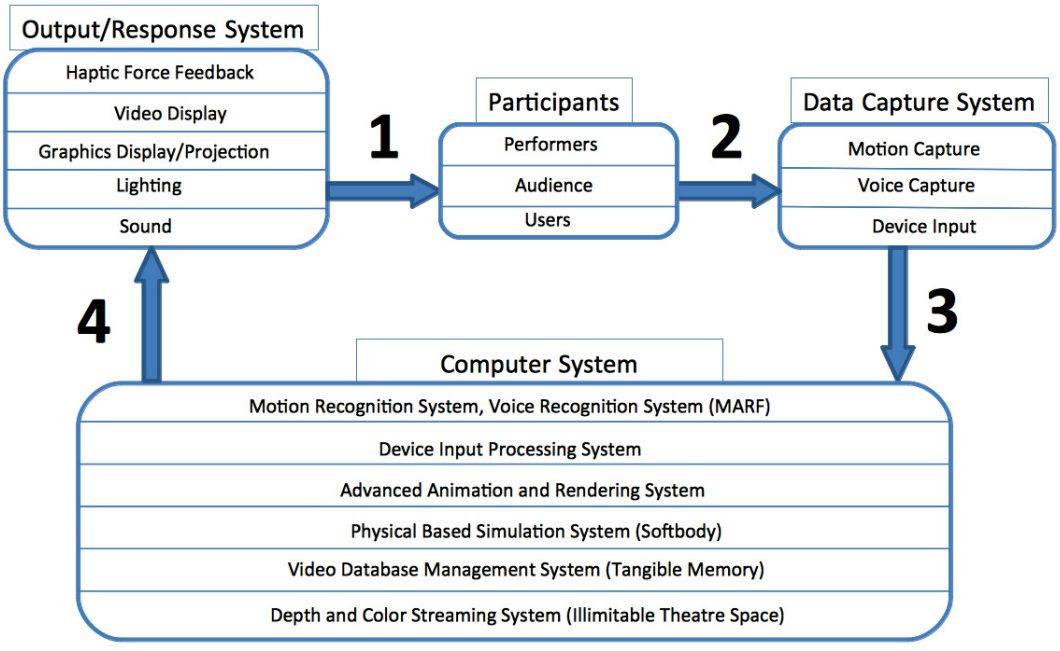
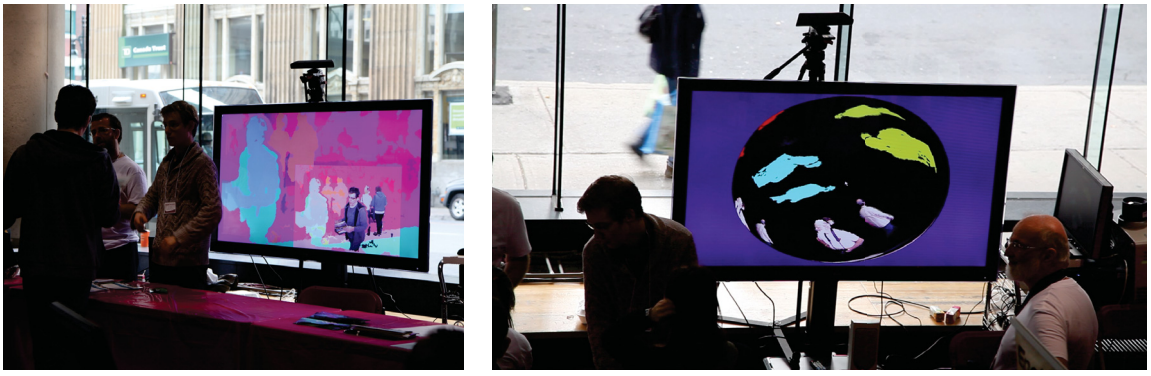


Figure 81: Conceptual Design of a Generalized Interactive Documentary and Performance Arts Installation

My installation turns the "Poor Theatre" to an extremely "Rich Theatre", in which one would create more dimensions than the traditional theatre form allows. It will allow the audience to experience the freedom of the illimitable performance environment surrounding them, rather than remain nearly fixed in their seats and stay at the passive observing level. Moreover, the performance of actors also would be effectively interactive to computer generated virtual environment and improvised by the communication with the audience.

Additionally, this contribution reveals that the overall design can be generalized as shown in Figure 81 to include and encompass all subsystems presented in this work so far, including the interactive documentary and jellyfish softbody simulation, and beyond. As a result, as a more massive and general aspect of the interactive concept design and technology have been transposed onto theatre production advanced MoCap, tangible media, and audience participation in realtime computer graphics effects.

This PoC installation has been featured in Concordia's Open House event in October 2012 (see Figure 82) and Exposcience 2012 at Stewart Hall, West Island, Montreal.



(a) (b)

Figure 82: Exhibition of the Installation and Concordia Open House, 2012, CSE

6.6.2 Theatre Elements

In this section we describe our coverage of the theatre elements in this interactive installation work. We decided to apply Aristotle's drama elements [189] to our contemporary theatre performance installation, such as *Plot*, *Character*, and *Spectacle* because they have retained their importance through time, are still the most widely used evaluative tools and general rules for artistic theatre performances.

6.6.2.1 *Plot.*

In the six essential theatre elements, *Plot* is arguably the most important one [189, 279]. If a performance does not carry a story, we would not call it a theatre performance. Therefore, a great story with dramatic storytelling approach requires artistic creative talents. Here, we won't give a concrete design of the story lines, but the storytelling approach. All plots have a beginning, a middle, and an end. In conventional theatre, actors could memorize their lines and have some flexibility in improvisation while they perform. However, in our illimitable theatre performance system, the story will not be told in a pre-planned linear way. It requires more acting skills and talents from acting for an unplanned expedient and event interrupted by audience.

One possible augmented example design could be that in the Cinderella's story, whether or not Cinderella tries on the glass shoe when the messenger knocks on her door. When she is hesitating via an interior monologue, the audience could apply gestures, such as wave or clapping to respond to her. The suspension here is that the audience could help Cinderella to make a decision, either she would succeed and so be with the prince forever, or she is doomed forever as her step-family slave or the story somehow deviates and resolves differently or arrives to the same or similar ending via a different path.

6.6.2.2 *Character.*

This element exists not only in a play, but also in TV, movies, and even video games. Characters, who are agents of the *Plot*, provide the motivations for the events of the

Plot. However, the traditional theatre art has more limitations to portray characters than movies and games, such as some fictionalized and deified figures with humanly impossible bodily or bodiless properties. In traditional Chinese *Nuo* opera, there are always ghosts and gods. The performer plays a valiant god who dispels ghosts and devils. The whole process how the god transfers into a human being's body, is through the actor's body movement, facial expression, voice change, and different gestures. With our illimitable system, we could project the computer generated ghost/god onto a media in front of actor, or directed onto the actor, to show the mystery play.

6.6.2.3 *Spectacle*.

Everything that could be seen or heard on stage (such as actors, sets, costumes, lights and sound) is a *Spectacle*. In Japanese theatre, on the *Noh* stage, the only ornamentation is the *kagami-ita*, a painting of a pine tree at the back of the stage to represent a distant view or prospect. It represents the tree through which *noh* was passed down from heaven to mankind. However, it is quite limited and boring in scenery expression. Our illimitable system could expand the time and space on stage, which may improve the *Noh*'s actors' performance. When a *Noh* actor walks far from *kagami-ita*, the background CG scenery could even move based on his motion, speed and gesture as if the space has been moving along with the actor from far to close. Many new movie technologies, such as large scale projection with landscape and ocean floor, also could be experimented with here.

6.6.3 Future Directions

In the future plans we make a theatre production design, a combination of all proposed components and elements below.

A logical extension and augmentation of the interactive documentary *Tangible Memories* would be the use of the illimitable space concept applied to it and the advanced interaction specifically from Chapter 5. By being both the viewer and artist, in the *Tangible Memories* project case, for the dancer, the dance-based interaction

with the memory bubbles is to playfully scatter them around, meanwhile being the viewer to observe. To the audience, she is an artist, and an actress.

The subsequent step is to give the illimitable space even more dimensions by bringing it into the realm of the web-based installation, using, e.g., Unity (e.g., 14 locks—a *Bart Bonte* game), HTML5, and Web 3 technologies combined with the database storage and retrieval and real-time media streaming along with the webcams on the participant's computers greatly scaling up the audience, similarly to the web-based interactive documentaries [24].

6.6.3.1 Augmenting Conceptual Design

At first, the audience and actors coexist in the closed stage space indicated (see Figure 83). It is not really a stage, but rather a space divided into different zones. The blue ring area is the four season zones: Spring, Summer, Autumn, and Winter. When participants step into a zone, the projected graphics on the surrounding environment (walls) will be transformed. The virtual seasons will be changed upon the participants' positions in the physical space. The central circle area is where participants could explore the dynamic lighting and sound.

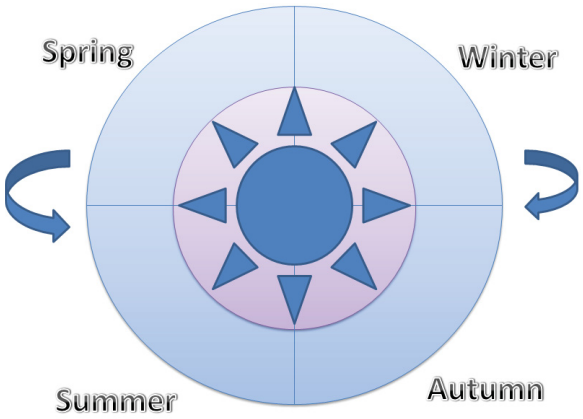


Figure 83: Theatre Production Performance Space Design

The theatre performance will take place in a blackbox with a maze to replace a standard stage. There are different shapes of surfaces, such as hanging curtains, altered floor, fog (used as projection screens for multimedia resources). Actors perform in three different positions within the maze: dance, sing, and monologue. Audience

walk freely in the environment and their joining to the actors' performance, in gesture, voice, or texts could directly contribute to the whole performance.

6.6.3.1.1 Particles. Particles are a very efficient model for computer rendering, especially with animation. Same groups of particles could be reused for Spring rain, Summer fireworks, Autumn leaves, and Winter snowflakes in the virtual environment, accomplished with the corresponding physical based algorithms. As mentioned in Section 5.2.3.6 in the absence of the interaction input from humans, the animation, lighting, and audio revert to their default "stable" states and scripted scenarios, e.g., such as snowflakes falling down to the ground if there is no wind.

6.6.3.2 Remaining Theatre Elements

For completeness, as a part of the future work, we need to finalize the review of the remaining three Aristotle's drama elements [189]: *Diction*, *Melody*, and *Thought*. The work on these has started already.

6.6.3.3 Tools

6.6.3.3.1 Max/MSP/Jitter/PureData. Delivering the installation and its implementation of interaction to the masses and artists can be achieved via Max/MSP and Jitter [300, 301] or their open-source alternative of PureData that create artistic media patches in their corresponding data flow languages. There is an *external* (plug-in) available for Max/MSP in Jitter to connect to Kinect—`jit.freenect.grab` [303]; as well as the Novint Falcon haptics sensor was used with PureData [302] [308] and OpenGL. Both can be used to augment the interaction and be available via Jitter or PureData data-flow *patches* (programs) for much easier perusal by the artists at large for more than the Microsoft's proprietary platforms allowing us to re-enhance back again the OpenGL *Tangible Memories* (see Section 6.5.3.1) as well as the C# version can be used with Mono and MonoXNA [304] (see Section 6.5.3.2).

6.6.3.3.2 Maya. Maya [9, 309, 310, 311] is a great tool to create virtual models of the 3D spaces and environments and do their animation that can be the dynamic virtual 3D props in any designed interactive theatrical projected installation (e.g. the maze). Additionally, starting from a specific version Maya began supporting stereo buffers for keyframed animation, which can be very helpful for creation of the environment and space similar to that of *Spectacle* [276], but in stereoscopic 3D. Maya achieves that by providing implementation for graphics cards that have hardware stereo buffers (as well as the plug-in we developed prior to that support for the cards that do not have the built-in stereo capability [8, 7]) for greater perception of the virtual and augmented reality in the performative space.

6.6.3.4 Augmenting Lighting and Audio

To enhance the perception and participatory experience further, I will use a few dynamic stage spot lights and some off-stage lights (in my production, they will be positioned within props based on the story scenarios) to create an artistic conceptual environment. As proposed earlier, the lighting and audio should be programmed to respond in real-time, dynamically stimulated by actors' or audience's actions instead of statically configured each time.

The sound visualization and response are also planned to be changed to be more representative of the mood, season, and tune of the audio, voice, tune, and the like, especially during the transition moments and dynamic music selection and playback made accordingly.

6.6.3.5 Augmenting Projection

The installation is planned to be projected in a blackbox or an enclosed large room alike. The space could be in any shape, which then will be divided into four season zones with the actors movement influencing the digital projections by their actions surrounding the audience seated in the center of such a theatre with motion-enabled chairs and all-dimensional 360° projections including ceiling and the floor projector or

glass screens and even onto fog. The projection requires multiple projectors with split video signals. A more advanced technique I intend to use is stereoscopic rendering and projection (see Section 6.6.3.3.2).

6.6.3.5.1 Holographic Displays. It is important to explore the newly emerged holographic display technology, which is currently very expensive, but with time the costs should come down and it will likely be an excellent performance tool among other things; in the mean time I will seek various sources of funding.

6.6.3.6 Nezha.

I rediscovered a Chinese animation film, *Nezha Conquers the Dragon King* (1979), which was made in 1970s by the Shanghai Animation Studio. It was incredibly beautiful and has accompanied me in my childhood in China. This fabulous artwork piece attained international reputation in the past in various festivals and reviews.

The *Plot* is based on a very ancient Chinese traditional myth story, which has only been, and could have only been, produced in animation film because it is impossible for traditional theatre to represent the mythology scenery of the story, such as the *Character* of Nezha (a boy was born during the Shang Dynasty, always depicted as an incarnation of brave, protagonist, and superior power) has three heads and six arms and has the ability to spit fire. There are many Chinese opera elements this film has borrowed, such as costumes, characters' gestures and movements, music background, and stage speech.

However, such a theatre performance could only be made with the *Rich Theatre* by using digital technology and the new media to create some special performance effects, such as an under-sea environment, heaven, fire, etc., and also to include the interaction between a 3D dragon and a Nezha-playing actor with the recent technological innovations such as Kinect, CG, projection, and others previously described.

Bibliography

[1] Miao Song (Director). *I Still Remember*. 1st BJIFF "See the world through films" Contest for Best Documentary Short, 2011. [Documentary film]; 13 minutes; Best Documentary Award. `http://www.bjiff.com/en/bjiffnews/n214618230.shtml`.

[2] Alison Reiko Loader. Making space. Master's thesis, Department of Design and Computation Arts, Concordia University, Montreal Canada, 2008.

[3] Novint Technologies, Inc. Novint Falcon. [online], 2011. `http://www.novint.com/index.php/products/novintfalcon`.

[4] B. C. Choi and A. W. Pak. Multidisciplinarity, interdisciplinarity and transdisciplinarity in health research, services, education and policy. [online], 2006. `http://www.ncbi.nlm.nih.gov/pubmed/17330451`, last viewed June 2010.

[5] Miao Song. Dynamic deformation of uniform elastic two-layer objects. Master's thesis, Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada, August 2007. ISBN: 978-0-4943-4780-5, `http://arxiv.org/abs/0907.4364`.

[6] Jason Lewis and OBX Labs. CitySpeak. `http://cspeak.net/`, 2008–2012.

[7] Miao Song, Serguei A. Mokhov, Alison R. Loader, and Maureen J. Simmonds. A stereoscopic OpenGL-based interactive plug-in framework for Maya and beyond. In *Proceedings of VRCAI'09*, pages 363–368, New York, NY, USA, 2009. ACM.

[8] Alison R. Loader, Serguei A. Mokhov, and Miao Song. Open Stereoscopic 3D Plugin Collection. SourceForge.net, 2008–2012. `http://sf.net/projects/stereo3d`, last viewed November 2010.

[9] Autodesk. Maya. [digital], 2008–2012. autodesk.com.

[10] Song Wang, Miao Song, Zhang Ling, and Maureen J. Simmonds. Pain and performance in virtual reality environments: A pilot feasibility study. Poster at the 3rd Pain, Mind and Movement Symposium, August 2010.

[11] Miao Song and Peter Grogono. Real-time modeling and physical-based animation of a jellyfish from softbody in OpenGL. In Bipin C. Desai, Sudhir P. Mudur, and Emil I. Vassev, editors, *Proceedings of the Fifth International C* Conference on Computer Science and Software Engineering (C3S2E'12)*, pages 123–124, New York, NY, USA, June 2012. ACM. Poster.

[12] Miao Song, Maureen J. Simmonds, and Peter Grogono. Innovative medical applications and beyond of 3D techniques in a responsive virtual reality lab: Experience report. Unpublished, December 2010.

[13] Miao Song. *Interactive Elastic Two-Layer Soft Body Simulation with OpenGL*. Lambert Academic Publishing, June 2010. ISBN: 978-3-8383-4137-8.

[14] Miao Song and Peter Grogono. A framework for dynamic deformation of uniform elastic two-layer 2D and 3D objects in OpenGL. In *Proceedings of C3S2E'08*, pages 145–158. ACM, May 2008.

[15] Miao Song, Peter Grogono, Jason Lewis, and Maureen J. Simmonds. A poor woman's interactive remake of the "I Still Remember" documentary with OpenGL. In Junia Anacleto, Sidney Fels, Nicholas Graham, Bill Kapralos, Magy Seif El-Nasr, and Kevin Stanley, editors, *Proceedings of ICEC 2011*, number 6972 in LNCS, pages 362–366. Springer, October 2011.

[16] Miao Song (Director). *I Still Remember*. Showcased in HTMlles 2010, 2010. [Documentary film]; 9 minutes; http://www.htmlles.net/2010/projects/emergence/.

[17] Miao Song. Interdisciplinary research presentation related to research-creation media arts. Concordia University Hexagram, December 2010.

[18] Miao Song and Peter Rist. Water ink animation film – a career of animation, complex for water-ink. Concordia University, 2011. An Essay Translation from book: "Between Looking up and Stooping: the Memory of Three Generation Female Chinese Film Photographers".

[19] Miao Song and Peter Grogono. Are haptics-enabled interactive and tangible cinema, documentaries, 3D games, and specialist training applications our future? In *Proceedings of GRAPP'09*, pages 393–398. INSTICC, February 2009.

[20] Miao Song. The role of computer graphics in documentary film. Presentation at FSAC (Film Studies Association of Canada) 2009, Carleton University, Ottawa, May 2009. `http://www.filmstudies.ca/FSAC_conference2009.pdf`.

[21] Miao Song. Unraveling Her Story: Miao. Theatre Production, 2007. Directed by Emily Burkes-Nossiter, Co-directed by Chia-Wen, Scriptwriting and acting by Lois Jones, Lucy Lu, Joy Ruben, Dorothy Singer, Miao Song, Talia Weisz, and Mimi Zhou.

[22] Miao Song and Peter Grogono. Application of advanced rendering and animation techniques for 3D games to softbody modeling and animation. In *Proceedings of C3S2E'09*, pages 89–100, Montreal, Quebec, Canada, May 2009. ACM.

[23] Miao Song and Peter Grogono. Deriving software engineering requirements specification for computer graphics simulation systems through a case study. In *Proceedings of the 3rd International Conference on Information Sciences and Interaction Sciences (ICIS2010)*, pages 285–291. IEEE Computer Society, June 2010.

[24] Wikipedia. Web documentary — Wikipedia, The Free Encyclopedia, 2012. [Online; accessed 3-August-2012].

[25] Marian F. Ursu, Vilmos Zsombori, John Wyver, Lucie Conrad, Ian Kegel, and Doug Williams. Interactive documentaries: A golden age. *Comput. Entertain.*, 7:41:1–41:29, September 2009.

[26] Florian Thalhofer. The Korsakow system: Database-driven interactive Korsakow films production. [online], 2008–2011. `http://korsakow.org`.

[27] Katerina Cizek (Director). HIGHRISE: Out of my window. [online], National Film Board of Canada, 2008–2010. A 360-degree interactive documentary, http://highrise.nfb.ca.

[28] Jimmy Wales, Larry Sanger, and other authors from all over the world. Wikipedia: The free encyclopedia. [online], Wikimedia Foundation, Inc., 2001–2012. http://wikipedia.org.

[29] Wayne Carlson. A critical history of computer graphics and animation. [online; accessed 2012], 2003. http://design.osu.edu/carlson/history/lessons.html.

[30] John A. Lent, editor. *Animation in Asia and the Pacific*. Indiana University Press, July 2001. ISBN: 978-025-334-035-1.

[31] Donald Hearn, M. Pauline Baker, and Warren Carithers. *Computer Graphics with OpenGL*. Prentice Hall, 4 edition, November 2010. ISBN: 978-0136053583.

[32] Tomas Akenine-Möller and Eric Haines. *Real-Time Rendering*. A.K. Peters Ltd., 2 edition, 2002. ISBN 1568811829, http://realtimerendering.com/index_rtr2.html.

[33] Natalya Tatarchuk, editor. *Advanced real-time rendering in 3D graphics and games*, 2006. http://developer.amd.com/media/gpu_assets/Course_26_SIGGRAPH_2006.pdf.

[34] Tan Tiow Seng. CS5243 3D game programming technology class notes. School of Computing, National University of Singapore, 2007. http://www.comp.nus.edu.sg/~cs5243/.

[35] Sudhir P. Mudur. COMP7661 advanced rendering and animation class notes. Concordia University, September 2007.

[36] Randi J. Rost. *OpenGL Shading Language*. Pearson Education, Inc., February 2004. ISBN: 0-321-19789-5.

[37] Frank Losasso. Surface reflection models. NVIDIA Corporation, 2004.

[38] Wikipedia. Toy Story — Wikipedia, The Free Encyclopedia. [Online; accessed 15-August-2008], http://en.wikipedia.org/w/index.php?title=Toy_Story&oldid=232017683, 2008. [motion picture].

[39] Paul Lincoln. Citizens comfort. [online]. http://www.mediartchina.org/recomb/citizen.

[40] Xu Zhongmin. Cloud. [online]. http://www.dgd.stu.edu.tw/DGD3/index-1-tst001.html.

[41] Fu ZhiYong. A swimming fish. [online]. http://newmediabeijing.org/md2004/exhibition_display.php?section=2&link_id=1&title=The+Swimming+Fish&artist_id=.

[42] E-GO Computer Graphics. Tsinghua university 100 anniversary celebration building projection, May 2011. http://www.e-go-cg.com/index.html.

[43] SIGGRAPH. *SIGGRAPH ASIA '10: ACM SIGGRAPH Asia 2010 papers*, New York, NY, USA, 2010. ACM.

[44] SIGGRAPH. *SA '10: ACM SIGGRAPH ASIA 2010 Computer Animation Festival*, New York, NY, USA, 2010. ACM.

[45] SIGGRAPH. *SA '10: ACM SIGGRAPH ASIA 2010 Art + Tech Showcase*, New York, NY, USA, 2010. ACM.

[46] SIGGRAPH. *SIGGRAPH ASIA '09: ACM SIGGRAPH ASIA 2009 Art Gallery & Emerging Technologies: Adaptation*, New York, NY, USA, 2009. ACM.

[47] SIGGRAPH. *SIGGRAPH ASIA '09: ACM SIGGRAPH ASIA 2009 Computer Animation Festival*, New York, NY, USA, 2009. ACM.

[48] SIGGRAPH. *SIGGRAPH Asia '08: ACM SIGGRAPH ASIA 2008 computer animation festival*, New York, NY, USA, 2008. ACM.

[49] SIGGRAPH. *SIGGRAPH Asia '08: ACM SIGGRAPH ASIA 2008 artgallery: emerging technologies*, New York, NY, USA, 2008. ACM.

[50] John C. Hart, editor. *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers*, New York, NY, USA, 2008. SIGGRAPH, ACM.

[51] Isaac Victor Kerlow. *The Art of 3-D Computer Animation and Imaging*. John Wiley & Sons, Inc., New York, NY, USA, 2 edition, 2000. ISBN: 0-471-36004-X.

[52] Lee Montgomery. *Tradigital Maya: A CG Animator's Guide to Applying the Classical Principles of Animation*. Focal Press, 1 edition, 2011. ISBN 0123852226, 9780123852229.

[53] Wikipedia. Procedural modeling — Wikipedia, The Free Encyclopedia. [online; accessed 27-August-2011], 2011. `http://en.wikipedia.org/w/index.php?title=Procedural_modeling&oldid=432153189`.

[54] Manfred Lau and James J. Kuffner. Behavior planning for character animation. *Eurographics ACM SIGGRAPH Symposium on Computer Animation*, 2005.

[55] William T. Reeves. Particle systems – a technique for modeling a class of fuzzy objects. *Computer Graphics*, 17:359–376, 1983.

[56] Paul Bourke. *Particle System Example*. The University of Wester Australia, 1998. `http://local.wasp.uwa.edu.au/~pbourke/modelling_rendering/particle/index.html`.

[57] Andrew Witkin. Particle system dynamics. In *SIGGRAPH97 Course Notes*. ACM, 1997. Online at `http://www.cs.cmu.edu/afs/cs/user/baraff/www/sigcourse/notesc.pdf`.

[58] Jernej Barbič, Marco da Silva, and Jovan Popović. Deformable object animation using reduced optimal control. *ACM Trans. Graph.*, 28:1–9, July 2009.

[59] Jernej Barbič. Vega – 3D deformable object library. [online], August 2012. `http://run.usc.edu/vega/`, viewed August 2012.

[60] Hugh Hancock. Machinima cutscene creation, part one. *Gamasutra*, September 2000. `http://www.gamasutra.com/view/feature/131537/machinima_cutscene_creation_part_.php`.

[61] Phylis Johnson and Donald Pettit. *Machinima: The Art and Practice of Virtual Filmmaking*. McFarland Press, 2011.

[62] Jenna Ng, editor. *Understanding Machinima: Essays in Film-making in Virtual Worlds*. Continuum Press, New York, NY, 2012.

[63] Wikipedia. Quake (video game) — wikipedia, the free encyclopedia. [Online; accessed 14-September-2012], 2012.

[64] Eike Falk Anderson. Off-line evolution of behaviour for autonomous agents in real-time computer games. In H.-P. Schwefel, J.-J. Merelo Guervós, P. Adamidis, H.-G. Beyer, and J.-L. Fernández-Villacañas, editors, *Parallel Problem Solving from Nature (PPSN VII)*, number 2439 in Lecture Notes in Computer Science (LNCS), page 689 ff. Springer-Verlag, 2002. `http://link.springer.de/link/service/series/0558/papers/2439/243900689.pdf`.

[65] Michael Nikonov and iπ Soft. iPi Desktop Motion Capture. [online], 2008–2012. `http://ipisoft.com`.

[66] Wikipedia. Soft body dynamics — Wikipedia, The Free Encyclopedia. [online; accessed 10-September-2012], 2012. `http://en.wikipedia.org/wiki/Soft_body_dynamics`.

[67] M. Moore and J. Wilhelms. Collision detection and response for computer animation. *Computer Graphics*, 22(4):289–298, August 1988.

[68] Ming Chieh Lin. *Efficient Collision Detection for Animation and Robotics*. PhD thesis, Department of Electrical Engineering and Computer Science, University of California at Berkeley, 1993. Online at: `ftp://ftp.cs.unc.edu/pub/users/manocha/PAPERS/COLLISION/thesis.ps.Z`.

[69] Chris Hecker. Physics, part 3: Collision response. *Game Developer Magazine*, pages 11–18, March 1997.

[70] Matthias Muller, Julie Dorsey, Leonard McMillan, and Robert Jagnow. Collisions and deformations: Stable real-time deformations. In *Proceedings of the 2002 ACM*

SIGGRAPH/Eurographics symposium on Computer Animation, pages 49–54. ACM, July 2002.

[71] Steve Capell, Seth Green, Brain Curless, and Zoran Popvic. Collisions and deformations: A multi resolution framework for dynamic deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer Animation*, pages 41–48. ACM, July 2002.

[72] Wikipedia. Rendering (Computer Graphics). Wikipedia, the free encyclopedia, 2012. `http://en.wikipedia.org/wiki/Rendering_(computer_graphics)`.

[73] Alan Watt and Fabio Policarpo. *3D Games: Real-time Rendering and Software Technology*, volume I of *ACM Press SIGGRAPH series*. Addison-Wesley, 1 edition, 2001. Edited by Stephen Spencer.

[74] Alan Watt and Fabio Policarpo. *3D Games Animation and Advanced Real-time Rendering*. Addison-Wesley, 2 edition, 2003. ISBN 0-201-78706-7.

[75] Jonathan Gibbs. *Rendering Skin and Hair*. ACM, March 2001. `http://silicon-valley.siggraph.org/MeetingNotes/shrek/hairskin.pdf`.

[76] Barbara Robinson. The first CG cast of human actors stars in *Final Fantasy: The Spirits Within*. *ComputerGraphics World*, August 2001. `http://cgw.pennnet.com/Articles/Article_Display.cfm?Section=Archives&\Subsection=Display&ARTICLE_ID=108473`.

[77] Kayvon Fatahalian and Tim Foley. Rendering jellyfish. [online], June 2004. `http://graphics.stanford.edu/courses/cs348b-competition/cs348b-04/jellies/index.html`, viewed January 2011.

[78] Patrick Coleman and Karan Singh. Ryan: rendering your animation nonlinearly projected. In *NPAR '04: Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, pages 129–156, New York, NY, USA, 2004. ACM.

[79] Janne Kontkanen. *Novel Illumination Algorithms For Off-Line And Real-Time Rendering*. PhD thesis, Helsinki University of Technology, 2007. http://lib.tkk.fi/Diss/2007/isbn9789512286102/isbn9789512286102.pdf.

[80] Tomas Moller and Eric Haines. *Real-Time Rendering*. A. K. Peters Limited, 2 edition, 2002. ISBN 1568811829.

[81] Michael Repplinger, Alexander Löffler, Dmitri Rubinstein, and Philipp Slusallek. URay: A flexible framework for distributed rendering and display. Technical Report TR-2008-01, Computer Graphics Group, Saarland University, December 2008. http://graphics.cg.uni-saarland.de/fileadmin/cguds/papers/2008/repplinger_2008/tr_2008-1_uray.pdfs.

[82] Wikipedia. Global illumination. Wikipedia, the free encyclopedia, 2008. http://en.wikipedia.org/wiki/Global_illumination.

[83] Wikipedia. Level of detail — Wikipedia, The Free Encyclopedia. [Online; accessed 29-October-2012], 2012. http://en.wikipedia.org/w/index.php?title=Level_of_detail&oldid=495068316.

[84] Carl M. Erikson. *Hierarchical Levels of Detail to Accelerate the Rendering of Large Static and Dynamic Polygonal Environments*. PhD thesis, University of North Carolina at Chapel Hill, 2000. https://wwwx.cs.unc.edu/~geom/papers/documents/dissertations/erikson00.pdf.

[85] Maryann Simmons and Dave Shreiner. Per-pixel smooth shader level of detail. [online], 2003. http://i31www.ira.uka.de/~semin05/LevelOfDetail/Material/p1-simmons.pdf.

[86] Daniel Rákos. GPU based dynamic geometry LOD. [online], October 2010. http://rastergrid.com/blog/2010/10/gpu-based-dynamic-geometry-lod/.

[87] Willem H. de Boer. Fast terrain rendering using geometrical mipmapping. [online], October 2000. http://www.flipcode.com/archives/article_geomipmaps.pdf.

[88] Kurt Akeley, Allen Akin, Ben Ashbaugh, Bob Beretta, John Carmack, Matt Craighead, Ken Dyke, Steve Glanville, Michael Gold, Evan Hart, Mark Kilgard, Bill Licea-Kane, Barthold Lichtenbelt, Erik Lindholm, Benj Lipchak, Bill Mark, James McCombe, Jeremy Morris, Brian Paul, Bimal Poddar, Thomas Roell, Jeremy Sandmel, Jon Paul Schelter, Geoff Stahl, John Stauffer, and Nick Triantos. ARB_vertex_program, Revision 46. [online], 2002–2007. NVIDIA Corporation, ARB Extension #26, http://www.opengl.org/registry/specs/ARB/vertex_program.txt.

[89] Bob Beretta, Pat Brown, Matt Craighead, Cass Everitt, Evan Hart, Jon Leech, Bill Licea-Kane, Bimal Poddar, Jeremy Sandmel, Jon Paul Schelter, Avinash Seetharamaiah, Nick Triantos, and contributors to the ARB_vertex_program working group. ARB_fragment_program, Revision 27. [online], 2002–2006. Microsoft Corporation, ARB Extension #27, http://www.opengl.org/registry/specs/ARB/fragment_program.txt.

[90] OpenGL. GLSL Quick Reference Guide. [online], 2008. http://www.opengl.org/sdk/libs/OpenSceneGraph/glsl_quickref.pdf.

[91] Alison R. Loader (Director). Folding. Concordia University and the National Film Board of Canada, 2008. [Motion picture].

[92] Michael Karp. Stereoscopic matchmove/layout for "journey to the center of the Earth" - 3D, using 3D equalizer and Maya. [online], December 2007. Retrieved January 2, 2008, from http://members.aol.com/mckarp/JCEstereoMatchmove.htm.

[93] Yu-Chung Chen, Dmitri Svistula, and Ratko Jagodic. Dancing jellyfish. [online], 2006. http://www.evl.uic.edu/rjagodic/animation/project3.htm, viewed January 2011.

[94] Shane Zamora and Xin Mao. Underwater scene. [online], 2010. http://charhut.info/cs280/, viewed January 2011.

[95] Flashbang Studios, LLC. Blush: a Unity-based 3D jelly fish game. [online], 2010. http://blurst.com/blush/, viewed January 2011.

[96] Blender Foundation. Blender. [online], 2008–2012. http://www.blender.org.

[97] William MacDonald Megill. *The Biomechanics of Jellyfish Swimming*. PhD thesis, The University of British Columbia, June 2002. http://cerf.bc.ca/megill/jfish/thesis/thesis.pdf.

[98] Janna C. Nawroth, Hyungsuk Lee, Adam W. Feinberg, Crystal M. Ripplinger, Megan L. McCain, Anna Grosberg, John O. Dabiri, and Kevin Kit Parker. A tissue-engineered jellyfish with biomimetic propulsion. *Nature Biotechnology*, 30:792–797, July 2012.

[99] Miao Song, Serguei A. Mokhov, and Peter Grogono. Teaching physical based animation via OpenGL slides. In Tarek Sobh and Khaled Elleithy, editors, *Innovations in Computing Sciences and Software Engineering; Proceedings of CISSE'09*, pages 483–488. Springer, December 2009. ISBN: 978-90-481-9111-6.

[100] Miao Song. The role of computer graphics in documentary film production. [online], April 2009. http://arxiv.org/abs/1101.0663.

[101] Mike Noel. *I, Robot*. http://www.vnoel.com/Movie-Reviews/iRobot.html, 2004.

[102] Martin Rieser and Andrea Zapp, editors. *New Screen Media: Cinema/Art/Narrative*. British Film Institute, London, 2002. ISBN: 085-170-864-1.

[103] Lev Manovich. *The Language of New Media*. MIT Press, 8 edition, 2007. ISBN: 9780262632553.

[104] Wikipedia. Jurassic park. [Online; accessed 23-April-2009], http://en.wikipedia.org/wiki/Jurassic_Park, 2009.

[105] Chris Robinson. Waking life: The truth is in the animation. *Montage Magazine*, 2004.

[106] Sheila Sofian. The truth in pictures, explores the multifaceted world of documentary animation. *The Animated Documentary*, pages 7–11, March 2005. http://fpsmagazine.com/mag/2005/03/fps200503hi.pdf.

[107] Stephen Rowley. Life reproduced in drawings: Realism in animation. *Animation Journal*, January 2005. http://www.highbeam.com/doc/1P3-1042427691.html.

[108] Wikipedia. Ken burns. [Online; accessed 23-April-2009], http://en.wikipedia.org/wiki/Ken_Burns, 2009.

[109] Paramount pictures. American teen. http://www.americanteenthemovie.com/, 2008.

[110] Wikipedia. American teen. [Online; accessed April-2009], http://en.wikipedia.org/wiki/American_Teen, 2009.

[111] Paul Wells. *Understanding Animation*. Routledge, illustrated, reprint edition, 1998. ISBN 0415115973, 9780415115971.

[112] Ulysses Ronquillo. Winsor McCay's *the sinking of the lusitania*. [Online; accessed 2-January-2000], http://drnorth.wordpress.com/2009/01/02/winsor-mccays-the-sinking-of-the-lusitania/, 2009.

[113] Wikipedia. The Sinking of the Lusitania. [Online; accessed April-2009], http://en.wikipedia.org/wiki/The_Sinking_of_the_Lusitania, 2009.

[114] Sybil DelGaudio. If truth be told, can 'toons tell it? documentary and animation. *Film History*, 9(2):189–199, 1997.

[115] Chris Robinson. Waking life: The truth is in the animation. *Montage Magazine*, 2004.

[116] Frank Thomas and Ollie Johnston. *Disney Animation: The Illusion of Life*. Abbeville Press, 1984.

[117] Unascribed. Reviving an ancient art. *The Times*, page 10, August 2006.

[118] Andrew Wilks. *Walking with Dinosaurs*. [documentary], 1999. http://en.wikipedia.org/wiki/Walking_with_Dinosaurs.

[119] Christian Darkin. 3D animation, motion graphics and CGI for documentary films. http://www.anachronistic.co.uk/tip-details/4-3d_animation,_motion_graphics_and_cgi_for_documentaries.htm, 2009.

[120] Ohad Landesman. In and out of this world: digital video and the aesthetics of realism in the new hybrid documentary. *Studies in Documentary Film*, 2(1):33–45, March 2008. http://www.atypon-link.com/INT/doi/abs/10.1386/sdf.2.1.33_1.

[121] USC Documentary Animation – USC Seminar Group. Interviews of documentary animation directors. [online], 2007–2008. http://docanimation.blogspot.com/.

[122] Sheila Sofian. The truth in pictures. *FPS Maganize*, March 2005. Online at http://fpsmagazine.com/mag/2005/03/fps200503hi.pdf.

[123] Roy G. Levin. *Documentary Explorations: 15 Interviews with Film Makers*. Doubleday, 1971.

[124] Bill Nichols. Documentary theory and practice. *Screen*, Winter(17):34–48, 1976–1977.

[125] Bill Nichols. *Ideology and the Image*. Indiana University Press, Bloomington, Indiana, 1981.

[126] Bill Nichols. *Introduction to Documentary*. Indiana University Press, Bloomington, Indiana, 2001.

[127] Bill Nichols. History, myth, and narrative in documentary. *Film Quarterly*, Fall:9–20, 1987.

[128] Bill Nichols. *Representing Reality*. Indiana University Press, Bloomington, Indiana, 1991.

[129] Lionel Rogosin. Interpreting reality. *Film Culture*, 21:20–28, 1960.

[130] Alan Rosenthal, editor. *New Challenges for Documentary*. University of California Press, 1988.

[131] R. Stebbins and J. Leyda. Joris ivens: Artist in documentary. *Magazine of Art*, 31:392, July 1938.

[132] Malin Wahlberg. *Documentary Time: Film and Phenomenology*. University of Minnesota Press, 2008.

[133] Hayden White. The value of narrativity in the representation of reality. *The Content of the Form*, pages 1–25, 1987.

[134] Mi Young Lee and Marielle Nitoslawska. Expanded documentary bibliography. [online], June 2009.

[135] Copper Heart and the National Film Board of Canada. Computer graphics research helps *ryan* come home with an Oscar. *Orion Research and Discovery News*, 3(2), April 2005.

[136] Morgan Spurlock. *Super Size Me*. [independent film], May 2004.

[137] Jairo Eduardo Carrillo. The making of *little voices*. [online], `http://www.locombia.net/voices/voien.html`, 2003. `http://www.youtube.com/watch?v=ZePSdP5s2ck`.

[138] Jeff Chastine and Ying Zhu. The cost of supporting references in collaborative augmented reality. In *Proceedings of Graphics Interface 2008 (GI'08)*, pages 275–282, Windsor, ON, Canada, May 2008. Canadian Human-Computer Communications Society.

[139] Stephen Cawood and Mark Fiala. *Augmented Reality: A Practical Guide*. Pragmatic Bookshelf, 2008.

[140] EDUCAUSE Learning Initiative (ELI). 7 things you should know about augmented reality, 2005. `http://connect.educause.edu/Library/`.

[141] Soha Maad, editor. *Augmented Reality*. InTech, January 2010. ISBN: 978-953-7619-69-5; online at `http://www.intechopen.com/books/augmented-reality`.

[142] Kevin Duncan. Speaking through Little Voices, video game tells the stories of displaced children. `http://www.banffcentre.ca/about/inspired/2009/winter/speak_through_life.asp`, 2008.

[143] Eduardo Carrillo. Born under fire. [online, motion picture], 2008. `http://www.locombia.net/`.

[144] Brett Morgan. *Chicago 10*. [independent film], January 2007.

[145] Sam Chen. Where great minds meet. http://www.animationtrip.com/item.php?id=55, 2005. The AnimationTrip Interview Series.

[146] Wendy Ide. Waltz with Bashir. *Times Online*, May 2008.

[147] Jerrin Zumberg. Israeli filmmakers head to cannes with animated documentary. http://www.israel21c.net, May 2008. ISRAEL21c News.

[148] Sony Classics. Waltz with Bashir presskit. http://www.sonyclassics.com/waltzwithbashir/pdf/waltzwithbashir_presskit.pdf, 2008.

[149] Nick Dawson. Drawing from memory. *FILMMAKER*, January 2009. http://www.filmmakermagazine.com/webexclusives/labels/OscarPreview2009.php.

[150] R. Molich and Jakob Nielsen. Improving a human-computer dialogue. *Communications of the ACM*, 33(3):338–348, March 1990.

[151] Wikipedia. Computer graphics. [Online; accessed 23-April-2009], http://en.wikipedia.org/wiki/Computer_graphics, 2009.

[152] Wikipedia. Human-computer interaction. [Online; accessed 23-April-2009], http://en.wikipedia.org/wiki/Human-computer_interaction, 2009.

[153] Mark Gawlinski. *Interactive Television Production*. Focal Press, 2003. ISBN: 0240516796, 9780240516790, http://books.google.com/books?id=TwKfJj9U3V8C.

[154] Neil Wilkes. Sky One to screen interactive documentary. *Digital Spy*, 2000.

[155] Konstantinos Chorianopoulos and Diomidis Spinellis. User interface development for interactive television: extending a commercial dtv platform to the virtual channel api. *Computers & Graphics*, 28(2):157–166, April 2004.

[156] Konstantinos Chorianopoulos and Diomidis Spinellis. A metaphor for personalized television programming. In *User Interfaces for All, LNCS 2615*, pages 187–194, New York, USA, 2003. Springer.

[157] Larry Press. Compuvision or teleputer? *j-CACM*, 33(9):29–36, 1990.

[158] J. Clark. A telecomputer. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, pages 19–23, New York, USA, 1992. ACM.

[159] Pablo Cesar, Konstantinos Chorianopoulos, and Jens F. Jensen. Social television and user interaction. *Comput. Entertain.*, 6:4:1–4:10, May 2008.

[160] Ian Willoughby. Groundbreaking Czechoslovak interactive film system revived 40 years later. `http://www.radio.cz/en/section/panorama/groundbreaking-czechoslovak-interactive-film-system-revived-40-years-later`, 2007.

[161] Jeffrey Shaw and Peter Weibel, editors. *Future Cinema: the Cinematic Imaginary After Film*. MIT Press, Cambridge, Mass., 2003. ISBN: 978-0262692861.

[162] G. Davenport, S. Agamanolis, B. Barry, B. Bradley, and K. Brooks. Synergistic storyscapes and constructionist cinematic sharing. *IBM Systems Journal*, 39(3-4):456–469, November 2000.

[163] Edward Angel. *Interactive Computer Graphics: A Top-Down Approach Using OpenGL*. Addison-Wesley, 2003.

[164] University of RMIT. Interactive online documentary. `http://media.rmit.edu.au/students/projects/wiki/index.php/Interactive_Online_Documentary`, 2007.

[165] AFC. Australian Film Commission (AFC). `http://www.screenaustralia.gov.au/`, 2009.

[166] ABC. Australian Broadcast Corporation (ABC). `http://www.abc.net.au/`, 2009.

[167] Régis Debray. *Media Manifestos*. Verso, 1994. ISBN: 978-1859840870.

[168] David Norman Rodowick. *The Virtual Life of Film*. Harvard University Press, Cambridge Mass., 2007.

[169] Bill Nichols. The voice of documentary. *Film Quarterly*, 3(36):17–30, 1983.

[170] Lon Barfield. Interactive documentaries. *SIGCHI Bull.: suppl. interactions*, 2003:14–14, January 2003. Real World column.

[171] Patrick Tarrant. Planet Usher: An interactive home movie. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA'04, pages 987–988, New York, NY, USA, October 2004. ACM.

[172] Michael Anderson, John Carroll, and David Cameron, editors. *Drama Education with Digital Technology*. Continuum, New York, 2009. ISBN: 978-184-706-266-6.

[173] Cynthia Katherine Poremba. *Real—Unreal: Crafting Actuality in the Documentary Videogame*. PhD thesis, Concordia University, Montreal, Canada, 2011.

[174] Richard Lachman. Diamond road online: A user-guided documentary experience. In *ACM SIGGRAPH 2008 new tech demos*, SIGGRAPH'08, pages 12:1–12:1, New York, NY, USA, August 2008. ACM.

[175] Nicolas Szilas, Jason Barles, and Manolya Kavakli. An implementation of real-time 3D interactive drama. *Comput. Entertain.*, 5, January 2007.

[176] Glorianna Davenport Researchers. Evolving documentary. `http://ic.media.mit.edu/`, 2004. Interactive Cinema Group.

[177] Perry Bard. Man with a movie camera: The global remake. [online], `http://dziga.perrybard.net/`, 2007. Interactive Documentary Project.

[178] Evelin Stermitz. Man with a movie camera: The global remake – interview with perry bard. `http://www.rhizome.org/discuss/view/41030`, 2008. Interactive Documentary Project.

[179] Jacqueline Wallace. NYT's bits blogger on digital storytelling. [online], March 2010. `http://www.cinerg.ca/archives/295`.

[180] Yiping Yuan. The history and status quo of traditional Japanese theater. Art Faculty, Nihon University, September 2011. lecture.

[181] Antonin Artaud. *The Theatre and Its Double: Essays*. Calder & Boyars, 1970.

[182] Jerzy Grotowski and Eugenio Barba. *Towards a Poor Theatre*. Routledge, New York, 1 edition, 1968. ISBN: 978-0878301553.

[183] Peter Brook. *The Empty Space*. Atheneum, New York, 1968.

[184] Bertolt Brecht. *Brecht on Theatre: The Development of an Aesthetic*. Hill and Wang, 13 edition, 1977. ISBN: 978-0809005420.

[185] Thomas Richards and Jerzy Grotowski. *At Work with Grotowski on Physical Actions*. Routledge, New York, 1995. ISBN: 9780415124928.

[186] Farley Richmond. *India*, pages 516–525. Banham, 1995.

[187] Elizabeth Halson. *Peking opera: a short guide*. Hong Kong; London: Oxford University Press, 1966.

[188] Brenda Laurel. *Computers as Theatre*. Addison-Wesley, Reading, Mass., 1993. ISBN: 978-020-155-060-3.

[189] Aristotle. *Poetics*. (1447a), 335 BCE. `http://ebooks.adelaide.edu.au/a/aristotle/poetics/`.

[190] Gabriella Giannachi. *Virtual Theatres: An Introduction*. Routledge, New York, 2004. ISBN: 041-528-378-7.

[191] Paolo Atzori and Kirk Woolford. Extended-body: Interview with Stelarc. [online], 1995. `http://www.ctheory.net/articles.aspx?id=71`.

[192] Christopher Salter. *Entangled: Technology and the Transformation of Performance*. MIT Press, Cambridge, Mass., 2010. ISBN: 978-026-219-588-1.

[193] Natasha Tsakos. Natasha Tsakos' multimedia theatrical adventure. [online], TED2009, February 2009. `http://www.ted.com/talks/lang/en/natasha_tsakos_multimedia_theatrical_adventure.html`.

[194] Marc Hollogne. L'illuminé. [online], Cinéma-Théâtre, 2010. `http://www.h07.org/marc-hollogne-makes-movies-in-the-theater-dejazet.html`, `http://fr.wikipedia.org/wiki/Marc_Hollogne`, `http://www.youtube.com/watch?v=kbTNtRiez6Q`.

[195] THE SILHOUETTES. MDA Show of Strength: The silhouettes–i believe. [online], September 2012. http://theoriginalsilhouettes.com.

[196] The Montreal Museum of Fine Arts. Interview of Denis Marleau and Stéphanie Jasmin. [online], June 2011. http://www.mbam.qc.ca/jpg/en/pdf/marleaujasmin.pdf.

[197] Wikipedia. Haptic technology — Wikipedia, The Free Encyclopedia. [online; accessed 4-February-2012], 2012. http://en.wikipedia.org/w/index.php?title=Haptic_technology&oldid=473275810.

[198] Rustam Stolkin, editor. *Scene Reconstruction Pose Estimation and Tracking*. InTech, June 2007. ISBN: 978-3-902613-06-6; online at http://www.intechopen.com/books/scene_reconstruction_pose_estimation_and_tracking.

[199] Miguel Sales Dias, Sylvie Gibet, Marcelo M. Wanderley, and Rafael Bastos, editors. *Gesture-Based Human-Computer Interaction and Simulation, 7th International Gesture Workshop, GW 2007, Lisbon, Portugal, May 23-25, 2007, Revised Selected Papers*, volume 5085 of *LNCS*. Springer, 2009.

[200] Mehrdad Hosseini Zadeh, editor. *Advances in Haptics*. InTech, April 2010. ISBN: 978-953-307-093-3; online at http://www.intechopen.com/books/advances-in-haptics.

[201] Wikipedia. Kinect — Wikipedia, The Free Encyclopedia. [online; accessed 4-February-2012], 2012. http://en.wikipedia.org/w/index.php?title=Kinect&oldid=474626703.

[202] OpenKinect Developers. The OpenKinect project. [online], 2012. http://openkinect.org.

[203] Microsoft. The Kinect for Windows SDK v. 1.5. [online], May 2012. Online at http://www.microsoft.com/en-us/kinectforwindows/develop/developer-downloads.aspx and http://msdn.microsoft.com/en-us/library/hh855347.

[204] Microsoft. The Kinect for Windows Developer Toolkit v. 1.5.2. [online], August 2012. http://go.microsoft.com/fwlink/?LinkId=259543.

[205] Microsoft. The Kinect Studio v. 1.5.0.1. [online], 2012. http://msdn.microsoft.com/en-us/library/hh855389.

[206] Microsoft. Human Interface Guidelines: Kinect for Windows v. 1.5. [online], 2012. http://go.microsoft.com/fwlink/?LinkId=247735.

[207] Novint Technologies, Inc. Haptic Device Abstraction Layer (HDAL) Programmer's Guide. [online; 14-August-2008], 2008. http://www.novint.com.

[208] Novint Technologies, Inc. Novint Falcon SDK. [online; version 2.1.3], 2008. http://www.novint.com/index.php/support/downloads.

[209] Novint Technologies, Inc. F-Gen – generalized community supported driver set for Falcon. [online], 2011. http://www.novint.com/index.php/fgen.

[210] Novint Technologies, Inc. Novint F-Gen SDK. [online; version 1.0.0.0_100818], August 2010. http://www.novint.com/index.php/support/downloads.

[211] OpenGL Architecture Review Board. OpenGL. [online], 1998–2012. http://www.opengl.org.

[212] Dave Shreiner and The Khronos OpenGL ARB Working Group. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Versions 3.0 and 3.1.* Addison-Wesley, 7 edition, July 2009. ISBN 978-0321552624.

[213] Dave Shreiner, The Khronos OpenGL ARB Working Group, Bill Licea-Kane, and Graham Sellers. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.1.* Addison-Wesley, 8 edition, March 2012. ISBN 978-0321773036.

[214] Sumanta Guha. *Computer Graphics Through OpenGL: From Theory to Experiments.* Chapman & Hall and CRC Computer Graphics, 1 edition, September 2010. ISBN: 978-1439846209.

[215] Mark J. Kilgard. All about OpenGL extensions. OpenGL.org, 1998–1999. http://www.opengl.org/resources/features/OGLextensions/.

[216] CodeColony. OpenGL Colony tutorials and demos. [online], 2008. http://www.codecolony.de/.

[217] Silicon Graphics Inc. GLUT examples. [online], 1997. http://www.opengl.org/resources/code/samples/glut_examples/examples/examples.html.

[218] Paul Rademacher, Nigel Stewart, and Bill Baxter. GLUI – A GLUT-based user interface library, version 2.35. [online], 1999–2006. http://glui.sourceforge.net/.

[219] Fabio Policarpo. Fly3D 2.0 SDK game engine. Included into the book "3D Games Animation and Advanced Real-time Rendering, ISBN: 0-201-78706-7", 2003.

[220] Alexandre Santos Lob ao, Bruno Evangelista, José Antonio Leal de Farias, and Riemer Grootjans. *Beginning XNA 3.0 Game Programming: From Novice to Professional*. Apress, 2009. ISBN 978-1-4302-1817-3.

[221] Microsoft. Beginner's guide to XNA Game Studio Express. your world. your game. DVD, 2007. http://creators.xna.com.

[222] Aaron Reed. *Learning XNA 3.0*. O'Reilly, 2009. ISBN 978-0-596-52195-0.

[223] Benjamin Nitschke. *Professional XNA Game Programming for XBOX 360 and Windows*. Wiley Publishing, 2007. ISBN 978-0-470-12677-6.

[224] Chad Carter. *Microsoft XNA Game Studio 3.0 Unleashed*. SAMS, April 2009. ISBN 978-0-672-33022-3.

[225] Microsoft. Microsoft.Kinect Namespace. [online], MSDN Library, 2012. http://msdn.microsoft.com/en-us/library/hh855419.

[226] radio42. BASS audio library .Net API v. 2.4.9. [online], 2012. http://www.un4seen.com/download.php?z/4/Bass24.Net.zip and http://bass.radio42.com/help/Index.html, last viewed August 2012.

[227] un4seen developments. BASS audio library v. 2.4.9. [online], 2012. http://www.un4seen.com/bass.html and http://www.un4seen.com/doc/, last viewed August 2012.

[228] Dave Grandbois. Bass.net / XNA music visualizer. [online], December 2011. http://gbois.hopto.org/view/34/Bass-net-XNA-Music-Visualizer.

[229] Jeff Molofee and Contributors. Lesson 35: Playing AVI movies in OpenGL. [online], Neon Helium (NeHe) Productions, 2006. http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=35; last viewed October 2011.

[230] Microsoft. Performance measuring sample. [online], MSDN, October 2010. http://create.msdn.com/en-US/education/catalog/sample/performance_sample.

[231] digitalerr0r. Kinect fundamentals #4: Implementing skeletal tracking. [online], December 2011. http://digitalerr0r.wordpress.com/2011/12/13/kinect-fundamentals-4-implementing-skeletal-tracking/.

[232] David Catuhe. Kinect Toolbox 1.2. [online], July 2012. http://kinecttoolbox.codeplex.com/ and http://blogs.msdn.com/b/eternalcoding/archive/2012/07/31/kinect-toolbox-1-2.aspx.

[233] Michael Tsikkos and James Glading. Writing a gesture service with the Kinect for Windows SDK. [online], August 2011. http://blogs.msdn.com/b/mcsuksoldev/archive/2011/08/08/writing-a-gesture-service-with-the-kinect-for-windows-sdk.aspx.

[234] Except on Tuesdays. Gestures with Microsoft Kinect for Windows SDK v1.5. [online], July 2012. http://blog.exceptontuesdays.com/post/27989563563/gestures-with-microsoft-kinect-for-windows-sdk-v1-5.

[235] Microsoft. Kinect Shape Game C# Sample. [online], MSDN Library, 2012. http://msdn.microsoft.com/en-us/library/hh855385.

[236] Alex Urbano Álvarez, Javier Cantón Ferrero, and David Mariscal Fernández. Fancy bubble shader sample, 2009. XNA3.1, online at `http://elgoe.blogspot.com` and `http://xnacommunity.codeplex.com/wikipage?title=BubbleShader`.

[237] Mike Eissele, Matthias Kreiser, and Thomas Erd. Context-controlled flow visualization in augmented reality. In *Proceedings of Graphics Interface 2008 (GI'08)*, pages 89–96, Windsor, ON, Canada, May 2008. Canadian Human-Computer Communications Society.

[238] F. P. Brooks Jr. What's real about virtual reality? *IEEE Computer Graphics And Applications*, 6(19):16, 1999.

[239] Yuval Boger. Virtual reality and head-mounted displays. [online], 2009. `http://vrguy.blogspot.com/`.

[240] Sensics. Sensics, the panoramic virtual reality display system. [online], 2010. `http://www.sensics.com/`, last viewed January 2010.

[241] Scott Malthouse. Merged worlds: When games invade reality. *Thirteen 1 - Online Games Magazine*, September 2009. Online at `http://www.thirteen1.com/Issues/1313909/mag.php`.

[242] Wendy A. Powell, S. Hand, S. Stevens, and Maureen J. Simmonds. Optic flow with a stereoscopic display: Sustained influence on speed of locomotion. *Annual Review of Cybertherapy and Telemedicine*, pages 65–70, 2006.

[243] Maureen J. Simmonds, G. L. Moseley, and J. Vlaeyen. Pain, mind and movement: An integrated and updated conceptualization. *Clin J Pain*, 24(4):279–280, 2008.

[244] Wendy A. Powell, Brett Stevens, and Maureen J. Simmonds. Treadmill interface for virtual reality vs. overground walking: A comparison of gait in individuals with and without pain. *Studies in health technology and informatics*, 144:198–203, 2009.

[245] Shahnaz Shahrbanian, Xiaoli Ma, Nicol Korner-Bitensky, and Maureen J. Simmonds. Scientific evidence for the effectiveness of virtual reality for pain reduction in adults with acute or chronic pain. *Stud Health Technol Inform*, 144:40–43, 2009.

[246] Wendy A. Powell, Brett Stevens, S. Hand, and Maureen J. Simmonds. Gain-matching and perception of self-motion: The relationship between visual flow and treadmill walking. In *12th Annual Cybertherapy 2007: Transforming Healthcare Through Technology*, June 2007.

[247] Wendy A. Powell, Brett Stevens, S. Hand, and Maureen J. Simmonds. Optimizing the rehabilitation potential of a treadmill interfaced to the virtual world. In *Proceedings of Computer/Human Interaction 2007*, April 2007.

[248] Shahnaz Shahrbanian and Maureen J. Simmonds. Effects of different virtual reality environments on experimental pain rating in post-stroke individuals with and without pain in comparison to pain free healthy individuals. In *13th Annual CyberTherapy Conference 2008: Changing the Face of Healthcare*, June 2008.

[249] Maureen J. Simmonds and Shahnaz Shahrbanian. Effects of different virtual reality environments on pain threshold in individuals with pain following stroke. In *7th International Conference Series on Disability, Virtual Reality and Associated Technologies with ArtAbilitation*, September 2008.

[250] Wendy A. Powell, Brett Stevens, and Maureen J. Simmonds. Treadmill interface for virtual reality vs. overground walking: a comparison of gait in individuals with and without pain. In B. K. Weiderhold and G. Riva, editors, *Annual Review of Cybertherapy and Telemedicine*, pages 198–203. IOS Press, 2009.

[251] Shahnaz Shahrbanian, Xiaoli Ma, Nicol Korner-Bitensky, and Maureen J. Simmonds. Scientific evidence for the effectiveness of virtual reality for pain reduction in adults with acute or chronic pain. In B. K. Weiderhold and G. Riva, editors, *Annual Review of Cybertherapy and Telemedicine*, pages 40–43. IOS Press, 2009.

[252] Miao Song, Peter Grogono, and Maureen J. Simmonds. Towards innovative application of computer graphics techniques in responsive virtual and augmented realities. Poster at the 1s ECSGA Colloquium, April 2010.

[253] WorldViz. Vizard3D VR Toolkit. [digital], 2002–2012. `http://www.worldviz.com/products/vizard`.

[254] The OGRE Team. Object-oriented graphics rendering engine (OGRE) 3D. [online], 2000–2012. http://www.ogre3d.org/.

[255] Miao Song and Peter Grogono. An LOD control interface for an OpenGL-based softbody simulation framework. In Tarek Sobh, editor, *Innovations and Advances in Computer Sciences and Engineering, Proceedings of CISSE'08*, pages 539–543. Springer Netherlands, December 2008. Published in 2010.

[256] Michael Fortin, Miao Song, David Gauthier, Sha Xin Wei, and Peter Grogono. Jellyfish simulation. Poster at the 2nd Grand Conference, May 2011.

[257] Michael Fortin. Interactive simulation of fluid flow. Master's thesis, Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada, April 2011.

[258] Peter Grogono. Concordia University Graphics Library (CUGL). [online], December 2005. http://users.encs.concordia.ca/~grogono/Graphics/cugl.html.

[259] 3D Labs. OpenGL Shading Language demo and documentation. 3D Labs, Inc., 2004. http://developer.3dlabs.com/openGL2/downloads/index.htm.

[260] 3D Labs. OpenGL Shading Language shader examples and source code. 3D Labs, Inc., 2004. http://3dshaders.com/shaderSource.html.

[261] Sam Lantinga and the SDL Contributors. SDL – Simple Directmedia Layer. [online], 2008–2012. http://www.libsdl.org/.

[262] Software Engineering Standards Committee of the IEEE. IEEE recommended practice for software requirements specifications, 1998. IEEE Std 830-1998.

[263] Alain Abran, James M. Moore, Pierre Bourque, and Robert Dupuis, editors. *Guide to The Software Engineering Body of Knowledge (SWEBOK): 2004 Version*. IEEE Computer Society, 2004. http://www.computer.org/portal/web/swebok/htmlformat.

[264] Hidetoshi Nonaka. Operation-based notation for archimedean graph. In Nagib Callaos, William Lesso, C. Dale Zinn, Jorge Baralt, Jaouad Boukachour, Christopher White, Thilidzi Marwala, and Fulufhelo V. Nelwamondo, editors, *Proceedings of*

the 12th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI'08), Orlando, Florida, USA, June 2008. IIIS.

[265] Richard Stallman, Roland McGrath, Paul Smith, and the GNU Project. GNU Make. Free Software Foundation, Inc., [online], 1997–2006. `http://www.gnu.org/software/make/`.

[266] Joaquim Jorge, Frank Hanisch, Frederico Figueiredo, and Rhonda Schauer. CG Educational Materials Source (CGEMS). [online], 2008–2012. `http://cgems.inesc.pt/`.

[267] Jack Ord. Basic physics with Java: Newton's law and the Feynman algorithm: The Euler and Feynman algorithms: a mass on a spring. igs.net, 2007. `http://www.kw.igs.net/~jackord/bp/f1.html`, Viewed November 2007.

[268] Craig Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*. Prentice Hall PTR, 2001. ISBN: 0131489062.

[269] Clockworkcoders. OpenGL Shading Language tutorials and demos. [online], 2007. `http://www.clockworkcoders.com/oglsl/`.

[270] Lighthouse 3D. GLSL tutorial. [online], 2008. `http://www.lighthouse3d.com/opengl/glsl/`.

[271] TyphoonLabs. TyphoonLabs' OpenGL Shading Language tutorials. [online], 2008. `http://www.opengl.org/sdk/docs/tutorials/TyphoonLabs/`.

[272] Serguei A. Mokhov. Real-time animation of hair and thread-like objects via deformation of layered meshes. Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada, 2004. Project and report.

[273] Peter Grogono. Getting started with OpenGL. [online], 2002. Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada.

[274] Peter Grogono. Lecture notes of advanced computer graphics. [online], 2002. Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada.

[275] Miao Song and Yolanda Ye. Interactive underwater sea world simulation in OpenGL. Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada, 2003. A computer graphics project.

[276] Miao Song. Spectacle. Animation Film, 2003. 1 minute 15 seconds.

[277] Miao Song. Tangible Memories: Multi-dimensional interactive documentary presentation, April 2011. [Concordia University Humanities Doctoral Student Annual Conference]; http://dislocationsconference.wordpress.com/schedule/.

[278] Microsoft. System.Speech.Recognition Namespace. [online], MSDN Library, 2011. http://msdn.microsoft.com/en-us/library/ms554855.

[279] Wikipedia. Poetics (Aristotle) — Wikipedia, The Free Encyclopedia. [Online; accessed 13-October-2012], 2012. http://en.wikipedia.org/w/index.php?title=Poetics_(Aristotle)&oldid=513909371.

[280] Miao Song (Director) and Deschanel Li (Dancer). Dance with me! My three dancers: Skeleton, Depth, and Color! [online], 2012. https://vimeo.com/49682696.

[281] Miao Song (Director) and Timmy. Timmy, the Virtual Audience, in his own dance competition. [online], 2012. https://vimeo.com/50069419.

[282] Miao Song (Director) and Deschanel Li (Dancer). When Chinese Water Sleeve Dance meets New Media! [online], 2012. https://vimeo.com/49399617.

[283] Miao Song (Director). Memory bubble. [online], October 2012. https://vimeo.com/51329588.

[284] Douglas O'Shaughnessy. *Speech Communications*. IEEE, New Jersey, USA, 2000.

[285] PUMA L.I.F.T. PUMA L.I.F.T. shoe commercial. [online], March 2009. http://www.youtube.com/watch?v=TM8DA830xng.

[286] Puma Running. L.I.F.T. behind the scenes with Bolt and Delilah. [online], March 2009. http://www.youtube.com/watch?v=fAYWhlm4MOc.

[287] Miao Song and Peter Grogono. Soft body simulation: Physical based animation with OpenGL. Poster at the 1s ECSGA Colloquium, April 2010.

[288] Serguei A. Mokhov, Miao Song, and Ching Y. Suen. Writer identification using inexpensive signal processing techniques. In Tarek Sobh and Khaled Elleithy, editors, *Innovations in Computing Sciences and Software Engineering; Proceedings of CISSE'09*, pages 437–441. Springer, December 2009. ISBN: 978-90-481-9111-6, online at: `http://arxiv.org/abs/0912.5502`.

[289] Serguei A. Mokhov and Miao Song. OpenGL project presentation slides interface and a case study. In *Proceedings of GRAPP'09*, pages 409–412, Lisboa, Portugal, February 2009. INSTICC.

[290] Miao Song, Serguei A. Mokhov, and Peter Grogono. Designing an interactive OpenGL slide-based presentation of the softbody simulation system for teaching and learning of computer graphics techniques. In *Proceedings of C3S2E'09*, pages 131–136, New York, NY, USA, May 2009. ACM.

[291] Miao Song. Feynman algorithm implementation for comparison with Euler in a uniform elastic two-layer 2D and 3D object dynamic deformation framework in OpenGL with GUI. [online], 2009. `http://arxiv.org/abs/0906.3074`.

[292] Miao Song. ABRA. CLSP Research Concordia Prom Video, 2009.

[293] Serguei A. Mokhov and Miao Song. An OpenGL-based interface to 3D PowerPoint-like presentations of OpenGL projects. In *Advanced Techniques in Computing Sciences and Software Engineering, Proceedings of CISSE'08*, pages 533–538. Springer Netherlands, December 2008. Published in 2010.

[294] Common Thread Productions. Unraveling Her Story. Theatre Production, Common Thread Productions, 2007. Directed by Emily Burkes-Nossiter, Co-directed by Chia-Wen, Scriptwriting and acting by Lois Jones, Lucy Lu, Joy Ruben, Dorothy Singer, Miao Song, Talia Weisz, and Mimi Zhou.

[295] Miao Song. Tangram. Interactive Flash Games, 2003.

[296] François Conti and Oussama Khatib. Spanning large workspaces using small haptic devices. In *WHC*, pages 183–188. IEEE Computer Society, 2005.

[297] News Staff. Synthetic Life? Silicone And Cardiac Tissue Becomes Medusoid – Swimming 'Jellyfish'. In *Science 2.0* [98]. Online at `http://www.science20.com/news_articles/synthetic_life_silicone_and_cardiac_tissue_becomes_medusoid_swimming_jellyfish-92307`.

[298] Jesse Emspak. Artificial jellyfish made from rat cells. In *Discovery News* [98]. Analysis; online at `http://news.discovery.com/tech/artificial-jellyfish-rat-cells-120723.html`.

[299] Miao Song (Director). *I Still Remember*. [Documentary film] 13 minutes. Top 10 in HTMlles 2010, `http://www.htmlles.net/2010/projects/emergence/`; 1st BJIFF "See the world through films" Contest for Best Documentary Short in 2011, Best Documentary Award, `http://www.bjiff.com/en/bjiffnews/n214618230.shtml`; The 6th Vancouver Women in Film Festival (VWIFF) 2012, Official Selection, `http://www.womeninfilm.ca/conceivingfamily.html`.

[300] Cycling '74. Jitter 1.5. [online], 2005. `http://www.cycling74.com/products/jitter.html`.

[301] Cycling '74. Max/MSP. [online], 2005. `http://www.cycling74.com/products/maxmsp.html`.

[302] Miller Puckette and PD Community. Pure Data. [online], 2007–2012. `http://puredata.org`.

[303] Jean-Marc Pelletier. `jit.freenect.grab` – a Max/MSP/Jitter external for Microsoft Kinect. [online], March 2012. RC5, `http://jmpelletier.com/freenect/`.

[304] Wikipedia. Mono (software) — Wikipedia, The Free Encyclopedia. [Online; accessed 24-September-2012], 2012. `http://en.wikipedia.org/w/index.php?title=Mono_(software)&oldid=511065984`.

[305] Lars Magnusson, Tim Pambor, Alan McGovern, C. J. Adams-Collier, et al. MonoXNA: Cross platform implementation of the Microsoft XNA framework. [online], 2009–2011. http://www.monoxna.org/ and code.google.com/p/monoxna.

[306] Alan Buis, Whitney Clavin, J. D. Harrington, and NASA. NASA telescope finds elusive buckyballs in space. [online], July 2010. http://www.nasa.gov/mission_pages/spitzer/news/spitzer20100722.html.

[307] Irene Klotz. Bushels of buckyballs found in space. [online], October 2010. http://news.discovery.com/space/buckyballs-carbon-space-nebula.html.

[308] Stephen Sinclair and Marcelo M. Wanderley. Using PureData to control a haptically-enabled virtual environment. In *Pd Convention '07*, Montreal, Quebec, Canada, August 2007. artengine.ca. http://artengine.ca/~catalogue-pd/7-Sinclair.pdf.

[309] Rob The Bloke. Maya API: MPxTools. [online]. http://maya.robthebloke.org/MPxTools.html.

[310] Maya Developers. Maya API: class MFnCamera. [online]. http://caad.arch.ethz.ch/info/maya/manual/DevKit/PlugInsAPI/classDoc/MFnCamera.html.

[311] Various Contributors. Maya plug-ins. [online]. http://highend3d.com/maya/downloads/plugins/.

[312] Mark J. P. Wolf, editor. *Encyclopedia of Video Games: The Culture, Technology, and Art of Gaming*. Greenwood, August 2012.

Index

Appendix

Appendix A

User Manuals

A.1 Curve-Based Animation Control Keys

The keys are as follows:

- `+` – zoom in by decreasing `fovy` (not part of camera)
- `-` – zoom out by increasing `fovy` (not part of camera)
- `W`, `8` – tilt camera up
- `X`, `2` – tilt camera down
- `D`, `6` – turn right
- `A`, `4` – turn left
- `ARROW KEY UP` – move camera forward
- `ARROW KEY DOWN` – move camera backward
- `ARROW KEY LEFT` – move camera left
- `ARROW KEY RIGHT` – move camera right
- `R` – place camera into the initial position (not a full reset)

A.2 Tangible Memories Kinect Interaction

A.2.1 Keyboard Controls

ESC – complete the viewing/interaction session

V – toggle video playback

K – toggle main Kinect video feed

L – toggle Kinect skeleton capture

D – toggle Kinect depth capture

O – initialize and make visible skybox environment or disable it

B – toggle the visibility of the skybox environment (if initialized)

G – toggle the visibility of the floor

P – toggle the camera projection of the floor

C – toggle clean screen

F – toggle full screen

1 – add simple LOD 2D memory bubbles to the scene

2 – add simple LOD 3D memory bubbles to the scene

3 – add fancy LOD 3D memory bubbles to the scene

F1 – exclusively add simple LOD 2D memory bubbles to the scene; exclude all others

F2 – exclusively add simple LOD 3D memory bubbles to the scene; exclude all others

F3 – exclusively add fancy LOD 3D memory bubbles to the scene; exclude all others

S – toggle the speech-based updates. Only active when speech processing was actually initialized (with 'T' or by default).

T – initialize or unload speech processing. Implies turning off speech updates on unload and the reverse on load. Speech updates 'S' can be selectively turned on and off with 'S' when 'T' is on.

M – toggle music visualizer

F11 – toggle wireframe mode for the computed memory bubbles

X – toggle collision detection among bubbles

F4 – simulate "red" command for testing

F5 – simulate "orange" command for testing

F6 – simulate "yellow" command for testing

F7 – simulate "green" command for testing

F8 – simulate "cyan" command for testing

F9 – simulate "blue" command for testing

F10 – simulate "violet" command for testing

0 – simulate "play" command for testing

F12 – simulate "stop" command for testing

Left Control – toggle white/black background

A.2.2 Mouse Controls

Rotation of the camera in the skybox world when it is enabled.

A.2.3 Voice Commands

red bubble, red bubble please – calls up red media bubble

orange bubble, orange bubble please – calls up orange media bubble

yellow bubble, yellow bubble please – calls up yellow media bubble

green bubble, green bubble please – calls up green media bubble

cyan bubble, cyan bubble please – calls up the light blue media bubble

blue bubble, blue bubble please – calls up blue media bubble

violet bubble, violet bubble please – calls up purple media bubble

purple bubble, purple bubble please – calls up purple media bubble

play – start/resume video playback

stop – pause video playback

A.2.4 Gestures

left hand over play button – start/resume video playback

left hand over stop button – release the bubble held in the right hand from the viewer and stop the video playback in it

Appendix B

Glossary

API — application programming interface[1] (see Section 2.4.3)

ARB — OpenGL Architecture Review Board

AVI — Audio Video Interleave format

BJIFF — Beijing International Film Festival

BMP — bitmap image format

BSP — binary separating planes (see Section 2.1.2.2)

BVHs — bounding volume hierarchies (see Section 2.1.2.2)

CD — collision detection (see Section 2.1.2.2)

CG — computer graphics

CGEMS — computer graphics gems [266]

CGI — computer-generated imagery[2]

CPU — central processing unit

[1] `http://en.wikipedia.org/wiki/API`
[2] `http://en.wikipedia.org/wiki/Computer-generated_imagery`

CUGL — Concordia University Graphics Library [258] (see Section 3.4.2.1.3)

DLOD — discrete LOD (see Section 2.1.3.3.3)

FFT — fast Fourier transform [284]

FPS — frames-per-second

GLSL — OpenGL Shading Language (see Section 2.1.3.3.4)

GLUI — GLUT-Based User Interface Library [218]

GLUT — OpenGL Utility Toolkit

GPGPU — general purpose GPU computing (see Section 2.4.3.2)

GPU — graphics processing unit (see Section 2.1.3.3.4)

GUI — graphical user interface

HCI — human-computer interaction

HD — high definition

HDL — haptics device library [208]

HLSL — High-Level Shader Language (see Section 2.4.3.3)

HMD — head-mounted display

ITV — interactive TV

LOD — level of detail (see Section 2.1.3.3.3)

MoCap — motion capture

MVC — Model-View-Controller architecture

OGLSF — OpenGL Slides Framework

OGRE3D — Object-oriented Graphics Rendering Engine [254]

OpenGL — open graphics library [211] (see Section 2.4.3.1)

OS — operating system

NPR — non-photorealistic rendering

PoC — proof-of-concept

RK4 — Runge-Kutta 4th order differential equation integrator

SDK — software development kit

SDL — Simple Directmedia Layer library [261]

SIGGRAPH — Special Interest Group on GRAPHics and Interactive Techniques[3]

SIP — Special Individualized Program

STB — set-top box[4]

VB — Visual Basic

VGA — video graphics array

VR — virtual reality

XML — Extensible Markup Language

XNA — recursive acronym for "**X**NA's **N**ot **A**cronymed" [312][5]

[3] http://en.wikipedia.org/wiki/SIGGRAPH
[4] http://en.wikipedia.org/wiki/Set-top_box
[5] http://en.wikipedia.org/wiki/Microsoft_XNA