Diss. ETH No. 23900

Augmented Reality Storytelling

A thesis submitted to attain the degree of

Doctor of Sciences of ETH Zurich

(Dr. sc. ETH Zurich)

presented by

Fabio Zünd

MSc in Computer Science, ETH Zurich

born on

23 September 1984

citizen of

Altstetten S.G., Switzerland

accepted on the recommendation of

Prof. Dr. Robert W. Sumner, examiner

Prof. Dr. Markus Gross, co-examiner

Prof. Dr. Mubbasir Kapadia, co-examiner

Prof. Dr. Rémi Ronfard, co-examiner

2016

Abstract

Modern storytelling technology enables producing and consuming stories anywhere and anytime. Multimedia content is authored collaboratively and shared with a large audience in the cloud. Although such tools are very accessible and easy to use — possibly the simplest form of expressing a feeling is to click the *like* button — the produced content is not only short form, but also short-lived. The majority of stories are of a superficial nature and deeper, more complex emotions are lost.

In this thesis, we aim to leverage novel digital technology to allow people to craft and experience rich stories together. We demonstrate that Augmented Reality (AR) is a promising partner to storytelling because, first, it can vastly enhance the immersiveness and interactivity of a narrative application and, second, it can create a social setting for multiple people to share the experience. The core challenges we address are (1) computer-assisted collaborative and interactive story authoring, (2) realism and interactivity in AR storytelling and, finally, (3) we present compelling, interactive AR applications that boost creativity in real-world activities.

(1) In the story authoring domain, we present novel and innovative tools that improve collaborative as well as interactive storytelling. Our distributed, collaborative story authoring system, Story Version Control (SVC), focuses on version control for stories as well as on graphical visualization techniques to enhance collaboration among authors. We propose a media-agnostic, graph-based story representation that acts as the foundation for version control operations. A user study validated the efficacy of SVC and provided directions for

future work. Our framework for authoring interactive AR narratives demonstrates how computer-assisted authoring enables content creators to easily author complex, branching narratives with multiple story arcs.

(2) In the AR technology domain, we demonstrate how motion effects influence AR experiences and present novel methods to develop an AR coloring book application. Three experiments were conducted to assess the impact of camera motion blur, image latency, and realism of lighting conditions. We observe that image latency records the strongest correlation with improved subjective enjoyment, satisfaction, and realism, and objective scoring performance. However, the motion effects employed are not significant in the overall user experience of mobile AR games, except where harmonious or convincing blended AR image quality is consciously desired by the participants. Our work on live texturing AR characters from colored drawings explores the idea of an AR coloring book and provides a 2D to 3D texture transfer process and a novel deformable surface tracking method, which both run in real-time on a mobile device. We conclude with two user studies that the novel texture transfer process is appreciated and the overall application strongly improves the sense of connection with the character and motivates people to draw more.

(3) In the applications domain, we propose the concept of Augmented Creativity as employing AR on modern mobile devices to enhance creative activities. Nine prototype applications demonstrate how AR can be a host for creativity, education, and interaction. The applications aim at digitally complementing real-world activities, such as puzzling, coloring, music arrangement, stamps, programming, and city-wide gaming through AR. Among the applications, a prototype city-wide trading game, Gnome Trader, is discussed in more detail. Gnome Trader allows players to trade virtual resources with gnomes hidden in newspaper boxes around Switzerland. An economy simulation framework to evaluate different vir-

tual market situations is discussed. Finally, two playtesting sessions indicated that the game is functional, fun, and well received.

Zusammenfassung

Moderne Technologien ermöglichen uns Geschichten unabhängig von Zeit und Ort zu produzieren und zu konsumieren. In der Cloud können diese Geschichten gemeinschaftlich verfasst und mit einem riesigen Publikum geteilt werden. Obwohl solche Techniken sehr zugänglich und einfach zu benutzen sind, ist der dabei produzierte Inhalt nicht nur kurzlebig, sondern auch knapp in der Länge, wie beispielsweise der Gefühlsausdruck per *Like*-Button zeigt. Der Großteil der Geschichten ist oftmals von oberflächlicher Natur während tiefere, komplexere Emotionen verloren gehen.

Das Ziel dieser Thesis ist es, mittels neuen digitalen Technologien das gemeinsame Gestalten und Erleben von gehaltvollen Geschichten zu fördern. Wir demonstrieren, dass AR ein vielversprechendes Werkzeug ist, um den Immersions- und Interaktionsgrad des Geschichtenerzählens zu steigern. Zudem bietet es Menschen ein soziales Umfeld, in dem sie interaktive Geschichten, wie zum Beispiel Computerspiele, zusammen erleben können. Die drei zentralen Herausforderungen, mit denen wir uns beschäftigen, sind die folgenden: (1) Das computergestützte, kollaborative und interaktive Verfassen von Geschichten, (2) die Steigerung des Realismusgrades von AR Applikationen und (3) das Erstellen von konkreten Applikationen, um die Kreativität in alltäglichen Aktivitäten fördern zu können.

(1) Im ersten Teil, welcher sich auf das Verfassens von Geschichten bezieht, präsentieren wir neue und innovative Werkzeuge, die das gemeinschaftliche Kreieren von Geschichten sowie das erstellen von interaktiven Geschichten erleichtern sollen. Unser kolla-

boratives, verteiltest Geschichtenverfassungssystem, SVC, konzentriert sich auf die Versionskontrolle so wie auf graphische Visualisierungstechniken, um die Zusammenarbeit unter Autoren zu fördern. Wir stellen eine media-agnostische, graph-basierte Repräsentation von Geschichten vor, die das Fundament für die Versionskontrolle ermöglicht. Eine zur Testung des SVC angelegte Nutzerstudie bestätigte hierbei deren Wirksamkeit und lieferte zugleich wegweisende Erkenntnisse für weiterführende Arbeiten in diesem Bereich. Das von uns gelieferte Grundgerüst für die interaktive Gestaltung von Erzählungen demonstriert, wie computergestütztes Geschichtenschreiben Autoren dazu befähigen kann, komplexe Erzählungen mit mehrfachen Geschichtszweigen zu kreieren.

(2) In einem zweiten Teil, welcher sich mit AR Technologien auseinandersetzt, demonstrieren wir, wie Bewegungseffekte das AR Erlebnis beeinflussen können. Desweiteren präsentieren wir eine neuartige Methoden zur Gestaltung eines AR Malbuches. Drei Experimente wurden durchgeführt, um die Auswirkung von Bewegungsunschärfe, Latenzzeit und Realismus bezüglich Lichtverhältnissen auf die Nutzer zu messen. Aus unseren Resultaten geht hervor, dass die Veränderung der Latenzzeit den grössten Einfluss auf das empfundene Vergnügen und die Zufriedenheit hat, sowie auch auf die Realismus-Einschätzung und die individuell erbrachte Leistung. In unserere AR Malbuchanwendung lassen wir AR Charaktere zum Leben erwecken. Wir präsentieren zwei echtzeit Prozesse, einen um Texturen von 2D nach 3D zu transferieren und einen für das Tracking von deformierbaren Oberflächen. Wir folgern aus zwei Studien, dass der neue Texturtransferprozess geschätzt wird, die Applikation insgesamt die Verbindung zwischen Nutzer und virtuellem Charakter fördert und somit Menschen zum Malen animiert.

(3) Im dritten Teil dieser Arbeit stellen wir im Rahmen konkreter Applikationen das Konzept Augmented Creativity vor. Dabei wird AR auf modernen Smartphones oder Tablets angewendet, um die Kreativität der Nutzer zu unterstützen. Neun Prototype Applikationen demonstrieren wie AR Kreativität, Bildung und Interaktion

fördern kann. Das Ziel der Applikationen ist es, Aktivitäten aus der realen Welt, so wie Puzzeln, Malen, Musikarrangieren, Stempeln, Programmieren und städteweite Spiele, mit digitalen Verbesserungen zu ergänzen. Unter den Anwendungen befindet sich ein städteweites Spiel, Gnome Trader, das detailierter diskutiert wird. In Gnome Trader handeln die Spieler virtuelle Rohstoffe mit Gnomen, die in Zeitungsboxen überall in der Schweiz versteckt sind. Eine Ökonomiesimulation mit dem Ziel, verschiedene Situationen der virtuellen Märkte zu untersuchen, wird vorgestellt. Zwei Probespielrunden haben gezeigt, dass das Spiel funktioniert, Spass macht und gut ankommt.

Acknowledgments

I wish to extend my gratitude foremost to my supervisors, Bob Sumner and Markus Gross, for their continuous guidance and mentoring, and for believing in me all those years. Bob and Markus are fantastic visionaries and excellent mentors, with whom I loved working every day.

I am also very thankful for my previous supervisors, Stefan Mangold and Thomas Gross. Stefan and Thomas have introduced me to the world of research and have taught me valuable skills to become successful.

I would like to thank my co-examiners, Mubbasir Kapadia and Rémi Ronfard, for serving on my thesis committee, reviewing my thesis, and providing valuable feedback.

I am happy and proud that most of my co-workers and some of my students eventually became co-authors. I wish to thank all my co-authors, Bob Sumner, Markus Gross, Mubbasir Kapadia, Thomas Gross, Stefan Mangold, Mattia Ryffel, Steven Poulakos, Maurizio Nitti, Alessia Marra, Marcel Lancelle, Chino Noris, Kenny Mitchell, Stéphane Magnenat, Jessica Falk, Marcel Marti, Miriam Tschanen, Maria Olivares, Milan Bombsch, Amit Bermano, Dat Ngo, Gerhard Röthlin, Pascal Fua, Guido Maiga, Andrea Schüpfer, Pascal Bérard, Alexandre Chapiro, Stefan Schmid, Derek Bradley, Thabo Beeler, Derek Nowrouzezahrai, Ilya Baran, Olga Sorkine-Hornung, Hans-Peter Pfister, Bernd Bickel, Yael Pritch, and Alexander Sorkine-Hornung, for their great effort and for working with me on fantastic projects.

I thank all my students, Lea Auf der Maur, Miriam Tschanen, Andrea Schüpfer, Michael Hutzli, Marcel Marti, Tizian Zeltner, Milan Bombsch, Isabelle Rösch, and Philippe Zindel, for their hard work and for trusting in me as their supervisor.

I owe special thanks to Stéphane Magnenat for his kind support to help me structure and revise the thesis and to Vivien Milz and Steven Poulakos for reading and correcting the manuscript. I am honored that Maurizio Nitti and Alessia Marra, both superb digital artists, embellished my projects with beautiful art and designs during the past years. More special thanks go to Martina Haefeli and Markus Portmann who were supporting me and my projects on an administrative level. I would like to thank my fellow PhD students, with whom I started this journey and who accompanied me during this wonderful time, Marios Papas, Antoine Milliez, and Kaan Yücer. They did not become co-authors but great friends.

Last but not least, I would like to express my deepest gratitude to my family, Pit, Andrea, and Silvio. Thank you for being there for me and providing a safe place to retreat. You made it all possible in the first place by supporting me all these years.

Contents

List of Figures

List of Algorithms

List of Tables

CHAPTER 1

Introduction

Figure 1.1: *A child reads a story in a book. Through Augmented Reality technology, the story's characters are awakened and the child is immersed and interacts with the narrative.*

Storytelling and Technology. Stories pervade our daily lives. They document our histories, they educate us, they entertain us, and they inspire us. As such, they have become omnipresent. Every day, we encounter stories in newspapers, books, movies, games, advertising, social media, on television, on the radio, or people simply tell us their stories. It seems, the urge to tell our stories is deep-rooted in human nature, across cultures and epochs.

Technology profoundly supports and advances storytelling. The invention of the printing press in the mid-fifteenth century propelled humankind out of the Middle Ages and ushered the modern era. Writing and copying books was not limited to scribes and monks anymore and the stories could be shared with almost everyone. Books were translated from Latin into different languages and hence made accessible to the masses. The first photographic images were taken around the mid-nineteenth century. Taking a photograph was drastically faster and easier than painting a picture. At the same time, the first device to record sound waves, the Phonautograph, was invented. Two significant new forms of storytelling were born, capturing stories in images and sounds. By the end of the nineteenth century, with the advent of cinematography, the first movie theaters were opened, introducing yet another host of storytelling using moving pictures. Scientific and technological innovation has continuously provided new metaphors and forms of storytelling.

How Storytelling Has Changed. And today, storytelling has changed yet again. The smartphone revolution and a worldwide digital connectivity, have led to a paradigm-shift in media creation since the new millennium. Smartphones, equipped with high-quality displays, cameras, microphones, speakers, and integrated high-speed Internet connection, allow us to produce a high output of media content and share it with the world in the cloud. Consumers, previously considered a mere audience, have become *prosumers*, consumers that are, as well, producing media content – text, images, audio, video – that is, they are creating their own stories. The focus is on short form content, fusing different media types, produced at a very low cost. Twitter, Facebook, YouTube, Internet forums, and blogs are the cornerstones of this new era. Nowadays, prosumers are creating stories voluntarily, profusely, and, most of all, relentlessly.

The Challenge. Although the prosumer's tools are very accessible and easy to use – possibly the simplest form of expressing a feeling is to click the *like* button – the produced content is not only short form, but also short-lived. The majority of stories are of a superficial nature and deeper, more complex emotions are lost. Profound reflection is not required or encouraged anymore and long-lasting narrative contributions are rare. At the same time, tools, such as Wikipedia, that are a host for richer, more profound information and stories compared to Twitter, Facebook, and YouTube, require the author to be much more skilled at composing and writing content and are less accessible to the masses.

Additionally, another disruption has happened in the storytelling industry, linked to the same paradigm-shift of consuming and producing media. Established traditional cinema and TV distribution processes lose ground to Internet streaming services, such as Hulu, Netflix, and Amazon Prime. Push real-time services, such as TV and radio broadcasting, are becoming obsolete, and audiences choose to consume content in the cloud. From the cloud, the stories can be experienced anywhere and, most importantly, anytime, as opposed to adhering to strict broadcasting schedules. Video-on-demand availability is reducing worldwide box office grosses, DVD and Blu-ray Disk sales, and threatens large storytelling and media content production corporations such as The Walt Disney Company, Sony, and 21st Century Fox. Hollywood is slowly adapting its business model, shifting its focus away from content production and towards technological innovation. At the core, new storytelling forms, metaphors, and tools are required urgently.

Augmented Reality. There exists a wide range of scientific and technological innovations that have the potential to support and enhance storytelling. Promising technologies are Augmented Reality (AR) [Azuma et al., 2001] and Virtual Reality (VR) [Steed, 1993]. Current AR technology employs novel displays, such as smartphones, tablets, or see-through Head Mounted Displays (HMDs), such as Microsoft's HoloLens, and renders virtual content onto the display seemingly integrated into the real world around the user. This is possible by tracking the position and pose of the display using various sensors, such as cameras, GPS, accelerometers, gyroscopes, and transforming the virtual content accordingly. On smartphones and tablets, the rear camera of the device captures the environment, which is then shown on the display, supplemented with virtual content. AR allows one bridge between real-world activities and digital enhancements, and immerses the user into a virtual world, illustrated in Figure 1.1. In VR applications, in contrast, virtual content only is presented to the user, but the display's pose and orientation is still tracked. The user is completely immersed in a digital virtual world, but he or she can look around and may even move around as in the real world, illustrated in Figure 1.2. In summary, AR applications bring the virtual world to the user while in VR applications, the user is brought into the virtual world.

The significant advantage of AR is that the experience can be easily shared with other people. This makes AR applications more social than VR applications. Multiple users, carrying mobile devices or wearing see-through HMDs and sharing the same physical space, can engage together in a social AR activity, for instance exploring a museum or playing a game in a room or even in a city. While it is technically possible to create multiplayer VR experiences, such applications are currently rare as they require elaborate full body tracking of each user to virtually embed them into the virtual world.

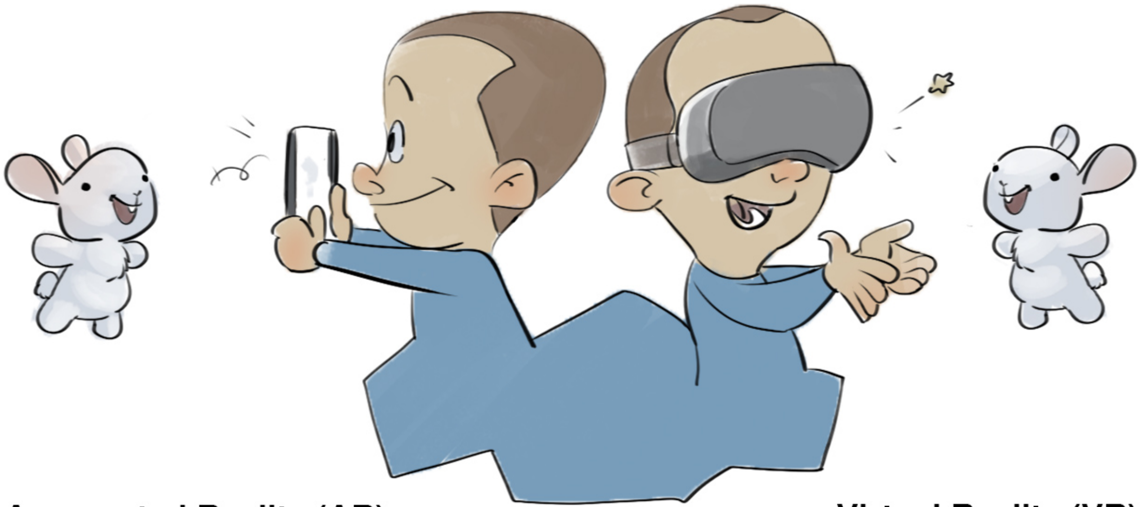


Figure 1.2: *In AR, a virtual world is brought to the user. In VR the user is brought into a virtual world.*

AR and VR technology have recently celebrated tremendous success. Only in 2015, almost 700 million USD[1] were invested in companies that specialize in developing AR and VR technologies. The most recent and vastly popular AR game, Pokémon Go, was downloaded an estimated 10 million times in the United States within a week of release and within a month it exceeded 100 million downloads worldwide. AR and VR are ideal partners to storytelling as they have the potential to immerse the user into the story in unparalleled ways and provide intuitive free-form interaction possibilities.

Our Vision. People lack a social environment including tools that encourages crafting and experiencing meaningful, rich, and lasting stories together. By providing tools to harness the prosumer's sheer abundance of creativity and productivity, we aim to help organize, shape, and improve storytelling. Our goal is to develop computer-assisted collaborative and interactive

[1] http://www.digi-capital.com/news/2016/01/arvr-investment-in-2015-breaks-out-near-700-million

story authoring tools that allow novice users to craft rich stories. Such tools democratize story authoring, as the art will no longer be confined to experts. Furthermore, using AR technology, we aim at establishing a creative environment that supports social, playful and interactive story experiences. They can easily be shared with multiple users and the intuitive and instinctive interaction possibilities provided by AR open up a new host of storytelling forms for stories created by prosumers, as well as by the industry. In essence, **our vision is to leverage digital technology to craft and experience stories together.**

Thesis Overview. Targeting this vision, we explore technologies focusing on collaborative and interactive storytelling as well as storytelling through AR in this thesis. The thesis is structured into three principal domains: story authoring, AR technology, and applications.

The next chapter, Chapter 2, presents prior art in the fields of story authoring and AR. Afterward, each of the three domains is discussed in detail:

First, in the **story authoring** domain, the core of AR storytelling is presented: a distributed, collaborative story authoring system that focuses on story version control as well as graphical visualization techniques to enhance collaboration among authors. A framework for authoring interactive AR narratives demonstrates how computer-assisted authoring enables content creators to easily author complex, branching narratives with multiple story arcs. Both frameworks aim at making story authoring more accessible to a general public. The story authoring domain is discussed in Chapter 3.

The second and the third domain examine rendering, presence, interactivity, and creativity topics related to shared AR experiences. In the **AR technology** domain, Chapter 4, we demonstrate how rendering effects influence the user experience in mobile AR games. We focus on evaluating responses to a selection of synthesized camera-oriented effects, such as motion blur, defocus blur, latency and lighting responsiveness. Our work on live texturing AR characters from colored drawings explores the idea of an AR coloring book and provides a 2D to 3D texture transfer process, a deformable surface tracking method, and validations for both. From a storytelling perspective, the coloring book application represents a use-case for user-created story content, such as characters, objects, and environments, while our story authoring tools focus on creating dramatic structure.

The application domain is discussed in Chapter 5. We propose the concept of Augmented Creativity as employing AR on modern mobile devices to enhance creative activities. Several prototype applications are presented that support creativity, education, interaction, and content creation. Among the applications, a prototype city-wide trading game, Gnome Trader, is discussed in more details.

Finally, Chapter 6 concludes the thesis by discussing the contributions and outlines potential future work.

1.1 Contributions

In this section, an introduction to each topic in the three domains, story authoring, AR technology, and applications, is presented and the core contributions are briefly outlined.

1.1.1 Story Authoring

The story authoring chapter discusses two core challenges in story authoring: authoring stories collaboratively using novel tools and graphical visualizations and authoring interactive narratives in an efficient fashion with the aid of computer-assisted authoring tools.

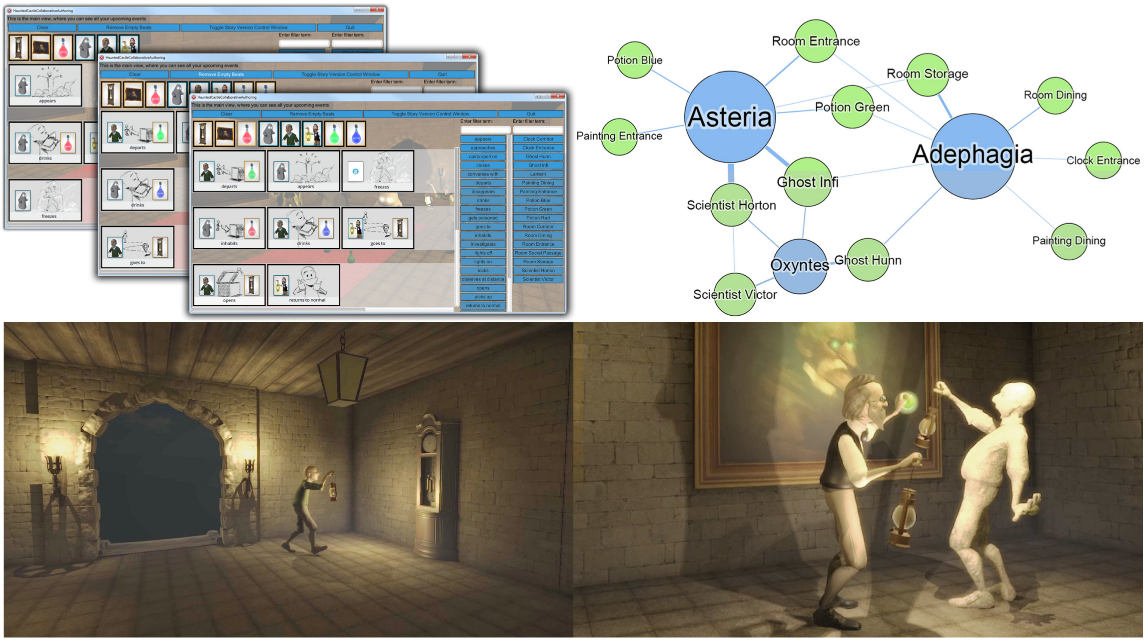


Figure 1.3: *Collaborative story authoring in the visual editor, graph-based story visualization, and playing back an animated haunted castle story.*

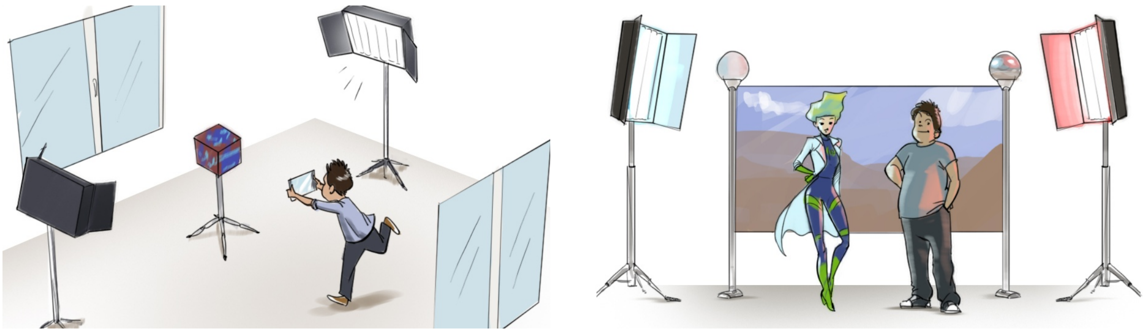
First, tackling the challenge of collaborative story authoring, we present a Story Version Control (SVC) and graphical visualization framework, depicted in Figure 1.3. We propose a media-agnostic story representation based on story beats, events, and participants that describes the flow of events in a storyline. Tree edit distance operations for this representation are developed and used to build the core features for story version control, including visual diff, conflict detection, and conflict resolution using three-way merge. Our system allows authors to work independently on the same story while providing the ability to automatically synchronize their efforts and resolve conflicts that may arise. The collaborative authoring process is further enhanced by using visualizations derived from the version control database that visually encode relationships between authors, characters, and story elements, during the evolution of the narrative. We demonstrate the efficacy of our system by integrating it within an existing visual storyboarding tool for authoring animated stories, and additionally use it to collaboratively author stories using video and images. We evaluate the usability of our system though two user studies. Our results reveal that untrained users are able to use and benefit from our system. Additionally, users are able to correctly interpret the graphical visualizations and perceive it to benefit collaboration during the story authoring process.

Figure 1.4: *Computer-assisted authoring tool for interactive narratives. The player can freely interact with characters in the authored story to dictate story progression. In this example story, spawning bees wreaks havoc on the bears and the player must use flowers to distract the bees.*

Second, focusing on authoring interactive narratives, we explore new authoring paradigms and computer-assisted authoring tools. We present a new design formalism, Interactive Behavior Trees (IBTs), which decouples the monitoring of user input, the narrative, and how the user may influence the story outcome. Automation tools for IBTs help the author detect and automatically resolve inconsistencies in the authored narrative, or conflicting user interactions that may hinder story progression. We compare IBTs to traditional story graph representations and show that our formalism better scales with the number of story arcs, and the degree and granularity of user input. The authoring time is further reduced with the help of automation, and errors are completely avoided. Our approach enables content creators to easily author complex, branching narratives with multiple story arcs in a modular, extensible fashion while empowering players with the agency to freely interact with the characters in the story and the world they inhabit. An example interactive story is depicted in Figure 1.4.

1.1.2 Augmented Reality

In the AR domain, two topics are discussed. First, we investigate the influence of motion effects in AR applications, and second, we present a coloring book application that tracks deformable book pages and transfers the color of 2D drawings to 3D models.



(a) Latency and blur experiment setup **(b)** Lighting responsiveness experiment setup

Figure 1.5: *Influence of motion effects in AR applications experiments. (a) The participant moves around a cubical marker in the latency and blur experiments. (b): A virtual character with realistic environment lighting is integrated into a group photo in the lighting experiment.*

To capture the influence of motion effects on user experience and task performance, our work focuses on evaluating responses to a selection of synthesized camera-oriented AR mixing techniques, such as motion blur, defocus blur, latency, and lighting responsiveness. In our cross section of experiments, shown in Figure 1.5, we observe that these measures have a significant impact on perceived realism, where aesthetic quality is valued. However, lower latency indicates the strongest correlation with improved subjective enjoyment, satisfaction, and realism, and objective scoring performance. We conclude that the reality mixing techniques employed are not significant in the overall user experience of a mobile AR game, except where harmonious or convincing blended AR image quality is consciously desired by the participants.

We present an AR coloring book app in which children color characters in a printed coloring book and inspect their work using a mobile device. The drawing is detected and tracked, and the video stream is augmented with an animated 3D version of the character, which is textured according to the child's coloring. This is possible thanks to several novel technical contributions. We present a texturing process that applies the captured texture from a 2D colored drawing to both the visible and occluded regions of a 3D character in real time, depicted in Figure 1.6. A deformable surface tracking method is developed, which is designed for colored drawings that uses a new outlier rejection algorithm for real-time tracking and surface deformation recovery. A content creation pipeline to efficiently create the 2D and 3D content is discussed. And, finally, we validate our work with two user studies that examine the quality of our texturing algorithm and the overall app experience.

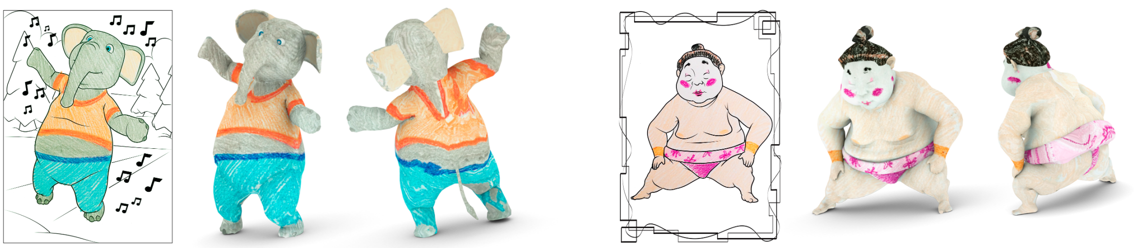


Figure 1.6: *Two examples of our AR coloring book algorithm showing the colored input drawings and the captured texture applied to both visible and occluded regions of the corresponding 3D characters.*

1.1.3 Applications

AR holds unique and promising potential to bridge between real-world activities and digital experiences, allowing users to engage their imagination and boost their creativity. In the applications chapter, we present the concept of Augmented Creativity, as employing AR on modern mobile devices to enhance real-world creative activities, support education, and open new interaction possibilities.

We present nine prototype applications in the domains of *seeing*, *hearing*, *strategizing*, *exploring*, *imagining*, and *learning*, that explore and develop Augmented Creativity in different ways, cultivating creativity through AR interactivity. Our puzzle apps enhance puzzles with computer-generated animation to facilitate the puzzle solving process. Our music app provides a tangible way for children to explore different music styles and instruments in order to arrange their own versions of popular songs. We examine how Augmented Creativity can provide a more compelling way to understand and teach complex concepts, such as computer programming. In the gaming domain, we show how to transform passive game interaction into active real-world movement that requires coordination and cooperation between players. Figure 1.7 depicts a selection of applications.

Furthermore, we discuss in more depth how AR can be applied to city-wide gaming concepts. We explore a location-based game concept that encourages real-world interactions and gamifies daily commuting activities. We create an immersive, pervasive trading game called *Gnome Trader* where multiple players engage with the game by physically traveling to predefined locations in the city and trading resources with virtual gnomes. Two screenshots are depicted in Figure 1.8. As the virtual market is a crucial component of the game, we take special care to analyze various economic game mechanics. We explore the pa-

Figure 1.7: *Augmented Creativity: cultivating creativity through* AR *interactivity. From left to right: puzzle app, music app, physical-interaction game, learning programming with robots app.*

rameter space of different economic models using a simulation of the virtual economy. The overall gameplay as well as the technical functionality is evaluated through several play tests.

Figure 1.8: *Gnome Trader game. The player trades resources with virtual gnomes integrated in newspaper boxes throughout the city.*

1.2 Publications

This thesis is based on the following accepted peer-reviewed publications:

[Zünd et al., 2016] F. Zünd, M. Tschanen, M. Lancelle, M. Olivares, M. Bombsch, M. Gross, and R. W. Sumner. Gnome trader: A location-based augmented reality trading game. In 9th International Conference on Game and Entertainment Technologies (GET). 2016.

[Magnenat et al., 2015] S. Magnenat, D. Ngo, F. Zünd, M. Ryffel, G. Noris, G. Rothlin, A. Marra, M. Nitti, P. Fua, M. Gross, and R. Sumner. Live texturing of augmented reality characters from colored drawings. Visualization and Computer Graphics, IEEE Transactions on, 21(11):1201–1210, Nov 2015.

[Kapadia et al., 2015a] M. Kapadia, J. Falk, F. Zünd, M. Marti, R. W. Sumner, and M. Gross. Computer-assisted authoring of interactive narratives. In Proceedings of the 19th Symposium on Interactive 3D Graphics and Games, i3D '15, pages 85–92. ACM, New York, NY, USA, 2015.

[Zünd et al., 2015b] F. Zünd, M. Ryffel, S. Magnenat, A. Marra, M. Nitti, M. Kapadia, G. Noris, K. Mitchell, M. Gross, and R. W. Sumner. Augmented creativity: Bridging the real and virtual worlds to enhance creative play. In SIGGRAPH Asia 2015 Mobile Graphics and Interactive applications, SA '15, pages 21:1–21:7. ACM, New York, NY, USA, 2015.

[Zünd et al., 2014] F. Zünd, M. Lancelle, M. Ryffel, R. W. Sumner, K. Mitchell, and M. Gross. Influence of animated reality mixing techniques on user experience. In Proceedings of the Seventh International Conference on Motion in Games, MIG '14, pages 125–132. ACM, New York, NY, USA, 2014.

Additionally, during the time period of this thesis, the following technical peer-reviewed papers were published:

[Poulakos et al., 2016] S. Poulakos, M. Kapadia, G. M. Maiga, F. Zünd, M. Gross, and R. W. Sumner. Evaluating accessible graphical interfaces for building storyworlds. In Interactive Storytelling: 8th International Conference on Interactive Digital Storytelling, ICIDS 2015, Los Angeles, California, November 15 - 18, 2016, Proceedings. Springer International Publishing, 2016.

[Poulakos et al., 2015] S. Poulakos, M. Kapadia, A. Schüpfer, F. Zünd, R. Sumner, and M. Gross. Towards an accessible interface for story world building. In AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. 2015.

[Zünd et al., 2015a] F. Zünd, P. Bérard, A. Chapiro, S. Schmid, M. Ryffel, M. Gross, A. H. Bermano, and R. W. Sumner. Unfolding the 8-bit era. In Proceedings of the 12th European Conference on Visual Media Production, CVMP '15, pages 9:1–9:10. ACM, New York, NY, USA, 2015.

[Kapadia et al., 2015b] M. Kapadia, F. Zünd, J. Falk, M. Marti, R. W. Sumner, and M. Gross. Evaluating the authoring complexity of interactive narratives with interactive behaviour trees. In Foundations of Digital Games, FDG'15. 2015.

[Bermano et al., 2014] A. H. Bermano, D. Bradley, T. Beeler, F. Zünd, D. Nowrouzezahrai, I. Baran, O. Sorkine-Hornung, H. Pfister, R. W. Sumner, B. Bickel, and M. Gross. Facial performance enhancement using dynamic shape space analysis. ACM Trans. Graph., 33(2):13:1–13:12, April 2014.

[Zünd et al., 2013] F. Zünd, Y. Pritch, A. Sorkine-Hornung, S. Mangold, and T. Gross. Content-aware compression using saliency-driven image retargeting. In Image Processing (ICIP), 2013 20th IEEE International Conference on, pages 1845–1849. Sept 2013.

CHAPTER 2

Related Work

This chapter introduces previous art for the three principal domains, story authoring, Augmented Reality (AR) technology, and their applications. Section 2.1 presents an introduction to the topic of storytelling and technology. The remaining relevant work is presented in the order in which it appears in this thesis.

2.1 Technology and Storytelling

The Virtual Showcase [Bimber et al., 2001] is an early AR platform focusing specifically on digital storytelling components. The showcase is targeted at museums and displays stereoscopic virtual content together with physical artifacts to multiple head-tracked simultaneous users. This is enabled through half-silvered mirror beam splitters to display the virtual content and projectors to illuminate the physical content on a per-pixel basis. To facilitate storytelling, the platform implements content generation, authoring, presentation, interaction, and content management components.

Alexander and Levine discuss how storytelling has evolved since the advent of Web 2.0 technologies [Alexander and

Levine, 2008]. Focusing on social and microcontent-driven software, Web 2.0 technology enables users to gradually – through small iterative contributions – and collaboratively create stories and story worlds using blogs, wikis, podcasts, MySpace, Flickr, Twitter, and YouTube. The authors explore fifty-seven Web 2.0 tools for storytelling and discuss how the Web 2.0 can be applied to higher education.

Robin presents an extensive introduction to digital storytelling and conveys how it can revolutionize classrooms [Robin, 2008]. His work discusses the historical educational uses of digital storytelling and presents a new pedagogical framework for it.

2.2 Story Authoring

We refer the readers to a comprehensive survey on narrative authoring [Riedl and Bulitko, 2013; Kapadia et al., 2013] and provide a brief review below.

From cave paintings to modern day storyboards [Hart, 2013] and comics [McCloud, 1994], telling stories with pictures has evolved to be more expressive and dynamic. Digital storytelling has come to represent a variety of multimedia formats, for instance video, still images, animation, and audio, that enable people to share personalized stories. The flexibility offered by digital formats makes it easier to experiment with the interaction between sequences of visual information, deriving new meaning for individual multimedia elements based on montage.

Technology also evolves to support story creation. Emergent forms of storytelling are enabled by technologies that guide the temporal ordering of shots, sequences and scenes based on attributes including characters, emotions, themes and story structure [Shen et al., 2009]. Artificial intelligence, in the form of automated planners, can also support consideration of cinematic actions and scene composition of those actions [Jhala and Young, 2010] as well as maintaining consistency between shots and enable shot reuse [Piacenza et al., 2011].

Interactive narrative systems are growing into applications in education, training and entertainment [Riedl and Bulitko, 2013]. Visual authoring tools such as CANVAS [Kapadia et al., 2016a; 2016b] leverage computer-assistance to visually author and synthesize multi-character animations from sparsely specified narrative events.

2.2.1 Collaborative Story Authoring

The continued evolution of web technologies has enabled more participatory and collaborative forms of content production. Early observations of collaborative writing in scientific communities have observed that the process of collaborative writing is a dynamic process with continuous negotiations related to both the written content as well as roles and responsibilities between the coauthors [Beck, 1993]. Technical as well as ethical and copyright challenges must be addressed [Mu et al., 2015].

Our collaborative storytelling system, Story Version Control (SVC), aims to enable authors to specify a sequence of narratively significant elements of multimedia content to generate stories in a media-agnostic way. At the same time, it supports the collaborative process by facilitating awareness of story contents and author participation.

Version Control Systems. Version Control Systems (VCSs) aim at maintaining a revision history and at facilitating and supporting collaboration between users. We refer the reader to the survey paper on model versioning approaches [Altmanninger et al., 2009], which describes various version control models and approaches in detail. Like modern VCSs used for code versioning, such as SVN[1] and Git[2], SVC implements an optimistic control mechanism with three-way merging.

Various VCSs for other domains have been developed in recent years. MeshGit [Denning and Pellacini, 2013] and skWiki [Zhao et al., 2014] are prime examples for non-traditional systems, they implement version control for editing meshes and for editing multimedia projects, respectively. At the core, they define an edit distance in that particular domain and provide version control functionality such as, checkout, commit, and update to the user, that operates on the repository based on this edit distance. MeshGit defines the mesh edit distance as a measure of dissimilarity between meshes, which can be use to transform a mesh into another mesh. skWiki expresses the edit distance between two media as a set of transformation operations, represented in domain specific language.

We introduce a version control system in the domain of stories. In contrast to MeshGit and skWiki, it defines an abstract domain representation for stories and remains media-agnostic such that any form of story, animation-based, video-based, comic-based, is compatible with the system.

Story Visualization. Humans perceive visual attributes very well, which motivates the mapping of different data to visual attributes such as color, size, and proximity [Mazza, 2009]. Information visualization is a cognitive activity facilitated by visual representations, which make it possible to explore relationships, to confirm hypothesis on data relationships, and to aid the construction of cognitive models [Mazza, 2009]. For instance, optimized storyline visualizations can be interactively inspected to understand the dynamic relationships between entities in a story [Liu et al., 2013b]. Our aim is to leverage our data representations and customize the graphical visualization capabilities of tools such as Gephi[3] to provide intuitive rep-

[1] https://subversion.apache.org/ [2] https://git-scm.com/ [3] https://gephi.org/

resentations of information about the collaborative authoring process.

Comparison to Prior Work. SVC complements existing work in digital story authoring and computational narrative by providing the tools for story authors to seamlessly and efficiently collaborate. This is accomplished using two important novel contributions: (1) A version-control system for stories that relies on a media-agnostic representation of stories, (2) Information visualization tools to help author glean meaningful information from multi-user story authoring sessions, in an effort to understand how the content of a story evolves, and how authors collaborate.

2.2.2 Authoring Interactive Stories

Scripted approaches [Loyall, 1997] describe behaviors as predefined sequences of actions where small changes often require far-reaching modifications of monolithic scripts. Approaches such as Improv [Perlin and Goldberg, 1996] and LIVE [Menou, 2001] describe behaviors as rules which govern how actors act based on certain conditions. These systems produce predefined behaviors corresponding to the current situation, and are not designed to generate complicated agent interactions with narrative significance. Facade [Mateas and Stern, 2003] executes authored beats to manage the intensity of the story and uses a generalized scripting language [Mateas and Stern, 2004] for manually authoring character interactions by encoding their preconditions for successful execution.

Story Graphs [Gordon et al., 2004] accommodate user interaction as discrete choices at key points in the authored narrative. Behavior Trees (BTs) are gaining popularity in the computer gaming industry for designing the artificial intelligence logic for non-player characters [Isla, 2005]. BTs offer graphical constructs for authoring modular, extensible behaviors, which can be extended to control multiple interacting characters [Shoulson et al., 2014]. For communication between nodes, BTs rely on a blackboard [Millington and Funge, 2009], which is a centralized, flat repository of data that can be accessed by nodes in the tree.

Automated Narrative. The use of domain-independent planners [Sacerdoti, 1975] is a promising direction for automated narrative synthesis, but requires the specification of domain knowledge. The work in [Kapadia et al., 2011b; 2011a] synthesizes complex multi-actor interactions while conforming to narrative constraints, which cannot be dynamically changed to accommodate user input. Narrative mediation systems [Riedl and Young, 2006] build on top of traditional story graph representations by considering the ramifications of possible user interactions, and automatically synthesizing sub stories to accommodate them. This produces story graphs with high branching that are difficult to edit by humans.

Virtual directors or drama managers [Magerko et al., 2004] are responsible for steering agents towards pre-determined narrative goals [Weyhrauch, 1997] while accommodating user input. Thespian [Si et al., 2005] uses decision-theoretic agents to create actors with social awareness, while PaSSAGE [Thue et al., 2007] guides a player through predefined encounters based on the system's estimation of the player's ideal experience. Event-centric planning [Shoulson et al., 2013a; 2013b] plans in the space of pre-authored narratively significant interactions, thus mitigating the combinatorial explosion of planning in the action space of individual character actions.

Comparison to Prior Work. Intelligent systems monitor the fictional world and intervene to drive the narrative forward, thus effectively replacing the human author. Since the underlying narrative still lies within traditional branching representations, it becomes intractable for the human author to iteratively build upon or replace automatically generated content.

Previous work either provides complete authorial control or relies on automation to synthesize emergent stories with minimal specification. In contrast, our goal is to empower content creators to author compelling free-form interactive narratives, by leveraging automation to facilitate the creative process instead of replacing it.

2.3 Augmented Reality

AR features in an increasing number of mobile games and interactive entertainment applications. In visual terms, the presence of real and virtual objects that coexist in the same space [Azuma, 1997] is conveyed through the seamless blending of the camera image with rendered computer graphics. Further, the realistic depiction of motion in interactive AR is affected by various reality mixing measures including matched motion blur, reduced latency, and responsiveness to changes in the camera image.

2.3.1 Augmented Reality Mixing Techniques

Many studies have analyzed the results of user experiments to assess cognition, perception, task performance and collaboration in AR [Swan(II) and Gabbard, 2005]. More recently, a collection of investigations from a visual observation and interpretation standpoint has been surveyed [Kruijff et al., 2010]. Whilst here, we target our investigations upon the influence on the overall user experience for interactive entertainment applications in mobile AR.

Camera Effects. The early work by Fischer et al. [Fischer et al., 2006] and a more comprehensive work by Klein and Murray [Klein and Murray, 2010], dealing with compositing camera effects within virtual rendered content, are most relevant for our experiments . They address the visually most important effects such as vignetting, noise, chromatic aberration, and blur caused by imperfect imaging and camera motion. As in our experiments, Park et al. [Park et al., 2012] use a video see through AR framework with mobile devices.

Instead of artificially decreasing the quality of virtual content, another approach is to improve the camera image quality. Unfortunately, many of these methods are ill posed and are usually computationally too expensive to run in real time on current mobile devices. Blur kernels in images are often estimated for subsequent deconvolution, i.e. restoration. A number of methods exists that handle only motion blur [Oh and Kim, 2014] or defocus blur [Tao et al., 2013]. Others do a joint estimation [Oliveira et al., 2014], but all of these image based techniques are computationally expensive. Other options include estimation of the kernel with an IMU with accelerometers or gyroscopes rigidly attached to the sensor [Joshi et al., 2010], [Bae et al., 2013]. We applied this idea to estimate the required motion blur for the virtual content.

Environment Lightning. For consistent material environment shading, a mirror sphere visible in the camera image can be used to extract lighting information [Kanbara and Yokoya, 2002] in real-time and render materials in a realistic way [Agusanto et al., 2003]. These methods also handle the white balance and color matched illumination to some degree. We follow this approach in our dynamic light environment experiment, except we use a diffuse sphere to effectively sample the optically preconvolved diffuse materials, which results in a reduced computation reflection mapping for low-powered mobile devices.

Psychological Effects. Sharan et al. [Sharan et al., 2013] show in experiments that motion blur in a racing game on a video game console is preferred by the users but has no influence on task performance. In a similar way, a psychological study in [Knez and Niedenthal, 2008] examined the influence of different in-game lighting conditions to player performance and feelings. They concluded that their participants solved maze levels in a first person shooter game fastest and best in warm (reddish) as compared to cold (blueish) in-game lighting condition. However, here we are concerned with the satisfaction, enjoyment and perceived realism of the dynamic effects of blur, latency and lighting for mobile AR games.

Display Latency. To measure the total system latency Jacobs et al. [Jacobs et al., 1997] use a blinking LED and its image displayed on the screen to measure their temporal offset with photo transistors and an oscilloscope. Friston and Steed [Friston and Steed, 2014] present and compare a number of latency measurements for VR environments. The delay from a known motion to the display reaction includes tracking, computation and display latencies. In our test cases, it is not necessary to measure motion. Instead, we adapt the two approaches and use a camera to record a modulated light source directly and on the screen at the same time.

Especially for Head Mounted Displays (HMDs), the effect of latency is critical for the user's comfort. Carmack [Carmack, 2013] describes a number of strategies to reduce lag. Sielhorst et al. [Sielhorst et al., 2007] measure latency of an AR see-through device. They assume that latency is a crucial factor for a user's task performance, but do not perform a user study to verify this statement.

2.3.2 Augmented Reality Coloring Book

Clark and Dünser [Clark and Dunser, 2012] explore the use of colored drawings to texture AR elements. Their work uses a pop-up book metaphor, providing animated augmented pop-ups that are textured using the drawing's colors. Their paper also shows two 3D models colored according to the drawings, and a commercial product derived from this work called colAR[4] shows many more. However, the paper does not include details of the texturing process. In addition to colAR, three other coloring book products, Crayola Color Alive, Paint My Cat and Disney's Color and Play[5], use AR in the context of coloring. Although no details of the texturing process are provided, we suspect that these products use manually-edited correspondences between triangles in UV space and drawing space. The use case for these products targets line art printed at home, which avoids the issue of curved pages due to a book's binding. For example, the colAR FAQ urges users to ensure that the printed pages are "lying flat". On the contrary, we support augmentation on deformed surfaces and propose an efficient content creation pipeline that provides a mostly automatic method to generate appropriate UV mappings. In addition, we describe quantitative user studies which contribute to the anecdotal observations offered by Clark and Dünser [Clark and Dunser, 2012].

Texture Generation for 3D Meshes. In the context of our app, a 2D colored drawing provides information about the texture of the portions of the 3D model that are visible in the drawing. Determining the texture for the occluded regions involves filling in the remaining portions of the model's texture map, which is an *inpainting* problem. Inpainting addresses the task of filling a portion of an image from other parts of the image. Methods can be split into two categories: diffusion-based and patch-based [Guillemot and Le Meur, 2014]. The former consist of propagating local information around the boundary between the known and the unknown parts of the image, under smoothness constraints. These methods work well for filling small unknown regions surrounded by known regions. However, they require many iterations and may exhibit artifacts when filling larger regions [Guillemot and Le Meur, 2014]. Methods in the second category copy patches from the known regions to

[4] http://colarapp.com/ [5] http://www.crayola.com/qr/package/2015/coloralive; http://www.paintmyzoo.com; http://www.colorandplaybooks.com/

the unknown ones until the unknown regions are filled completely. The order of filling is critical for the visual result [Criminisi et al., 2004], but even the best filling order can lead to discrepancies, especially when an unknown region lies in the middle of a known region [Liu and Caselles, 2013]. Indeed, in that case there will likely be non-matching patches in the center of the unknown region, creating visual artifacts. Recently, techniques such as graph cuts have been applied to alleviate this problem. The resulting algorithm produces good visual results but takes about one minute for a typical image [Liu and Caselles, 2013]. Recent work on video inpainting [Herling and Broll, 2014] achieves real-time performance, but uses a desktop processor and fills only a small area of the image. A modern approach exploits the fact that local structures are typically repeated in the image and therefore structural priors can be captured and used for reconstruction [Liu et al., 2013a]. These global methods work well for filling many small holes, but are not designed to fill larger areas.

Although these methods work well for image processing applications, they are not designed for real-time performance on mobile devices. In the context of texturing meshes from colored drawings, the critical element is that the generated texture is continuous across silhouette boundaries and texture seams. Therefore, we express texture generation as a static problem whose aim is to create a mapping, for every point of the texture, to a point in the drawing. Our proposed algorithm is inspired by both diffusion-based and patch-based methods, as it both extends the known parts of the image to the unknown ones, and copies pixels know regions to unknown ones.

Deformable Surface Tracking. Reconstructing the 3D shape of a non-rigid surface from monocular images is an under-constrained problem, even when a reference image of the surface in its known rest shape is available. This is the problem we address in the context of live texturing from colored drawings, as opposed to recovering the shape from sequences as in many recent monocular Non-Rigid Structure from Motion methods such as [Fayad et al., 2010; Garg et al., 2013].

Given correspondences between a reference image and a live input image, one can compute a 2D warp between the images and infer a 3D shape from it, assuming the surface deforms isometrically [Bartoli et al., 2012; Chhatkuli et al., 2014]. The reconstruction has the potential to run in real time because it is done in closed form and point-wise or by linear least-squares. However, the accuracy of the recovered shape is affected by the quality of the 2D warp, which does not take into account the 3D deformation properties of the surface. In our coloring book application, this problem is more severe because a large part of the surface is homogeneously blank, making 2D image warping imprecise.

An alternative is to go directly from correspondences to a 3D shape by solving an ill-conditioned linear-system [Salzmann and Fua, 2010], which requires the introduction of additional constraints to make it well-posed. The most popular added constraints involve preserving Euclidean or Geodesic distances as the surface deforms, which is enforced either by solving a convex optimization problem [Brunet et al., 2010; Perriollat et al., 2011; Salzmann and Fua, 2011] or by solving sets of quadratic equations [Salzmann et al., 2008; Moreno-Noguer et al., 2009] in closed form. The latter method is typically implemented by linearization, which results in very large systems and is no faster than minimizing a convex objective function, as is done in [Salzmann and Fua, 2011].

The complexity of the problem can be reduced using a dimensionality reduction technique such as principal component analysis (PCA) to create morphable models [Cootes et al., 1998; Blanz and Vetter, 1999], modal analysis [Moreno-Noguer et al., 2009], free form deformations (FFD) [Brunet et al., 2010], or 3D warps [Delbue and Bartoli, 2011]. One drawback of PCA and modal analysis is that they require either training data or specific surface properties, neither of which may be forthcoming. Another is that the modal deformations are expressed with respect to a reference shape, which must be correctly positioned. This requires the introduction of additional rotation and translation parameters into the computation, preventing its use in live AR applications.

The FFD approach [Brunet et al., 2010] avoids these difficulties and relies on parameterizing the surface in terms of control points. However, its complex formulation is quite slow to optimize as reported in [Brunet et al., 2014]. The work of Ostlund et al. [Ostlund et al., 2012] takes inspiration from the Laplacian formalism presented in [Sorkine et al., 2004] and the rotation-invariant formulation of [Sumner and Popović, 2004] to derive a rotation-invariant regularization term and a linear subspace parameterization of mesh vertices with respect to some control vertices. This technique leads to the first real-time 3D deformable surface tracking system as reported in [Ngo et al., 2015], which can run at 8–10 frames per second (FPS) on a MacBook Pro 2014 laptop. However, the high memory consumption and still heavy computation prohibit it from running in real-time on mobile devices.

To the best of our knowledge, there have been no reports so far describing a real-time deformable object tracking system on mobile devices. The presented contribution is an improvement upon previous work [Ngo et al., 2015]. We propose a new outlier rejection mechanism, reformulate the reconstruction energy function to gain speed while not sacrificing accuracy, as well as rely on frame-to-frame tracking to gain frame rate and only apply the feature detection and matching periodically to retrieve back lost tracked points and accumulate good correspondences. These together allow real-time tracking on a tablet.

2.4 Applications

We refer the reader to an exhaustive survey on AR applications, specifically focusing on gaming [Thomas, 2012].

A prominent early example of AR storytelling is the MagicBook [Billinghurst et al., 2001]. Large markers are integrated into a book's pages, which enable viewing virtual content through Virtual Reality (VR) glasses, based on which page of the book is open. The user can see only the virtual content and not the book as the VR glasses are opaque. Grasset and colleagues employ a mixed reality approach to further improve the user's experience [Grasset et al., 2008]. In their application, the user can see both the virtual content as well as the physical book. They add various visual and auditory effects to an existing illustrated book to enhance the reader's immersion. The Haunted Book [Scherrer et al., 2008] is a prime example of well-integrated AR content. The camera is mounted on a lamp on the table and the augmented book is viewed through a computer screen. Their focus lies on interaction between the virtual content and the physical book.

2.4.1 Augmented Creativity

Much work has been dedicated to understanding creativity and how it can be inspired and supported [Amabile, 1998; Runco, 2004]. A variety of research efforts aim to cultivate creativity through the help of computers and AR. In particular, many music related projects, projects targeting AR books, storytelling, as well as teaching projects exist, as presented in this section.

An early example of fostering creativity in the digital age is the work by Folkestad and colleagues, who present a long-term empirical study that analyzes and evaluates computer-supported music composition by teenagers [Folkestad, 1995]. The Music Table [Berry et al., 2003], based on the Augmented Groove project [Poupyrev et al., 2000], shows how early AR technology can support music composition. Markers act as composition operations generating MIDI events that are fed into a composition system. Compared to our music application, their markers represent operations that make changes to the composition while our makers directly represent an instrument. An advantage of our system is that it allows one to directly play the song in real-time and explore various combinations of instruments without requiring direct knowledge of music theory.

While the MagicBook as well as the Haunted Book present attractive examples of immersive AR applications, they lack some user interaction components. Hence, the content is static such that each reading of the book results in the same experience. Pushing the boundaries of interactivity, AR façade [Dow et al., 2007] is an AR version of the renowned desktop-based interactive drama, Façade. Similarly, our authoring framework for interactive narratives employs AR to overcome the limitations present in a previous authoring framework [Kapadia et al., 2011a], which can hinder creativity due to poor interaction.

Recently, AR technology has been employed and analyzed in a teaching context at schools [Wei et al., 2015; Karamanoli and Tsinakos, 2015]. In one instance, a teaching scheme for high school students to learn about the creative design processes was developed. The teaching scheme includes two AR-based teaching aids for the classroom, supporting both the teachers and the students. Their study shows that student attention and motivation improved significantly due to the introduction of AR technology.

2.4.2 City-Wide Augmented Reality Trading Game

Creating a stable virtual economic model is a non-trivial process because the distribution and temporal progression of prices can greatly affect gameplay in unforeseen ways. In fact, constructing balanced economies is an ongoing topic in the gaming industry. Market crashes or hyperinflation are common ailments that can afflict even blockbuster productions, such as Diablo 3 [Earle, 2013]. Research related to game economies has been conducted with the focus on online trading games or massively multiplayer online games (MMOs), such as Eve Online or Everquest [Reeder et al., 2008; Castronova et al., 2009]. However, virtual markets in online games often rely heavily on free trading between players and the ability to exchange in-game currency or goods for real money [Yamaguchi, 2004; Debeauvais et al., 2012].

Research in the context of virtual economies often refers to agent-based modeling, an approach that consists of computational objects that interact according to predetermined rules. This approach allows one to consider richer environments with more complex behavioral perspectives and dynamic market prices. Agent-based modeling has been employed for the financial sector by modeling asset prices [Rekik et al., 2014], for food consumption [Deguchi et al., 2001], and for the Swiss wood market [Kostadinov et al., 2014]. Most of these agent-based simulations attempt to mimic real-world markets as closely as possible, which differs from our goal of exploring the virtual market of a trading game in a more open-ended fashion. In Gnome Trader, our simulation models two core components. First, the simulation controls gnomes that represent non-player characters and do not exhibit individual needs. Second, it models player agents whose sole goal is to accumulate in-game currency by trading resources. The level of control over prices in our virtual trading game is unique and is not necessarily comparable to models of real markets. Therefore, special care must be given to the economic model used in the game design of Gnome Trader. We describe our exploration of dynamic and spatially varying pricing models that reflect the emergent behavior of real markets.

CHAPTER 3

Story Authoring

Figure 3.1: *Different story media types as employed with our Story Version Control system: (a) procedural animation system, (b) video clips, and (c) comic panels.*

3.1 Introduction

This chapter presents the story authoring domain of this thesis. First, a collaborative story authoring system that focuses on story version control as well as on graphical visualization techniques to enhance collaboration among authors is proposed in Section 3.2. Second, our framework for authoring interactive Augmented Reality (AR) narratives demonstrates how computer-assisted authoring enables content creators to easily author complex, branching narratives with multiple story arcs. Authoring interactive narratives is discussed in Section 3.3. Both works aim at assisting and supporting the story author by providing intuitive editing tools.

3.2 Story Version Control and Visualization

With today's mature connected platforms for content authoring, collaboration naturally plays a central role. Current story authoring tools, however, are not explicitly designed for collaborative authoring and suffer from two primary limitations. First, without native support for collaboration, authors must explicitly communicate with one another to manually synchronize their content and resolve conflicts. Second, understanding the overall progression of contributions and the relationships to creative content in a collaborative setting with multiple authors is nearly impossible due to inherent complexities in such a setup. As a result, patterns in authoring procedures that might enhance collaborative creativity are difficult to find.

Our research enhances collaborative story authoring by providing a Version Control System (VCS) and graphical visualization framework. Our system allows authors to work independently on the same story while providing the ability to automatically synchronize their efforts and resolve conflicts that may arise.

Additionally, we provide visualization tools that extract and visualize information from the version control database to convey the creative intent of authors, their contributions to the story over the course of its evolution, and their relationships with characters and other authors. Together, these capabilities enhance collaborative authoring and provide meaningful insights into the creative process.

We address several challenges in developing our version control and visualization framework. First, we present a media-agnostic story representation based on story beats, events, and participants that describes the flow of events in a storyline. Given this representation, we next develop a tree edit distance operation to compute the distance between two stories and the edit operations needed to transform one story to another. Given this foundation, we then build the core features for story version control, including visual diff tools, conflict detection, and conflict resolution using three-way merge operations.

In addition to developing the base functionality for story version control, we further demonstrate that the data within the version control database holds great potential to enhance the collaborative authoring process through visual representations of story evolution. We propose three types of visualizations that span story content and author participation. First, story-centric visualizations represent either character or author involvement in story progression. Second, relationship visualizations depict the connectivity between authors, characters, and story elements. For example, co-authors may observe the characters an author most frequently edits. Finally, meta-relationship visualizations characterize the correlation between authors based on their interactions with story characters or story events. Two authors whose edits frequently include the same character may have a strong meta-relationship.

We demonstrate our version control system and information visualization tools on collaboratively authored stories using an

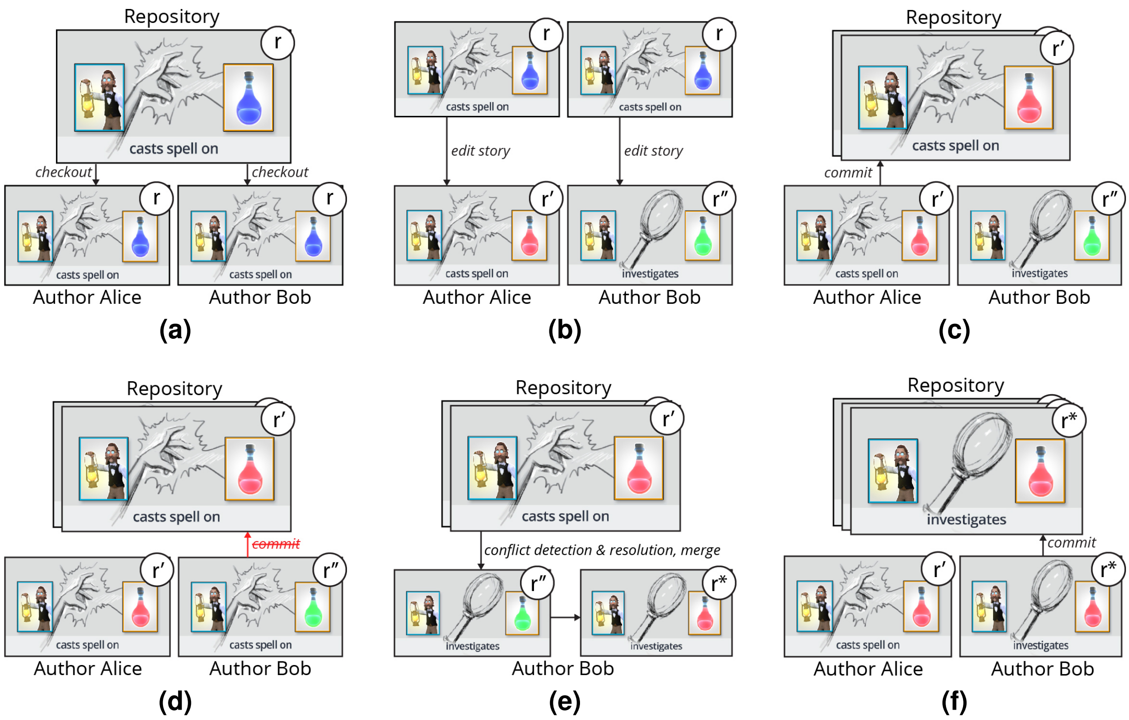


Figure 3.2: *Example scenario illustrating* SVC*'s workflow utilizing an optimistic versioning approach.*

animation synthesis engine. To show the media-agnostic nature of our method, we further show stories based on raw images and video. We validate the efficacy of our tools with a user study in which teams of authors collaboratively created novel stories. Additionally, users are able to correctly interpret visualizations to extract meaningful information and perceive it to be a useful tool to promote collaboration in storytelling.

3.2.1 Preliminaries

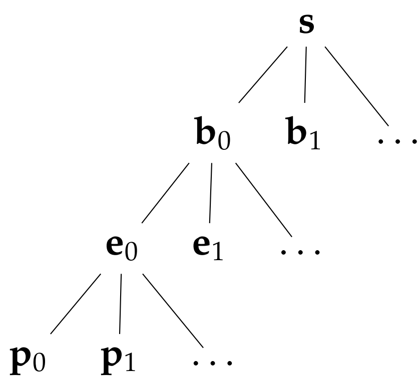
SVC utilizes the optimistic versioning paradigm [Altmanninger et al., 2009] in which a three-way merging is applied. Three-way merging is common in most of today's version control systems and allows the system to identify editing operations of authors more precisely than using raw-merge and two-way merge.

We illustrate the three-way merging authoring paradigm by an example scenario in Figure 3.2 in which authors Alice and Bob are collaboratively editing a story about the Scientist Horton who casts a spell on a blue potion. Later, the story is changed such that Horton instead investigates the blue potion. (a) The authors checkout the latest revision $\mathbf{r}$ from the repository, which will serve as their base revision. The current revision $\mathbf{r}$ contains the story *Horton casts a spell on a blue potion*. (b) The authors each make changes to the story, creating new revisions $\mathbf{r}'$ and $\mathbf{r}''$, respectively. Alice changes the *Potion Blue* to a *Potion Red*. Bob changes the *Potion Blue* to a *Potion Green* and the event from *casts a spell* to *investigates*. These changes are partially overlapping (editing the potions) and thus conflicting. (c) Alice commits her changes $\mathbf{r}'$ back to the repository. (d) Bob cannot commit his changes $\mathbf{r}''$ to the repository in the current state, as his base revision $\mathbf{r}$ is older than the current revision $\mathbf{r}'$ on the server. (e)Instead, he is required to first update and merge his story $\mathbf{r}''$ with the latest revision $\mathbf{r}'$. As a conflict is present, SVC assists Bob in resolving the conflict and merging his revision with the latest, thereby creating revision $\mathbf{r}^*$. (f) Bob can now commit his updated revision $\mathbf{r}^*$ to the repository. Afterward, Alice can update her revision $\mathbf{r}'$ conflict-free to the latest revision $\mathbf{r}^*$.

3.2.2 Story Representation

SVC assumes an abstract, media-agnostic representation of a story. A story is structured into story beats, story events, and event participants. A participant is a character or prop object that takes part in the story in one or multiple events. An event is a context-specific interactions between any number of participants, where each instance of an event can have a vastly different outcome depending on the participants. Events serve as the building blocks for authoring complex narratives. A beat combines multiple simultaneously happening events into one unit of advancement in story time. SVC implicitly assumes progression of time in the story as progression of beats. Events within a beat are unsorted, and happen in parallel.

Formally, a story $\mathbf{s}$ is represented as an ordered tree in which the story node $\mathbf{s} = \langle I, \{\mathbf{b}_0, \mathbf{b}_1, \ldots\}\rangle$ is the root node containing beat nodes $\mathbf{b}$ as ordered children. A node ID I is contained in every node. A beat node $\mathbf{b} = \langle I, \{\mathbf{e}_0, \mathbf{e}_1, \ldots\}\rangle$ contains an ordered list of event nodes. An event node $\mathbf{e} = \langle I, g, \{\mathbf{p}_0, \mathbf{p}_1, \ldots\}\rangle$ contains a signature name g, which defines the type of event, and participant nodes $\mathbf{p}$. The participant nodes $\mathbf{p} = \langle I, h\rangle$ are leaf nodes, they contain a participant ID h. As such, the structure is defined as follows:



A path $\mathbf{P_n} = \{I_0, I_1, \ldots, I_k\}$ of a node $\mathbf{n}$ is defined as the concatenation of node IDs I from the root node $\mathbf{s}$ at level 0 to node $\mathbf{n}$ at level k. The notation $\mathbf{n} \succ \mathbf{n}'$ expresses that a node $\mathbf{n}$ is a descendant and lies in the sub-tree of $\mathbf{n}'$. The node $\mathbf{n}$ is a descendant of $\mathbf{n}'$ if its path is longer $|\mathbf{P_n}| > |\mathbf{P_{n'}}|$. Finally, we use the symbol $\mathbb{S}$ to denote the space of all possible stories. By using this tree representation, we benefit from existing tree editing algorithms to compare, modify, and merge stories.

3.2.3 Story Version Control

In SVC, version-control functionality such as checkout, commit, update, and merge, is provided through tree edit operations. Tree edit operations are employed as a representation of dissimilarity between two stories and thus constitute the core of the repository. The following sections describe how tree edit operations are applied to support version control mechanisms.

Story Edit Distance

Using the Robust Tree Edit Distance (RTED) [Pawlik and Augsten, 2011], SVC calculates an injective mapping $\sigma : \mathbb{N}^+ \to \mathbb{N}^+$ that maps post-order node indices in a story $\mathbf{s} \in \mathbb{S}$ to post-order node indices in another story $\mathbf{s}' \in \mathbb{S}$. In the example story trees depicted in Figure 3.3, the mapping is $\sigma = \{(1 \mapsto 1), (3 \mapsto 2), (4 \mapsto 4)\}$.

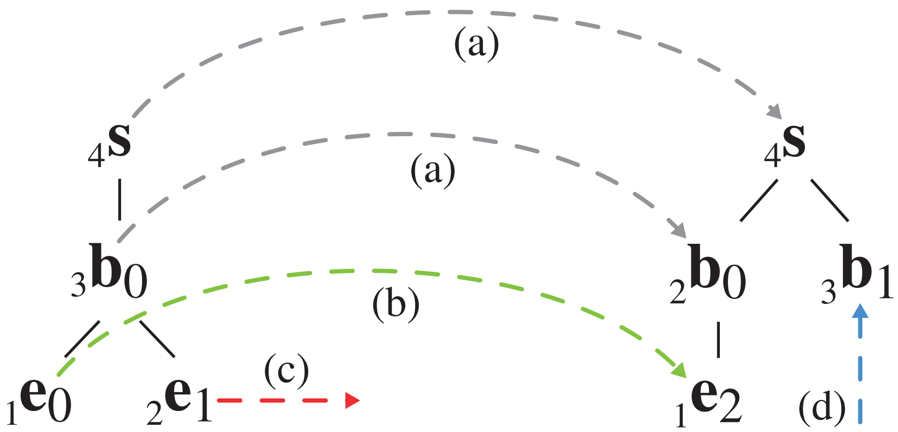


Figure 3.3: *Example mapping of two story trees to illustrate (a) ignored mappings, (b) rename operations, (c) remove operations, and (d) insert operations.*

Given a mapping, the corresponding tree operations can be synthesized as follows:

(c) For each node in $\mathbf{s}$ that is not mapped to a node in $\mathbf{s}'$, a **remove operation** is synthesized. A remove operation $o^{\text{rem}}(\mathbf{s}, \mathbf{P_n}) : \mathbb{S} \to \mathbb{S}$ removes the node $\mathbf{n} \in \mathbf{s}$ at path $\mathbf{P_n}$.

(d) For each node in $\mathbf{s}'$ that has no mapping from a node in $\mathbf{s}$, an **insert operation** is synthesized. An insert operation $o^{\text{ins}}(\mathbf{s}, \mathbf{P_n}, c, \mathbf{n}') : \mathbb{S} \to \mathbb{S}$ inserts a child node $\mathbf{n}'$ to the parent node $\mathbf{n} \in \mathbf{s}$ at path $\mathbf{P_n}$ at child index c.

(b) For each mapping between a node $\mathbf{n} \in \mathbf{s}$ and $\mathbf{n}' \in \mathbf{s}'$ a **rename operation** is synthesized if the nodes are not equal, that is, if their ID is not identical $I_\mathbf{n} \neq I_{\mathbf{n}'}$. A rename operation $o^{\text{ren}}(\mathbf{s}, \mathbf{P_n}, \mathbf{n}') : \mathbb{S} \to \mathbb{S}$ updates the node $\mathbf{n} \in \mathbf{s}$ at path $\mathbf{P_n}$ to a node $\mathbf{n}'$.

(a) Mappings between equal nodes are ignored.

These three types of operations are sufficient to transform an arbitrary story $\mathbf{s}$ into an arbitrary new story $\mathbf{s}'$ [Zhang and Shasha, 1989]. The difference between two stories $\mathit{diff}(\mathbf{s}, \mathbf{s}') = \{o_0, o_1, \ldots, o_n\}$ is a set of n operations o_i that express the atomic changes required to transform $\mathbf{s}$ into $\mathbf{s}'$.

Repository representation

The SVC repository stores and keeps track of all operations the authors have performed over time. We define a story repository $\mathbf{R} = \{\mathbf{r}_0, ..., \mathbf{r}_N\}$ as an ordered list of revisions. Inspired by [Zhao et al., 2014], a revision $\mathbf{r}_i = \langle u_i, o_i, b_i, t_i, \mathit{diff}_i \rangle$ is a tuple consisting of a unique identifier u, an owner o, a reference to a parent (base) revision b, a time stamp t, and a diff diff. The repository does not store snapshots of a story but instead accumulates the delta transformation operations. Applying the tree edit operations in a diff diff_i to the story $\mathbf{s}_{i-1}$ from the previous revision $\mathbf{r}_{i-1}$ recreate the current story $\mathbf{s}_i$. Hence, recreating a specific story $\mathbf{s}_k$ can be expressed as a function composition series $\mathbf{s}_k = \mathit{diff}_k \circ \mathit{diff}_{k-1} \circ \ldots \circ \mathit{diff}_0(\mathbf{s}_\emptyset)$.

Version Control Conflicts

In a three-way merge scenario, two users try to commit their story to the repository, both stories based on the same base revision story, as illustrated in Figure 3.2. Two diffs need to be merged, that is, the two sets of operations need to be combined. During that process, SVC detects conflicting operations and lets the authors resolve the conflict manually using the client. Two operations are conflicting under the following conditions:

Rename Conflict A rename operation $o_i^{\text{ren}}(\mathbf{s}, \mathbf{P_n}, \mathbf{n}')$ conflicts with a rename operation $o_j^{\text{ren}}(\mathbf{s}, \mathbf{P_n}, \mathbf{n}'')$ if both operations update the same node $\mathbf{n} \in \mathbf{s}$ to a different node, $\mathbf{n}' \neq \mathbf{n}''$.

Insert Conflict An insert operation $o_i^{\text{ins}}(\mathbf{s}, \mathbf{P_n}, c, \mathbf{n}')$ conflicts with an insert operation $o_j^{\text{ins}}(\mathbf{s}, \mathbf{P_n}, c, \mathbf{n}'')$ if both operations insert a different node $\mathbf{n}' \neq \mathbf{n}''$ at the same child index c at the same parent node $\mathbf{n}$.

Remove Conflict A remove operation $o_i^{\text{rem}}(\mathbf{s}, \mathbf{P_n})$ conflicts with any rename operation $o_j^{\text{ren}}(\mathbf{s}, \mathbf{P_{n'}}, \mathbf{n}^*)$ or insert operation $o_j^{\text{ins}}(\mathbf{s}, \mathbf{P_{n'}}, c, \mathbf{n}^{**})$ if $\mathbf{n}' \succ \mathbf{n}$ holds, that is, if the latter operations target a descendant node of $\mathbf{n}$.

In the example illustrated in Figure 3.2 (c), Alice's and Bob's revisions $\mathbf{r}'$ and $\mathbf{r}''$ contain operations that conflict with each other. Alice's revision contains a rename operation changing the *Potion blue* to the *Potion Red* and Bob's revision contains a rename operation changing the same *Potion Blue* to a *Potion Green* instead. The other operation in Bob's revision $\mathbf{r}''$ is a rename operation changing the event *casts spell on* to *investigates*, which does not conflict with any of Alice's operations.

System Architecture

Figure 3.4 summarizes the proposed system architecture. Story repositories are centralized and persisted on a server and can be accessed online by both story editing as well as story visualization clients.

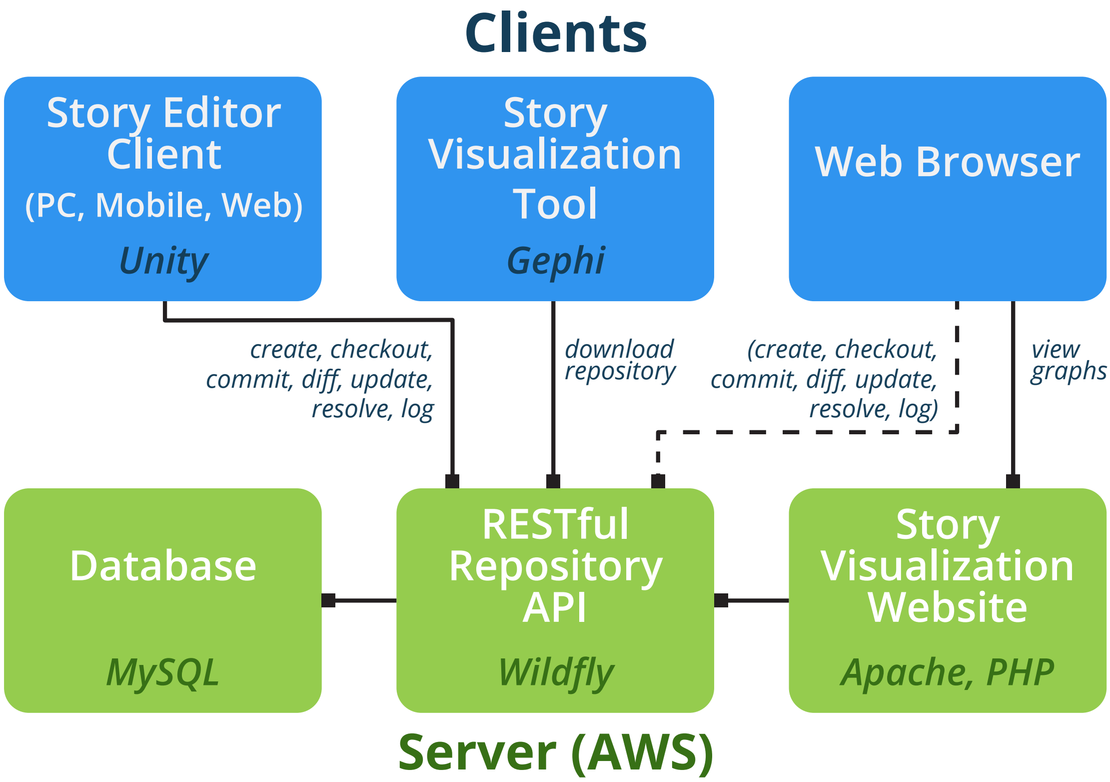


Figure 3.4: SVC *System Architecture. Story repositories are centralized and persisted on a server and can be accessed online by both story editing as well as story visualization clients.*

SVC Server. The SVC server, running on Ubuntu in the Amazon Web Services cloud[1], comprises a Wildfly[2] application that provides a RESTful API for performing operations on the repository and that transmits json-serialized repository data. All data is persisted in a local MySQL database[3]. The Wildfly server ap-

[1] https://aws.amazon.com/ [2] http://wildfly.org/ [3] http://www.mysql.com/

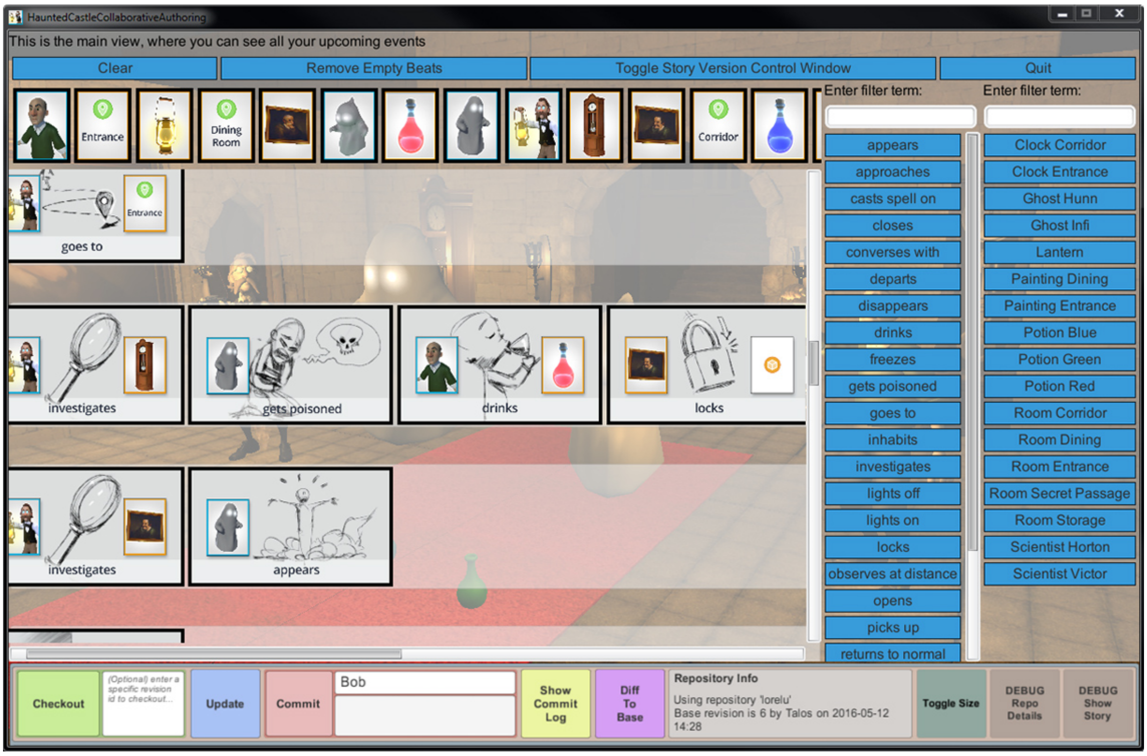


Figure 3.5: *The* SVC *Story Editor Client providing an intuitive drag-and-drop interface for authoring stories by arranging beats, events, and participants. A version control UI is integrated.*

plication provides the following JAX-RS web services: *checkout*, *commit*, *diff*, *resolve*, *update*, *log*, *full*, *create*.

The **full** request is only used by the visualization tools to download the entire repository including the operations in all revisions as well as a story snapshots for each revision. Using this information, the tools have complete freedom to visualize any aspect of the repository.

During a **checkout** request for a revision $\mathbf{r}_k$, all sets of operations (diffs) are applied to the empty story in order to synthesize the requested story $\mathbf{s}_k = \mathit{diff}_k \circ \mathit{diff}_{k-1} \circ \ldots \circ \mathit{diff}_0(\mathbf{s}_\emptyset)$.

A **diff** request calculates and returns the diff of two stories. The stories can be submitted or referred to using revision IDs.

During a **commit** request, a diff between the submitted story

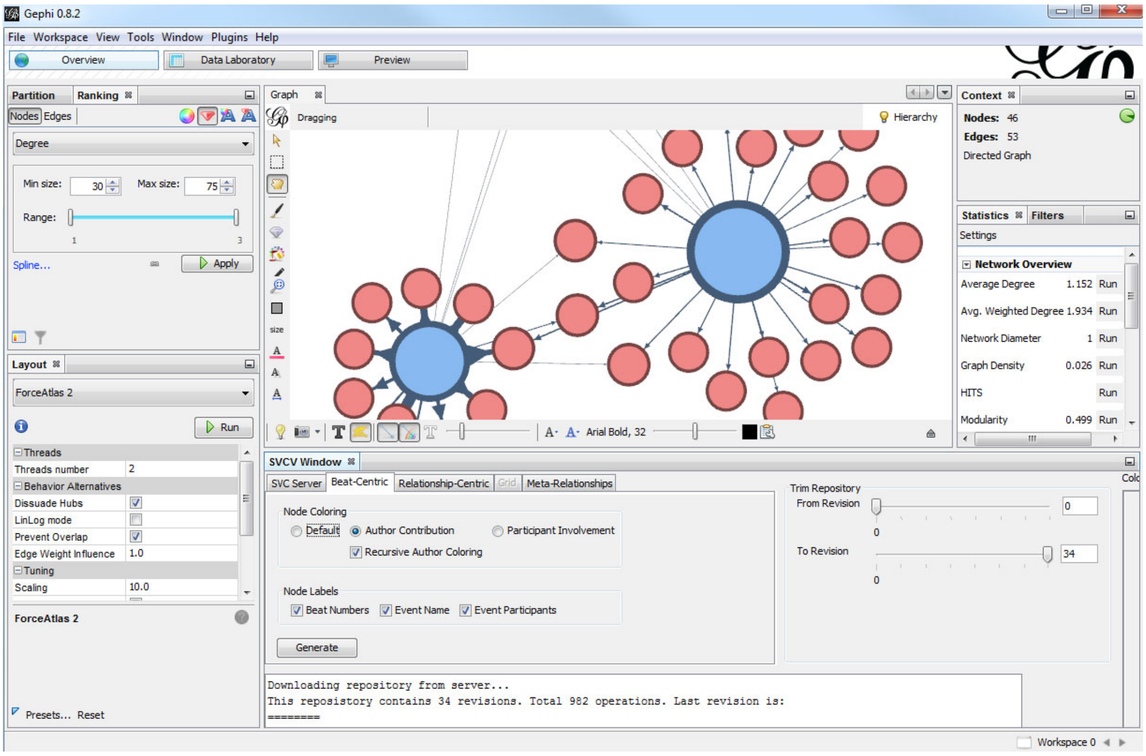


Figure 3.6: *The* SVC *Gephi-based story visualization tool with integrated* SVC *connectivity.*

and the story is calculated. The diff is then appended to the repository as a new revision.

An **update** request from the client contains the author's latest story $\mathbf{s}_{\text{author}}$ as well as the author's base revision $\mathbf{r}_{\text{base}}$ with story $\mathbf{s}_{\text{base}}$. Using the story $\mathbf{s}_{\text{head}}$ in the latest (HEAD) revision in the repository the server calculates the diffs $\mathit{diff}(\mathbf{s}_{\text{head}}, \mathbf{s}_{\text{base}})$ as well as $\mathit{diff}(\mathbf{s}_{\text{author}}, \mathbf{s}_{\text{base}})$ and calculates the conflicts thereof. If there are no conflicts, the operations in $\mathit{diff}(\mathbf{s}_{\text{head}}, \mathbf{s}_{\text{base}})$ are applied to $\mathbf{s}_{\text{author}}$ and the resulting story is returned to the client. If conflicts are present, they are sent to the client. The client presents the conflicts to the author in a visual fashion and, for each pair of conflicting operations, let the author resolve each conflict by deciding if her or the operation from the other author should be used. It is possible that operations from the same author are dependent from each other. A dependency exists between an operation $o(\mathbf{n}, \cdot)$ on node $\mathbf{n}$ and another operation $o'(\mathbf{n}', \cdot)$ on node $\mathbf{n}'$ if $o(\mathbf{n}, \cdot)$ is a rename or an insert operation and $\mathbf{n}' \succ \mathbf{n}$ holds. After each resolved conflict, the client applies the same decision for all dependent remaining conflicts. For instance, author Alice inserts a new event and a participant into a beat. These operations conflict with Bob's operation to remove the entire beat. Alice is resolving the conflicts. Her operation that inserts the participant depends on the operation that inserts the event and thus cannot be chosen independently. If Alice decides to keep her operation that inserts the participant, the client would automatically decide to keep as well the operation that inserts the event. After all conflicts have been resolved, the client sends a **resolve** request back to the server containing a decision for each conflict. The server then applies each selected operation in the conflicting operations pairs as well as the remaining non-conflicting operations in $\mathit{diff}(\mathbf{s}_{\text{head}}, \mathbf{s}_{\text{base}})$ and $\mathit{diff}(\mathbf{s}_{\text{author}}, \mathbf{s}_{\text{base}})$ to the base story $\mathbf{s}_{\text{base}}$ and sends the resulting story back to the client.

Story Editor Client. The Unity-based[4] story editor client application extends an existing animated story authoring framework [Kapadia et al., 2016a]. It provides an intuitive interface and employs a storyboarding paradigm for authoring stories. Storyboard images serve as event objects and portrait images serve as participant objects. The story is authored by arranging participants, events, and beats in the 2D screen space. The authoring editor is extended with a UI for performing version control tasks and visualizing story differences. After an update as well as when diffing two revisions, changed events and participants are highlighted using different colors. Figure 3.5 depicts a screenshot of the editor while an author is working on a story.

Story Visualization Tools. The Gephi[5] visualization platform was extended to connect to the SVC server's API and load the

[4] https://unity3d.com/ [5] https://gephi.org/

repository data structure. Using the **full** request, an entire repository including all operations and a story snapshot for each revision is constructed on the server and downloaded. The user can trim the repository to visualize only a specific interval of revisions. Gephi is specialized in graph-based visualizations and provides the user with a variety of statistical metrics and filters. A screenshot of the standalone graph visualization tool is depicted in Figure 3.6. Various types of graphs, as described in Section 3.2.4, can be generated and explored by the user.

Additionally, we provide a website, which lets the author generate a predefined subset of graphs offered in the standalone visualization tool. This website targets users who decide to relinquish the features offered by Gephi but instead prefer a faster and simpler solution to explore graph visualizations of their stories.

3.2.4 Visualization

Collaboratively authored stories serve as a multi-dimensional space of interconnected information. When we author stories collaboratively, it is valuable to explore the stories through different lenses. In this section we explore such different perspectives. Our dimensions are authors, revisions, participants, beats, and events. Each slice of information in this multi-dimensional space characterizes valuable insights into the evolution of the story, characters, authors, and their relations, and is a precursor to facilitating collaboration.

Graphs offer intuitive visual metaphors to convey this information, where each of the dimensions can be assigned a visual attribute, such as node color, node size, or edge width. Different graphs can be constructed by varying visual attributes and aggregating dimensions. To illustrate different types of graphs, Figure 3.7 depicts a simple repository containing three revisions from three authors, Alice, Bob, and Charlie. Purple highlighting

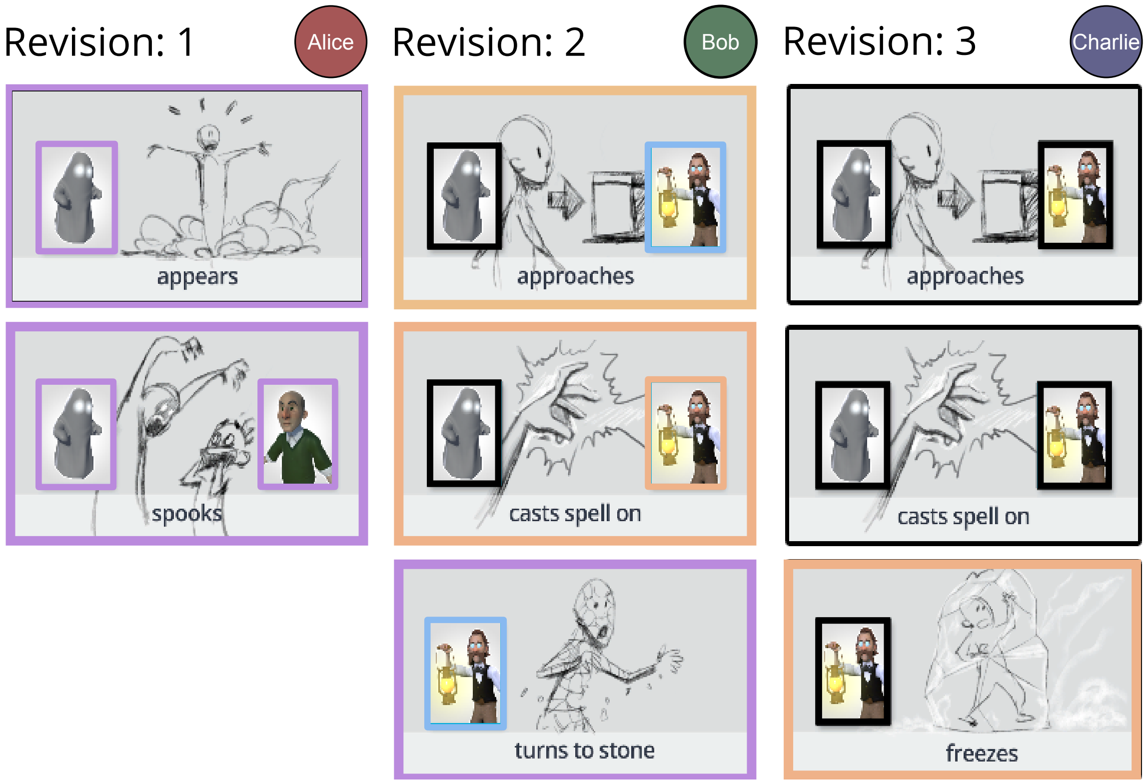


Figure 3.7: *Simple story repository with three revisions and three authors, visualized using events and participants. Purple highlighting indicates inserted objects while orange highlighting indicates changed objects since the last revision. The corresponding visualizations are depicted in Figure 3.8.*

indicates inserted events and participants while orange highlighting indicates changed events and participants since the last revision. Remove operations are not visualized. We refer to this exemplary repository in the following sections while introducing the graphs supported by SVC.

Story-Centric Graph. In a story-centric graph, beat and event nodes form the graph layout, as depicted in Figure 3.8 (a). Author contribution is aggregated over a revision interval and displayed inside the nodes as pie charts. Contributions are calculated recursively to include edits to events and participants within the beat. The story at the last value of the revision inter-

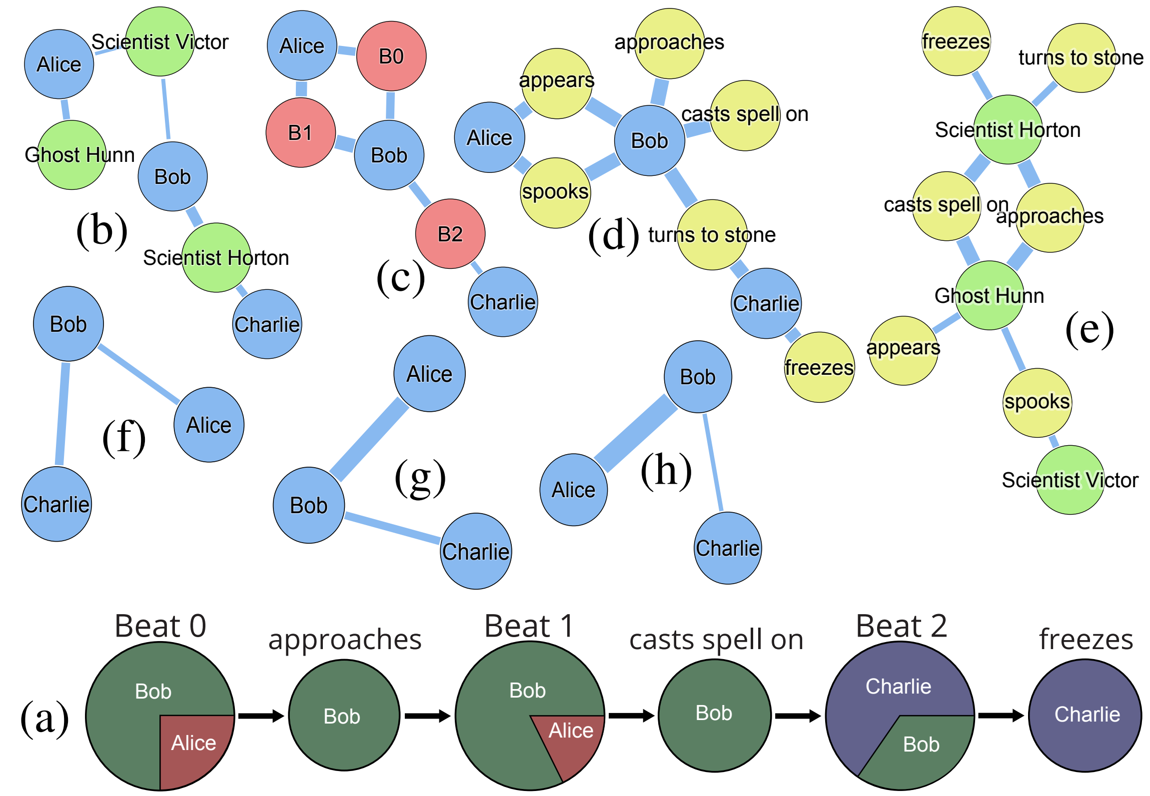


Figure 3.8: *Different graph types generated from the repository of the example story presented in Figure 3.7. Beat-centric graph (a), author-participant (b), author-beat (c), author-event (d), and participant-event (e) relation-centric graphs. Graphs (f), (g), and (h) depict meta-relation-centric graphs corresponding to (b), (c), (d), respectively.*

val is visible. This graph visualizes the length, number of beats, as well as the width, number of parallel events, of a story. Figure 3.8 (a) illustrates that, while Alice created the first beat, she has only little contribution because multiple operations were applied to it by Bob afterward. In contrast, Charlie is the sole owner of the *freeze* event in the last beat as no other author touched that event or participants within.

Relations. Relations are bipartite graphs that characterize the correlation between two dimensions in the story repository.

Author-Participant Relations. This graph visualizes the relations

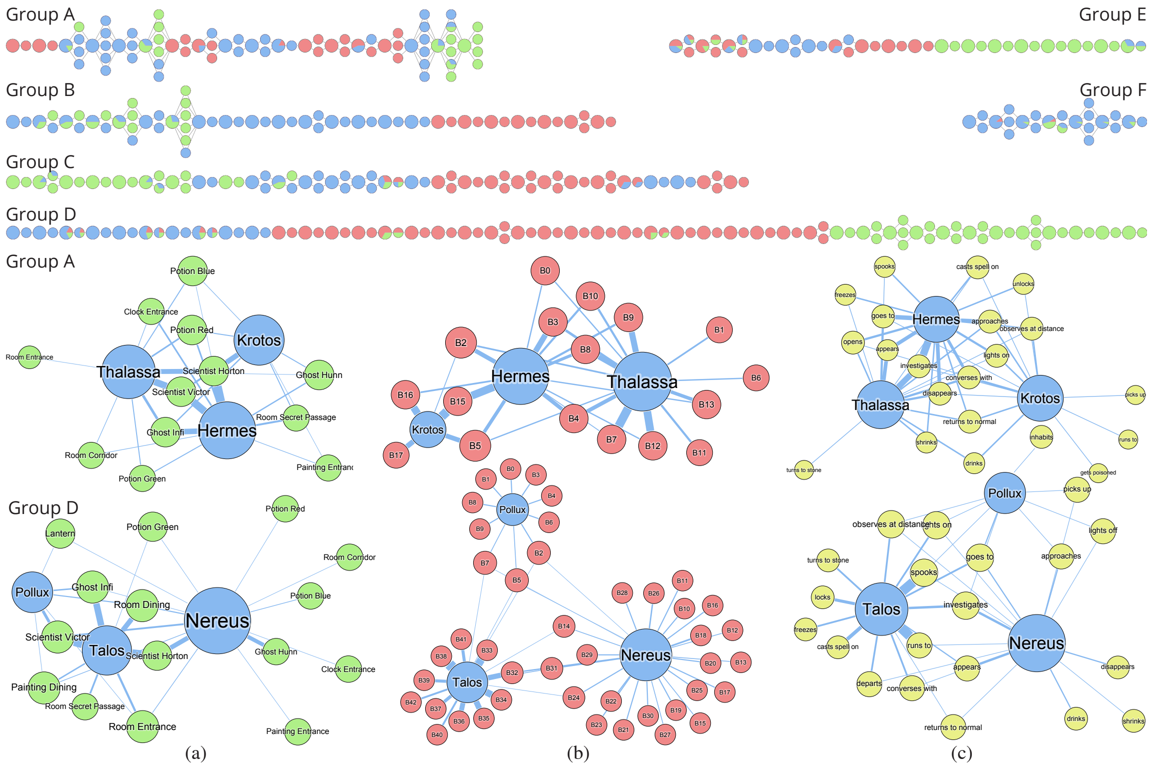


Figure 3.9: *Selected graphs from the story repositories created during the user study. Top contains all story-centric graphs in scale. Bottom (a), (b), and (c) show author-participant, author-beat, and author-event relation-centric graphs for groups A and D, respectively.*

between authors and participants, as depicted in Figure 3.8 (b). The edge width between an author node and a participant node corresponds to the number of times an author has used the participant over all revisions of the story. In the exemplary repository Bob edited the participants *Scientist Horton* and *Scientist Victor*.

Author-Beat Relations. This graph visualizes the relation between authors and beats. The edge width between an author node and a beat node corresponds to the number of times an author has edited the beat or events and participants within the beat over all revisions of the story. The graph in Figure 3.8 (c) indicates that Alice and Bob were both editing the first two beats and Bob and Charlie were editing the last beat.

Author-Event Relations. This graph visualizes the relations between authors and events types. The event instances from all revisions are collected and then grouped by event type. The edge width between an author node and an event node corresponds to the number of times an author has edited the event type over all revisions of the story. The graph in Figure 3.8 (d) indicates that Bob was clearly editing the most different event types.

Participant-Event Relations. This graph visualizes the relations between participants and events types. The author instances and participant instances from all revisions are collected and then grouped by event type and participant type, respectively. The edge width between an author node and a participant node corresponds to the number of times a participant instance occurs within an event instance over all revisions of the story. Figure 3.8 (e) illustrates that participant *Scientist Horton* and *Scientist Hunn* were characters that appeared in the most different event types and are likely the lead characters in the story.

Meta-Relations. Meta-relations characterize relations between entities in the same dimension through their interaction with another dimension. For example, we can characterize the relation among authors based on how they interacted with participants. It visualizes the relations between all authors based on how much they used the same participants throughout all revisions. The edge width corresponds to the number of interactions. Similarly, we can quantify the meta-relation between authors based on which events they used, or how much they edited the same objects within the beats throughout all revisions.

Figure 3.8 depicts three meta-relation-centric graphs author-participant-author (f), author-beat-author(g), and author-event-author (h), corresponding to the graphs (b),(c), and (d), respectively. Meta-relations are especially valuable for large repositories with a great number of entities per dimension, for instance, many different authors or many different event types, where relation graphs become extremely complex and illegible.

Meta-relations are particularly useful when interpreting repositories that involve many authors. Example meta-relation graphs for large repositories with seven authors are depicted in Figure 3.10. Graph (a) indicates that author Aleus is dominating and collaborating with all other authors, while the remaining authors have low connectivity among themselves. Two subgroups of authors are visible in graph (b). Authors Pallas and Orion as well as Hermes and Memnon seem to be highly engaged on the same entities of the story. Krotos is and independent author who is not editing anything that others author have created. Finally, graph (c) conveys that all authors are strongly connected and almost all edit the same story entities.

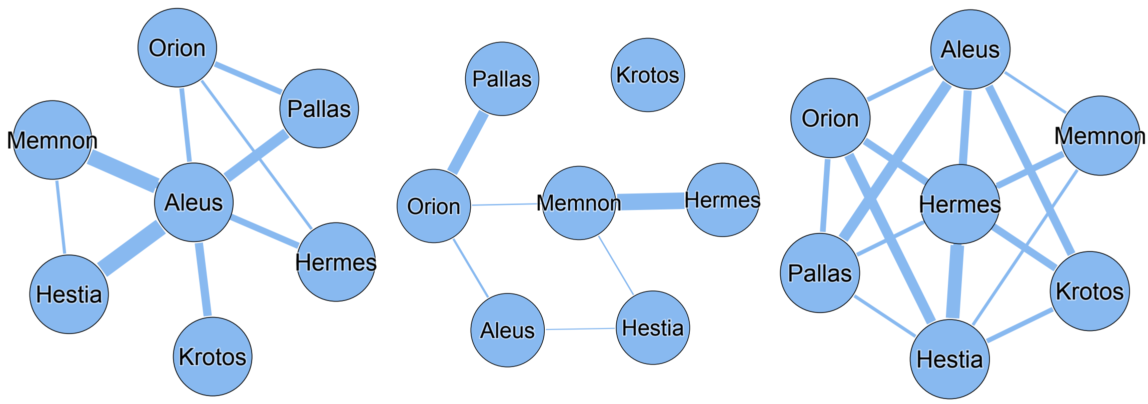


Figure 3.10: *Meta-relation graphs characterize relations between entities in the same dimension, here authors, through their interaction with another dimension.*

User Study Repository Visualizations

Figure 3.9 and Figure 3.11 depict selected graphs from the repositories created as part of the user study, during which six groups (A to F) with three authors each were authoring stories using the SVC system over the course of a day. The user study is described in detail in Section 3.2.5. Figure 3.9 (top) contains the story-centric graphs for each group. The author color keys are omitted for simplicity. Story lengths as well as the author's organization regarding who is assigned to which parts of the story is clearly visible. For instance, in group D, the story is divided into three parts and each part is assigned to one author. Author red made some edits in the previous part to conform it with his or her part. Analogously, author green made edits in the first two parts. Figure 3.9 (a), (b), and (c) contain author-event, author-beat, and author-participant relation-centric graphs for groups A and D. A force atlas [Jacomy et al., 2014] layout was applied to the graphs, which arranges the nodes based on a constant gravity force as well as on the force induced by the edge widths. Additionally, the nodes are resized based on the node degrees. We found that these visual attributes increase readability greatly, as was shown in the graph visualization user study discussed in Section 3.2.5.

In group A *Scientist Horton* and *Scientist Victor* were clearly the most often used characters as is depicted in graph (a). Group A, the story is mostly happening in *Room Dining* and *Room Entrance*.

Graph (b) for group D confirms our observations that the authors did non share edits on the same beats. Nereus was editing the most number of beats. In contrast, the authors in group A share most of their beats.

Graph (c) for group A illustrates that the authors were using many of the same beat types, which confirms again that the authors were collaborating more closely than group D.

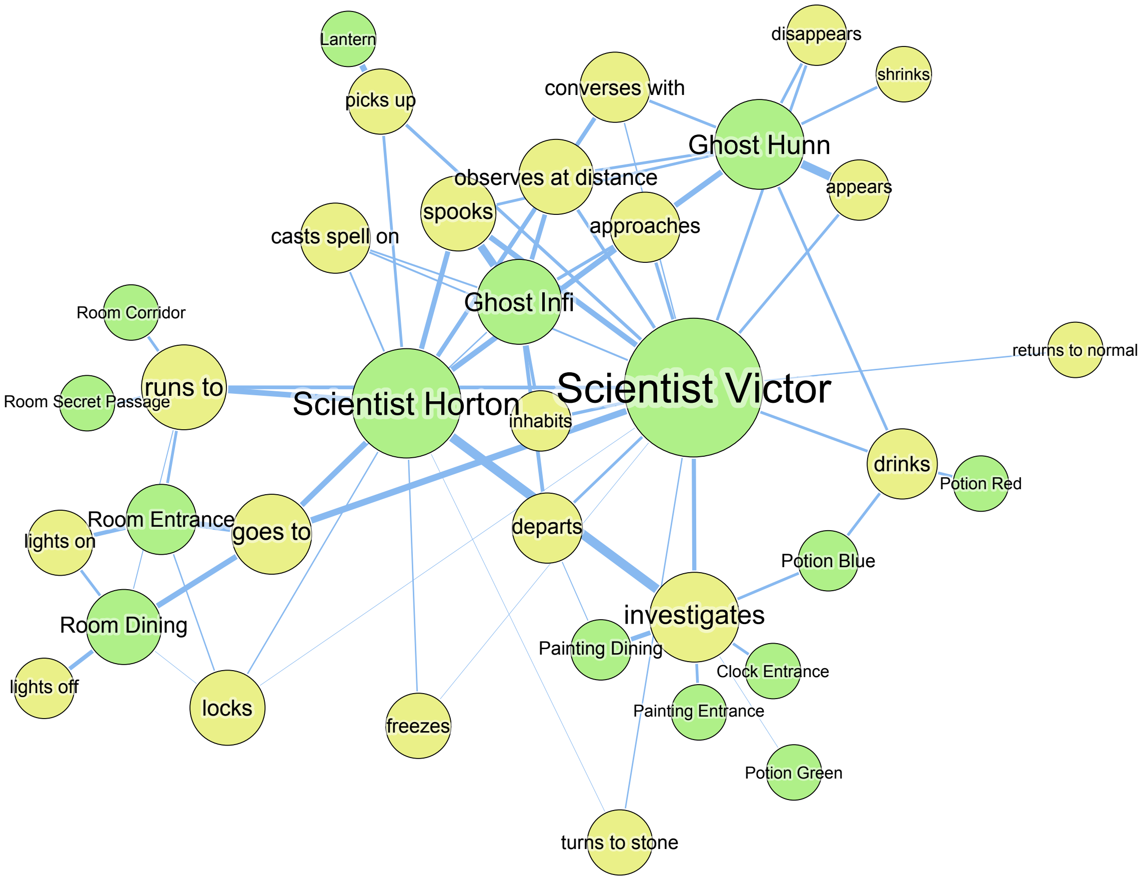


Figure 3.11: *Participant-event relation-centric graph for group D.*

Finally, Figure 3.11 depicts a participant-event relation-centric graph for group D. Various details about the story can be extracted from the graph. *Scientist Victor* and *Scientist Horton* are clearly the main protagonist of the story as they appear in the most different event types. Interestingly, *Potion Green* is investigated but never drunk. Unlike both humans, both ghosts are never frozen and never turn to stone. *Ghost Hunn* is the only ghost that appears and disappears.

Observations. It is particularly useful to observe how the different groups worked together using the visualizations described above. We observe that group A authors were particularly collaborative, where they often worked together to iterate and finalize on the same segments of the story, and contribute to the participants story progression. This is evident in their meta-relationships where we see strong relationships develop between all three authors. The structural properties of the authored stories also emerge from these visualizations where groups A and B author complex stories with many ongoing narrative events. The other groups chose to opt for simpler, linear narratives. The authors in group D collaborate in interesting ways, as evident from their meta-relations. In terms of story beats, the authors prefer to work independently, sketching out different portions of the stories by themselves without much crosstalk. Nereus is central in determining the progression of the various participants.

3.2.5 Evaluation

We conducted two user studies to evaluate both the usability of the system and the value of the graphical visualizations for collaboration. The first study included eighteen participants composed of 14 Computer Scientists and 4 Digital Artists. Participants were grouped into 6 teams of 3. Each team was tasked with the goal to collaboratively co-author a story with our system. The experiment duration was 24 hours, and subjects were asked to distribute their contributions over the experiment duration. All teams were provided with the core SVC system, including functionality to author stories as visual storyboards. The user study concluded with a primary questionnaire to evaluate the entire system. A second follow-up study was conducted to evaluate graph visualizations.

System Evaluation

The primary questionnaire aimed to evaluate the usability of the system in addition to the usefulness of specific version control and visualization functionalities to support collaboration.

To evaluate the usability of the system, we included the 10 question from the System Usability Scale (SUS) [Brooke, 1996]. This questionnaire was selected because it is easy to administer and can provide reliable results for small sample sizes. Our system received a score of 74.6, which demonstrates above average usability.

The remaining items in the primary questionnaire are listed in Table 3.1. These questions were selected to observe qualitative aspects about the collaborative process as well as specific components of SVC. The results are presented in Figure 3.12. Regarding the collaborative process, participants tended to agree that the system facilitates collaborative authoring, awareness of how coauthors were contributing and how the story evolved over time. For the basic version control functionality, subjects strongly agreed that they understood the basic functionality and tended to agree that the basic functionality operated as expected and was helpful in the collaborative process. Participants also tended to agree that they understood how to use the visual difference and update preview views, and that these functionalities operated as expected. However, the visual difference was neutrally observed to be helpful in the process of collaborative authoring. Participants tended to agree that they understood how to use the Conflict Resolution Tool and that the tool helped them resolve conflicts with coauthors. Finally, participants tended to agree that the provided version control functionality was sufficient. We generally observed that the system was useful as well as usable.

Graph Visualization Evaluation

Another aim of the user study is to evaluate the influence of graph visualizations on the collaborative process, specifically in terms of improving awareness of author participation and story content. We designed a follow-up study to specifically

Table 3.1: *Questionnaire excluding SUS questions. Likert scale response: (1) "Strongly disagree" to (5) "Strongly agree".*

Item	Questions
	Questions about Collaborative Process
Q1	The system facilitates collaborative authoring.
Q2	I felt connected to my coauthors.
Q3	I was aware of how my coauthors were contributing.
Q4	I was aware of how the story evolved over time.
	Questions about version control basic functionality
Q15	I understood how to use the basic functionality (checkout, update, and commit)
Q16	The basic functionality operated as expected.
Q17	The basic functionality helped me in the process of collaborative authoring.
	Questions about version control visualizations
Q18	I understood how to use the visualizations for DiffToBase View and Update Preview View.
Q19	The DiffToBase View and Update Preview View operated as expected.
Q20	The visualizations for DiffToBase and Update Preview View helped me in the process of collaborative authoring.
	Questions about Conflict Resolution Tool
Q21	(If you used it) I understood how to use the Conflict Resolution Tool.
Q22	(If you used it) The Conflict Resolution Tool operated as expected.
Q23	(If you used it) The Conflict Resolution Tool helped me resolve edit conflicts with coauthors.
	Question about Version Control Functionality
Q24	The version control functionality provided is sufficient.

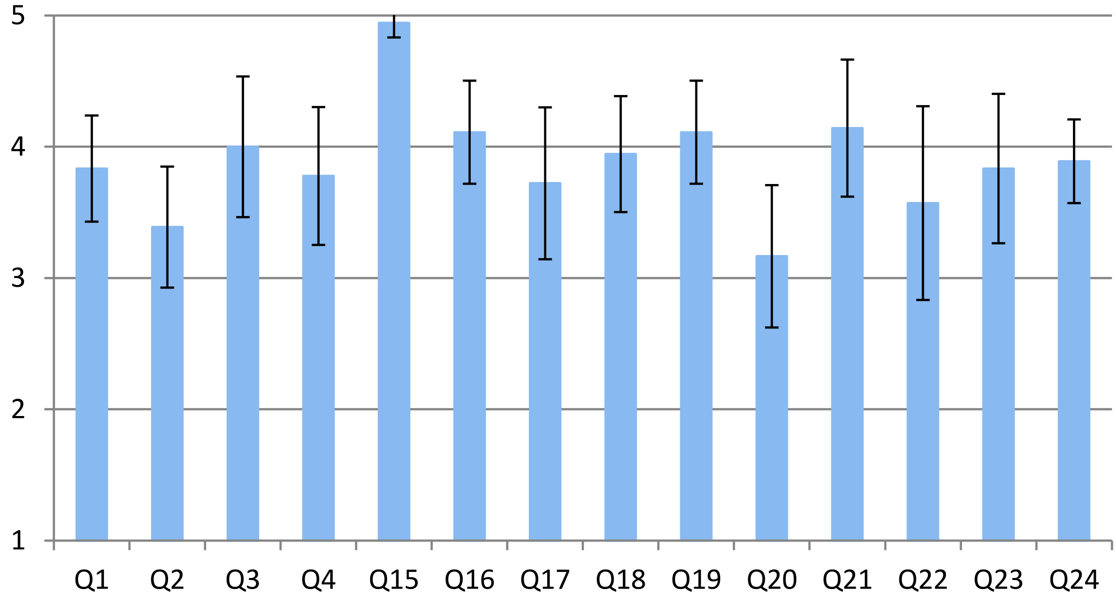


Figure 3.12: *Questionnaire Results excluding SUS questions. All remaining questions regarding the core Story Version Control system. Likert scale: (1) "Strongly disagree" to (5) "Strongly agree". Error bars represent 95% confidence intervals.*

explore quantitative and qualitative aspects of graph visualizations. The quantitative questions provide the context for participants to analyze the graphs. After analyzing two graphs of a given type, the participant is more informed to assess whether the graph type improves awareness of author participation or story content.

We selected the 6 different types of graph visualizations listed in Table 3.2. All graphs, except for the meta-relation-centric graphs, were generated from the completed stories authored during the first study. For each graph type, we present two graphs from two different coauthor groups and ask five questions. The first question requires comparing the two graphs of given type. Questions 2 and 3 can be answered by analyzing graphs 1 and 2, respectively. Question 4 explores relevance of the graph type for supporting collaboration ("This type of graph improves awareness of author collaboration in the story."). Question 5 explores the relevance of the graph type for supporting content understanding ("This type of graph improves awareness of story content.").

We collected responses from 15 of the original user study participants. Although some of these questions were particularly challenging , responses to quantitative questions were correct 90% of the time. The primary aim of the quantitative questions was to motivate the participant to think critically about each graph. Figure 3.13 summarizes the responses to qualitative questions per graph type. Subjects agreed that the Story-Centric Authors-Beats graphs (G1, see example in Figure 3.9, top) improved awareness of author collaboration, and disagreed that it improves awareness of story content. Interestingly, the other two graphs showing Authors-Beats in either Relation-Centric (G4) or Meta-Relation-Centric (G6) had similar responses. We interpret this to mean that visualizations involving authors and beats provide a high-level representation to improve understanding of author participation. Respondents either tended to agree or agreed that all graphs visualizing author information improved awareness of author participation. The Relation-Centric Participants-Events graph lacked author information resulting in the opposite effect, subjects agreed it improved awareness of story content.

Applications

We demonstrate the possibilities of our media-agnostic abstract story representation in SVC by creating stories of various media types. Figure 3.1 depicts three examples, including screenshots from the Haunted Castle animated story, which was automatically synthesized from our visual storyboard editor. Additionally, we created stories using (b) short movie clips as events to produce a movie and (c) photographs as narrative events to pro-

Table 3.2: *Graph Types in Visualization Questionnaire.*

	Types of Graphs in Questionnaire
G1	Story-Centric
G2	Authors-Participant Relations
G3	Authors-Event Relations
G4	Authors-Beat Relations
G5	Participants-Event Relations
G6	Author-Beat Meta-relations

duce a comic representation. The final movie and comic rendering were manually created based on story definitions produced by our authoring system. The movie clips were extracted from the open source film Tears of Steel[6].

3.2.6 Conclusion

We have presented our work to enhance collaborative story authoring by providing a story version control and graphical visualization framework. To support story version control, we propose a graph-based story representation as well as a tree edit distance operation that provides the foundation for story version control operations. We present a suite of visualizations that encode the relationships between story elements, characters, and authors, as the story evolves. These visualizations capture meaningful information that cannot be observed from a single story snapshot, or from the raw data contained in a repository, and provides a means for authors to interpret each others creative intent. Our user study validates the efficacy of our tools and our results show a variety of stories authored by untrained users.

[6] (CC) Blender Foundation - mango.blender.org

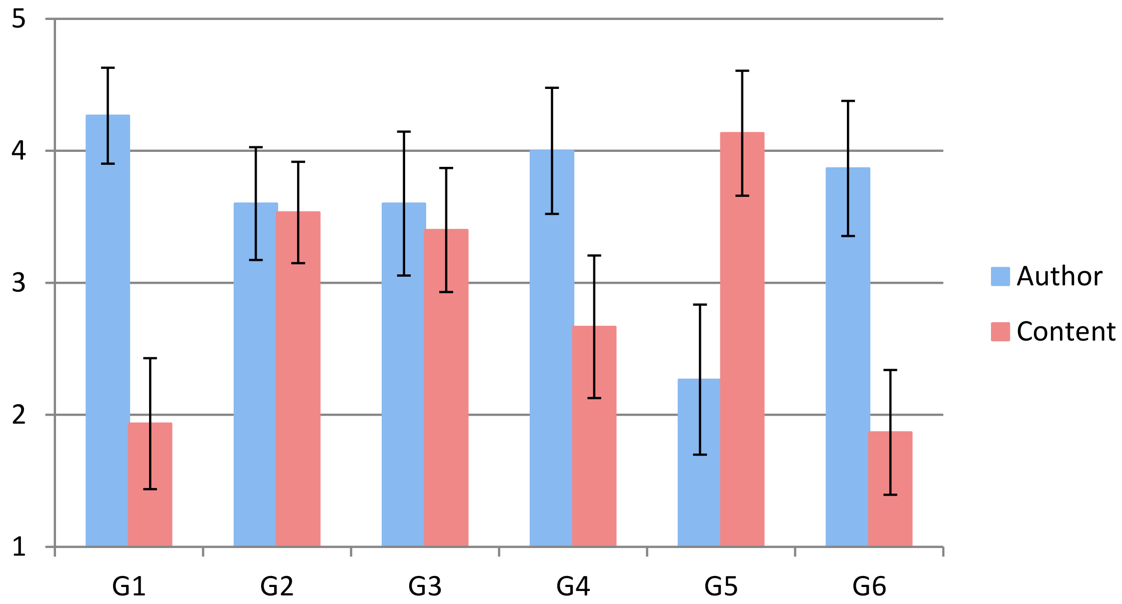


Figure 3.13: *Graph Visualization Qualitative Questions. Responses to questions about how the respective graph improves awareness of author collaboration and story content. Likert scale: (1) "Strongly disagree" to (5) "Strongly agree". Error bars represent 95% confidence intervals.*

Limitations in our current system motivate a number of future research directions. Currently, our system does not support branching narratives. Several users in our user study requested this feature. While we have not implemented such functionality, it could be incorporated via additional engineering effort. Users also requested commenting (annotations to communicate with other authors) and spoken character dialog. Incorporating these features could be achieved by including an additional layer on top of our system that relies on more traditional text-based version control. Directly coupling textual representations with our graph-based representation would provide an interesting area of future work.

Additional participant suggestions and limitations hint at more far-reaching future work. Timing in our system is not represented explicitly but defined implicitly by the given story beats.

A more robust and flexible timing model could provide an additional level of control to story authors. In our work, conflict resolution operates only at the syntactic level when tree operations fail. Exploring semantic, rather than syntactic, conflicts is an exciting research direction. Such semantic detection could flag anachronisms such as a character appearing in a story after he or she has passed away. Detecting, communicating, and offering resolution suggestions for such semantic issues is a challenging future direction. Even more exciting future work could focus on extending semantic understanding to provide deeper assistance to story authors, such as suggesting portions of the story that could be most interesting to develop further.

3.3 Authoring Interactive Narratives

Interactive narratives strive to offer immersive digital experiences in which users create or influence a dramatic storyline through their actions in interactive virtual worlds. The far-reaching goal is to immerse users in a virtual world where they become an integral part of an unfolding narrative and can significantly alter the story's outcome through their actions.

Traditional linear narratives provide little user agency to influence the outcome of the story. Computer games often use linear plots interspersed with isolated interactive segments, with all users experiencing the same plot during successive sessions. Branching narratives [Gordon et al., 2004], where the narrative outcome depends on the user's decisions provide a discrete set of choices that influence the story. The authoring complexity of these approaches grows exponentially with the number of story arcs, the number of interaction possibilities, and the granularity of interaction. Story arcs are tightly coupled and new interactions require monolithic changes where the authoring complexity is kept tractable only by severely limiting user agency to discrete choices at key points in the story. Hence, traditional interactive narrative applications such as games either provide strong narrative experiences with limited user agency or provide compelling interactive experiences with simple narrative structure.

The growing maturity of AR technologies opens up a new host of interaction possibilities and bridges the gap between the user's real world and virtual content to create immersive experiences. However, the ability to author interactive narrative content has not kept pace with other AR technology. Novel and compelling interaction together with deep narrative experiences is seldom achieved.

Our goal is creating an authoring platform that enables content creators to author free-form interactive narratives with multiple story arcs where the players can influence the narrative outcome, while using automation to facilitate the authoring process and not hinder it. There are two main challenges that we face: (1) An appropriate language for authoring interactive narratives that scales with story complexity, and freedom of interaction. (2) Integrated automation solutions to facilitate the story authoring process without sacrificing author control.

In this thesis, we explore the use of Behavior Trees (BTs) [Isla, 2005] for authoring free-form interactive narratives. The hierarchical, graphical nature of BTs is ideal for authoring complex, branching narratives in a modular fashion. However, traditional BT formalisms are not suitable for handling state persistence and free-form user interaction, which are essential for interactive narratives. To meet these requirements, we extend the BT formalism to decouple the monitoring of user input, the narrative, and how the user may influence the story outcome. This decoupling provides users much greater freedom in their ability to interact with the characters in the story and the world they inhabit, and empowers content creators to create compelling narrative experiences with free-form user interaction. Independent of the specification language, the author has to consider all possible responses for each interaction during any point of the narrative to provide complete freedom of user interaction. To make this problem tractable, we provide a suite of automation tools to detect and resolve story inconsistencies, as well as potential conflicts where user interactions may invalidate the story plot. This empowers the author to truly focus on authoring stories, and rely on automation to make the problem of integrating interactivity tractable. Additionally, the same automation tools can be used to dynamically detect and repair stories during an actual play session, in response to unforeseen user intervention that was not detected during authoring.

We demonstrate the potential of Interactive Behavior Trees (IBTs) and automation by authoring an interactive narrative for an AR application. Our approach enables content creators to easily author complex, branching narratives with multiple story arcs in a modular, extensible fashion while empowering players with the agency to freely interact with the characters in the story and the world they inhabit.

3.3.1 Interactive Narratives

An interactive narrative is traditionally represented as a branching story graph where the vertices correspond to story atoms during which the user has no outcome on the narrative, and the directed edges represent a discrete set of choices, which allow the user to influence the story outcome. To provide the user a dramatic storyline in which he can heavily influence the progression and outcome of the narrative, it is important to offer many decision points and a high branching factor. However, increasing the involvement of the user also heavily increases the combinatorial complexity of authoring such a story graph. We identify three main requirements towards authoring free-form interactive narrative experiences:

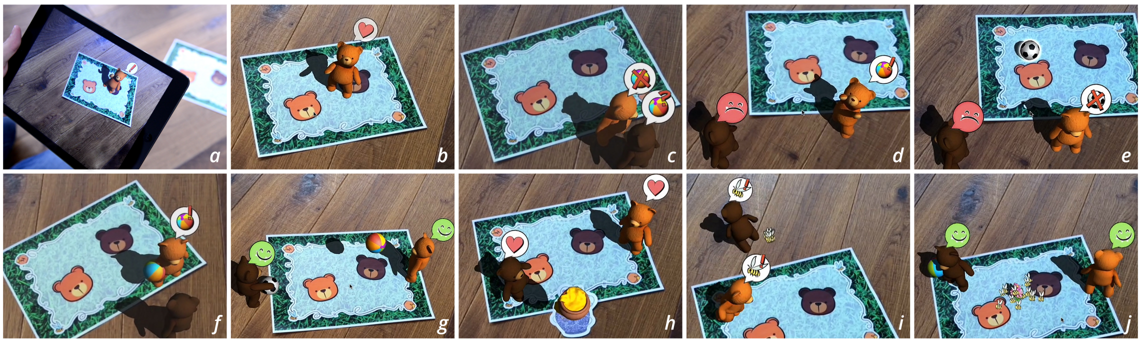


Figure 3.14: *Example narrative: (a,b) The player can freely interact with characters in the authored story to dictate story progression. (c) The bear is distracted when his friend enters the scene and asks him to play ball. (d) When no ball is to be found, the first bears turns to the player for help. (e) The bears aren't happy with the soccer ball and only want to play with the beach ball. (f, g) With the players help, the first bear gives his friend the ball so they can play a game of catch. (h) At any time, the player can use different interactions (e.g., adding a honeypot into the scene) to branch the story in a different direction. (i,j) Spawning bees wreaks havoc on the bears and the player must use flowers to distract the bees.*

1. **Modular Story Definition.** Complex interactive narratives have many interconnected story arcs that are triggered based on user input leading to widely divergent outcomes. The complexity of authoring narratives must scale linearly with the number of story arcs, which can be defined in a modular and independent fashion.
2. **User Interactions.** User interaction should be free-form, and not limited to discrete choices at key stages of the story, with far-reaching ramifications on the outcome of the narrative. Monitoring user input and story logic should be decoupled to facilitate the modification of user interactions without requiring far-reaching changes to the story definition.

3. **Persistent Stories.** The actions and interactions between the user and characters over the entire course of the narrative must persist and influence story progression.

3.3.2 Interactive Behavior Trees

BTs provide a graphical paradigm for authoring complex narratives in a modular and extensible fashion. Story arcs can be independently authored as subtrees and then connected together using BT control nodes to author branching narratives. Recent extensions [Shoulson et al., 2014] facilitate the authoring of complex multi-actor interactions in a parametrizable fashion, enabling the reuse of modular plot elements, and ensures that the complexity of the narrative scales independently of the number of characters. These properties of BTs make them ideally suited for authoring complex, branching narratives (Requirement 1). However, BTs cannot easily handle free-form interactions (Requirement 2) and don't have any means of explicitly storing the past state of characters involved in the narrative (Requirement 3).

To meet these requirements, we introduce a new BT design formalism that facilitates free-form user interaction and state persistence. IBTs, as illustrated in Figure 3.15(a) are divided into 3 independent sub-trees that are connected using a Parallel control node. An IBT $\mathbf{t}_{\text{IBT}} = \langle \mathbf{t}_{\text{ui}}, \mathbf{t}_{\text{state}}, \mathbf{t}_{\text{narr}} = \{\mathbf{t}_i^{\text{arc}} | \mathbf{t}_1^{\text{arc}} \ldots \mathbf{t}_m^{\text{arc}}\}, \mathbf{fi} \rangle$ where: (1) $\mathbf{t}_{\text{narr}}$ is the narrative definition with modular story arcs $\{a_i\}$, each with their own independent subtree $\{\mathbf{t}_i^{\text{arc}}\}$. (2) $\mathbf{t}_{\text{ui}}$ processes the user interactions. Figure 3.15(b) illustrates the story subtree. (3) $\mathbf{t}_{\text{state}}$ monitors the state of the story to determine if the current story arc needs to be changed. Figure 3.15(b) illustrates the story subtree. (4) The blackboard $\mathbf{fi}$ stores the state of the story and its characters. (5) A fourth subtree $\mathbf{t}_{\text{cr}}$

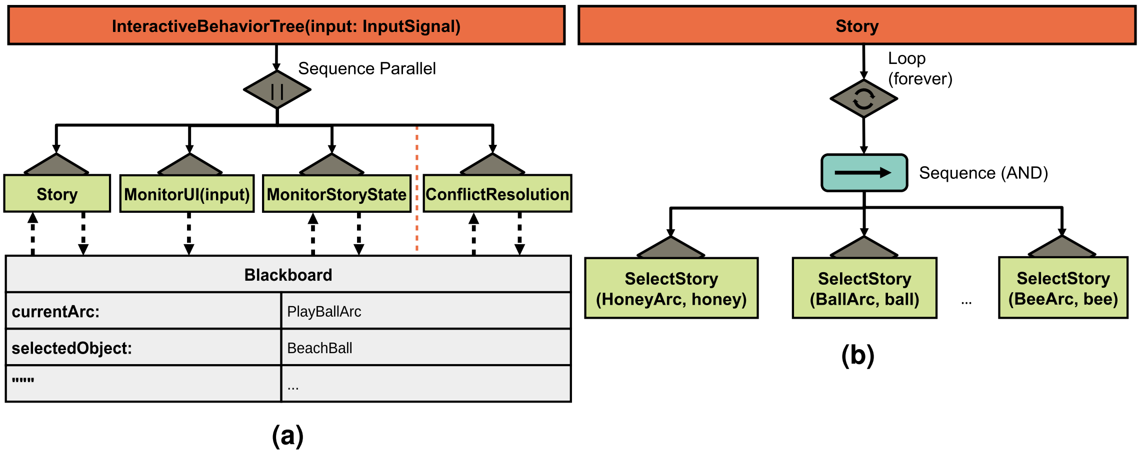


Figure 3.15: *(a) Design formalism of* IBTs *with decoupled specification of user input, narrative definition, and the impact of user input on story state. (b) Narrative subtree with modular story arcs.*

is added for conflict resolution, and will be described in Section 3.3.3.

Story Definition. $\mathbf{t}_{narr}$ is responsible for handling the narrative progression and is further subdivided into subtrees that represent a separate story arc. Figure 3.15(b) provides an example of $\mathbf{t}_{narr}$ while Figure 3.16(a) illustrates each arc definition $\mathbf{t}^{arc}$, which is encapsulated as a separate subtree. This introduces an assertion node, which is checked at every frame whether the current arc is still active before proceeding with its execution. This minor extension to the story arc definition allows the story to instantaneously switch arcs at any moment in response to the user's interactions.

Monitoring User Input. $\mathbf{t}_{ui}$ monitors the different interactions that are available to the user and can be easily changed depending on the application or device. Once an input is detected, it sets the corresponding state in the blackboard **fi**, which is queried by $\mathbf{t}_{state}$ to determine the current state of the story, and the active story arc. Since $\mathbf{t}_{ui}$ is executed in parallel with the

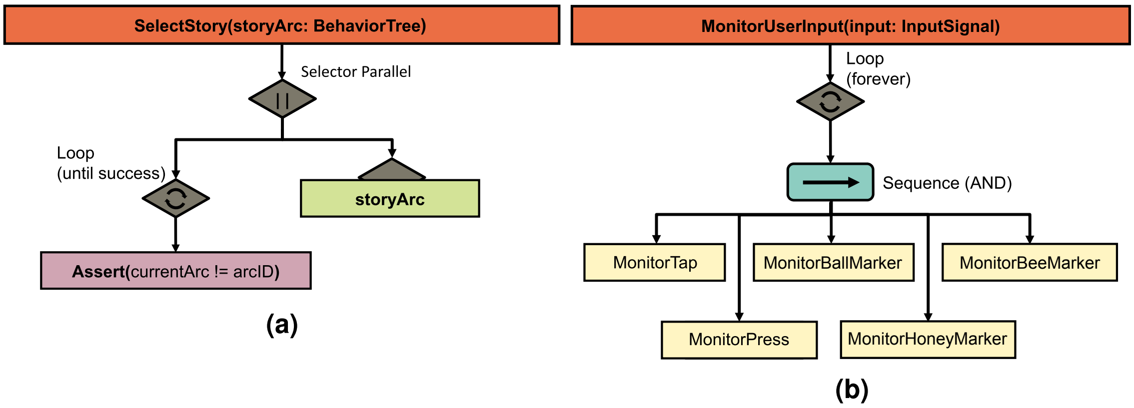


Figure 3.16: *(a) Each story arc definition is encapsulated in its own independent subtree, which first checks if this is the current active arc before proceeding with the narrative execution. (b) Subtree to monitor user input.*

other subtrees, we are able to immediately respond and register the interactions of the user and use it to influence the narrative outcome. Figure 3.16(b) illustrates an example.

Monitoring Story State. $\mathbf{t}_{\text{state}}$ contains separate subtrees for each story arc, which checks if the precondition for the particular arc is satisfied. If so, **fi** is updated to reflect the newly activated story arc, which is used to switch the active story in $\mathbf{t}_{\text{narr}}$. Figure 3.17(a)(b) illustrates $\mathbf{t}_{\text{state}}$ and a subtree used for checking the preconditions for an example story arc. It may be possible for the preconditions of multiple story arcs to be satisfied at any instance, in which case the story arcs are activated in order of priority (the order in which they appear in $\mathbf{t}_{\text{narr}}$). It is also possible for multiple story arcs to be active simultaneously if they are operating on mutually exclusive characters and objects.

Message Passing and State Persistence. The overall design of the IBT results in three subtrees that execute independently in parallel with one another. The blackboard **fi** stores internal state variables (e.g., the current active story arc) to facilitate communication between the subtrees, and maintains state persistence.

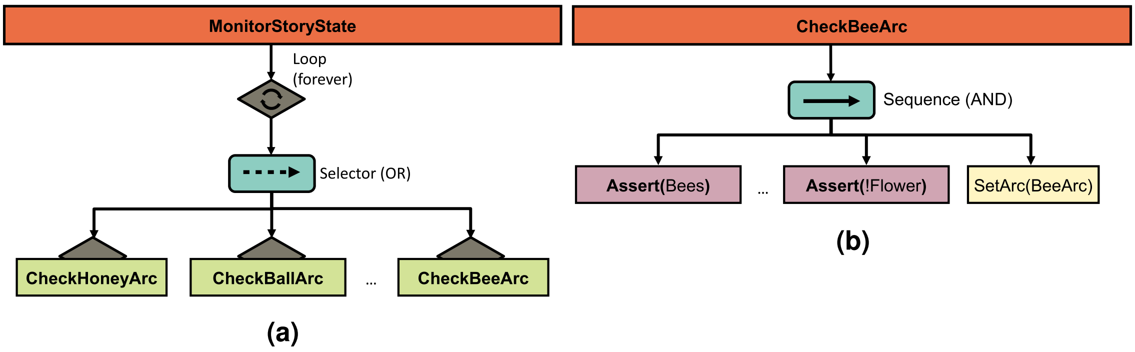


Figure 3.17: *(a) Subtree that changes story state based on user input, which triggers branches in story arc. (b) An example subtree from (a), which checks if all the preconditions for a particular story arc are satisfied before setting it as the current active arc.*

$\mathbf{t}_{\text{ui}}$ updates **fi** when any input signal is detected. Tree $\mathbf{t}_{\text{state}}$ monitors **fi** to check if the preconditions of a particular story arc are satisfied, and updates the current arc. Finally, each arc subtree in $\mathbf{t}_{\text{narr}}$ checks if it is the current active arc before continuing. Also, the user input and the narrative execution can update the story and character state to influence the progression of the narrative at a later stage.

3.3.3 Automation Tools for Authoring Interactive Narratives

Independent of the specification language (story graphs, behavior trees or any other authoring paradigm), interactive narratives require the author to consider the ramifications of all possible user interactions at all points in the story, which is prohibitive for complex stories with many different interaction modalities. To address this challenge, we introduce a suite of automation tools that exploit domain knowledge to automatically identify and resolve invalid story specifications (Section 3.3.3), potential user actions that may invalidate story arcs

(Section 3.3.3, Section 3.3.3), and even automatically synthesize complete stories (Section 3.3.3).

Domain Knowledge

In order to use automated planning tools to facilitate the authoring process, we need to add additional domain knowledge, which can be used by an intelligent system for inference. This includes annotating semantics that characterize the attributes and relationships of objects and characters in the scene (state), different ways in which they interact (affordances), and how these affordances manipulate their state. Our framework is no different from other intelligent systems [Riedl and Bulitko, 2013] in this regard. However, the cost of specifying domain knowledge is greatly mitigated by the ability of computational tools to consider all possible interactions between smart objects, how user input may change story state, and use it to detect and resolve invalid stories. Below we describe our representation of domain knowledge, which balances ease of specification and efficiency of automation.

Smart Objects. The virtual world $\mathbf{W}$ consists of smart objects with embedded information about how an actor can use the object. We define a smart object $\mathbf{w} = \langle \mathbf{F}, \mathbf{Œ} \rangle$ with a set of advertised affordances $f \in \mathbf{F}$ and a state $\mathbf{Œ} = \langle \theta, R \rangle$, which comprises a set of attribute mappings θ, and a collection of pairwise relationships R with all other smart objects in $\mathbf{W}$. An attribute $\theta(i, j)$ is a bit that denotes the value of the j^{th} attribute for $\mathbf{w}i$. Attributes are used to identify immutable properties of a smart object such as its role (e.g., a ball or a bear) which never changes, or dynamic properties (e.g., `IsHappy`) which may change during the story. A specific relationship R_a is a sparse matrix of $|\mathbf{W}| \times |\mathbf{W}|$, where $R_a(i, j)$ is a bit that denotes the current value of the a^{th} relationship between $\mathbf{w}$i and $\mathbf{w}$j. For example, an `IsFriendOf` relationship indicates that $\mathbf{w}$i is a friend of **w**j. Note that relationships may not be symmetric, $R_a(i,j) \neq R_a(j,i)\ \forall\ (i,j) \in |\mathbf{W}| \times |\mathbf{W}|$. Each smart object's state is stored as a bit vector encoding both attributes and relationships. The overall state of the world **W** is defined as the compound state $\text{Œ} = \{\text{Œ}1, \text{Œ}2 \cdots \text{Œ}|\mathbf{W}|\}$ of all smart objects $\mathbf{w} \in \mathbf{W}$, which is encoded as a matrix of bit vectors. $\text{Œ}_\mathbf{w}$ denotes the compound state of a set of of smart objects $\mathbf{w} \subseteq \mathbf{W}$.

Affordances. An affordance $f = \langle w_o, \mathbf{w}_u, \mathbf{\Phi}, \Omega \rangle$ is an advertised capability offered by a smart object that takes the owner of that affordance w_o and one or more smart object users $\mathbf{w}_u$, and manipulates their states. For example, a smart object such as a ball can advertise a *Throw* affordance, allowing another smart object to throw it. A precondition $\mathbf{\Phi} : \text{Œ}_\mathbf{w} \leftarrow \{\texttt{TRUE}, \texttt{FALSE}\}$ is an expression in conjunctive normal form on the compound state $\text{Œ}_\mathbf{w}$ of $\mathbf{w} : \{w_o, \mathbf{w}_u\}$ that checks if f can be executed based on their current states. A precondition is fulfilled by **w** if $\mathbf{\Phi}_f(\mathbf{w}) = \texttt{TRUE}$. The postcondition $\Omega : \text{Œ} \to \text{Œ}'$ transforms the current state of all participants, Œ to Œ$'$ by executing the effects of the affordance. When an affordance fails, $\text{Œ}' = \text{Œ}$. Figure 3.18 describes the general definition of an affordance.

An affordance instance $\mathbf{h} = \langle f, \mathbf{w} \rangle$ includes a set of smart objects $\mathbf{w} \subset \mathbf{W}$ such that $\mathbf{\Phi}_f(\text{Œ}_\mathbf{w}) = \texttt{TRUE}$. To map affordance instances as leaf nodes of a BT, execution of an affordance returns a status that takes three possible values. It returns *Running* if the affordance is still executing. If it *succeeds*, the postconditions Ω_f are applied to the state of all smart object participants. If it *fails*, there is no change in state. This ensures that affordances are considered as atomic units.

Affordance f $(w_o, \mathbf{w}_u)$:
Precondition $\mathbf{\Phi}_f$**:**
CNF expression on compound state **Œ** of w_o and $\mathbf{w}_u$
Postcondition Ω_f**:**
Change in state of w_o and $\mathbf{w}_u$ after successfully executing f
Status γ**:**
Running: Continue executing f
Success: $\mathbf{Œ}' = \Omega_f(\mathbf{Œ})$;
Failure: $\mathbf{Œ}' = \mathbf{Œ}$;

Figure 3.18: *Affordance definition.*

User Interactions. We define a set of user interactions $u \in \mathbf{U}$, which define the different ways in which a user can interact with smart objects in the world $\mathbf{W}$. User interactions are treated as special kinds of affordances where the user is one of the affordance participants. This allows any underlying planning framework to accommodate user interactions during the planning process.

Problem Definition

Given the terminology defined in Section 3.3.3, we define a general problem description $\mathbf{P} = \langle \mathbf{s}_0, \mathbf{\Phi}_g, \mathbf{A} \rangle$ consisting of an initial state $\mathbf{s}_0$, a set of preconditions to satisfy the goal state $\mathbf{\Phi}_g$, and the set of affordance instances $\mathbf{A} = \{\mathbf{h}_i\}$, which may include instances of user interactions. The problem instance $\mathbf{P}$ will be defined in a variety of ways to resolve inconsistencies in the story, or potential conflicts, described in the remainder of this section.

Causal Links. We introduce the concept of a causal link to symbolize a connection between two affordance instances such that executing the postconditions of one affordance satisfies a clause in the preconditions of the other. Causal links are represented as $\mathbf{l} = \langle \mathbf{h}_1, \phi_2^i, \mathbf{h}_2 \rangle$. ϕ_2^i defines the i^{th} clause in $\mathbf{\Phi}_2$ such that $\phi_2^i(\Omega_1(\mathbf{Œ})) = \texttt{TRUE}$.

We interpret $\mathbf{P}$ as a search problem in the space of possible partial plans. We define a partial plan $\pi = \langle \mathbf{H}, \mathbf{\Phi}_{open}, \mathbf{L}, \mathbf{O} \rangle$ where $\mathbf{H}$ is the set of affordance instances currently in π, $\mathbf{\Phi}_{open}$ is a set of pairs $\phi_{open} = \langle \mathbf{h}, \phi_{\mathbf{h}} \rangle$ where $\mathbf{h} \in \mathbf{H}$ and $\phi_{\mathbf{h}}$ defines one condition in the precondition expression $\mathbf{\Phi}_{\mathbf{h}}$. $\mathbf{L}$ defines the set of all causal links between pairs of affordance instances in $\mathbf{H}$, and $\mathbf{O}$ defines a set of transitive and asymmetric partial orderings of affordance instances $\{\mathbf{h}_i \prec \mathbf{h}_j\}$ representing a "before" relation, where $\mathbf{h}_i, \mathbf{h}_j \in \mathbf{H}$. This means that $\mathbf{h}_i$ must occur before $\mathbf{h}_j$ in the partial order plan. A partial plan π_{p} is a plan that has not yet satisfied all open preconditions: $|\mathbf{\Phi}_{open}| > 0$, while a complete plan π_{c} has no open preconditions: $\mathbf{\Phi}_{open} = \emptyset$.

Partial Order Planning. We use a partial order planner (POP) [Sacerdoti, 1975] to compute a plan $\pi_{\mathrm{c}} = \mathbf{Plan}(\mathbf{P})$ that generates an ordering of affordance instances from $\mathbf{s}_0$ which satisfies the preconditions $\mathbf{\Phi}_g$. While POP requires more computational power for processing a single node, it has been shown to outperform total-order planning (TOP) approaches [Pearl, 1984] when dealing with goals that contain subgoals [Minton et al., 1994], allowing the planner to efficiently operate in the search space of partial plans. POP employs the Principle of Least Commitment where affordance execution is ordered only when strictly needed to ensure a consistent plan. This ensures efficiency when dealing with problems where there may exist multiple possible solutions that differ only in their ordering of affordance execution. In contrast, TOP strictly sequences actions when finding a solution. POP is also able to efficiently work for problem definitions where the goal state is partially specified – containing only the desired preconditions that must be satisfied. We provide a brief overview of the algorithm below.

At each iteration, POP selects a clause $\phi_{open} = \langle \mathbf{h}_i, \phi_i \rangle$ from the set of open preconditions $\mathbf{\Phi}_{open}$ and chooses an affordance instance $\mathbf{h}_j \in \mathbf{A}$ that satisfies ϕ_i. If $\mathbf{h}_j$ is not already present, it is inserted into the partial plan π_{p}. $\mathbf{h}_j$ must execute before $\mathbf{h}_i$, which is specified by adding a causal link $\mathbf{l} = \langle \mathbf{h}_j, \phi_i, \mathbf{h}_i \rangle$. Any instance $\mathbf{h} \in \mathbf{H}$ that contradicts ϕ_i must happen either before $\mathbf{h}_j$ or after $\mathbf{h}_i$, and is resolved by introducing additional causal links, as defined by the method **Protect**(). If $\mathbf{h}_j$ is added for the first time, its preconditions are added to $\mathbf{\Phi}_{open}$, and the process continues until all preconditions are satisfied: $\mathbf{\Phi}_{open} = \emptyset$. Alg. 1 outlines the details of the algorithm and we refer the readers to [Sacerdoti, 1975] for additional details.

Algorithm 1 Partial Order Planner

function FUNCTION (Plan) ($\mathbf{P} = \langle \mathbf{s}_0, \mathbf{\Phi}_g, \mathbf{A} \rangle$)
    $\mathbf{\Phi}_{open} = \{\langle \mathbf{h}_{\mathbf{\Phi}_g}, \phi \rangle | \; \forall \phi \in \mathbf{\Phi}_g\}$
    $\mathbf{H} = \{\mathbf{h}_{\mathbf{s}_0}, \mathbf{h}_{\mathbf{\Phi}_g}\}$
    $\mathbf{O} = \{\mathbf{h}_{\mathbf{s}_0} \prec \mathbf{h}_{\mathbf{\Phi}_g}\}$
    $\mathbf{L} = \emptyset$
    while $\mathbf{\Phi}_{open} \neq \emptyset$ **do**
        $\langle \mathbf{h}_c, \phi_c \rangle = \mathbf{SelectAndRemoveCondition}(\mathbf{\Phi}_{open})$
        if $\phi_c(\mathbf{h}) == \texttt{FALSE} \; \forall \; \mathbf{h} \in \mathbf{H}$ **then**
            $\mathbf{h}_s = \exists \mathbf{h} \in \mathbf{A}$ s.t. $\phi_c(\mathbf{h}) = \texttt{TRUE}$
            $\mathbf{H} = \mathbf{H} \cup \mathbf{h}_s$
            $\mathbf{O} = \mathbf{O} \cup (\mathbf{h}_{\mathbf{s}_0} \prec \mathbf{h}_s)$
            for all $\mathbf{l} \in \mathbf{L}$ **do**
                $\mathbf{O} = \mathbf{Protect}(\mathbf{l}, \mathbf{h}_s, \mathbf{O})$
            end for
            $\mathbf{\Phi}_{open} = \mathbf{\Phi}_{open} \cup \{\langle \mathbf{h}_s, \phi_s \rangle | \; \forall \; \phi_s \in \mathbf{\Phi}_s\}$
        else
            $\mathbf{h} = \exists \; \mathbf{h} \in \mathbf{H}$ s.t. $\phi(\mathbf{h}) = \texttt{TRUE}$
        end if
        $\mathbf{O} = \mathbf{O} \cup (\mathbf{h}_s \prec \mathbf{h}_c)$
        $\mathbf{L} = \mathbf{L} \cup \langle \mathbf{h}_s, \phi_c, \mathbf{h}_c \rangle$
        for all $\mathbf{h} \in \mathbf{H}$ **do**
            $\mathbf{O} = \mathbf{Protect}(\langle \mathbf{h}_s, \phi_c, \mathbf{h}_c \rangle, \mathbf{h}, \mathbf{O})$
        end for
    end while
    $\pi = \langle \mathbf{H}, \mathbf{\Phi}_{open}, \mathbf{L}, \mathbf{O} \rangle$ **return** π
end function

Integrating Plan into IBT. The plan π_c generated by POP represents an ordering $\mathbf{O}$ of affordance instances $\mathbf{H}$, which can be easily integrated into an existing behavior tree definition by choosing appropriate control nodes that constrain the order in which affordances in the plan can be executed. Figure 3.19 illustrates an example of how a plan is converted into its corresponding BT definition.

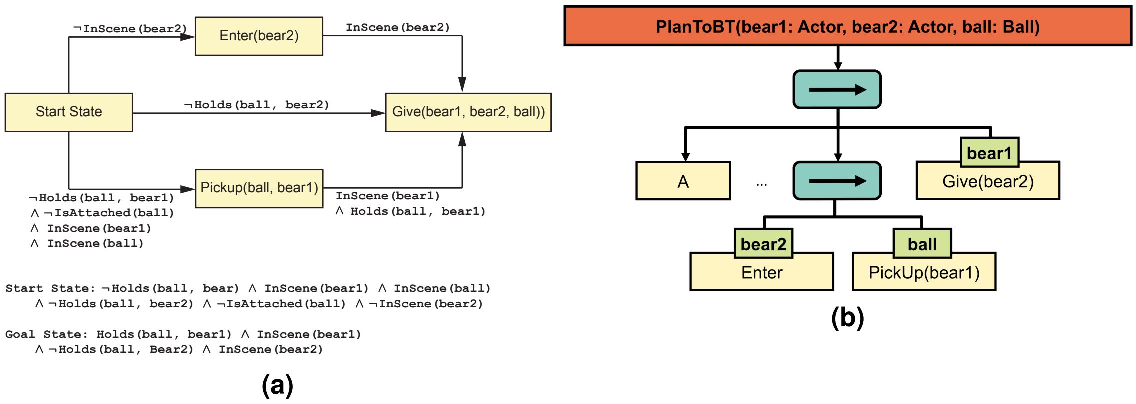


Figure 3.19: *(a) Illustrates a sample plan constructed by POP. The edges represent the causal links between the different affordances. (b) A concrete mapping of the plan to a* BT*.*

Narrative Consistency

A consistent narrative does not violate the preconditions of any affordance instances that are executed during the narrative. A story author may not consider all possible preconditions when defining a story, leading to the definition of an inconsistent story. We define an inconsistent node $\mathbf{n}$ in a story arc as an affordance instance $\mathbf{h}_t$ associated with $\mathbf{n}$ such that $\mathbf{\Phi}_t(\mathbf{Œ_n}) = \texttt{FALSE}$ where $\mathbf{Œ_n}$ is the compound state of smart objects in the scene obtained by executing all nodes in the IBT leading to $\mathbf{n}$.

Belief States. Interactive narratives authored using IBTs can branch in many directions, depending on user interaction and the execution trace of nodes in the IBT. Hence, there may be many possible states that the smart objects are in at a current node $\mathbf{n}$. Therefore, we introduce the notion of a belief state

$\mathbf{b_n} = \{\mathbf{Œ}^1_\mathbf{n}, \mathbf{Œ}^2_\mathbf{n} \cdots \mathbf{Œ}^n_\mathbf{n}\}$, which represents a set of partially specified states of all smart objects that may be reached at $\mathbf{n}$ due to different possible execution traces of the IBT. A partially specified state may contain attributes whose values cannot be determined. By considering the belief state of all possible executions of the narrative that led to $\mathbf{n}$, we can determine whether the preconditions of an affordance instance might be violated by any possible execution of the story arc.

Inconsistency Detection and Resolution. When an inconsistent node $\mathbf{n}$ is detected in the story definition $\mathbf{t}_{\text{narr}}$ of an authored IBT $\mathbf{t}_{\text{IBT}}$, we compute the belief state $\mathbf{b}$ of all possible states that could arise from different execution traces of $\mathbf{t}_{\text{IBT}}$ up to $\mathbf{n}$. For each of these states $\mathbf{Œ} \in \mathbf{b}$, we define a problem instance $\mathbf{P} = \langle \mathbf{Œ}, \mathbf{\Phi_n}, \mathbf{A} \rangle$ and generate a plan π to add additional nodes in the tree such that $\mathbf{\Phi_n}$ is satisfied. Alg. 2 outlines the algorithm for inconsistency detection and resolution.

Algorithm 2 Inconsistency Detection and Resolution

function FUNCTION (DetectAndResolveInconsistencies) ($\mathbf{t}_{\text{IBT}}$)
    for all $\mathbf{t}^{\text{arc}} \in \mathbf{t}_{\text{narr}}$ **do**
        for all $\mathbf{n} \in \mathbf{t}^{\text{arc}}$ **do**
            $\mathbf{b} = \mathbf{ComputeBeliefState}(\mathbf{n}, \mathbf{t}_{\text{IBT}})$
            for all $\mathbf{Œ} \in \mathbf{b}$ **do**
                if $\mathbf{\Phi_n}(\mathbf{Œ}) == \texttt{FALSE}$ **then**
                    $\mathbf{P} = \langle \mathbf{Œ}, \mathbf{\Phi_n}, \mathbf{A} \rangle$
                    $\pi = \mathbf{Plan}(\mathbf{P})$
                    $\mathbf{t}_{\text{IBT}} = \mathbf{IntegratePlan}(\pi, \mathbf{n}, \mathbf{t}_{\text{IBT}})$
                end if
            end for
        end for
    end for
    return b
end function

Conflicts

The players actions may invalidate the successful execution of consistent narratives, and the author must consider the ramifications of all possible interactions at all possible points in the narrative definition. In order to make this problem tractable, we present automation tools that automatically detect potential user interactions that may invalidate affordance preconditions at any stage in the narrative, and provide resolution strategies to accommodate user interference, while still ensuring that the narrative is able to proceed down the intended path.

Before we define a conflict, we first differentiate between two sets of causal links. A link $\mathbf{l} = \langle \mathbf{h}_1, \phi_2^i, \mathbf{h}_2 \rangle$ is *active* if the affordance instance $\mathbf{h_n}$ associated with the current node $\mathbf{n}$ has the following ordering: $\mathbf{h}_1 \prec \mathbf{h_n} \preceq \mathbf{h}_2$. These include all the links that are active when considering the current node $\mathbf{n}$. Keeping track of active causal links allows us to maintain a list of conditions on state attributes which may not be violated by any user interactions while executing $\mathbf{n}$. A link is *needed* to ensure the progression of the narrative at a particular node $\mathbf{n}$ in the IBT if $\mathbf{h}_1 \preceq \mathbf{h_n} \prec \mathbf{h}_2$. These include links that are active *after* the execution of the current node $\mathbf{n}$, and determine conditions on attributes which need to be met even after the execution of an user interaction, to ensure progression of the narrative.

Conflicts. This allows us to formally define a conflict $\mathbf{c}$ as a pair $\langle u, \mathbf{l} \rangle$ where $\mathbf{l} = \langle \mathbf{h}_i, \phi_j^i, \mathbf{h}_j \rangle$ is an active causal link, such that if the user performs a particular interaction $u \in \mathbf{U}$ during the execution of $\mathbf{h}_i$, ϕ_j^i may be violated. Conflicts are detected at a particular node $\mathbf{n}$ if any active causal links at $\mathbf{n}$ are violated and can be resolved by generating a plan that satisfies the conditions of all needed links. Conflicts can be handled in two ways: (1) **Accommodation**. We allow the user to interfere with the narrative by successfully executing u such that $\mathbf{h}$ fails. In this case, we need to generate a conflict resolution strategy that is able to accomplish the same result, as executing $\mathbf{h}$. (2) **Interference**. The affordance instance $\mathbf{h}$ is successfully executed and u fails. No plan is needed in this case. It is up to the author to decide whether to accommodate or interfere for a particular conflict. For conflicts where no plan is possible, we are limited to interference where the user interaction is perceived to be unsuccessful.

Conflict Resolution Subtree. We add a new subtree into the IBT formalism $\mathbf{t}_{\text{cr}}$ that is automatically populated and contains the conflict resolution strategies (plans) for all potential conflicts. During narrative execution, whenever a conflict occurs, control is transferred to the corresponding subtree in $\mathbf{t}_{\text{cr}}$ that contains the plan for resolving that particular conflict.

Conflict Detection and Resolution. Alg. 3 provides algorithmic details of detecting and resolving conflicts at a particular node $\mathbf{n}$ in the IBT $\mathbf{t}_{\text{IBT}}$. We check if any interaction violates the active links at that node. For a potential conflict $\mathbf{c} = \langle u, \mathbf{l} \rangle$, we consider the belief state $\mathbf{b}$ up to the execution of the current node $\mathbf{n}$ in the IBT. For each state $\text{Œ} \in \mathbf{b}$, we define a problem instance $\mathbf{P} = \langle \mathbf{s}_0 = \Omega_u(\text{Œ}), \mathbf{\Phi}_g = \mathbf{\Phi}_{\text{needed}} \rangle$, where $\mathbf{\Phi}_{\text{needed}}$ are the combined conditions of all needed links. A plan π is generated for $\mathbf{P}$ and inserted into the conflict resolution subtree $\mathbf{t}_{\text{cr}}$ to accommodate u. If no plan is found, then we choose to interfere where u is said to fail. The appropriate conflict resolution strategy is added into $\mathbf{t}_{\text{cr}}$.

Dynamic Conflict Detection and Resolution. Static analysis of the IBT is not able to detect *all possible* conflicts that may occur during execution of the narrative. In particular, we cannot detect conflicts (1) that occur while executing nodes in the conflict resolution subtree, (2) due to user actions in one story arc that violate the preconditions of nodes in another story arc. These unforeseen conflicts can be handled by using a modified version of Alg. 3 during the execution of the narrative to dynamically detect and resolve conflicts. This works well in prac-

Algorithm 3 Conflict Detection and Resolution

function FUNCTION (DetectAndResolveConflicts) $(\mathbf{n}, \mathbf{L}, \mathbf{t}_{\text{IBT}})$
    $\mathbf{b} = \mathbf{ComputeBeliefState}(\mathbf{n}, \mathbf{t}_{\text{IBT}})$
    for all $\mathbf{Œ} \in \mathbf{b}$ **do**
        $\mathbf{\Phi}_{\text{active}} = \emptyset$
        $\mathbf{\Phi}_{\text{needed}} = \emptyset$
        for all $\mathbf{l} = \langle \mathbf{h}_i, \phi_j, \mathbf{h}_j \rangle \in \mathbf{L}$ **do**
            if $(\mathbf{h}_i \prec \mathbf{h}_\mathbf{n} \wedge \mathbf{h}_\mathbf{n} \preceq \mathbf{h}_j) == \texttt{TRUE}$ **then**
                $\mathbf{\Phi}_{\text{active}} = \mathbf{\Phi}_{\text{active}} \cup \phi_j$
            end if
            if $(\mathbf{h}_i \preceq \mathbf{h}_\mathbf{n} \wedge \mathbf{h}_\mathbf{n} \prec \mathbf{h}_j) == \texttt{TRUE}$ **then**
                $\mathbf{\Phi}_{\text{needed}} = \mathbf{\Phi}_{\text{needed}} \cup \phi_j$
            end if
        end for
        for all $u \in \mathbf{U}$ **do**
            if $\mathbf{\Phi}_{\text{active}}(\Omega_u(\mathbf{Œ})) == \texttt{FALSE}$ **then**
                $\mathbf{P} = \langle \Omega_u(\mathbf{Œ}), \mathbf{\Phi}_{\text{needed}}, \mathbf{A} \rangle$
                π = **GeneratePlan** $(\mathbf{P})$
                if $\pi \neq \emptyset$ **then**
                    Accommodate$(\mathbf{t}_{\text{cr}}, \mathbf{n}, u, \pi)$
                else
                    Interfere$(\mathbf{t}_{\text{cr}}, \mathbf{n}, u)$
                end if
            end if
        end for
    end for
end function

tice as only a small number of conflicts remain undetected during static analysis and the algorithm for conflict resolution is very efficient and able to instantly generate plans for reasonably complex problem domains.

User Inaction

The user may choose not to execute actions that are required to progress the narrative further. For example, a narrative may require the user to throw a ball into the scene for two bears to play catch. To account for potential user inaction, our automation framework generates contingency plans where the characters in the story may adopt alternate means to accomplish the same effect of the user interaction. For each node $\mathbf{n}$ corresponding to an interaction u, we define a problem instance $\mathbf{P} = \langle \mathbf{Œ}, \Omega_u(\mathbf{Œ}), \mathbf{A} - \mathbf{U} \rangle$ where the action space $\mathbf{A} - \mathbf{U}$ only considers affordance instances with smart objects and discounts user interactions. This is used to generate a plan that achieves the same effect as $\Omega_u(\mathbf{Œ})$ and is integrated into the original IBT definition, as shown in Figure 3.20. During narrative execution, if the user does not perform the desired interaction within a reasonable time threshold, it is said to fail and the contingency plan is executed.

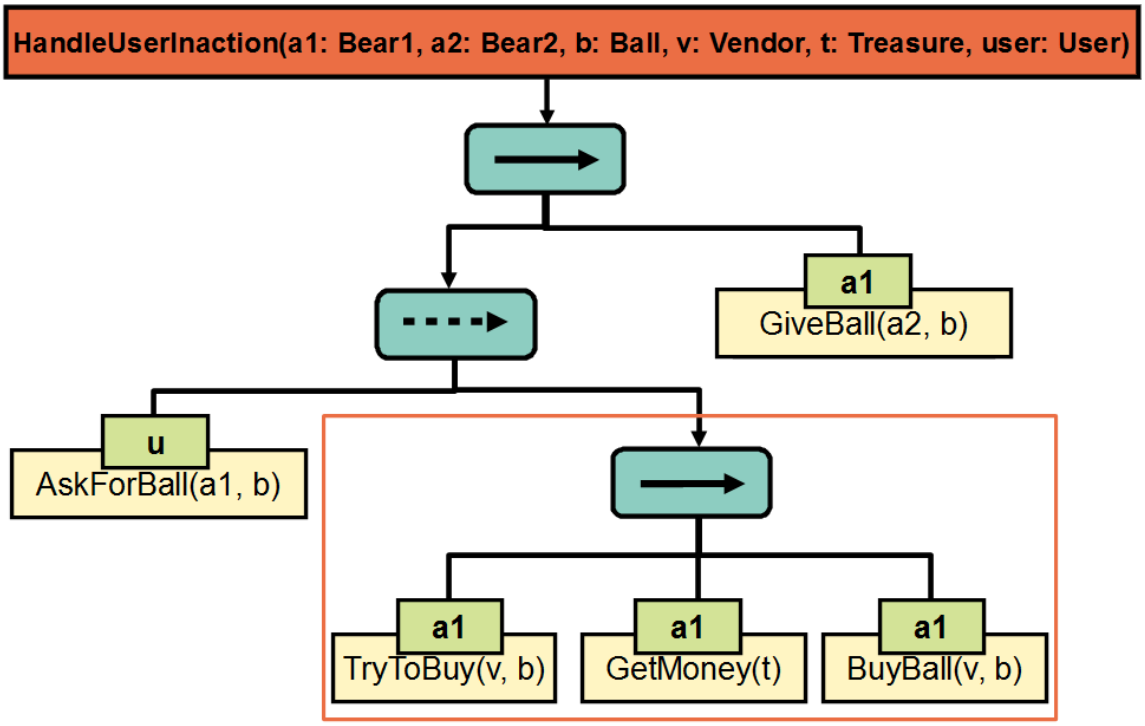


Figure 3.20: *Plan generation to accommodate user inaction. Our system automatically generates an alternate strategy (highlighted nodes) to accommodate potential user inactions that may hinder narrative progression.*

Automated Narrative Synthesis

Authors may harness the power of automation to automatically synthesize narratives which can be integrated into the IBT and edited to meet author requirements. At a given node $\mathbf{t}$ in the IBT, the author simply specifies a desired set of preconditions $\mathbf{\Phi}_g$. This translates into multiple problem instances $\mathbf{P} = \langle \mathbf{Œ}, \mathbf{\Phi}_g, \mathbf{A} \rangle$ for each state $\mathbf{Œ}$ in the belief state $\mathbf{b}$, obtained as a result of executing the IBT up to $\mathbf{t}$. A plan π is generated for each problem instance $\mathbf{P}$ and inserted into the IBT, to provide a narrative that accommodates author-specified preconditions $\mathbf{\Phi}_g$.

3.3.4 Conclusion

This work demonstrates the potential benefits of new design formalisms for authoring interactive narratives based on BTs, and the use of computer-assisted tools for reducing the authoring burden without sacrificing control. Compared to traditional story graph representations, IBTs better scale with story complexity and freedom of user interaction, and authoring stories takes lesser time with reduced number of errors. The authoring complexity is further reduced with the help of automation, and errors are completely avoided.

Limitations and Future Work. The use of computational intelligence to facilitate the authoring process requires the specification of domain knowledge, which is currently limited to domain experts. For future work, we will explore better representations and automatic acquisition of domain knowledge. While IBTs significantly reduce the authoring complexity, it has a learning curve and is not immediately accessible to end users without a computer programming background. The task of authoring interactive narratives requires expertise and takes effort and time. Improved design formalisms are thus needed to meet the growing demand of casual content creation.

The proposed automation features can in principle be applied to other formalisms as well, and future studies may use improved story graphs with hierarchical node structures to study the benefits of automation with alternate story representations. For future work, we will conduct a user study to assess the authoring complexity of the different design formalisms, and the benefits of automation during the authoring process. Additional experiments should be conducted to capture and analyze the playability of the authored narratives. Strong playability metrics that reflect the semantic quality of the authored narratives need to developed. Finally, we would like to evaluate the scalability of our approach to handle the complexity and diversity of narratives present in commercial games.

CHAPTER 4

Augmented Reality

Figure 4.1: *This* AR *picture app takes pictures of a user together with Eva, the super scientist. Using two spheres next to the image marker, the app captures the environment lighting of the real world and applies similar lighting conditions to the virtual character to enhance realism.*

4.1 Introduction

This chapter presents the AR technology domain of this thesis. Work contained in this chapter focuses on techniques to enhance AR experiences. First, we demonstrate how motion effects influence the user experience in mobile AR games. We focus on evaluating responses to a selection of synthesized camera-oriented effects, such as motion blur, defocus blur, latency, and lighting responsiveness. Second, AR techniques to create an AR coloring book app are presented. AR characters are live textured from colored drawings. We propose a 2D to 3D texture transfer process, a deformable surface tracking method, and validations for both.

4.2 Augmented Reality Mixing Techniques

AR features in an increasing number of mobile games and interactive entertainment applications. In visual terms, the presence of real and virtual objects that coexist in the same space [Azuma, 1997] is conveyed through the seamless blending of the camera image with rendered computer graphics. Further, the realistic depiction of motion in interactive AR is affected by various reality mixing measures including matched motion blur, reduced latency, and responsiveness to changes in the camera image.

Recently, the perceptual and practical impact on the players' user experience of motion blur in racing games has been studied rigorously with the, perhaps surprising, outcome that the effect is itself not essential to the enjoyment of the game overall [Sharan et al., 2013].

In a Virtual Reality (VR) scenario, all animated graphical content takes place within the bounds of a display showing the virtual environment, and so the resulting effect of motion blur in this scenario is purely synthetic. However, given that AR brings into play the context of the players' real visual surroundings, we are interested to determine whether the same outcome would be observed. That is, would similar responses arise from a user study examining the effects on enjoyment, sense of realism, satisfaction, and task performance with reality mixing methods.

Real-time computation performance is more challenging for AR than VR in that the balance and blend of rendered graphics must be targeted to extract and process the real camera image to mix and match content seamlessly. Therefore, it is important to determine which reality mixing measures matter for user experience and to what degree must they be accurate. In this thesis, we reprise a motion blur user study [Sharan et al., 2013], but for the AR domain and then further investigate the motion effects of varying levels of latency and responsiveness to dynamic environment lighting changes.

4.2.1 Experiments

We conducted three experiments to assess the impact of different camera and lighting effects in AR games on task performance and perception. Our subjects were 24 to 55 years old and around 80% of them are male.

In the first experiment the users played an AR game *ARTravelers* multiple times with different amounts of artificial camera blur added to the background video image and to the foreground virtual content. The second experiment used the same game with different amounts of added artificial image latency. After each short game, the users answered questions about enjoyment, satisfaction on their resulting score, and perceived realism and immersion.

The AR game used in the first two experiments uses a cubical marker, which is mounted about chest high on a tripod in the center of a room, as depicted in Figure 1.5a. The game is explained in more details in Section 5.2.1. For both experiments, each player plays five rounds. In each round, targets spawn randomly around the cube. The player needs to align the mobile device with the target and the cube in order to destroy the target. The cube has a side length of 20 cm and it was mounted on a tripod with about 15 degrees incline. With the cube inclined, the players need to raise and lower the device to achieve high scores, which increases the difficulty of the game. 100 points are awarded for destroying a target with alignment errors up to 1 degree and 30 points for alignment errors of 60 degrees, linearly interpolated in between. Faster shooting allows players to destroy more targets and thus collect more points. The duration of each round was 60 seconds in the first experiment and 40 seconds in the second experiment.

In the third experiment we employed *ARPix*, an AR application that lets someone take a photo of the user together with a virtual character, as depicted in Figure 1.5b. Four different versions of the image with different rendering techniques were presented to the user. He or she was then asked to choose the image that looks the most realistic.

In order to statistically evaluate the experiments, we designed them as follows: In the first experiment (blur) the independent variable is the amount of blur, which is nominally defined within three configuration scenarios A, B, and C. The dependent variables where the achieved score (ratio) as well as the answered questions about enjoyment (interval), satisfaction (interval), realism (interval), and matching (interval). The individual questions are discussed in Section 4.2.2. For the second experiment (latency) the independent variable was the amount of artificially added latency (ratio), whereas the dependent variables were the achieved score (ratio) and the answered questions about enjoyment (interval), satisfaction (interval), and responsiveness (interval).

The games ARTravelers and ARPix were developed in Unity[1] using Qualcomm's Vuforia[2] package for AR. They employ natural image marker tracking as performed by Vuforia for the camera pose estimation. ARTravelers ran on an iPad Air (MD785GP/A), iOS version 7.0.3, for the first two experiments, while ARPix ran on an iPad 3rd Generation (MD371FD), iOS version 5.1.1.

4.2.2 Camera Motion Blur Experiment

Cameras capture an image by exposing the photographic film or digital sensor to photons. Depending on the brightness of the scene being photographed, that is, the number of photons arriving at the film or sensor, the camera settings must be adapted in order to capture a well exposed image. The camera settings comprise the exposure time (shutter speed for video), the film sensitivity or sensor gain, and the aperture of the lens. For darker scenes, as the sensitivity and the aperture are typically more limited (especially in small form cameras as integrated into mobile devices), the exposure time is often increased. This inevitably leads to blurred images if objects in the scene move relative to the camera.

AR applications rely on immersion created through seamlessly mixing the rendered virtual content (usually in the foreground (FG)) with the camera image (usually in the background (BG)). If the FG image does not visually match the BG image, we expected that the immersion may break. Our goal was to test situations where the FG blur matched the BG blur and situations where it did not match. As it is more feasible to add motion blur to the virtual rendered content than to remove it from the camera image, we chose the former method.

In this experiment, we chose a bright room, lit by daylight

[1] http://unity3d.com/ [2] http://www.qualcomm.com/solutions/augmented-reality

through windows and by lamps (see Figure 1.5a). In that way the exposure time is short, resulting in images with very little motion blur. Then, we artificially blurred the BG and the FG independently to simulate a longer exposure time and examined the impact on the task performance and the players' perceptions. The amount and direction of the motion blur depends on the camera motion and is calculated as a translation relative to the marker.

Experiment Setup

During the experiment, each participant played 5 rounds, 60 seconds each. In each round, a blur configuration scenario was randomly selected. We presented the game to the player without modification (scenario A), with artificial motion blur only added to the BG (scenario B), and with artificially added blur to both the FG as well as the BG (scenario C), see Table 4.1. Scenario A is the default case in a bright environment. Scenarios B and C simulate a darker scene when the device's camera switches to a longer exposure time. However, in scenario C there is also artificial FG blur added to match the BG blur, which we expected to create a more realistic mixing of virtual and real images than in scenario B. The player was not informed about the intention of the experiment and thus was not aware that there would be artificial blur added to the game.

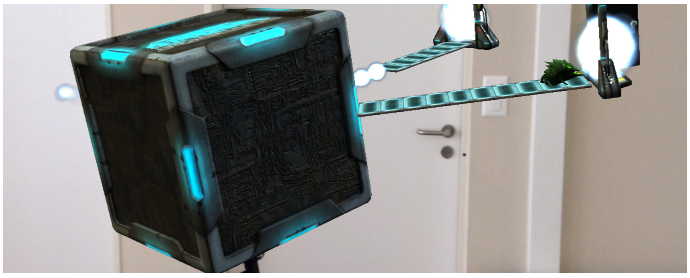
Table 4.1: *Configurations scenarios for the motion blur experiment.*

	Foreground Blur	Background Blur
Scenario A	Off	Off
Scenario B	Off	On
Scenario C	On	On

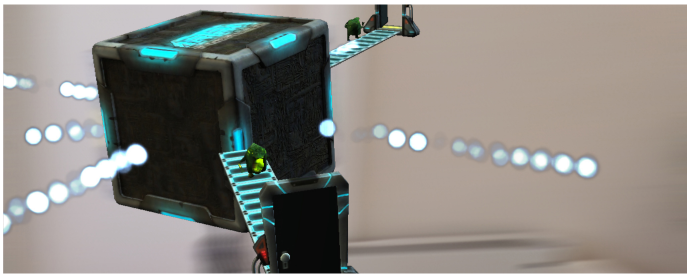
Figure 4.2 depicts screenshots from the ARTravelers game, modified for this experiment. During the game, the user sees

the cube with targets spawning from its center. These targets have to be destroyed by walking to the target and aligning the device with the target and the marker cube. The head-up display shows the collected points, the remaining time, and the time during which the tracking was lost.

We chose to add a relatively strong blur, compared to the naturally occurring blur in dark settings, because we wanted to investigate if the participants would notice the blur at all and if the blur had any influence on task performance.



(a) A: FG blur = Off; BG blur = Off



(b) B: FG blur = Off; BG blur = On

(c) C: FG blur = On; BG blur = On

Figure 4.2: *ARTravelers in-game screenshots with different blur configuration scenarios (A, B, C) during the Camera Motion Blur experiment.*

After each round, the participants were asked to answer the following questions:

- **Enjoyment:** "How much did you enjoy this run?"
- **Score satisfaction:** "How satisfied are you with your score in this run?"
- **Realism:** "How realistic did the game look in this run?"
- **Matching:** "How well did the virtual content (foreground) match with the camera image (background)?"

Each question could be answered on a scale from 1 (not at all) to 5 (very much).

Experiment Results

The first out of five rounds was considered a training round and was omitted in the further analysis to remove a strong influence of the learning effect. This learning effect can clearly be observed in Figure 4.3b. We recorded 12 participants and thus, 48 rounds in total.

The box plots of all 48 rounds' scores ordered into the blur configuration scenarios in Figure 4.3a visually show a slight negative trend from scenario A to B to C. However, the ANOVA yielded no statistically significant connection, as described in Table 4.2. Thus, the hypothesis that the blur configuration scenario does not influence the player's performance or experience cannot be discarded ($p > 0.05\%$). This may be due to the fact that there were not enough participants recorded or that all participants experienced the blur scenarios very differently. Some players might have looked over the mobile device to orient themselves in the real world and ran directly to the target's position. Others might have been more confused by the blur and scored less.

Table 4.2: *ANOVA statistic for the influence of blur scenarios on player performance (scores) and survey answers (enjoyment, score satisfaction, realism, matching). In our experiment we could not observe a significant influence of the blur scenario on the dependent variables.*

Dependent variable	SS	df	MS	F	p
Scores	75761	2	37880	0.64	0.532
Enjoyment	2.115	2	1.057	0.95	0.392
Score satisfaction	2.447	2	1.223	0.78	0.463
Realism	0.731	2	0.365	0.48	0.620
Matching	1.066	2	0.533	0.59	0.560

Figure 4.4 depicts the correlation matrix for the the independent variable, the blur configuration scenario, and the dependent variables, the scores and the answers. Confirming the results from Table 4.2, none of the independent variables correlates in a statistically significant way with the dependent variable. Interestingly, the correlation between the scores and the enjoyment is also not a significant correlation, even though intuitively, one would expect a correlation. However, there seems to be, next to the obvious correlation between enjoyment and scores satisfaction, a strong ($r > 0.4$) correlation between enjoyment and matching as well as very strong ($r > 0.69$) correlation between enjoyment and realism. We observe that, independent of the achieved scores and how the blur was actually configured, the participants answered that they enjoyed the last round if they voted for a high perceived realism of the game as well as a high matching of BG and FG.

4.2.3 Latency Experiment

In video see-through AR games and applications running on mobile devices and Head Mounted Display (HMD) the virtual

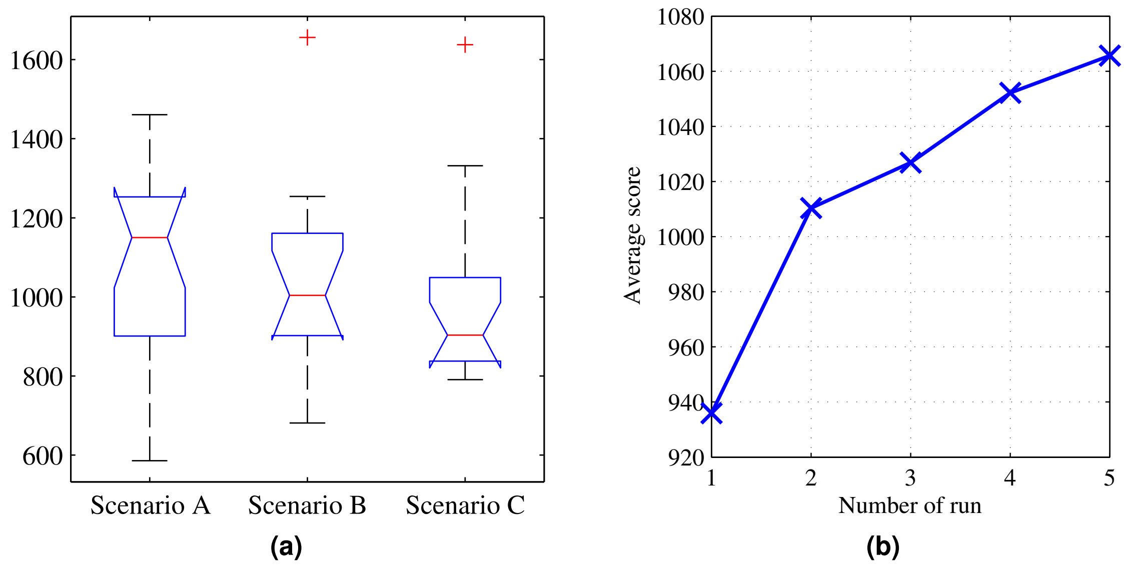


Figure 4.3: *(a) depicts box plots of scores for each scenario. Visually, a decreasing trend is noticeable from scenario A to B to C. That is, the stronger the total blur, the less the players scored. However, the trend is not statistically significant. (b) shows the average score per round for all players. A learning effect is clearly visible.*

content is rendered onto the camera image. Until the final image is visible on the device's display, processing, synchronization and signal transmissions cause delay and add up to the total system latency.

This latency is an important factor for reactive games as well as for augmented reality applications. We used the same game as in the motion blur experiment (Section 4.2.2) to investigate the influence of latency on the task performance.

In this experiment, we artificially increase the latency of the frames presented to the players and record again task performance as well as usage experience. We hope to gain insight about the degree to which latency influences these measures in order to judge the importance of reducing sources of latency.

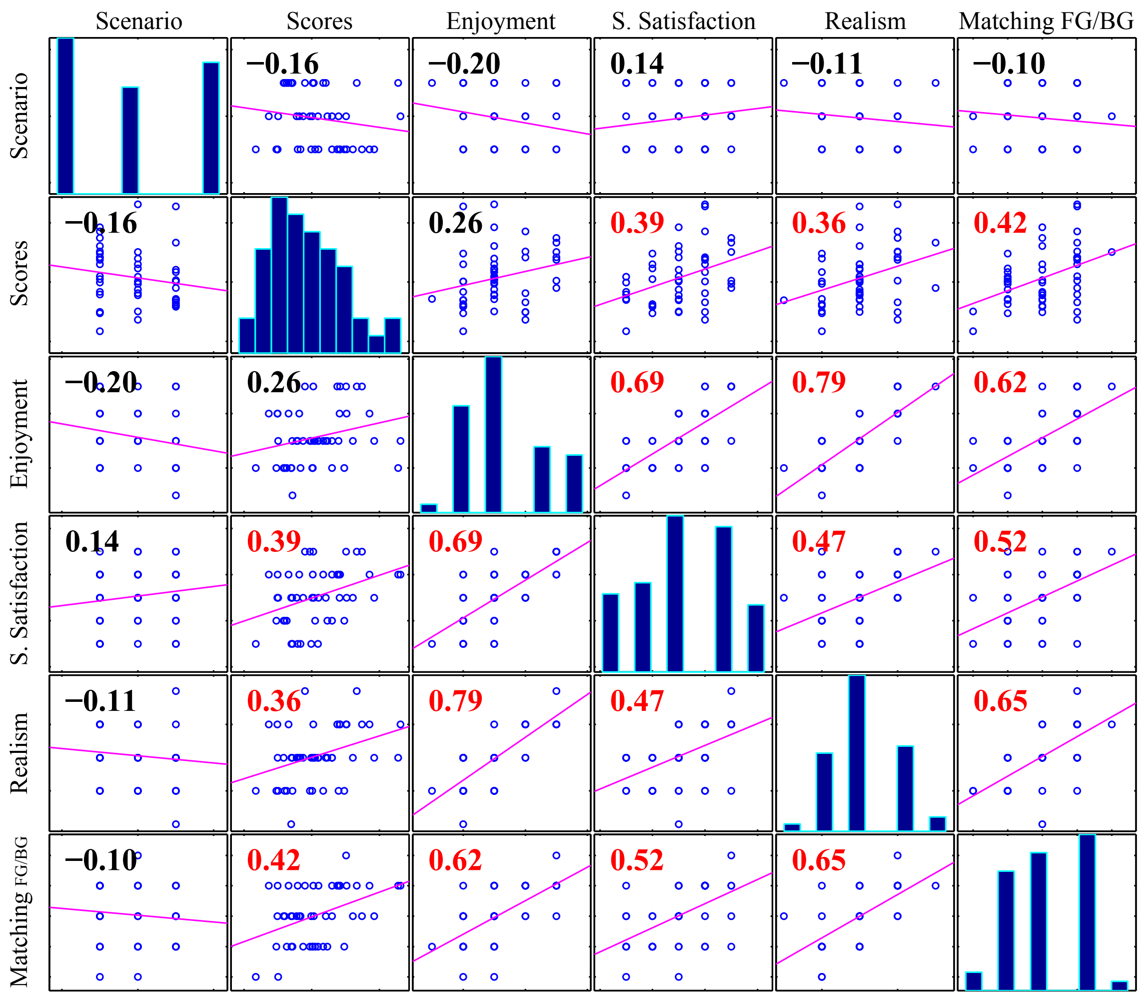


Figure 4.4: *Correlation matrix for the independent variable (scenarios) and all dependent variables (scores and answers). A red correlation value indicates a statistically significant correlation.*

Native System Latency Measurements

To measure the total native latency caused by the hardware (camera) and software (operating system and middleware) in ARTravelers, we place blinking LEDs in the view of the device's camera and record both the LEDs on the screen as well as the LEDs directly with a 200 Hz camera, as depicted in Figure 4.5. This approach is similar to Jacobs et al.'s [Jacobs et al., 1997] method.

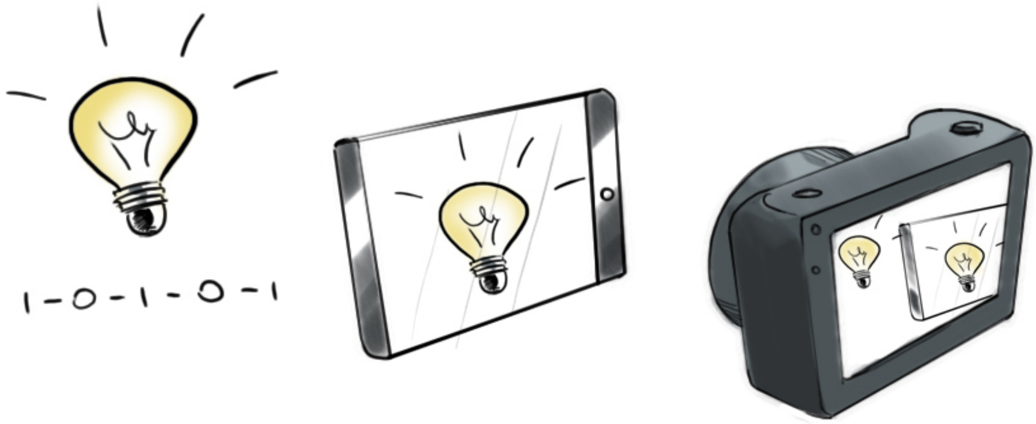


Figure 4.5: *Latency measuring setup. The mobile device running AR-Travelers is pointed at blinking LEDs. A high-speed camera captures both, the LEDs and the LEDs on the display. The recorded video can then be analyzed to calculate the delay between the LEDs and the LEDs on the display.*

We then analyze the brightness of both LEDs' image regions in the recorded video over time by cross correlating the two pixel brightness signals (thresholded) to find the system latency. In Figure 4.6, the two pixel brightness signals for an example measurement are shown. To get a stable signal, an average of a 10×10 pixel area is used. To avoid a measurement error from rolling shutter, which is used in both cameras, we tried to vertically center the lamps in the image. A textured background and the marker cube were also visible to the device's camera in order to measure a realistic latency like during the game. The offset between the two signals is clearly visible. In this measurement it is about 100 ms.

We conducted measurements for several settings that each had different influences on the latency, summarized in Table 4.3. The lighting column denotes how bright the scene was lit. It is important because low light conditions may lead to increased exposure time and therefore introduce larger latencies. The features column denotes if the device's camera was pointed at a

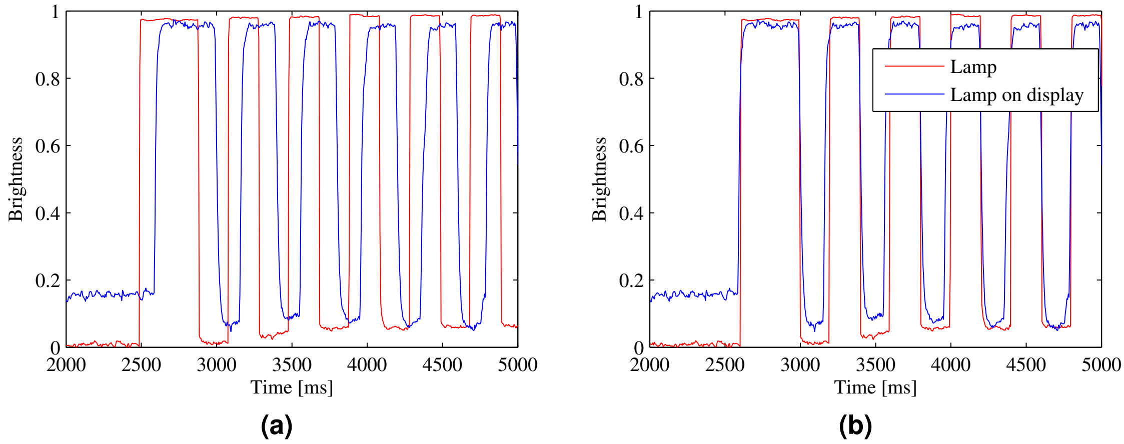


Figure 4.6: *Two brightness signals over time. In (a) the offset between the signals is clearly visible, about 100 ms. In (b) the offset has been calculated and removed from the signals with the method discussed in Section 4.2.3.*

background with a high number of image features (e.g. SIFT features [Lowe, 2004]). With more potential features, the feature matching takes longer as there are more features with which to compare. The application column refers to the application that was running while recording the measurement video.

Measurement A used the camera app preview and bright lighting to measure the minimal possible latency (around 90 ms). The conditions in B, C and D were chosen to cover different situations or setups for our game. In normal lighting (B, C) the latency is around 100 ms and in a low lighting situation it grows to around 130 ms.

Artificial Latency Measurements

We modified ARTravelers such that it delays all shown frames by N frames to introduce additional artificial latency (AAL), that is, latency additional to the existing native system latency. The application runs at constant 30 fps and therefore every AAL

Table 4.3: *Measured native system latency for different settings.*

Setting	Lighting	Features	App	Avg latency
A	bright	-	iOS camera app	89.9 ms
B	normal	high	ARTravelers	101.5 ms
C	normal	low	ARTravelers	97.3 ms
D	dark	low	ARTravelers	131.1 ms

frame adds an additional 33.3 ms latency to the native latency. To verify, we measured the AAL using the same method as in Section 4.2.3. In Figure 4.7a the measured total latency over the intended artificial latency is plotted. As expected from the results in Table 4.3, there is a minimal total latency of about 115 ms. For higher AAL, the native latency decreases to about 50 ms to 60 ms.

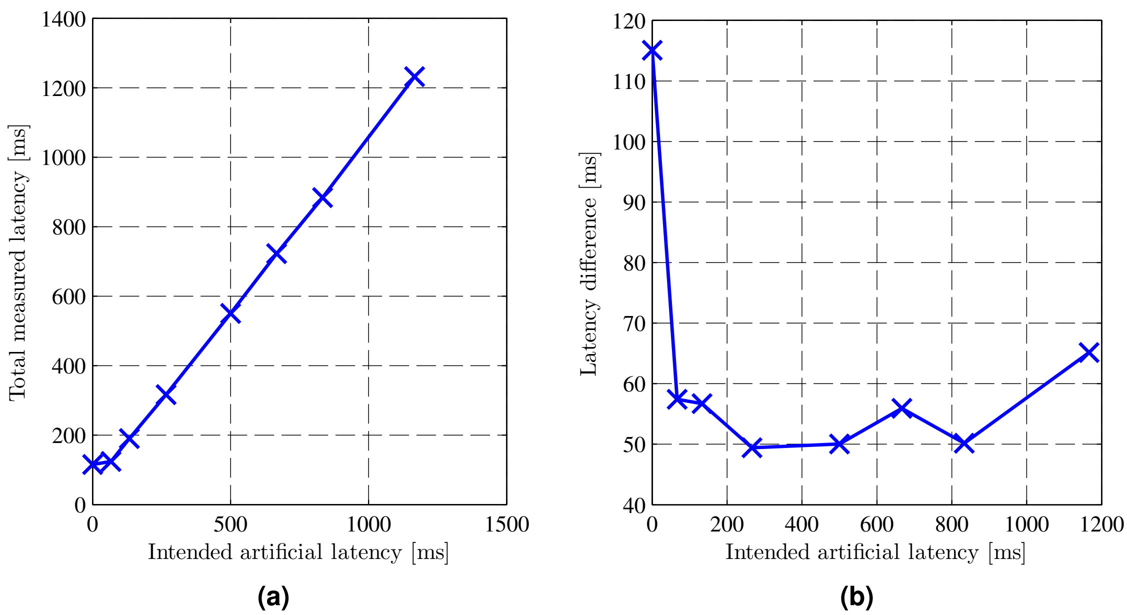


Figure 4.7: *(a) depicts the total measured latency over the intended AAL. (b) shows the native latency (i.e. the difference between the total measured latency and the intended AAL) over the intended AAL.*

Figure 4.7b depicts the native latency (i.e. the difference between the total measured latency and the intended AAL) over the intended AAL. In the experiment, this curve is used as a look up table to calculate the true total latency based on the intended AAL.

Experiment Setup

Similar to the blur experiment, each participant played multiple rounds of ARTravelers with different amounts of AAL and tried to achieve as high a score as possible. The participant was not informed about the intention of the experiment, i.e. that there will be increased latency in the game. First, the participant played two introductory rounds: one without AAL and one with a high (333 ms) AAL. These two introductory rounds were not used for further analysis. Then the participant played another five rounds, each lasting 40 seconds, with AAL bucket randomly assigned to each round. We intended to sample the lower AAL space denser and defined the buckets as follows:

- **Bucket 1:** 0 to 1 frames (0 ms to 33 ms) AAL
- **Bucket 2:** 2 to 4 frames (66 ms to 132 ms) AAL
- **Bucket 3:** 5 to 9 frames (165 ms to 297 ms) AAL
- **Bucket 4:** 10 to 16 frames (333 ms to 528 ms) AAL
- **Bucket 5:** 17 to 35 frames (561 ms to 1155 ms) AAL

Over the five rounds, each player was assigned each bucket once. Inside each bucket a random value was generated for each player. After each round, the participants were asked to answer the following questions:

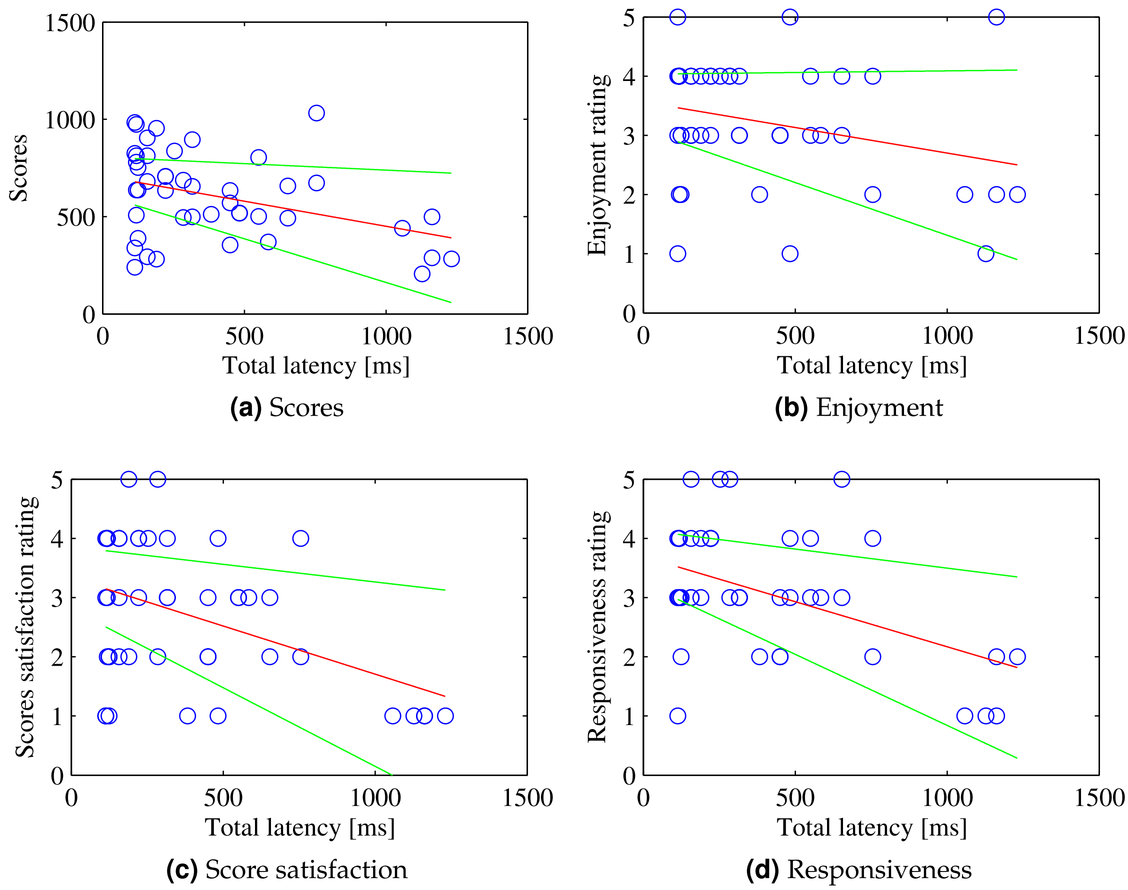
- **Enjoyment:** "How much did you enjoy this run?"
- **Score satisfaction:** "How satisfied are you with your score in this run?"

- **Responsiveness:** ”How responsive did the game feel?”

Each question could be answered on a scale from 1 (not at all) to 5 (very much).

Experiment Results

We used a linear regression model to explain the connection between latency, as the independent variable, and enjoyment, score satisfaction, and responsiveness, as dependent variables.



(a) Scores

(b) Enjoyment

(c) Score satisfaction

(d) Responsiveness

Figure 4.8: *Linear regressions of scores, enjoyment, scores satisfaction and responsiveness over total latency. The red lines depict the fitted linear curves and the green lines define the lower and upper limit of the 95% confidence interval.*

Table 4.4: *Regression analysis results for the dependent variables scores, enjoyment, score satisfaction, and responsiveness over the independent variable, the total latency.*

Dependent variables	R^2	F	p
Scores	0.147	7.40	0.009
Enjoyment	0.077	3.59	0.064
Score satisfaction	0.191	10.13	0.002
Responsiveness	0.221	12.21	0.001

The experiment showed a significant ($p < 0.01$) connection between latency and scores as well as between latency and responsiveness. As expected, but as opposed to the blur experiment, the player could perceive a delay in the game and at the same time his or her performance suffered during high latency rounds, see Figure 4.9 and Table 4.4. However, latency had no significant influence ($p > 0.05$) on the reported enjoyment of the participants. Even in ARTravelers, which requires high concentration but little reaction skills, a negative effect of latency on task performance is apparent.

4.2.4 Realistic Lighting and Camera Artifacts Experiment

As discussed in [Wood et al., 2004], the success of video games depends on a high degree of visual realism. High-quality realistic graphics were rated as important by four-fifths of the participants in their comprehensive study.

In this experiment we investigated the impact of different mixing techniques on the users' perception of realism for AR applications. ARPix is an application that lets a person take a picture of another person posing with an augmented virtual character, Eva, as depicted in Figure 1.5b. The goal of ARPix is to blend

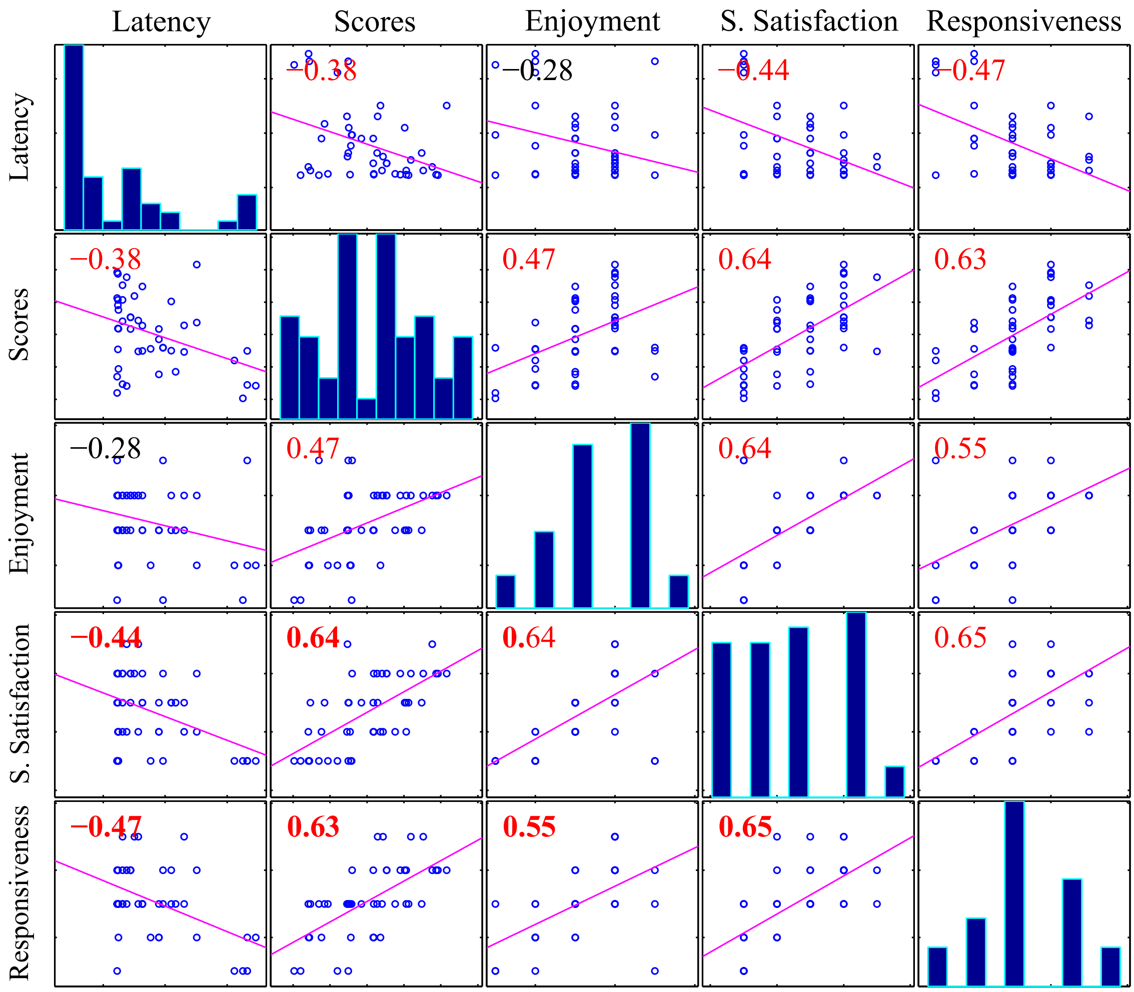


Figure 4.9: *Correlation matrix for the independent variable (latency) and all dependent variables (scores and answers). A red correlation value indicates a significant correlation.*

Eva into the camera image with a high degree of realism. To achieve this goal, two effects are integrated:

A diffuse sphere and a specular sphere mounted at each side of the image marker reflect the lighting condition of the current real scene. ARPix, knowing the positions of the spheres, calculates the lighting condition and applies it to the virtual scene.

Cameras in cell phones and tablets often have a limited lens and sensor quality, resulting in slightly blurry, distorted, and noisy images. These effects can be summarized as camera artifacts. To enhance the realism of our virtually rendered images, artificial camera artifacts (ACA) are added. In our experiment, with enough light, sensor noise was not an issue, and distortion was not visible as Eva is positioned close to the center of the frame. The ACA in the experiment only included artificial blur. The radius of the Gaussian blur filter was manually adjusted to best match the camera image.

Experiment Setup

We installed ARPix, including the marker and the two spheres, and positioned a bright white/blueish lamp to the left side and a smaller reddish lamp to the right side of the user, as depicted in Figure 1.5b. We conducted this experiment at two different locations (1 and 2), with different participants and similar lighting conditions. The participants at location 1 were less experienced in visual computing and games as the participants at location 2.

After taking the picture with the participant, four different image versions with different combinations of the aforementioned effects were generated and presented to the participant. Table 4.5 describes the mapping from the scenario to the contained effects. Correct lighting refers to the lighting captured from the spheres. For the incorrect lighting, we simply created a light shining from the bottom up at Eva.

For comparison, all four images were presented to the user with a randomized position on the screen. The user was then asked to choose the most realistic image (1 to 4). Figure 4.10 depicts a screenshot taken from the image selection screen during the running application. The red letter was added after the screen was captured to indicate the scenario.

Table 4.5: *The effect configurations for the four images presented to the user.*

Scenario	Lighting	Artificial Camera Artifacts
A	Correct	On
B	Correct	Off
C	Incorrect	On
D	Incorrect	Off

Experiment Results

The preferred image votes for both locations are depicted in Figure 4.11. In both locations the participants mainly voted for the image with correct lighting and enabled ACA (scenario A), which confirms our expectation. In location 2 a greater part of the participants voted for configuration B compared to location 1. We believe that this is due to the fact that the percentage of technical people that are working in the field of visual computing was significantly higher in location 2 than in location 1. Participants with a visual computing background may be more sensitive to image effects and may recognize the slightly blurry virtual content and consider it a visual defect, whereas less technical participants would not directly spot the blur but unconsciously feel that the image with ACA enabled blends better into the background.

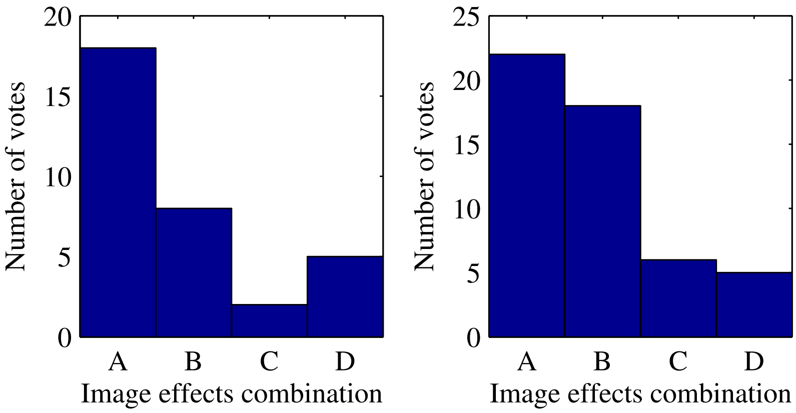
4.2.5 Conclusion

Consistent with an earlier study of motion blur on player's experience of a racing game [Sharan et al., 2013], we observed that even strong motion blur in our mobile Augmented Reality game, ARTravelers, may be noticed by players but does not affect the player's performance or feeling of immersion.

Figure 4.10: *Screenshot from ARPix. Here, the user can select his or her preferred version of the image taken with the virtual character. In 1 and 4 the virtual character is correctly lit and in 1 and 2 she is slightly blurred.*

This was also true in our variations of applied foreground and background blur, specific to AR scenarios. In common with many console action games, our AR game is demanding and requires continuous player attention. Most players, when asked if they noticed differences from round to round, would answer that they were too distracted focusing on the game and did not pay any attention to effects. For such intense games, the players are less likely to notice such camera image composition effects.

Given that immediate camera image synthesis reality mixing effects, such as motion blur, did not impact enjoyment, we further assessed an alternative animation related factor, latency.



(a) Location 1 **(b)** Location 2

Figure 4.11: *Votes for the most realistic picture from a selection of four effect combination scenarios A: correct lighting and artificial camera artifacts, B: correct lighting and no artificial camera artifacts, C: incorrect lighting and artificial camera artifacts, D: incorrect lighting and no artificial camera artifacts.*

As the role of latency in mobile AR applications had not previously been dealt with in terms of overall impact in the user experience, we introduced a latency measuring method based on [Jacobs et al., 1997] and applied a range of additional exaggerated artificial frame delays. This resulted in a strong correlation between lower latency and positive user experience. Significant observable impact of reduced latency for AR mobile games was recorded on realism, enjoyment, satisfaction, matching and score. The strongest of these was realism, and overall, we validate that low-latency is critical to the sense of presence and engagement in mobile AR games.

Finally, we investigated the impact of visual realism in mobile AR for the category of applications where aesthetic quality is a factor. In this test, we measure the impact of responding to dynamic changes in the lighting environment and camera sensor resolution blur matching. Here we found a strong preference to the inclusion of more realistic lighting environment matching, but no noticeable preference for resolution matching between the camera and the rendered AR graphics.

In summary, a series of user experience evaluations of dynamic augmented reality blending approaches were presented. For image effects we detected no significant impact on the user experience for mobile AR games. However, latency was highly significant in this domain. Where the entertainment application domain deals with the importance of aesthetic quality, such as in a self portrait AR photo app, the realism oriented reality mixing measures were found to be important.

In our targeted set of experiments, we have shown several interesting observations and conclusions. Latency clearly warrants further scrutinization, as AR systems improve performance, the effects of varying latencies below 50 ms is particularly interesting as signaled by this study. More generally, this work motivates further research, such as a more extensive study of the influence of motion characteristics across different genres of AR games.

4.3 Augmented Reality Coloring Book

Coloring books capture the imagination of children and provide them with one of their earliest opportunities for creative expression. However, given the proliferation and popularity of television and digital devices, traditional activities like coloring can seem unexciting in comparison. As a result, children spend an increasing amount of time passively consuming content or absorbed in digital devices and become less engaged with real-world activities that challenge their creativity. AR holds unique potential to impact this situation by providing a bridge between real-world activities and digital enhancements. AR allows us to use the full power and popularity of digital devices in order to direct renewed emphasis on traditional activities like coloring.

In this thesis, we present an AR coloring book app that provides a bridge between animated cartoon characters and their colored drawings. Children color characters in a printed coloring book and inspect their work using a consumer-grade mobile device, such as a tablet or smart phone. The drawing is detected and tracked, and the video stream is augmented with an animated 3D version of the character that is textured according to the child's coloring. Figure 1.6 shows two 3D characters textured with our method based on the input 2D colored drawings.

Accomplishing this goal required addressing several challenges. First, the 2D colored drawing provides texture information only about the visible portions of the character. Texture for the occluded regions, such as the back side of the character, must be generated. Naïve approaches, such as mirroring, produce poor results since features like the character's face may be mirrored to the back of their head. In addition, without special treatment, texture mapping will exhibit visible seams where different portions of the parameterization meet. Second, our method targets live update, so that colored changes are immediately visible on the 3D model as the child colors. Thus, the texturing challenges must be solved with a very limited computation budget. Third, the pages in an actual printed coloring book are not flat but exhibit curvature due to the binding of the book. As a result, tracking algorithms and texture capture must be robust to page deformation in order to properly track the drawings and lift texture from the appropriate 2D regions. Finally, the practical consideration of authoring costs requires an efficient content creation pipeline for AR coloring books.

Our coloring book app addresses each of these technical challenges. We present a novel texturing process that applies the captured texture from a 2D colored drawing to both the visible and occluded regions of a 3D character in real time while

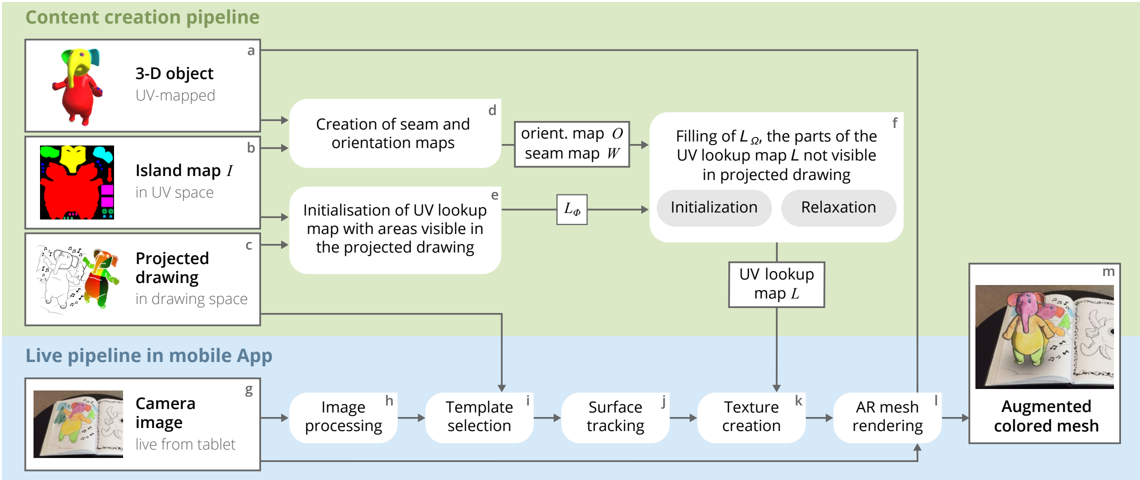


Figure 4.12: *The static content creation and the in-app live surface tracking, texturing, and rendering pipelines.*

avoiding mirroring artifacts and artifacts due to parameterization seams. We develop a deformable surface tracking method that uses a new outlier rejection algorithm for real-time tracking and surface deformation recovery. We present a content creation pipeline to efficiently create the 2D and 3D content. Finally, we validate our work with user studies that examine the quality of our texturing algorithm and the overall experience.

4.3.1 Problem Formulation

Our method for live texturing an AR character from a colored drawing updates the texture of the 3D character at every frame by copying pixels from the drawing. To do so, we create a UV lookup map, that, for every pixel of the texture, indicates a pixel coordinate in the drawing. As the drawing lies on a deformable surface, the later procedure operates on a rectified image of the drawing. We split this process into two separate pipelines, as shown in Figure 4.12. A static *content creation* pipeline creates a lookup map from the work of artists. This pipeline needs to be run only once. A *live* pipeline tracks the drawing and overlays the augmented character on top of it, with a texture created from the drawing in real time using the lookup map.

Content Creation Pipeline

This process aims at finding suitable lookup coordinates on the drawing for the parts of the mesh that are not visible in the drawing, given a UV-mapped mesh and its projection as a drawing. We want to find a lookup map that generates a texture that is continuous over the boundary between hidden and visible parts of the mesh in the drawing and over seams. We also want to fill hidden parts with patterns similar to the ones on visible parts, with minimal distortions between them.

Production Process and Variabes. The artist produces (1) a UV-mapped mesh (Figure 4.12a); (2) an island map I in UV space that defines regions that shall receive similar coloring (Figure 4.12b); and (3) a drawing, which is a combination of a view of the mesh through an edge shader – for printing – and a mapping from coordinates on this drawing to points in the UV map (Figure 4.12c).

Based on these three items, the following objects are created: (1) the wrapping seams W^i for island I^i, mapping some pixels of I^i together; (2) the orientation map O in the UV space, giving a local orientation of the projected mesh structure; and (3) the lookup map L, indicating for every pixel of the character texture which drawing pixel to read. $L = L_\Phi \cup L_\Omega$, where L_Φ are regions visible in the drawing (source regions) and L_Ω are regions not visible in the drawing (target regions).

Lookup Map. The most-challenging part of the content creation pipeline is the creation of L, based on the mesh, the projected drawing, and the island map I. To do so, W, O (Figure 4.12d), and L_Φ (Figure 4.12e) are first created. Then, for every island I^i, a first approximation of L^i_Ω is generated by copying coordinates from L^i_Φ. This approximation is very rough and violates the continuity desideratum, in particular across seams. Therefore, the next step is to enforce this continuity (Figure 4.12f).

We can frame this problem in a similar way as the one of energy minimization in a spring system. Assuming that every pair of neighboring points of L – including across seams – are connected by a virtual spring, relaxing all springs will lead to fulfill the continuity constraints. To do so, we have to minimize the total energy of the system:

$$\sum_{(p,q)\in N_L} k_{p,q}(||L[q] - L[p]|| - 1)^2 \tag{4.1}$$

for all pairs p, q which are either direct neighbors or seam neighbors in L; $k_{p,q}$ being a constant specific to the pair (p, q) that compensates the texel density in the UV map, which is not constant compared to the density in the model space. This correction is necessary for the generated texture motifs to be unstretched.

Live Pipeline

Given a set of colored drawing templates (Figure 4.12c), the app takes the live camera stream as the input (Figure 4.12g), detects (Figure 4.12i) and tracks (Figure 4.12j) in 3D one of the templates appearing in the camera image. In this work, we explicitly account for the fact that the drawing paper may not be flat and may change its shape during the coloring, which is a common situation for books. Once the drawing shape is tracked in 3D, accurate color information from the colored drawing can be retrieved for our texturing algorithm, unlike other existing works [Clark and Dunser, 2012] that can only deal with planar book pages. The augmented character is then overlaid on the input image by using the retrieved colors and the 3D pose of the drawing (Figure 4.12m).

Image Processing. Given the camera image stream of the colored drawing, we want to process the input image so that the colored drawing appearance is as close to the original template as possible (Figure 4.12h). Our approach achieves this by exploiting the fact that it is a line art drawing. This step is necessary because the appearance of the drawing changes significantly due to the coloring.

Template Selection. After the input image is processed to be close to the original line art drawing templates, the system automatically detects which template is appearing in the camera stream (Figure 4.12i). The selected drawing template is used as the template image in our template-based deformable surface tracking algorithm and later for drawing the augmented character.

Deformable Surface Tracking. Allowing the page to deform creates many challenges for the tracking algorithm since the number of degrees of freedom in deformable surface tracking is much higher than that in rigid object tracking. Our deformable surface tracking (Figure 4.12j) builds upon previous work [Ngo et al., 2015] and makes it fast enough to run in real time on mobile devices and robust enough to handle colored line art drawings. We will formulate the shape reconstruction problem as shown in [Ngo et al., 2015].

We represent the drawing page by a predefined 3D triangular mesh with vertex coordinates stacked in a vector of variables $\mathbf{x} = [\mathbf{v}_1; \ldots; \mathbf{v}_{N_v}]$. The drawing page in its rest shape corresponds to the triangular mesh in the flat rectangular configuration with the paper size as dimensions. To make the template appear similar to the one captured by the camera, we synthetically render a reference image in the camera view in which the template image occupies 2/3 of the camera view.

To reduce the problem dimension, instead of solving for all mesh vertex coordinates, we use a linear subspace parametrization that expresses all vertex coordinates as a linear combinations of a small number N_c of control vertices, whose coordinates are $\mathbf{c} = [\mathbf{v}_{i_1}; \ldots; \mathbf{v}_{i_{N_c}}]$, as demonstrated in a previous work [Ostlund et al., 2012]. There exists a linear relation $\mathbf{x} = \mathbf{Pc}$, where $\mathbf{P}$ is a constant parametrization matrix.

Given point correspondences between the reference image generated above and an input image, recovering the colored drawing shape in this image amounts to solving a very compact linear system

$$\min_{\mathbf{c}} \quad \|\mathbf{MPc}\|^2 + w_r^2 \|\mathbf{APc}\|^2, \text{ s.t. } \|\mathbf{c}\| = 1 \,, \tag{4.2}$$

in which the first term enforces the re-projection of the 3D reconstructed mesh to match the input image data encoded by matrix $\mathbf{M}$, the second term regularizes the mesh to encourage physically plausible deformations encoded by matrix $\mathbf{A}$, and w_r is a scalar coefficient defining how much we regularize the solution. This linear system can be solved in the least-square sense up to a scale factor by finding the eigenvector corresponding to the smallest eigenvalue of the matrix $\mathbf{M}_{w_r}^T \mathbf{M}_{w_r}$, in which $\mathbf{M}_{w_r} = [\mathbf{MP}; w_r \mathbf{AP}]$. Its solution is a mesh whose projection is very accurate but whose 3D shape may not because the regularization term does not penalize affine deformations away from the reference shape. The initial shape is further refined in a constrained optimization that enforces the surface to be inextensible

$$\min_{\mathbf{c}} \ \|\mathbf{MPc}\|^2 + w_r^2 \|\mathbf{APc}\|^2, \text{ s.t. } C(\mathbf{Pc}) \leq 0 \,, \tag{4.3}$$

giving the final 3D pose of the drawing in the camera view. $C(\mathbf{Pc})$ are inextensibility constraints that prevent Euclidean distances between neighboring vertices from growing beyond their geodesic distance in the reference shape.

Texture Creation and Mesh Rendering. Once the 3D shape of the colored drawing has been recovered in the view of camera, the mesh is re-projected onto the image plane. This re-projection defines a direct mapping between the pixels on the original drawing template and the pixels on the image of the colored drawing. We then can generate the texture for the character mesh using the lookup map L (Figure 4.12k). Using the live view as the background image for the 3D scene, and using proper parameters for the virtual camera, we can render the augmented character in the 3D pose of the page using the generated texture from the drawing (Figure 4.12l).

In Section 4.3.3, we will present in detail how we tackle the steps described above and compare to [Ngo et al., 2015] our improved method to achieve a robust, accurate, and real-time tracking system for colored drawings.

4.3.2 Implementation

Generation of the UV Lookup Map L

To create the visible part L_Φ of the UV lookup map:

1. An artist prepares the texture-mapped mesh and the island map I. Each island has a unique color, unused boundary pixels are marked as well.
2. An optional mask can be used to exclude some pixels of L_Φ. This allows, for example, ignoring the printed black outlines in the coloring book. The artist is responsible for deciding which pixels to exclude.
3. A *unique color* UV texture is generated, in which every pixel has a color that encodes its position.
4. The mesh is rendered using the unique color texture, producing an image where each pixel encodes the UV position that was used to create it.
5. L_Φ is computed by traversing each pixel in the rendered image, computing its position in the UV space, and recording its destination at that location.

Seam Map. The seam map W is created from the input mesh and the island map I. The resulting seams describe which points in the UV space correspond to the same points on the mesh. Only seams within a UV island are considered.

1. The mesh is transformed into a graph in UV space; each vertex is mapped to one or more nodes having exactly one UV coordinate and an island index.
2. All sets of nodes that correspond to the same vertex are added to a candidate set. Edges connecting nodes that are part of the candidate set are added to the graph.
3. A set of pairs of seams is extracted by traversing these edges starting at the candidate points, while ensuring that a seam pair consists of corresponding nodes and has the same topology.
4. Seam edges are rasterized so that not only vertices, but all points on seams are linked together in W.

Orientation Map. The orientation map O encodes a locally-consistent orientation derived from the edges of the input mesh in UV space.

1. The mesh is transformed into a graph in UV space, where each vertex is mapped to one or more nodes.
2. For each node, we normalize the directions of connecting edges and cluster them into direction groups. If there are two clusters, we store the average directions, otherwise we ignore the node.
3. The orientation is made consistent by traversing the graph and by rotating the directions of nodes.
4. The projected mesh in UV space is rasterized to assign to each pixel the orientation of the nodes of the corresponding face.

UV Lookup Map. The algorithm for creating an initial lookup map is shown in Algorithm 4. It takes a two-step approach.

First, it creates an initial mapping by copying coordinates from the known region (source) into the unknown one (target) in a way that maintains continuity constraints at the border between these regions. Second, it removes discontinuities at seams by expressing neighboring constraints as a spring system, whose total energy must be minimized.

The creation of the initial mapping works by copying coordinates from one side of the border to the other side. The algorithm first propagates a gradient from the border of the source region into the target region. The generation of that gradient uses the E* algorithm [Philippsen, 2006], which is a variation of A* on a grid with interpolation. Then, starting from points close to the source, for every point in the target the algorithm counts the distance to the source, by tracing a path following the gradient. Once in the source, the algorithm continues to trace the path in the already-mapped area until it has run for the same distance as it did in the target. If the tracing procedure encounters the end of the already-mapped area, it reverses the orientation of the path. This procedure leads to rough copies of the source area being written into the target area.

Algorithm 4 shows the most common execution path. In addition to this normal path, the island I and the mesh can be arbitrarily pathological: some areas in Ω might be unconnected to Φ or there can be saddle points in the gradient. Therefore, the algorithm needs a procedure to recover from exception cases; this is represented by FIXBROKENPOINTS(). While iterating over points in Ω, the algorithm collects all points that fail to reach a valid point in Φ, and stores by island i them for further processing. Then, for every of them it checks whether one of its neighbor is valid, and if so, it copies its mapping. For the remaining points, which typically belong to unconnected regions in Ω^i, it groups them in connected blobs and tries to copy a consistent mapping based on the center of the largest region in Φ^i.

If some pixels cannot be copied, the algorithm assigns a point from Φ^i.

The algorithm then solves Eq. (4.1) by performing iterative relaxation. For each point, it attaches a spring to its neighbors (4-connectivity) and, if this point is on a seam, to the point on the other side of the seam (using the seam map W). The spring constant is adjusted to account for the distortion in the UV map. The algorithm iteratively relaxes all springs, using a simple iterative gradient descent method. The relaxation stops if the error does not diminish for a continuous number of steps. In our experiments, we set this number to 8; we set the maximum number of iterations $c_{\max}$ to 100,000 but the algorithm typically stops early after 4,000–20,000 steps.

4.3.3 Deformable Surface Tracking

In this section, we describe our algorithm to detect and track in 3D a possibly non-flat deformable colored line drawing paper. We rely on wide-baseline feature point correspondences between the reference image and the input image. For this, we propose to use BRISK [Leutenegger et al., 2011] in place of the memory-intensive Ferns [Ozuysal et al., 2010] used in [Ngo et al., 2015]. Since many of the correspondences are erroneous, we propose a new outlier rejection algorithm, which is faster and more robust than the one used in [Ngo et al., 2015]. We reformulate the reconstruction energy function to gain speed while not sacrificing accuracy. We rely more on frame-to-frame tracking to gain frame rate and only apply the feature detection and matching once every N frames to retrieve back lost tracked points and accumulate good correspondences. A mask indicating the region of interest in which to look for feature points is also constructed to speed up the feature point detection and limit gross outliers.

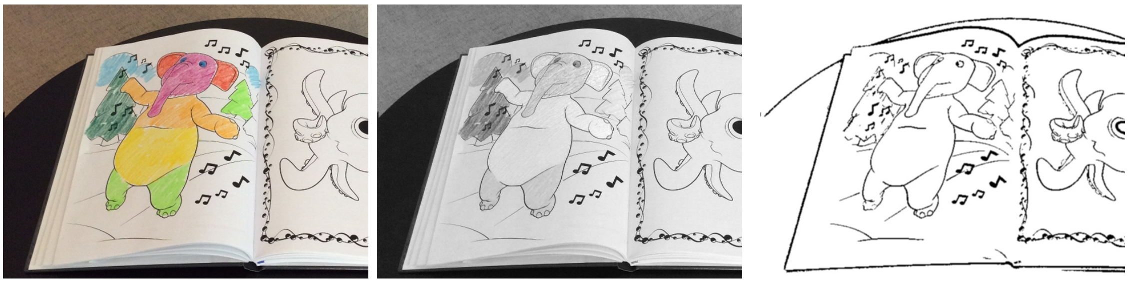


Figure 4.13: *Image processing: (a) Input image of the colored drawing. (b) Luminance channel in* HSV *color space. (c) Resulting line draw image.*

Image Processing

In this step, we apply image processing techniques to the input image so that the appearance of the colored drawing becomes as close to the original template as possible. Because in the next step of the pipeline we describe feature points by binary descriptors using intensity comparisons, this step is especially important to be able to better detect which drawing template is selected and to provide more inlier wide-baseline feature point correspondences between the input image and the reference image.

In the coloring book context, we want to remove colors and keep the black lines in the image of the colored drawing. Therefore, we transform the input image from RGB to HSV color space, in which only the luminance channel is used because it captures most information about the original black line draws. Line draws are more visible in the HSV luminance channel than in the gray scale image. We then apply adaptive thresholding to the luminance image to get a binary line draw image. Small noisy connected components are removed and Gaussian smoothing with standard deviation $\sigma = 1$ pixel is used to remove the staircase effect of binarization. This process is demonstrated in Figure 4.13.

Our color removal procedure is completely automatic without having to adjust parameters for different capturing scenarios, in contrast to [Clark and Dunser, 2012], where the thresholds must be adjusted for different lighting conditions and for different cameras.

Template Selection

To detect which template appears in the camera, we compare the processed input image with a set of given templates. We detect a sparse set of feature points and use a voting scheme to determine which template is currently visible and should be selected for further steps.

Later in the shape reconstruction phase, we also need to establish feature point correspondences between the selected template and the input image. In our problem context, it is necessary to choose a suitable feature point detection, description, and matching method that has reasonable memory and CPU consumption, and provides good quality matches. We have tested a number of methods including Ferns [Ozuysal et al., 2010], BRIEF [Calonder et al., 2010], ORB [Rublee et al., 2011], FREAK [Alahi et al., 2012], BRISK [Leutenegger et al., 2011], SURF [Bay et al., 2006], and found BRISK to be the most suitable choice for our application. BRISK detects scale and rotation invariant feature points, the descriptors are binary vectors, which can be compared very quickly using NEON instructions widely available in mobile processors nowadays, and the matching quality is good enough for our purpose. Clark et al. [Clark and Dunser, 2012] uses SURF in their coloring book application, which we found too computationally intensive. Differently, in the context of deformable surface tracking, [Ngo et al., 2015] extracts scale and rotation invariant interest points as maxima of the Laplacian and then uses the Ferns classifier [Ozuysal et al., 2010] to do the matching. However, Ferns is so memory intensive that it is not suitable for our real-time tracking on mobile devices.

We detect about 500 feature points on each generated reference image, and extract their binary descriptors. Doing so allows all templates to have an equal chance of being selected in the voting scheme. We automatically adjust parameters to extract between 300–2000 features points on the processed input image. The idea of our voting scheme is that each feature descriptor in the input image will vote for one template which has the closest descriptor according to the Hamming metric.

Although comparing two binary descriptors can be done very quickly using NEON instructions, performing brute-force search for the nearest neighbor in the template descriptor database is prohibitive. We instead project all binary descriptors to a low d-dimension space ($d = 7$ in our settings) using a pre-generated random projection matrix. K nearest neighbors ($K = 100$ in our settings) in this low dimension space can be searched very quickly using k-d trees. It also scales well with respect to the number of templates. We use the *libnabo* library [Elseberg et al., 2012] to perform K-nearest-neighbor search. We then go back to the original binary space and select the nearest descriptors among these K candidates. A Hamming threshold is used to filter truly good matches. The template with highest votes is selected.

Outlier Rejection

Once the template has been selected, wide-baseline correspondences are established between the template and the processed input image. This matching step for only two images is done very quickly using brute-force search in the Hamming metric. We obtain a set of putative correspondences, and outliers must then be removed.

Outlier Rejection in 3D. The method in [Ngo et al., 2015] eliminates erroneous correspondences by iteratively solving Eq. (4.2), starting with a relatively high regularization weight

w_r and reducing it by half at each iteration. The current shape estimate is projected on the input image and the correspondences with higher re-projection error than a pre-set radius are removed. This radius is then reduced by half for the next iteration. Repeating this procedure a fixed number of times results in a set of inlier correspondences.

This outlier rejection procedure can reject up to about 60 percent of outliers as reported in [Ngo et al., 2015]. One of its main limitations is that it solves for a 3D mesh, which involves depth ambiguity and a larger number of variables compared to a 2D mesh. We propose to perform outlier rejection in 2D similarly to [Pilet et al., 2008] but our algorithm can work with both regular and irregular meshes and is much faster thanks to the linear subspace parameterization.

Outlier rejection in 2D. We represent the 2D deformable surface by a 2D triangular mesh and use the regularization matrix **A** mentioned in Section 4.3.1 on the x and y components to regularize the mesh.

Unlike [Pilet et al., 2008], which requires a regular 2D mesh and uses the squared directional curvature of the surface as the smoothing term, our regularization term can work on both regular and irregular meshes. We solve for a 2D mesh, which is smooth and matches the input image. The linear subspace parameterization $\mathbf{x} = \mathbf{P}\mathbf{c}$ still works on a 2D triangular mesh and is used to reduce the complexity of the problem. We solve the following optimization problem:

$$\min_{\mathbf{c}} \quad \rho(\mathbf{B}\mathbf{P}\mathbf{c} - \mathbf{U}, r) + \lambda_s^2 \|\mathbf{A}\mathbf{P}\mathbf{c}\|^2 , \tag{4.4}$$

where **c** represents 2D control vertices, **A** is the regularization matrix, **B** represents the barycentric coordinates of the feature points in matrix form, and **U** encodes the feature point locations in the input image. Further, ρ is a robust estimator whose radius of confidence is r and is defined as

$$\rho(\delta, r) = \begin{cases} \frac{3(r^2-\delta^2)}{4r^3} & -r < \delta < r \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

Instead of introducing a viscosity parameter α and iteratively solving two coupled equations with a random initial solution as in [Pilet et al., 2008], we solve Eq. (4.4) directly using a linear least squares approach with a big starting radius of confidence and reduce it by half at each iteration. The result of this iterative process is a both robust and very fast outlier rejection algorithm.

Note that our 2D outlier rejection does not prevent us from tracking the surface in 3D. We use the obtained set of inlier correspondences to track and reconstruct the surface fully in 3D.

Surface Reconstruction by Detection

Once outlier correspondences are eliminated, we solve Eq. (4.2) only once and scale the solution to give initialization for the constrained optimization in Eq. (4.3). The works in [Ostlund et al., 2012; Ngo et al., 2015] formulate the inequality constraints $C(\mathbf{Pc}) \leq 0$ as equality constraints with additional slack variables whose norm is penalized to prevent lengths from becoming too small and the solution from shrinking to the origin. They solve a complex optimization problem involving extra variables and hard constraints:

$$\begin{aligned} \min_{\mathbf{c},\mathbf{s}} \quad & \|\mathbf{MPc}\|^2 + w_r^2\|\mathbf{APc}\|^2 + \mu^2\|\mathbf{s}\|^2, \\ \text{s.t.} \quad & C(\mathbf{Pc}) + \mathbf{s}^2 = 0 \, . \end{aligned} \tag{4.6}$$

In contrast, we use soft constraints that allow the edge lengths to slightly vary around their reference lengths. We obtain a simpler optimization problem with fewer variables and still arrive at sufficiently accurate reconstructions for AR purposes. We further use a motion model to temporally regularize the solution.

Since the tracking frame rate is high, a linear motion model is enough. We solve

$$\begin{aligned} \min_{\mathbf{c}} \; & \|\mathbf{MPc}\|^2 + w_r^2 \|\mathbf{APc}\|^2 + \lambda^2 \|C\left(\mathbf{Pc}\right)\|^2 \\ & + \gamma^2 \|\mathbf{c}_{t-2} - 2\mathbf{c}_{t-1} + \mathbf{c}\|^2 \,, \end{aligned} \tag{4.7}$$

in which $\mathbf{c}_{t-1}$ and $\mathbf{c}_{t-2}$ are solutions to previous two frames.

Using the linear motion model, we can predict the 3D pose of the drawing in the next frame and create an occlusion mask where the surface projection should be in the next input image. This technique helps to speed up the feature point detection and matching. It also improves the robustness of the system because gross outliers are limited.

Surface Reconstruction by Tracking

In the tracking mode, we make use of the fact that both the surface shape and the camera pose change only slightly between two consecutive frames. We use the motion model to predict the shape for the current frame and use the result to initialize the reconstruction.

Similar to [Ngo et al., 2015], we track inlier correspondences from frame to frame on grayscale images using the standard Lukas-Kanade algorithm [Bouguet, 2001]. This step brings a great help that allows the system to track under extreme tilts and large deformations, successfully.

We rely on frame-to-frame tracking to gain frame rate and only apply the feature detection and matching once every $N = 10$ frames to retrieve back lost tracked points and accumulate good correspondences. This simple technique turns out to be efficient for our problem.

Interactive Coloring Book

The interactive coloring book app is built using the Unity game engine[3] and runs on Android and iOS. It uses Unity to access the camera through the `WebCamTexture` object, and fetches the pixels into the main memory. These are then passed to a C++ library implementing the deformable surface tracking algorithm. This library tells Unity whether a drawing template is detected and if so returns the 3D shape of the possibly non-flat colored drawing in the camera coordinates. The app then rectifies the image of the colored drawing in the canonical fronto-parallel pose. It does so by projecting the vertices of the triangular mesh representing the drawing 3D shape into the image plane, and using these projected points as texture coordinate to draw a rectified grid with the camera image as texture, in an offscreen render target. Finally, the app renders the camera image using a screen-aligned quad, and overlays the corresponding animated 3D character. The virtual camera for the 3D scene is positioned in function of the location of the tracked surface in the real world. The character is rendered with a texture filled by getting colors from the the rectified camera image through the lookup map L.

4.3.4 Evaluation

Texture synthesis

Performance. Our artist required less than one hour per drawing to create the inputs for the algorithm. Then, the computational cost is about half an hour on a MacBook Pro 2014 2.5 GHz. The implementation currently uses Cython, and could be accelerated massively, should it use the GPU.

Method. We conducted a web-based quantitative user study to evaluate the subjective performance of our lookup-based tex-

[3] http://unity3d.com

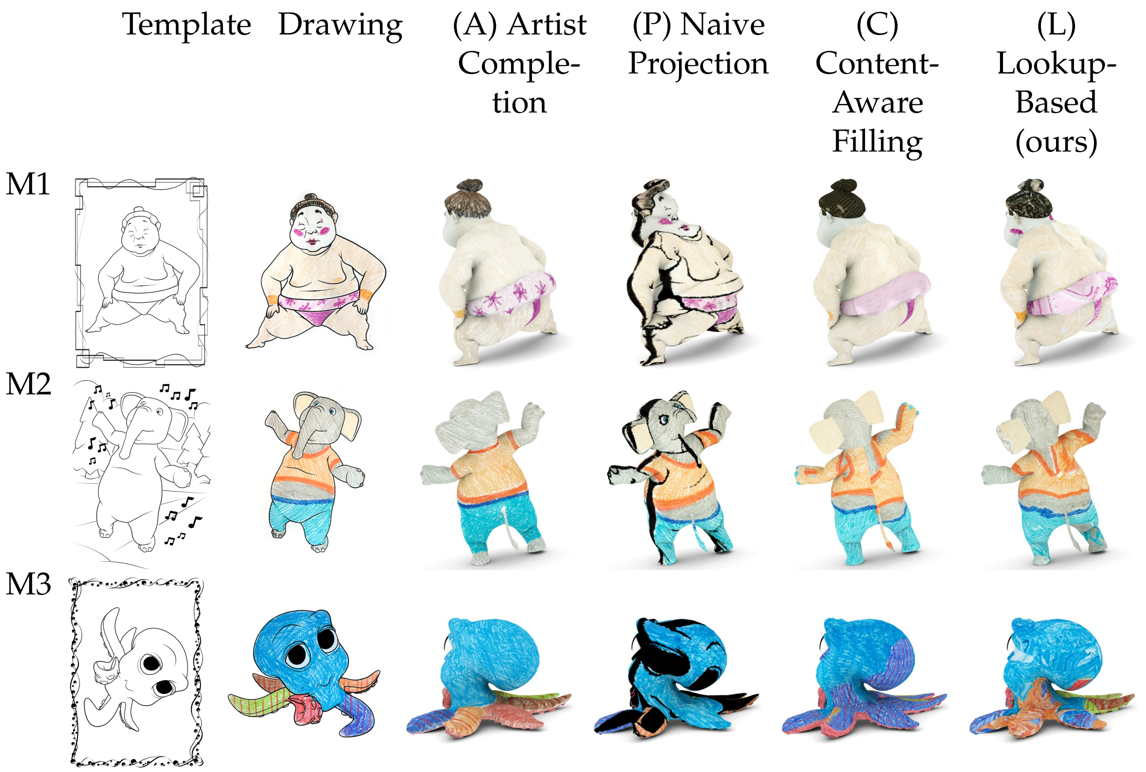


Figure 4.14: *Completion of the texturing of 3D models from a drawing, for different models (M1–M3) and methods (A, P, C, L).*

ture generation method (L), in comparison to three other methods: completion of the texture by a professional artist (A); projection of the drawing to the model from the point of view of the drawing (P); and completion using the Photoshop CS6 *content-aware* filling tool, by applying it to each island separately (C). Methods L and P share the same run-time component: copying pixels from the drawing to the texture using a lookup map, but they differ in how this map is generated. Method C synthesizes pixels in function of the existing patterns, and is thus considerably slower. Finally, method A is a manual operation that takes about 20 minutes when applied by a professional artist.

The participants were presented with 36 screens, each showing an image of a 2D drawing along with two images of a 3D model, each textured with 2 different methods. A short text explained that the coloring of the 3D models were automatically generated from the 2D drawing and the participant was asked to choose the image that looked aesthetically more pleasant. This was repeated for 3 different models, 2 colorings and 4 methods; all combinations were presented in random order. Figure 4.14 shows a selection of these combinations. We collected the number of total votes for each method n_A, n_P, n_C, and n_L for each participant, interpreted as dependent variable (DV). Additionally, we collected the age, gender, and we asked the question whether the participant plays video games at least once a month in average (to measure the participant's general expertise with animated content), as independent variable (IV). The user study was disseminated through professional and personal contacts and by announcement on reddit's *gamedev*, *augmentedreality*, *computervision*, and *SampleSize* channels. We collected the results over a duration of two days.

Results. A total of 314 persons completed the study: 84% male and 16% female, 74% with expertise and 26% without expertise. The median age was 28 (min: 18, max: 78); we use the median statistics as the number of votes is a discrete space. A Friedman test was run to determine if there were differences in n_A, n_P, n_C, and n_L. Pairwise comparisons were performed with a Bonferroni correction for multiple comparisons. The number of votes for the methods were statistically significantly different ($X^2(3) = 816.07, p < 0.001$). Post-hoc analysis revealed statistically significant differences ($p < 0.001$) among all pairwise combinations of methods. Figure 4.15 plots the median and the 95% confidence interval of n_A, n_P, n_C, and n_L, ordered by gender and expertise.

The preferred completion method is the drawing by a professional artist (A). The second preferred method is the content-aware filling of Photoshop CS6 (C), probably because it tends to generate more smooth textures than ours (Figure 4.14, M1-C vs M1-L). However, it does not handle seams, which can lead to dramatic results in some cases (Figure 4.14, M2-C). On the con-

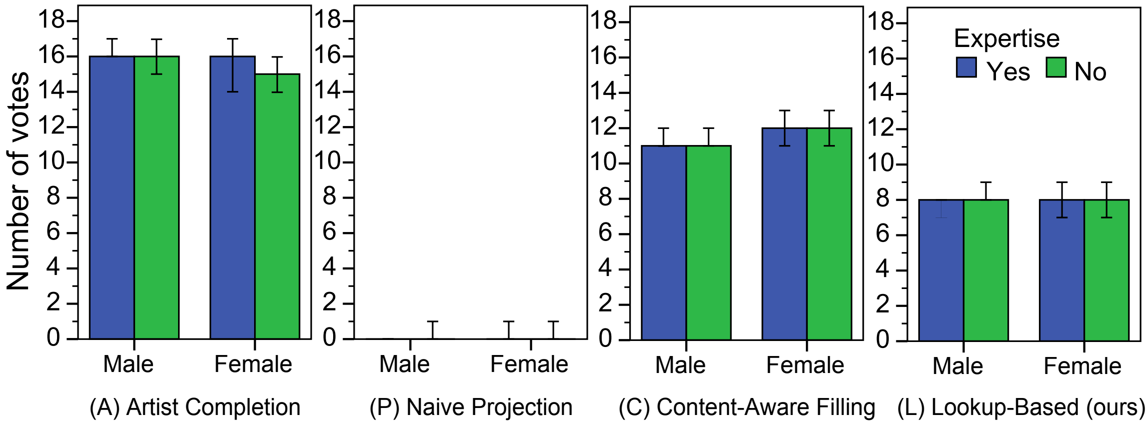


Figure 4.15: *Results from the user study to evaluate our texture synthesis method. The median and the 95% confidence interval are shown for the number of votes for the compared methods (A, P, C, L).*

trary, our method has more graceful degradations (Figure 4.14, M2-L). The naive method P is consistently disliked by everyone. A Chi-Square test revealed that no significant association between all pairwise combinations of IV and DV could be observed ($p > 0.05$), that is, no influence of age, gender, or expertise on the number of votes could be confirmed. This is consistent with the strong agreement of participants and shows that our method, albeit a bit worst than Photoshop's content-aware filling (which is not real-time), is far better appreciated than a naive approach.

Our proposed texture generation algorithm was necessary to meet the real-time requirements of our application. However, being lookup-based, it has some limitations, as being unaware of the users' coloring at run-time. Nonetheless, it provides superior results to a naïve approach, comparable to much slower offline methods. We conclude that it is good enough for real-time application in a coloring book.

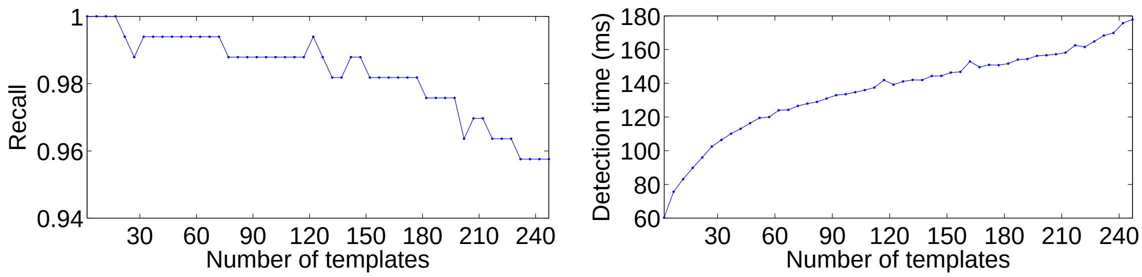
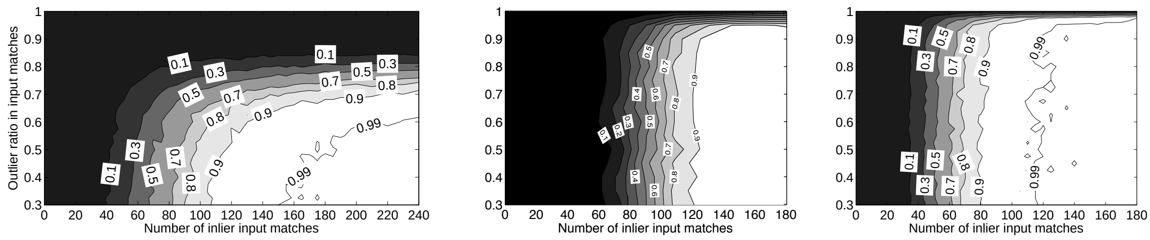


Figure 4.16: *Scalability of the template detection algorithm. Left: recall measure. Right: average detection time.*

Deformable Surface Tracking

In this section, we experimentally validate the robustness, accuracy, and computational efficiency of our methods and show that they are suitable to our coloring book application on mobile devices.

Template Selection. We captured an image sequence of colored drawings in normal settings of book coloring and input it into the template selection module to test how accurately we can detect a template if it appears in the camera. To study the scalability of our detection algorithm, we vary the number of templates from 2 to 250, which is much more than the number of pages a coloring book usually has. We define the recall measure as the ratio of correct detection among the total number of test input images. In our case, false positives are not very severe because these false alarms will be rejected in the outlier rejection step. As shown in Figure 4.16, our template selection algorithm can detect the template accurately and its computational performance scales well with the number of templates. Since we detect the template only once before passing the result to the tracking module, this detection can be run across several frames.



(a) [Ngo et al., 2015] (b) [Pilet et al., 2008] (c) Ours

Figure 4.17: *Probability of having at least 90 % of mesh vertices reprojected within 2 pixels of the solution as a function of the number of inlier matches and proportion of outliers, on the x-axis and y-axis respectively. The lines are level lines of the probability.*

Robustness

We demonstrate the robustness of our outlier rejection scheme described in Section 4.3.3 for planar surfaces and compare it to [Ngo et al., 2015; Pilet et al., 2008]. To evaluate our approach, we used the same image from the standard paper dataset as in [Ngo et al., 2015], in which the surface undergoes large deformations. We used both the corresponding Kinect point cloud and SIFT correspondences to reconstruct a mesh that we treated as the ground truth. Similar to [Ngo et al., 2015; Pilet et al., 2008], we then produced approximately one million sets of correspondences by synthetically generating 10 to 180 inlier correspondences spread uniformly across the surface, adding a zero mean Gaussian noise ($\sigma = 1$ pixel) to the corresponding pixel coordinates to simulate feature point localization errors, and introducing proportions varying from 0 to 100 % of the randomly spread outlier correspondences.

We ran our proposed outlier rejection algorithm with 20 regularly sampled control vertices on each one of these sets of correspondences and defined a criteria to assess the effectiveness of our outlier rejection scheme. The criterion for success is that at least 90 % of the reconstructed 3D mesh vertices project within 2 pixels of where they should. Figure 4.17 depicts the success rates according to this criteria as a function of the total number of inliers and the proportion of outliers. The results reported in Ngo et al. [Ngo et al., 2015] and Pilet et al. [Pilet et al., 2008], which have similar experiment settings as ours, are included for comparisons. Note that unlike us and Ngo et al. [Ngo et al., 2015], Pilet et al. [Pilet et al., 2008] does not add Gaussian noise to the input correspondences. Nevertheless, it takes our method approximately only 80 inlier correspondences to guarantee that the algorithm will tolerate up to 0.95 ratio of outliers with 0.9 probability. The algorithm decays nicely with respect to the number of inlier input matches. It is still 50 % successful given only 50 inlier matches and a 0.95 outlier ratio. Compared to [Ngo et al., 2015; Pilet et al., 2008], our rejection algorithm requires fewer inlier correspondences to detect the surface and can handle a larger portion of outliers.

Figure 4.18 demonstrates the robustness of our outlier rejection algorithm on real images. The algorithm can filter 37 inlier matches out of 422 putative ones. As a result, accurate color information can be retrieved for the texture generation algorithm (see the virtual characters in Figure 4.18).

Pilet et al. [Pilet et al., 2008] reports the total execution time of 100 ms per frame in a 2.8 GHz PC including feature point detection and matching time. Our outlier rejection algorithm is similar except that we solve a much smaller least squares problem. For the outlier rejection algorithm alone, on a MacBook Pro 2014 2.6 GHz laptop, it takes Ngo et al. [Ngo et al., 2015] about 25 ms per frame while it only takes our method about 2 ms per frame on the same input correspondences.

Reconstruction accuracy. We compared the reconstruction accuracy of our method described in Section 4.3.3 against [Ngo et al., 2015] which is representative of the current state-of-the-art

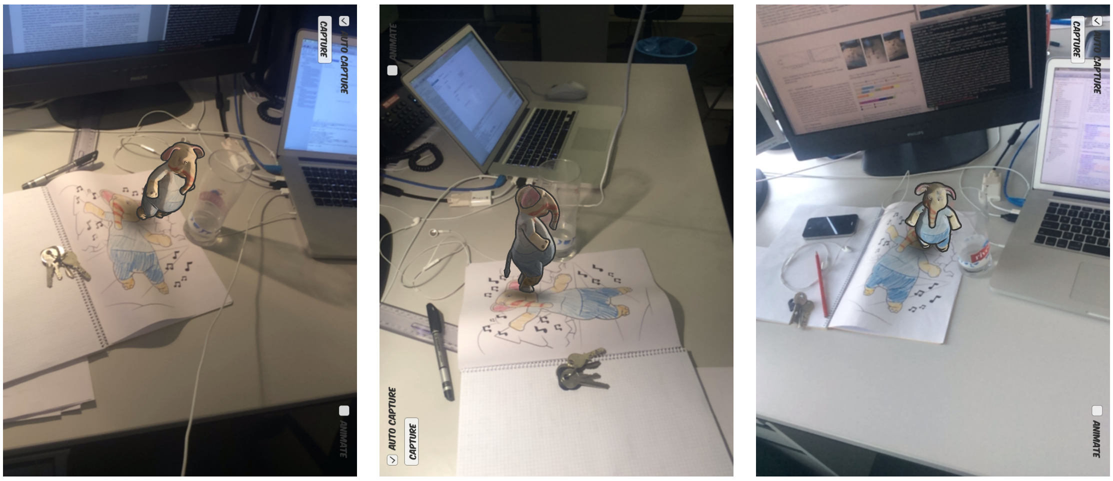


Figure 4.18: *Screenshot from the app showing the robustness of our outlier rejection algorithm.*

in template-based deformable surface reconstruction. We used the same dataset as for the Robustness evaluation (see preceding section). This dataset provides 193 consecutive images acquired using a Kinect camera [Varol et al., 2012]. The template is constructed from the first frame, and 3D reconstruction is performed for the rest of the sequence using the image information alone. Both methods use 20 regularly-placed control vertices. We used the Kinect point cloud and SIFT correspondences to build ground truth meshes, and compute the average vertex-to-vertex distance from the reconstructed mesh to the ground truth mesh as the measure of reconstruction accuracy. The average reconstruction accuracy of [Ngo et al., 2015] is 2.83 mm while our average reconstruction accuracy is 2.74 mm, which is comparable. However, on a MacBook Pro 2014 2.6 GHz laptop, it takes [Ngo et al., 2015] about 70 ms per frame to solve the optimization problem in Eq. (4.6), while it takes our method only 3.8 ms to solve the optimization problem in Eq. (4.7).

Timing. In this section, we look in detail at the timing of the major steps (see Section 4.3.3) of our deformable surface detection and tracking algorithm on an iPad Air on which we prototyped our app. We used a sample coloring book with 3 drawing tem-

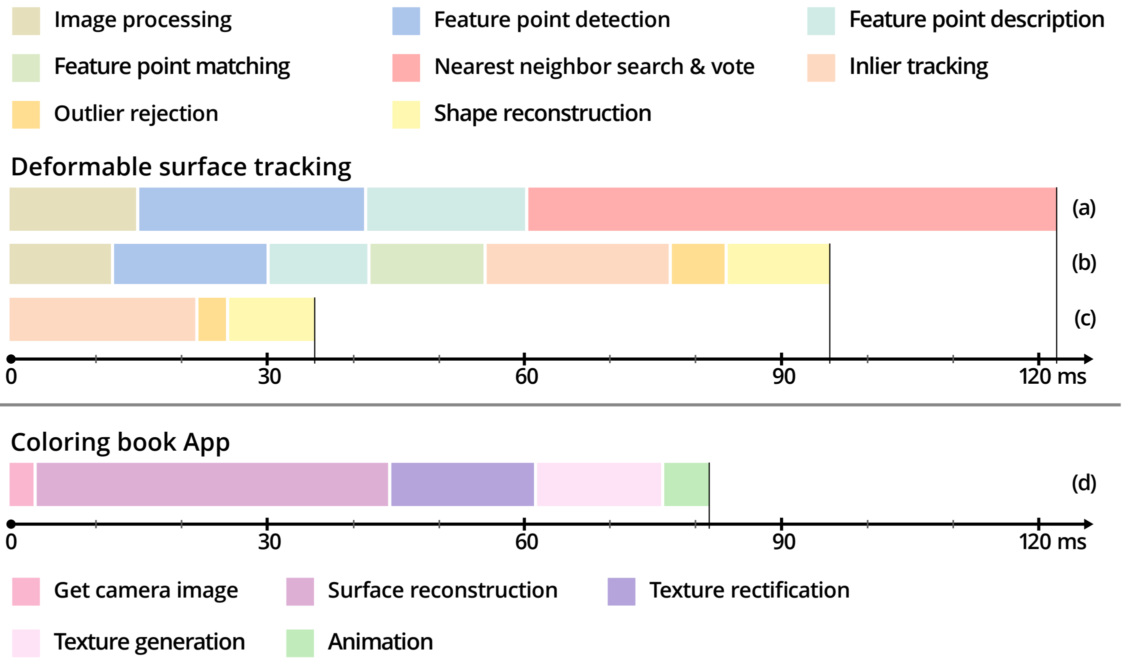


Figure 4.19: *Timing measurements of major steps of the deformable-surface tracking algorithm and the coloring book app. (a) Template selection. (b) Surface reconstruction in the detection plus tracking mode. (c) Surface reconstruction in the tracking mode. (d) Coloring book app.*

plates and measured the average time spent on each step. The results are shown in Figure 4.19.

In the template selection mode, the total time spent per frame is 122 ms (Figure 4.19a). These measurements were averaged over 480 frames. In the surface reconstruction by detection plus tracking mode, both surface reconstruction by detection and frame-to-frame surface tracking are performed. The reconstruction time per frame is 96 ms, averaged over 50 frames (Figure 4.19b). Once the colored drawing is properly tracked, using a mask indicating the region of interest to look for feature points decreases the time spent on feature point detection and description because only a small region of the input image is processed and disturbing feature points are filtered out. In the surface reconstruction by tracking mode, image processing, feature detection, description, and matching are not required. The average total reconstruction time per frame is 36 ms (Figure 4.19c). These measurements were averaged over 250 frames.

Most of the time of our algorithm is spent on feature point detection, description, and matching. We perform surface reconstruction by detection once every 10 frames, and perform surface reconstruction by tracking on the rest of the frames. On average, it takes the reconstruction 41 ms per frame. Our approach has some limitations: the computational cost varies for different frames and we have not fully exploited the tracking context yet. To avoid feature point detection, description and matching, a future work is to incorporate active search to locally look for correspondences in tracking mode. Doing so, we believe our deformable surface tracking method will have a performance closer to its planar rigid object tracking counterpart.

Interactive Coloring Book

App Performance. The app runs at about 12 frames per second (FPS) on an iPad Air (first generation). Figure 4.19(d) shows the breakdown of the time for one frame, averaged over 600 measurements. Most of the time, 41 ms, is spent on tracking the deformable surface. Rectifying the texture to match the drawing template takes 17 ms, while creating the texture using the lookup map L takes 15 ms. The latter step could be strongly optimized by doing it through a shader. Finally, acquiring the image from the camera and rendering the scene takes 9 ms.

Since, in the context of a coloring book, the deformation of the drawing paper is relatively small, we experimentally observed that 20 control vertices out of total 110 vertices are enough. The corresponding app frame rate is 12 FPS. The frame rate decreases to 9, 6, 5, 3 FPS with 40, 60, 80, 110 control vertices, respectively.

Our approach relies on feature points, whose discriminativeness in line art drawings is not as high as in natural textures. It thus requires the drawing to have sufficient detail. To reliably track the whole page, we also rely on feature points detected around the character, for instance in decorations printed in margins. Occlusions due to the hand of the user are handled well because besides tracking feature points from frame to frame, which causes drifts and point losses, we also periodically perform feature points detection and matching to retrieve back lost tracked points, correct drifts, and accumulate good correspondences. The frame rate of 12 FPS, however, is not enough to handle sudden motions. In future work, optimization of implementation would improve the frame rate and provide better user experience.

Method End-User Evaluation. We conducted a user study to assess the subjective performance of our app. Each participant was asked to choose one out of three templates (see Figure 4.14) and color it with a set of colored crayons. There was no time limit and multiple subjects were allowed to sit together, chat, and color their chosen template. An iPad Air running our app was available and the participants were asked to occasionally inspect their coloring through the app. Upon finishing coloring, the participants were asked to answer the following questions: Tablet Usage: *How often do you use a tablet in average?*, Drawing Skill: *Does this augmented reality app affect your drawing skills?*, Drawing Motivation: *Does this augmented reality app affect your motivation to draw?*, Connection to Character: *Does this augmented reality app affect your feeling of connection with the cartoon character?*, and Would Use App: *If this app was available for free, would you use it to draw characters?*. Additionally, we recorded their age, gender, and their comments. In the analysis we treated Tablet Usage, age, and gender as IV and Drawing Skill, Drawing Motivation, Connection to Character, and Would Use App as DV.

Results End-User Evaluation. We recorded answers from a total of 40 participants, 38 % male and 62 % female. The median age was 23 (min: 18, max: 36). While our app is designed to be used by children, due to delays with ethical approval for the experiment, we did not quantitatively test it with this age group. Despite this limitation of our study, informal tests showed a strong appreciation and engagement. In a further work, additional testing with children is desirable to quantify these. Figure 4.20 shows a histogram of the answers, for the different questions. Chi-Square tests revealed that no statistically significant association could be observed between any pairwise combination of IV and DV ($p > 0.05$).

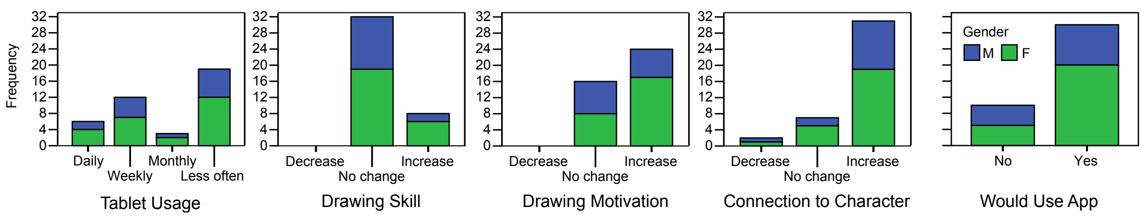


Figure 4.20: *Histograms of the answers to the questionnaire evaluating the end-user appreciation of the AR coloring book app.*

The majority of participants (60 %) felt that the app would increase their motivation to draw coloring books, while a large minority (40 %) said that their motivation was not affected. This shows that while the app should be made more engaging, it already has a positive effect on motivation. The app is perceived as having a minor pedagogical value, as 20 % of the participants said that it affected their drawing skills positively. Almost all participants (80 %) felt that the app increased their feeling of connection to the character, except for 2 people who actually felt the contrary. This may be due to imperfect tracking for their actual drawings, ruining the experience. A majority (75 %) of the participants would use this app, if it was available for free. About a third of the participants gave additional comments, mostly a short sentence of appreciation. Two participants noted that the texture synthesis was not perfect, but found it impressive nevertheless. One suggested to add the possibility to control the virtual character. These results show that the proposed solution is mature and well appreciated, and contributes to enrich the coloring experience.

4.3.5 Conclusion

In this work we have demonstrated an AR app that allows children to color cartoon characters in a coloring book and examine an animated 3D model of their drawing. We concluded from a user study that this app strongly improves the sense of connection with the character, and motivates people to draw more. This app is built upon two novel technical contributions in the field of (1) real-time texture synthesis and (2) deformable surface tracking. For the first, we formulated the problem of generating a texture in real time and proposed a solution by creating a lookup map to copy existing pixels from the drawing. We proposed a novel algorithm for artists to generate this map automatically. For deformable surface tracking, we provided a new template selection and a new outlier rejection mechanism, as well as a lighter problem formulation, which together allow real-time tracking on a mobile device. While we have used these contributions to enable our AR coloring book, we believe they have the potential to impact many more applications in the field of visual computing.

Algorithm 4 Initial Creation of the Lookup Map

```
procedure GenerateLookup(I, L_Φ, W, O)
    ▷ First approximation
    L ← ∅
    for i in enumerate(I) do
        G ← generate gradient for island i
        L^i ← L^i_Φ                                ▷ initialize with source
        for p in sorted(Ω^i, G) do                 ▷ for points in target
            d, p′ ← 0, p
            while p′ ∉ L^i_Φ do                    ▷ until source reached
                p′ ← descend G from p′
                d ← d + 1                          ▷ count distance in target
            end while
            v ← incoming direction                 ▷ enter source
            while d > 0 do                         ▷ trace same dist. as in target
                if p′ ∉ L^i then                   ▷ unknown mapping
                    v ← −v                         ▷ reverse direction
                else
                    L^i[p] ← L^i[p′]               ▷ copy mapping
                end if
                rotate v using O[p′]               ▷ follow source orientation
                p′, d ← p′ + v, d − 1
            end while
        end for
        L ← L ∪ L^i                                ▷ merge lookup for this island
    end for
    FixBrokenPoints()
    ▷ Relaxation
    e ← ∞                                          ▷ iterative relaxation
    for c in range(c_max) do
        L, e′ ← RelaxLookup(L, W)
        if e′ > e then                             ▷ if error increases...
            break                                  ▷ ...stop early
        end if
        e ← e′
    end for
    return L
end procedure
```

CHAPTER 5

Applications

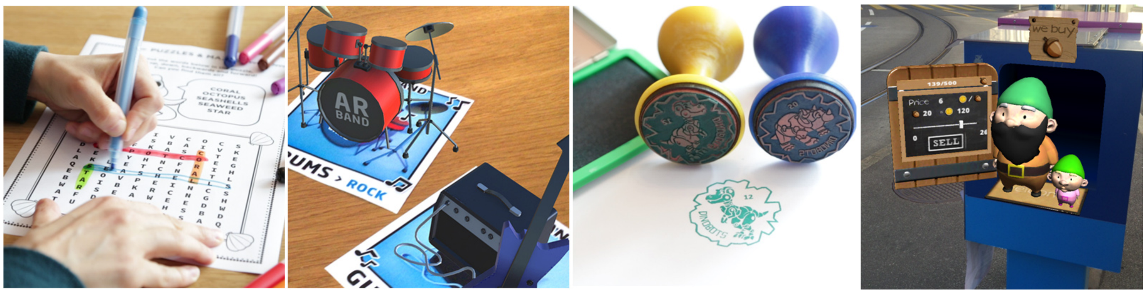


Figure 5.1: *Augmented Reality apps that explore creativity and interactivity. From left to right: AR puzzle app, AR music app, AR stamps app, AR multiplayer trading game.*

5.1 Introduction

This chapter presents the applications domain of this thesis. Various prototype applications that explore Augmented Reality (AR) and storytelling are presented and discussed in Section 5.2. The focus lies on the concept Augmented Creativity, which employs AR to enhance creative activities. In the second part, in Section 5.3, our AR city-wide trading game, Gnome Trader, along with its economy simulation and evaluation is discussed in more detail.

5.2 Augmented Creativity

Creative play allows children to engage their imagination as they explore and interact with the world around them. This process of active discovery contributes to a rich and memorable childhood and provides a foundation for creative problem solving skills that are crucial throughout an individual's life. Physical interaction with one's surroundings is a critical component of the experiential learning that lies at the heart of creative play. However, as prepackaged content and digital devices become more popular and ubiquitous, children spend an increasing amount of time passively consuming content or absorbed in the digital world. Television and video games dominate while the creative activities of exploration and discovery suffer in this increasingly digital world.

AR holds unique potential to impact this situation by providing a bridge between real-world activities and digital enhancements. AR allows us to use the full power and popularity of mobile devices to direct renewed emphasis on the traditional activities of creative play. We refer to this concept of enhancing real-world interaction, discovery, exploration, and imagination through AR as Augmented Creativity.

Creativity can be expressed and experienced in various domains. We discuss the concept of Augmented Creativity within six domains, *seeing*, *hearing*, *strategizing*, *exploring*, *imagining*, and *learning*, and present nine working prototype applications that each form a unique bridge between the real and digital worlds in order to enhance creativity, support education, and open new interaction possibilities.

Educational games pose the challenge of combining educational values with the fun of playing a video game. The result can be unbalanced, with the educational content overshadowing the fun gaming experience. Augmented Creativity aims to combine educational and entertainment values together by integrating them into an AR experience that explores new ways of interacting with the application. Our Augmented Creativity puzzles app provides a seamless bridge between the real-world activity of puzzling and computer-generated animated characters. Our music app provide a tangible way for children to explore different music styles and compose their own version of popular songs. In the gaming domain, we show how to transform passive game interaction into active real-world movement that requires city-wide coordination and cooperation between players. And, finally, we examine how Augmented Creativity can provide a more compelling way to understand and teach complex concepts, such as computer programming.

5.2.1 Applications

Our work focuses both on technical advances in mobile graphics and AR as well as on the creative use of this technology and aims at enhancing the interactions that are possible between the real and virtual worlds. The following sections develop the idea of Augmented Creativity in the context of the six domains of seeing, hearing, strategizing, exploring, imagining, and learning. In each case, we highlight creative and educational goals as well as potential impact.

Seeing Domain

We present a prototype AR applications that focuses on creativity from a visual perspective and enhances puzzle activities found in a children's activity book. Our coloring book app, as presented in Section 4.3, belongs to this domain.

Puzzles App. Traditional coloring and activity books often include different types of puzzles. These activities cultivate concentration by encouraging the reader to focus and to have fun solving puzzles at the same time. Puzzles are often simple games, such as finding words, spotting differences between similar images, or finding a way out through a maze. Extending our repertoire from coloring to activity books, we present an AR puzzle app to enhance puzzling experiences.

As the child solves a printed puzzle in a traditional way by drawing the solution on the page of the book, the app aids and guides the child and verifies the solution, as depicted in Figure 5.2, bottom row. Solving the puzzle triggers graphical effects and animations overlaid on the page when viewed through a mobile device to reward the child. We have currently implemented two kinds of puzzles: find-the-words and mazes. In the former, the user is asked to find and highlight words from a list within a grid of letters, while in the latter, the user has to draw the correct path through a maze. Using HSL color space segmentation of the camera image and subtracting the known marker template, the app detects what the user has drawn in the page and verifies the solution. The corresponding 3D figure is overlaid in AR for every correct word in the find-the-words puzzle. An animated virtual character appears and walks through the maze along the correct path drawn by the

Creative Goal Fosters imagination, allows character individualization, helps to express feelings about character.

Educational Goal Improves coloring skills, 3D perception, challenges imagination, concentration, and puzzle solving skills.

Potential Impact User-painted characters and levels, scripting virtual worlds through coloring.

Figure 5.2: *Seeing domain: Top: Children color 2D characters and inspect their work in 3D on a mobile device (Section 4.3). Bottom: The* AR *puzzle app guides children in solving puzzles and rewards them with effects and animations.*

user. If the drawn path deviates from the correct path, the character stops walking at the last correct point, indicating that the remaining part of the path should be revised.

This app supports, guides, and rewards children during puzzle activities and hence stimulates concentration and creativity. The technology used in this app can easily be extended to a large variety of different paper puzzle games.

Hearing Domain

Music is an integral part of human life and an important aspect of childhood development. We present an application that combines the endless creative dimensions of music with the intuitive spatial interactions offered by AR.

Music Arrangement App. When composing music, the artist comes up with melodies and rhythms, decides on keys, scales, chords, and tempi, and chooses instruments for each part of the composition. At the same time, the composer will adhere to a specific music style while writing the song. While there exist hundreds of styles, the common categories of styles are Alternative, Blues, Classical, Dance, Hip-Hop, Pop, Jazz, Soul, Rock, and World. Music styles often dictate specific rhythmic elements, instruments, and scales. Nevertheless, every song can be re-arranged with different instruments to express a different style. For example, playing a known Punk-Rock song as Reggae, or a Hip-Hop song as Jazz, keeping the theme and the feeling of the song, might be a wonderful and even educational experience for a musician or an audience. With our application, we aim to provide a similar experience to the user.

Our application allows the user to arrange an existing song using different musical elements. We split the available musical elements into two independent dimensions: style and instrument. The user can choose instruments and styles independently and recreate the song as imagined. In our prototype app, we provide Rock/Pop, Jazz, and Latin styles and Piano, Guitar, Drums, and Strings as instruments. To arrange the song, the user is given a collection of physical image markers, each representing a specific instrument for a given music style. By placing a marker on a physical board, an augmented version of the instrument is shown and the corresponding audio is played, as depicted in Figure 5.3.

All markers play the same song in the same key and in the same tempo. The physical position of the marker on the board affects the composition of the corresponding instrument: moving an image marker closer to the camera will result in a louder sound of that instrument while placing instruments on the sides will affect the stereo composition. Hence, the user can intuitively explore different spatial compositions of the band components to produce a unique arrangement of the song.

The application has strong creative and educational aspects. The user learns about music styles and the concept of arrangement and how the same song can be played in varying styles and with different instruments. Additionally, the application shows the importance of the spatial position of the instruments in a band. AR allows the user to change the position and the volume of the instruments while the song is playing, allowing her or him to direct the virtual band. Finally, this project opens up a wide range of future possibilities. For example, an extended multiplayer version of the application would allow users to collaborate in larger bands with different sections to create unique songs.

Strategizing Domain

This section illustrates two AR game prototypes, which require the players to develop creative strategies. The physical-interaction game focuses on cooperative strategy while the AR stamps game boosts competitive strategy.

Physical-Interaction Game. The interaction components in computer games are often limited to screens and input devices and, while the games immerse the player in a compelling virtual world, they remove the player from the real world and thus from physically interacting with other players. In this application, we demonstrate how we can overcome this limitation by employing AR technology in a team-oriented game.

Creative Goal Experiment with different instruments and styles to rearrange your favorite song.

Educational Goal Teaches concepts of arrangements, styles, and the disposition of the band components.

Potential Impact Collaborative music arrangement experience, learn about the disposition of an orchestra.

Figure 5.3: *Hearing domain: The* AR *music arrangement app allows users to re-arrange songs using a variety of instrument markers.*

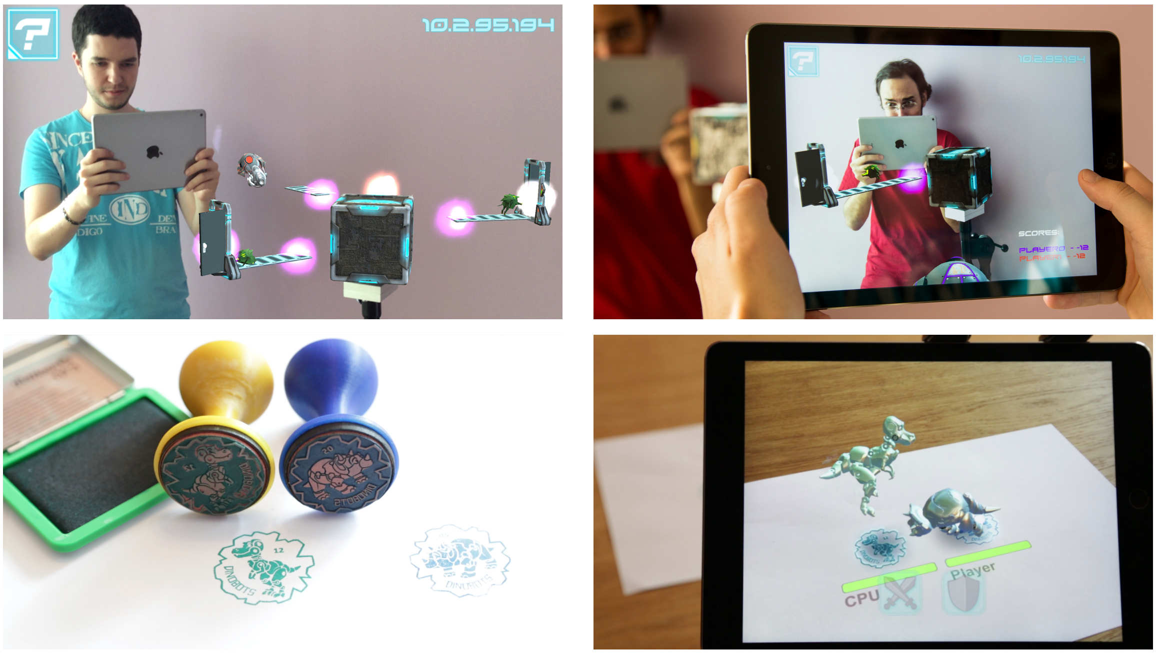
We have developed a networked multiplayer AR game, *ARTravelers*, which offers an immersive digital collaborative experience. In this game, players work together, communicate, coordinate, and synchronize to apply clever strategies to win the game. As an educational instrument, our game teaches children to quickly evaluate the current situation and decide if they should claim points for themselves or if the chance of winning is increased by leaving the points for other team members. Coordination and cooperation is necessary in order to achieve the highest score.

The players surround a cubical AR marker placed in the center of the room, as depicted in Figure 5.4, top row. Each player carries a tablet to virtually interact with the game and the other players. Up to four players can join a match, in which they have to prevent an invading alien force from entering our world. Inside the game, portals are spawned around the cube and the players need to physically align their tablet with each portal to close it before an alien creature passes through it. Players and portals are assigned random colors. Every player can close a portal to score points but a player closing a portal with the matching color receives more points. It is thus in the interest of the team to match the colors of players and portals, the difficulty being that only a player facing a portal can recognize its color. Therefore, the players need to synchronize and optimize their movement, not hindering each other, to successfully close as many portals as possible.

Our game bridges the gap between virtual and real as players simultaneously maneuver around the cube in both the digital and the real world. It explores the concept of Augmented Creativity from the perspective of finding creative strategies to collaboratively win the game as a team. It bears the potential to positively affect the interpersonal relationship between players and to facilitate team-building.

Stamps App. We present a game that bridges physical activity and interaction with animated content through AR using custom made, laser-cut stamps. AR marker images are laser-cut into the bottom of stamps, as depicted in Figure 5.4, bottom row. The current implementation supports two images. In the game, each stamp image represents a different creature and the player is asked to pick one for himself or herself and one for the computer opponent. After applying the stamps on paper, the corresponding virtual creatures appear in AR on top of the stamp image when viewed through a mobile device. In addition, the color of the ink used in the physical stamp is detected and applied to the corresponding virtual creature. The game starts as soon as both creatures appear. In every turn, each player can choose to either attack or defend with his or her creature. This decision is followed by an animation of the fight. The outcome of the turn is computed using a payoff matrix and the defeated creature loses health points. Turns are repeated until one of the creatures runs out of health points. The gameplay is based on the iterated prisoners dilemma [Axelrod, 1993] with a modified payoff matrix and with the *cooperate* and *defect* actions replaced by *attack* and *defend*. The AI opponent strategy implements a slight variation of the famous tit for tat algorithm [Axelrod, 1980].

This game explores a new host of interactions taking advantage of stamps through AR and promotes strategic, competitive play. We envision that a variety of different games can be extended from this concept by incorporating additional stamps that represent new characters and abilities.



Creative Goal Devise creative strategies to collaborate or compete with your team members.

Educational Goal Teaches cooperation and team communication.

Potential Impact Facilitates team-building and opens new ways of interacting through stamp images, shapes, and colors.

Figure 5.4: *Strategizing domain: Top: Our* AR *physical-interaction game requires player to engage in the physical world to win as a team. Bottom: Laser-cut stamps provide a new host of* AR *interactive games.*

Exploring Domain

AR holds the potential to provide a novel interaction at city-wide scales. By overlaying interactive elements on top of the rich existing structure of buildings, parks, and roads, novel virtual environments and innovative experiences can be created. In this domain, we showcase a city-wide treasure hunting framework to aid developers building city-wide treasure hunting games. Additionally, we present an AR city-wide resource trading game, called *Gnome Trader*. As Gnome Trader is discussed in depth in Section 5.3.

Treasure Hunt Framework. Our AR city-wide gaming framework focuses on scavenger hunt games. The player is sent on a quest that leads through the city to find a hidden treasure. The treasure hunt is divided into several chapters. In each chapter, the player visits a locations, which is found with the help of textual directions and clues. Arriving at a locations, the player needs to solve AR puzzles in order to continue to the next chapter, as depicted in Figure 5.5, top row. A building facade or any other visually prominent geometry acts as the AR image marker that is tracked by the app. The puzzles can be arbitrarily complex and are freely implemented by the developer. Various puzzle components are provided in the framework, such as levers, buttons, a rotation lock, a number lock, and a sliding puzzle. The chapters guide the player through historical sites and famous locations of the city and the puzzles are inspired by the location's historic events.

Our city-wide gaming framework was tested and validated in the context of a hackathon challenge during the Ludicious Zürich Game Festival 2014[1]. The participants formed five small teams of one to three people; they were given a predefined location and pictures of the buildings surrounding it. The pictures served as AR markers. The participants were asked to develop

[1] http://www.ludicious.ch/

a chapter of the game story and a puzzle. Every team was able to design a chapter within the few hours of the challenge. All chapters were then gathered and merged into a final app that was playable during the festival.

This work enables creativity and educates on two levels: the novice developers are given a gentle introduction to the challenging field of AR and city-wide gaming, while the players are provided with a playful tool to discover the cultural background of a city. The successful deployment of this technology opens the way to a multitude of city-wide AR games.

Imagining Domain

Imagination lies at the heart of creative storytelling. Authoring interactive stories bears a promising opportunity to apply a wide spectrum of creative thinking. Our work on authoring interactive narrative is discussed in depth in Section 3.3.

Learning Domain

The availability of inexpensive electronic components has led to a variety of robotic toys, ranging from pets to cars. Most of these toys only allow simple, or even passive, interactions. Within this range of toys, edutainment robots have a special role, by carrying the promise of improving the skills of children in computational thinking. However, their actual educational value is not always scientifically assessed. In particular, the way in which these toys allow children to understand the process of program execution is not clear. To improve this understanding, we have developed an application combining an edutainment robot and AR.

Robot Programming. In this application, the child programs the behavior of the robot by creating event handlers through

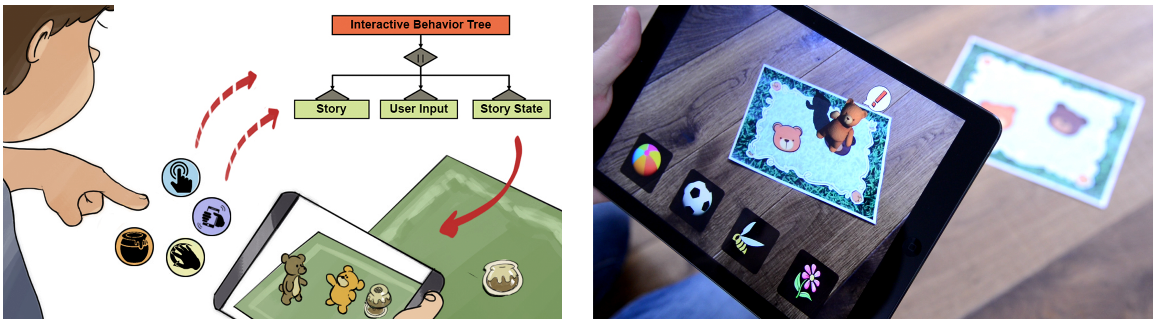
Creative Goal Aids novice developers building their own city-wide games and supports players to explore and discover cities.

Educational Goal Teaches concepts on AR and city-wide games development and virtual markets in games.

Potential Impact Large variety of AR city-wide games.

Figure 5.5: *Exploring domain: Top: Players are sent on a quest through the city to solve puzzles in our city-wide AR treasure hunt game. Bottom: In Gnome Trader (Section 5.3), the player locates trading locations embedded in newspaper boxes using a map and trades virtual resources with gnomes.*

the pairing of event and actions blocks. For example, such a pair could mean "when the robot sees an object on its front sensor, it must stop and light in red". Pairs consist purely of visual elements. Previous studies have shown that this method is effective in allowing novices to program the behavior of the robot, but that without additional aid not all children acquire a deep understanding of the program [Magnenat et al., 2014], because they do not have a clear picture of which event is executed in which condition. We address this problem by taking advantage



Creative Goal Reduces difficulty of authoring interactive narratives for authors, experience interactive AR stories as a player.

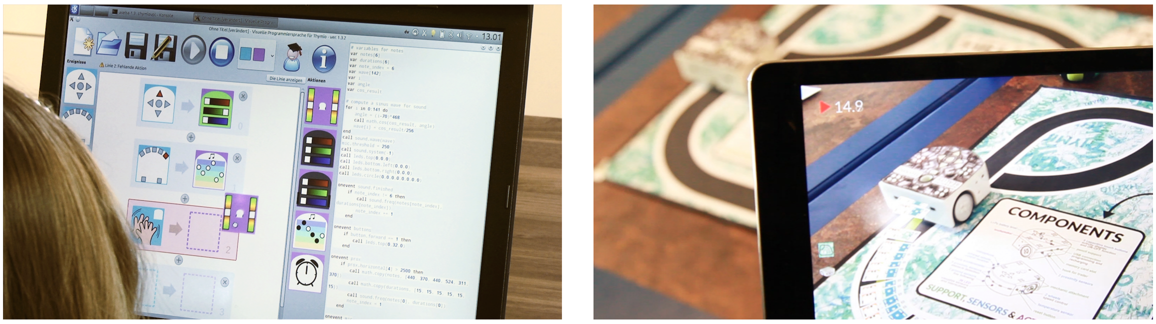
Educational Goal Teaches concepts and complexity of non-linear interactive story telling.

Potential Impact Create complex non-linear interactive stories that induce high user agency with far-reaching consequences.

Figure 5.6: *Imagining domain: The AR integration for our interactive narratives authoring framework, presented in Section 3.3, facilitates natural and intuitive interaction for the reader.*

of AR, to allow the child to see which event has been executed where in the physical world, depicted in Figure 5.7. With this application, we have shown that the use of AR improves the score on a test for deep understanding and changes the programming behavior of the child, although further studies are required to precisely assess these effects [Magnenat et al., 2014].

Recently, we have been working on gamifying the programming experience using AR, following two paths. On the one hand, we are using AR to augment the capabilities of the robot. For example, while the sensors of the robot cannot by themselves distinguish different types of objects, if the objects have AR markers on them, AR can allow the robot to give an identity to the object it perceives, greatly enhancing its programming possibilities without adding any cost to the robot's hardware. On the other hand, we are embedding a programming tutorial within an AR game, such that the full engagement capacity of gaming can be used for learning. For example, in this sys-



Creative Goal Create user-defined behaviors for robots and virtual agents.

Educational Goal Teaches programming and computer science concepts to understand the dynamics of program execution.

Potential Impact User-programmed non-playing characters, minions or pets in games; games based on user-programmed agents.

Figure 5.7: *Learning domain: Gamified robot programming using* AR *visualizes the robot event execution and guides the student.*

tem, solving programming puzzles brings new programming blocks as achievement, directly linking the quality of learning to the reward mechanism. Some blocks, called *vanity blocks*, have purely-virtual effects, such as displaying a fireworks, and can only be collected by creating programs of outstanding quality. This mechanism is an example of how the richness of the augmented world can be a lever to improve learning of core knowledge of general use.

This work shows that visually programming a robot with AR, besides fostering the creativity of children by allowing them to program their own behaviors into the robot, educates them about core computer science concepts. AR plays an important role in this deep comprehension, by making the dynamics of program execution visible. In addition, it opens many new opportunities in the field of gaming, for instance the creation of user-programmed non-playing characters, minions or pets; and a whole new class of competitive and collaborative games based on user-programmed agents.

5.2.2 Conclusion

We have outlined the concept of Augmented Creativity and shown how the graphics capabilities of mobile devices can be used to enhance real-world activities. Our work is supported by nine prototype applications that explore and develop the concept of Augmented Creativity in different domains, cultivating creativity through AR interactivity. Our work strives to make innovative use of mobile graphics in order to create compelling experiences that direct attention to classic real-world creative activities while offering engaging and exciting digital enhancements, fostering education, and opening further possibilities in the field of AR gaming.

5.3 City-Wide Augmented Reality Trading Game

Through their unique combination of visual, narrative, auditory, and interactive elements, video games provide an engaging medium of expression within our society. The proliferation of mobile devices that combine computation and graphics processing with video and GPS sensors holds great potential to enhance gaming. Furthermore, augmented reality AR provides the ability to blend virtual and real-world experiences so that location-aware games extend past our televisions, weaving the magic of gameplay into cultural locations such as the cities in which we live. In this way, games become an augmented version of real settings, bridging between imaginative worlds and the reality around us. In this thesis, we describe the design, testing, simulation, and implementation of an online multiplayer location-based AR game called *Gnome Trader*. In Gnome Trader, newspaper boxes throughout the country of Switzerland are augmented with virtual gnome characters who trade resources – nuts and peas – with players. Gnomes inhabit the boxes and vary their resource prices according to supply and demand. The core game mechanic is based on players increasing their wealth by buying and selling resources at different locations for different prices, allowing them to make a profit and to purchase upgrades. The use of AR enforces the impression that a gnome is physically located in a particular box at a particular location, and thus can have a unique price compared to gnomes located elsewhere.

Although Gnome Trader follows core game design principles [Fullerton, 2008] with a focus on being fun and entertaining, it has a deeper component that connects to serious gaming. At its heart, Gnome Trader represents a virtual economy. The prices of nuts and peas evolve depending on location-based transactions. Since the game is designed for large-scale deployment, it holds a unique promise to validate different economic models and better understand how the forces of supply and demand interact with pricing policies and player actions. By experimenting with the game economy, we hope to gain valuable insights about real-world economic challenges.

5.3.1 Game Design

In Gnome Trader, the player embodies a virtual trader, equipped with a bag to carry resources and gold pieces. The goal of the player is to travel within the country and trade resources at specific locations to make a profit. Each newspaper box across Switzerland represents a registered trading location. The logo of the newspaper box acts as an AR marker. Combined with the mobile device's GPS location, our software can uniquely identify every box. We choose this marker and GPS setup as it obviates the costly task of physically altering the boxes in order to add QR codes, near-field communication beacons, or other disambiguation technologies. Upon arriving at a newspaper box, the player opens the game app and points the smartphone's integrated camera at the box's logo. A virtual gnome with a trading interface appears on the screen, integrated into the video, to give the impression that the gnome is physically located inside the box. Figure 5.8 illustrates the Gnome Trader gameplay using screenshots from our prototype implementation.

(a) (b) (c) (d)

Figure 5.8: *Screenshots of the prototype game. (a) The game map indicating trading locations in the streets. (b) A Gnome family waiting to purchase nuts from the player; they offer 6 gold pieces per nut. (c) A gardener Gnome offering nuts to the player at a price of 5 gold pieces per nut. (d) The bag shop menu; the player has enough money to unlock the next larger bag.*

Four types of trading locations exist: each newspaper box either contains a gardener gnome selling peas or nuts or a gnome family buying peas or nuts. Gardener gnomes produce resources at a constant rate until their storage is full. Analogously, family gnomes consume resources at a constant rate until their storage is empty. A crowdsourced approach is employed for adding new trading locations to the game. After the marker on an unknown newspaper box has been recognized by the game client of sufficiently many different players, a new trading location is inserted at that position in the city. Thus, the population of gnomes grows as players from new regions join the game. The price of each resource is dynamically calculated by an economic model. We implemented two economic models, both of which are described in Section 5.3.1. Generally, a player can buy resources for a low price from a producer gnome with high storage and sell resources for a high price to a consumer gnome with low storage. For simplicity, in the current prototype, players cannot trade directly with each other. Their interaction is limited to the computer-controlled gnomes. The player can access a city map depicting all gnome locations and information about their resources. This feature encourages the player to find an appropriate gnome to sell the currently carried resources at a higher price than purchased. The resource carrying capacity of the player is limited by the size of a resource bag. With enough gold, the player can purchase larger bags to carry more resources, therefore increasing trading efficiency. Thus, with increased wealth, the player can buy more resources to make an even higher profit. The player can compare his or her performance to the other players on a global leaderboard, providing motivation to compete and continue playing. The total number of traded resources and gold is summed up for each player and displayed as a score.

Economic Models

A controllable and sustainable economic model is key to the success of the game as it defines how prices evolve over time and adapt to the behavior of the players. The goal of the economic model is to create a stable supply and demand behavior for the gnomes. Gnomes who are visited frequently by players should raise their prices, while gnomes who are rarely visited should lower their prices until a minimum price is reached. This behavior models a gnome's desire to maximize profit, while also introducing competition to the market. If an individual gnome's selling price is higher than those in the vicinity, players will buy elsewhere, forcing the gnome to lower prices. Such pricing models open up fascinating possibilities for gameplay. How-ever, they are very hard to configure in a way that achieves a well-balanced market. We developed two specific models, inspired by existing work [Smith, 1994; Davis and Williams, 1986], that model the effects of asymmetric supply and demand configurations on prices converging toward a competitive equilibrium.

Model A: Production-Consumption. In the Production-Consumption model, each Gnome continuously produces or consumes its resource at a fixed rate until the storage is full or empty, respectively. The resource price p is directly calculated from the storage ratio $r = \frac{S_{cur}}{S_{max}}$, where S_{cur} is the number of resources currently in storage and S_{max} is the storage size. In a fixed interval, each Gnome updates its price

$$p = w \cdot v(r) + k, \tag{5.1}$$

$$v(r) = \begin{cases} -2(r-1), & r < 0.5 \\ 1.5 - r, & \text{otherwise} \end{cases}, \tag{5.2}$$

where w is the real world market influence described in Section 5.3.1, which is a constant scaling factor across all trading locations, and k is a randomization term that varies the price. The randomization term adds a certain amount of chance to the game, which increases suspense for the players. In the prototype, k was defined such that it decreases or increases the price up to ten percent. The price modifier term $v(r)$ doubles the price if the storage is empty and halves the price if the storage is full. A producer gnome with low storage is considered successful and should increase its price to make more profit.

Analogously, a consumer gnome with a low storage will starve soon and should increase the price it is willing to pay.

Model B: History-Based. In the History-Based model, gnomes keep track of their trading success over time and adjust prices according to trends. A trading history logs how many resources were sold and purchased over the last couple of days. At each update step, the current earnings are compared to the history and evaluated for performance. If a producer gnome is successful, that is, if it was recently able to sell more resources than before, the selling price is increased. If the gnome sold less, the price is reduced. A consumer gnome is considered successful if it traded many resources recently, in which case it tries to lower the buying price, otherwise it is increased. Producer and consumer prices at time t are calculated as

$$p^t_{\text{prod}} = p^{t-1}_{\text{prod}} + m \cdot \left(\sum_{i=0}^{d-1} n_s^{t-1} / \sum_{i=1}^{d} n_s^{t-1} - 1 \right) + (w^t - w^{t-1}) \quad (5.3)$$

$$p^t_{\text{cons}} = p^{t-1}_{\text{cons}} - m \cdot \left(\sum_{i=0}^{d-1} n_b^{t-1} / \sum_{i=1}^{d} n_b^{t-1} - 1 \right) + (w^t - w^{t-1}) \quad (5.4)$$

where the parameters n_s^t and n_b^t denote the amount of resources sold or bought by the Gnome at time t, d controls the history depth, and m is a scalar model parameter. The term w is the real market influence described in Section 5.3.1. An important difference to the Production-Consumption model is that in this model gnomes have access to an unlimited number of resources.

Real World Economy Influence. To create a more realistic playing experience, the game's virtual economy is loosely tied to the real world economy. This feature gives the player a feeling of immersion and suspense as he or she can utilize real world measurements and estimates of the economy to make decisions in the game. This connection is created by introducing a gnome market index term w, which is calculated from a real market index, such as Dow Jones, NASDAQ, or Nikkei. As a result, fluctuations in the real market index are mirrored in the virtual market and visible in the individual gnome's price calculations. We calculate the gnome market index term

$$w^t = 3 + 1.9 \cdot atan(0.005(\mathrm{I}^t - \mathrm{I}^t_{\mathrm{AVG60}})), \tag{5.5}$$

for a time t by scaling the real market index I and removing low frequencies. The term $\mathrm{I}^t_{\mathrm{AVG60}}$ denotes the 60-day moving average of I. In our prototype, the Swiss Market Index (SMI) was employed for I. Specific values for the equation were found empirically. Figure 5.9 depicts the SMI and the corresponding Gnome market index w between the years 2013 and 2015.

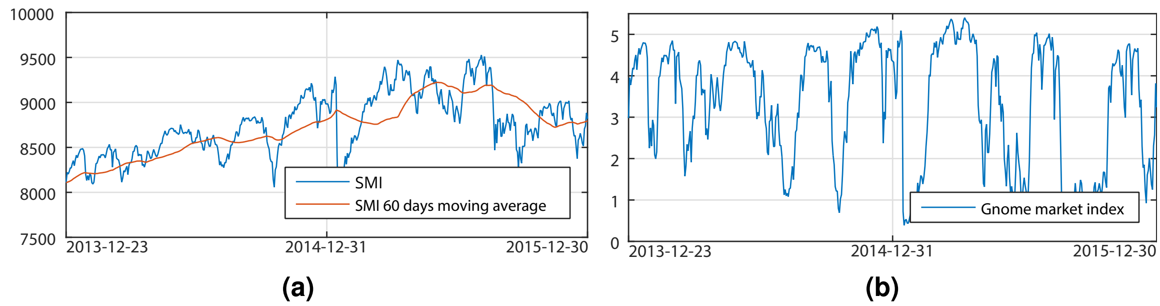


Figure 5.9: *(a) Swiss Market Index (SMI) and (b) in-game gnome market index between the year 2013 and 2015. The SMI is used to influence the game's virtual economy.*

Implementation

Figure 5.10 depicts the client-server architecture. The game client is implemented using the Unity[2] game engine, relying on the Vuforia SDK[3] for AR processing. A real-time websocket communication architecture based on socket.io[4] is employed to

[2] https://unity3d.com/ [3] https://developer.vuforia.com/ [4] http://socket.io/

pass messages between the Unity client and a node.js[5] server application. The server application stores user data, transaction data, and trading location data in a MongoDB[6] database. Maintenance scripts perform updates on the database in regular time intervals to, for example, recalculate prices and distribute resources. The game map is made accessible to the players on a website running on Apache[7]. The Google Maps API[8] allows our system to overlay gnome locations, their prices, and storage levels over the city street map. Financial data is retrieved from Quandl[9].

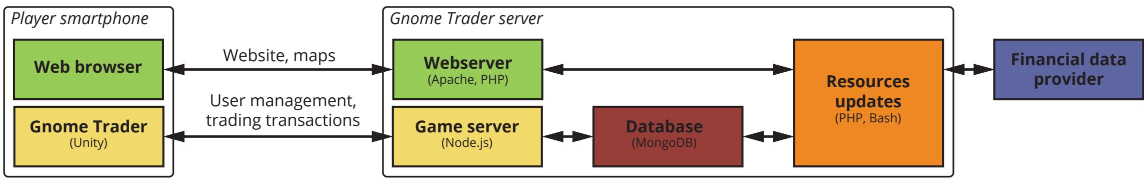


Figure 5.10: *Gnome Trader client-server architecture overview. The game server provides user and trading location data to the Unity-based game client app. The game map, provided by a webserver, is viewed on any browser. Regular resource update and maintenance tasks are performed directly on the database. Financial data is retrieved from an external data provider.*

5.3.2 Simulation

It is crucial to be able to analyze potential pricing schemes of a trading game in a controlled manner before launch to reduce the risk of problematic behaviors such as massive inflation or market crashes. For this purpose, we propose an agent-based framework that simulates the virtual economy of Gnome Trader. The simulator uses a simplified model of the game. Players cannot buy upgrades or spend money in any way other

[5] https://nodejs.org/ [6] https://www.mongodb.org/ [7] http://www.apache.org/
[8] https://developers.google.com/maps/ [9] https://www.quandl.com/

than buying resources. There is only one type of resource to trade. The city map is generated at random, using an algorithm that is inspired by scale-free graphs [Li et al., 2005] and exhibits similar properties. Part of a example generated city map is depicted in Figure 5.11. The simulator was implemented using MASON[10], an agent-based modeling toolkit for Java.

Player Behavior Model

Players are represented by agents in the simulator. The model distinguishes player movement and trading behavior and assumes the two to be independent. Agents never move to a gnome for the explicit purpose of making an advantageous trade. The reasoning for this choice is that traveling takes real effort. Thus, players are unlikely to go out of their way just to play the game. Instead, they will go about their business as usual, and only stop to play when a convenient opportunity presents itself. Because of this assumption, the agent behavior model used for the simulation tries to mimic the movement patterns of average people going about their daily lives in an urban environment. Players are assumed to commute between two fixed locations once per day, such as going to school or work. Agents can trade with gnomes they visit during their daily commute. They try to maximize profits by conservatively selecting the best trading opportunities along that path. Agents only sell when there is no better offer along their path. They also avoid selling a resource for less than it was bought, which is represented by the resistance price. The resistance price decays slowly over time to avoid locking agents out of trading indefinitely. Agents buy when a producer's price is lowest and there is a guaranteed profit to be made elsewhere by selling the resource for more money. The last condition prevents player agents from buying up worthless resources simply because they are cheap. The agent player behavior is summarized in Alg. 5.

[10] https://cs.gmu.edu/ẽclab/projects/mason/

Algorithm 5 Simulator Player Agents Decision-Making

procedure TRADESTEP
  set resistance price $p_{\mathrm{r}} = 0$
  for all timestep Δt **do**
    calculate $p_{\mathrm{prod}}^{\mathrm{min}}$ from the list of planned producer Gnome visits in Δt
    calculate $p_{\mathrm{cons}}^{\mathrm{max}}$ from the list of planned consumer Gnome visits in Δt
    decrease p_{r}
    for all trading location g **do**
      if (g is a consumer) and ($p_{\mathrm{cons}} \geq p_{\mathrm{cons}}^{\mathrm{max}}$) and ($p_{\mathrm{cons}} \leq p_{\mathrm{r}}$) **then**
        sell as many resources as possible
      end if
      if (g is a producer) and ($p_{\mathrm{prod}} \leq p_{\mathrm{prod}}^{\mathrm{min}}$) and ($p_{\mathrm{prod}} \leq p_{\mathrm{cons}}^{\mathrm{max}}$) **then**
        buy as many resources as possible
        $p_{\mathrm{r}} = max(p_{\mathrm{r}}, p_{\mathrm{prod}})$
      end if
    end for
  end for
end procedure

Results

Both economic models described in Section 5.3.1 were simulated under various configurations to explore the parameter spaces. The real world influence and the randomization term were omitted for the simulation, to observe the properties of the economic model without the interference of these added effects

Model A: Production-Consumption. The behavior of the Production-Consumption model depends on the ratio of consumer and producer gnomes as well as their production and consumption rates. If the parameters are chosen such that the inflow of resources into the system equals the outflow, then the economy is stable and prices fluctuate around a constant value. However, by upsetting this balance we can simulate a variety of interesting real market phenomenon, such as shortages and oversupply. For example, Figure 5.12 shows the output of a scenario with a slightly higher number of consumer gnomes than

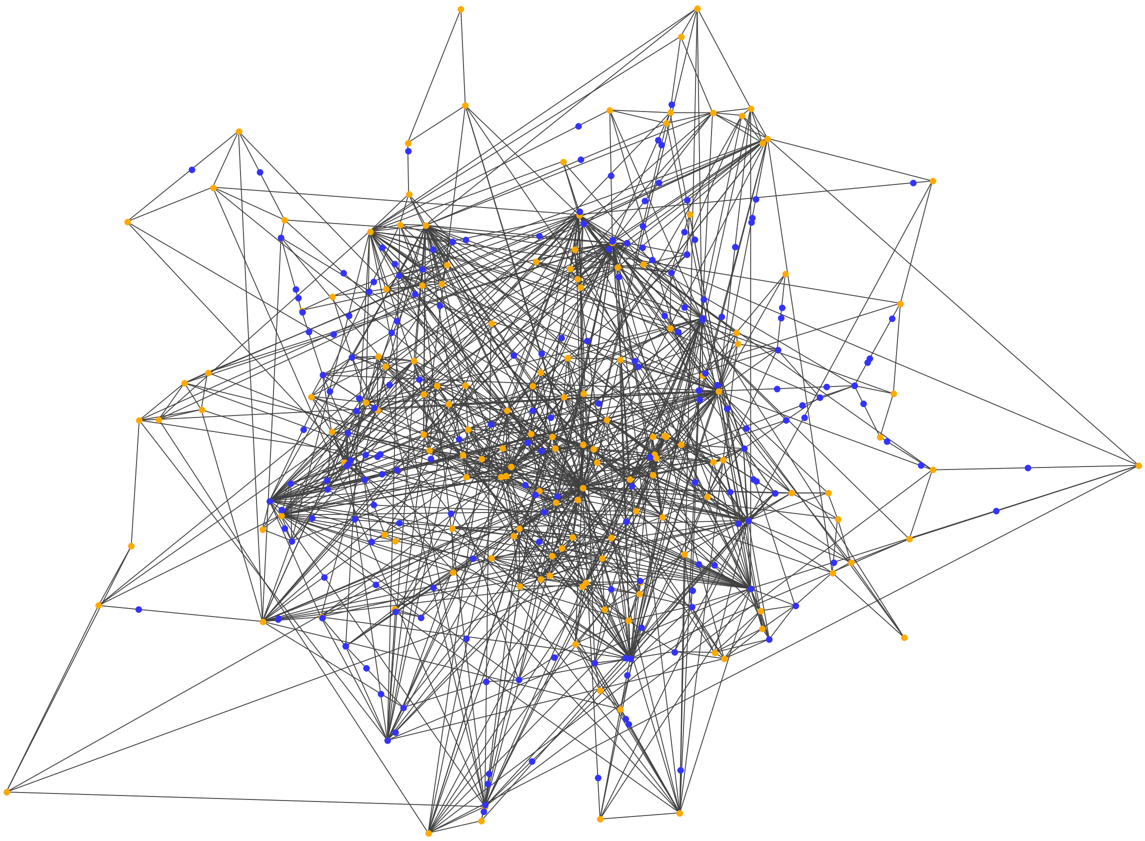


Figure 5.11: *A randomly generated city map with 200 player agents (blue dots) and 200 gnomes (orange nodes). A majority of the agents are on the way home from work moving toward the outside of the city.*

producer gnomes. This imbalance results in an undersupply of resources that causes the producer prices to rise continuously. The shortage causes consumer prices to spike after about 140 days. Player agents recognize the opportunity for profit, as evidenced by the subsequent increase in trading activity. Player wealth drops momentarily due to the investment. Prices fluctuate for a few days before the simulation continues normally. By day 150, the high demand causes producers to run out of resources and their prices cap. Meanwhile, consumer gnomes have been well supplied and their prices drop. The system settles in a stable state after a few more fluctuations. The scenario

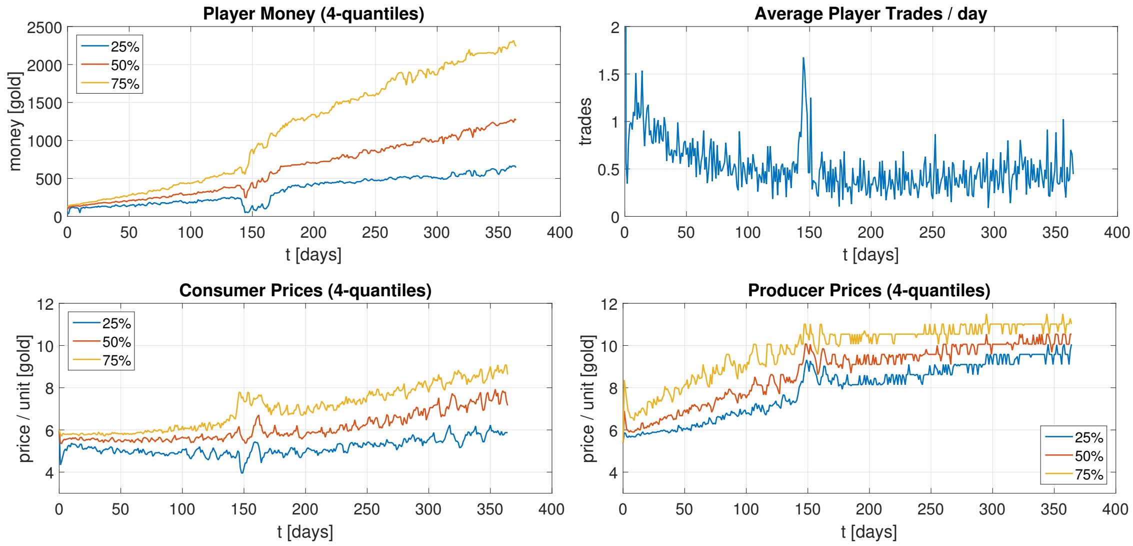


Figure 5.12: *Simulation output of an under-supply scenario with Production-Consumption economic model. The city contains 102 consumer Gnomes and 98 producer Gnomes. The market index is fixed at* $w = 6.0$ *and Gnomes have a maximum capacity of 500 units. Production and consumption rates are at 20 units per day.*

could be further stabilized by injecting more producer gnomes into the system or by adjusting the production rates.

The Production-Consumption model can be fine-tuned dynamically and offers developers a high level of control over the market behavior. The ability to reproduce real world phenomenons makes it an interesting choice for a game economy.

Model B: History-Based. Figure 5.13 shows a simulation run of the same scenario using the History-Based economic model. The imbalance between consumers and producers has no apparent effect on this economic model, because its prices are only dependent on the sales and purchase volumes, not on the absolute amount of resources in the system. As such, it is more robust to variations in population sizes, but also easier to influence by certain player strategies. While the unlimited supply and demand for resources causes players to gain wealth at a much faster pace than in the Production-Consumption model, this phenomenon can be accounted for by balancing the initial prices accordingly.

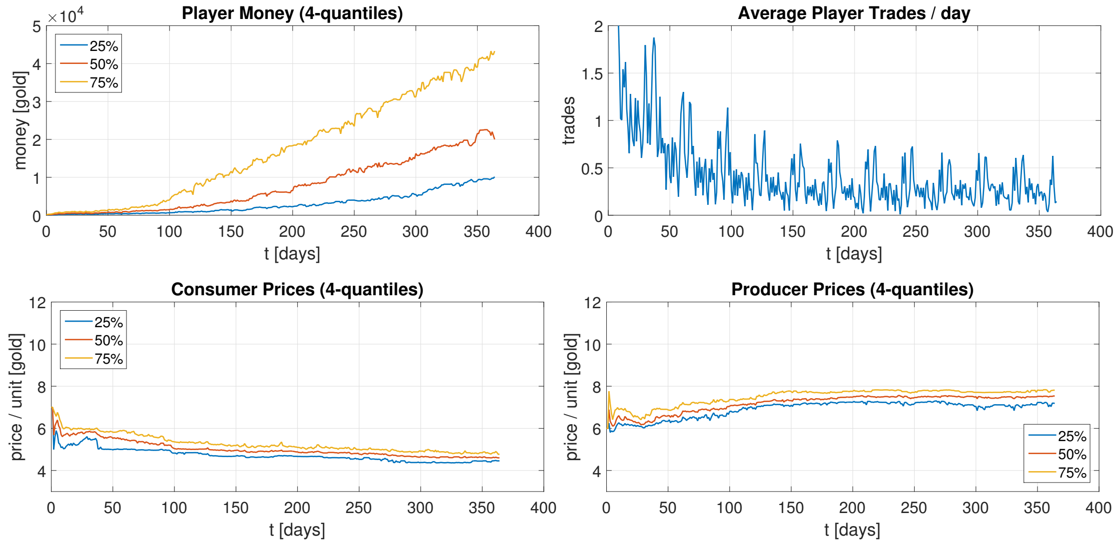


Figure 5.13: *Simulation output with the History-Based economic model. Gnomes store trade histories for 30 simulation days and the multiplier is set to* $m = 2.0$.

5.3.3 Playtesting

We tested the game prototype to evaluate the technical functionality and the appeal of gameplay. Following the agile development concept, we started with tests early during the development and finally conducted two more formal playtesting sessions. In the first playtesting session 8 participants (5 male and 3 female, aged 17 to 26 years) played the prototype in Zurich, Switzerland. Each participant was engaged in the game for 2 hours in total over the course of a week. Afterward, the participant's feedback was collected with a questionnaire. Most users needed some time to appreciate the concept of traveling physically to successfully play the game. While the participants did not lose interest in the game during that week, some mentioned that without any changes it may become less interesting to continue playing. The limited accuracy and robustness of the measured GPS location sometimes led to inconsistent trading location information, which negatively influenced the gameplay. Some participants also suggested that direct trading between players would be desirable. A second playtesting session was conducted in Barcelona, Spain, with 19 participants over the course of an afternoon. The game was played again successfully and additional insights could be gained through a questionnaire. To generate a single figure that would encapsulate the player's satisfaction with the game, the Net Promoter Score was used. The average results for the score were very positive, standing at 73.5%. After playing, 79% of the participants indicated that they would play the game a few times per week or more. Feedback from the participants included that the AR approach requires good lighting conditions, which can be problematic at night time. Participants also mentioned that the high battery consumption should be addressed.

5.3.4 Conclusion

In this work, we demonstrated a game prototype for a city-wide trading game. AR and real world market influence contribute to a rich and immersive gaming experience. Our simulator showed that the Consumption-Production economy model is well suited for implementation because it can be fine-tuned as required and reproduces the phenomenon of real markets. Alternatively, the History-Based economy model is more robust to variations in player and gnome populations but, at the same time, is also vulnerable to certain player strategies. We conclude from two play testing sessions that the game concept is functional and well received.

CHAPTER 6

Conclusion

This chapter concludes the thesis by summarizing and discussing the main contributions and providing directions for future work.

6.1 Summary

In Chapter 1 of this work, an introduction to storytelling and Augmented Reality (AR) was presented, and the current challenges and our vision was outlined. Relevant previous work related to AR storytelling was discussed in Chapter 2.

In order to support our vision of leveraging digital technology to craft and experience stories together, this thesis has presented frameworks to support collaborative as well as interactive story authoring in Chapter 3, and we have examined rendering, presence, interactivity, and creativity related for shared AR storytelling experiences in Chapter 4 and Chapter 5.

Story Authoring. In the story authoring domain, in Chapter 3, we have presented tools to improve collaborative as well as interactive storytelling.

Our Story Version Control (SVC) system enables authors to directly collaborate on narratives. We proposed a media-agnostic, graph-based story representation that allows tree-edit distances and operations to be directly calculated among stories. The tree-edit operations provide the foundation for version control operations, such as checkout, update, commit, and merge. The evolving stories are stored in the cloud as repositories, while authors gradually perform operations. A SVC client is part of CANVAS, our story authoring framework. Graph visualization techniques are integrated into SVC and applied to story repositories. These visualizations allow the authors to inspect relations and meta-relations between story elements contained in the repositories, such as characters, events, and authors. They provide valuable information and a means for the authors to interpret each other's creative intent. A user study validated the efficacy of SVC and provided ideas for future directions. The participants contributed valuable feedback and suggestions for improvements.

Our approach for authoring interactive narratives using Interactive Behavior Trees (IBTs) enables content creators author interactive, branching narratives with multiple story arcs in a modular, extensible fashion. Through the AR implementation, the players gain the agency to freely interact with the characters in the story and the world they inhabit. We demonstrated how IBTs and the use of computer-assisted tools can reduce the authoring complexity without sacrificing authoring control. Compared to traditional story graph representations, IBTs better scale with story complexity and freedom of user interaction, and authoring stories takes lesser time with reduced number of errors. The authoring complexity is further reduced through automation methods, which can completely avoid authoring errors.

Augmented Reality. In the AR domain, in Chapter 4, we have presented multiple experiments to explore the influence of AR mixing techniques on the user experience. Furthermore, we proposed a suite of algorithms and methods to enable an AR coloring book application.

Three experiments were conducted to assess the impact of camera motion blur, image latency, and realism of lighting conditions. For this purpose, two different AR applications were employed: ARTravelers, a fast-paced action game and a camera application, ARPix, which takes pictures of the user and integrates a virtual character into the image. In each experiment, artificial blur, latency, or varying lighting conditions were introduced to the application and the users' performance was recorded. A questionnaire was filled out by the users at the end of each experiment. In the first experiment, we observed that strong camera motion blur may be noticed by players but does not significantly affect player performance or their feeling of immersion. This result may seem unintuitive but it is consistent with earlier motion blur user studies. In contrast, in the second experiment, lower image latency strongly correlated with a positive user experience. Additionally, significant impact of latency on realism, enjoyment, satisfaction, and player scores were recorded. Finally, in the third experiment, we found a strong preference to a virtual lighting environment that closely matches the physical lighting environment.

An AR coloring book application was presented that is based upon two main contributions: First, 3D texture synthesis from 2D drawings is achieved in real-time on a tablet by employing a UV lookup map approach. For this purpose, a novel algorithm for artists to generate this map automatically is provided. Second, deformable surface tracking is performed in real-time by introducing a new template selection and outlier rejection mechanism, as well as a lighter problem formulation. We concluded from two user studies that the novel texture transfer process is appreciated and the overall application strongly improves the sense of connection with the character and motivates people to draw more often. Furthermore, we observed during the studies that, even though coloring is a solitary activity, through the application, multiple children engaged together alternating coloring and inspecting each others 3D models, rendering the AR coloring book a social experience.

Applications. Chapter 5 introduced the concept of Augmented Creativity.

We have demonstrated nine prototype AR applications in the domains of seeing, hearing, strategizing, exploring, imagining, and learning that each leverage the camera and graphics processing capabilities of modern mobile devices to facilitate novel, compelling, and immersive AR experiences. The experiences aim at digitally enhancing real-world activities, such as puzzling, coloring, music arrangement, stamps, programming, and city-wide gaming, through AR to boost creativity. For each application, creative goals, educational values, and potential impacts were explored.

Our AR city-wide trading game, Gnome Trader, allows players to trade virtual resources with gnomes hidden in newspaper boxes around Switzerland. Online multiplayer capabilities, AR, and a virtual economy that is influenced by the real world market contribute to a rich and immersive gaming experience. Two economic models were developed and simulated to explore different economic scenarios. Two playtesting session, in Zürich and in Barcelona, confirmed that the game is functional and well received.

6.2 Discussion and Outlook

This section discusses key insights on AR storytelling and proposes future directions.

Story authoring highly benefits from digital collaboration

tools. The Internet is the greatest enabler of collaboration. Today, it connects almost 3.5 billion people around the globe and almost every website provides some way of collaboration, often in the form of user-generated content. Examples are Content Management Systems (CMSs) (Wordpress), allowing multiple users to collaboratively author websites, Wikis (Wikipedia), communication tools (Slack), blogs (Blogger, Tumblr), project management tools (Trello, Basecamp, Smartsheets, Binfire), generic design or production tools (InVision, Github, Google Drive including Docs, Sheets, Slides, Forms, and Sites), and, finally, social media and news sites (Reddit, YouTube, Twitter, Facebook, 9Gag). In contrast to these tools, which facilitate collaborative storytelling in a general sense, our SVC system provides intuitive, visual version control directly tailored to narratives in a media-agnostic fashion. It can detect story conflicts between multiple authors and automatically merge two stories to resolve conflicts. At the same time, meaningful information is extracted from the raw continuously expanding story data, visualized, and presented to the authors. While not all common version control features are implemented, missing for instance is branching, we believe that SVC is a valuable leap towards controlled and accurate collaborative story authoring, which is also accessible to novice story authors.

Future collaborative storytelling should include authoring tools that assist authors not only with syntactic error detection and resolution, such as in SVC, but assist the authors handling higher-level, semantic challenges as well. This will require the system to acquire a deeper understanding of the narratives, including state of story elements as well as capturing dramatic structure, sentiment, and the author's creative intent.

Modern AR creates unprecedented immersiveness that can greatly boost shared narrative experiences. While reading, hearing, or watching any story, readers become absorbed into the storyworld. They start to identify themselves with the environments, the situations, and the characters in the story. AR can drastically improve this aspect of storytelling as it establishes a strong connection between users and the storyworld by giving users the feeling that they are physically present inside the story or that the story is unfolding around them in the real world. Additionally, and in contrast to Virtual Reality (VR), AR applications can be easily shared among multiple users in a common physical space, which enables a wide range of social experiences. Our storytelling applications demonstrated how creative activity books, games, and even educational applications can greatly benefit from the immersiveness AR techniques create. In our coloring book application, the flat and motionless character in the book becomes seemingly alive and dances in front of the users. We noticed during our playtesting sessions that children create a strong bond very quickly to the awakened character. Our city-wide games place the player into a mixed reality comprising a real city with its streets and buildings, and digital augmentations, such as characters, puzzles, or other story elements. The city that was known beforehand can be re-explored in the game world. Building facades and other ubiquitous city elements that are normally completely static become interactive puzzles or gates to a fantastic virtual world.

Today, the majority of consumer AR applications rely on monocular cameras using image based markers, GPS, and accelerometers to synchronize the virtual with the physical world. Newer portable devices that are just entering the consumer market, such as Google's Tango and Microsoft's Hololens, fuse multiple cameras to detect depth and track geometry directly without markers. Future applications will not require predefined objects labeled with markers to interact with the virtual world, but can integrate arbitrary objects into the story. Toys, furniture, people, architecture, and even fauna and flora could be automatically detected, embedded into, and synchronized with the virtual world, boosting immersiveness of such applications to extraordinary levels.

AR offers a host of novel and intuitive interaction possibilities that improve interactive storytelling experiences. Interactive storytelling is a topic that has been well explored in the past. Especially, computer games provide highly interactive narratives offering strong user agency to influence the plot. AR applications running on tablets or Head Mounted Displays (HMDs) allow users to move the virtual camera intuitively, as they would move their head or hand. They can interact with virtual objects by simply moving and touching the display in the physical world. This bears great advantages over traditional, rather cumbersome computer input mechanisms such as a keyboard and a mouse. In our interactive bear story, users can inspect the story from any angle as it unfolds on the table before them. The story elements, that is, the balls, bees, and flowers, can be picked up and placed instinctively. Also user-created content processes benefit from the interactivity AR provides. In our music arrangement application, users spatially position the virtual instruments in the physical world, observe them through the tablet display, and listen to the music with matching volume and stereo panning, as if the instruments were placed in front of them in the real world. Even children can intuitively inspect 3D characters they or other children colored using the AR coloring book application. Finally, interaction is not limited to directly handling virtual objects through AR. Our city-wide games make use of the most fundamental way of interaction. The users physically position themselves in the city, allowing GPS coordinates to directly influence the story. For instance, in the treasure hunt game, finding and arriving at the key locations in the city is the main game mechanic.

The vast majority of current AR applications are limited to mono-directional control mechanisms, that is, the user controls the application through various input mechanisms but the software only returns visual but no physical feedback to the user. There has been various work on haptic technology enabling users to feel virtual environments and allowing the applications to exercise physical contact and force to the users through, for instance, force-feedback devices [Salisbury et al., 2004], electro-vibrations [Bau et al., 2010], electrical stimulation [Lopes and Baudisch, 2013], or air [Sodhi et al., 2013]. Interaction possibilities in future AR applications would be vastly increased through such haptic technologies. Users would not only see virtual objects integrated into the real world, but they could feel and touch them.

Outro. In our digital age, stories are not bound to follow simple trajectories anymore, they may be open-ended, non-linear, transmedia, interactive, exploratory, or completely unpredictable [Alexander and Levine, 2008]. There exists already a large spectrum of tools and platforms that enable storytellers to create various types of narratives, to author them collaboratively, and to share them with a vast audience. At the same time, consuming and experiencing stories has never been more convenient, ubiquitous, and captivating. This thesis presented advancements in the field of AR storytelling focusing on collaboration, interaction, immersiveness, and creativity. We believe that this thesis inspires future research and constitutes a next step towards rich, interactive, and enthralling storytelling.

References

[Agusanto et al., 2003] Kusuma Agusanto, Li Li, Zhu Chuangui, and Ng Wan Sing. Photorealistic Rendering for Augmented Reality Using Environment Illumination. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '03, pages 208—-, Washington, DC, USA, 2003. IEEE Computer Society.

[Alahi et al., 2012] A Alahi, R Ortiz, and P Vandergheynst. {FREAK}: Fast Retina Keypoint. In *CVPR*, pages 510–517, 2012.

[Alexander and Levine, 2008] Bryan Alexander and Alan Levine. Web 2.0 storytelling: Emergence of a new genre. *EDUCAUSE review*, 43(6):40–56, 2008.

[Altmanninger et al., 2009] Kerstin Altmanninger, Martina Seidl, and Manuel Wimmer. A survey on model versioning approaches. *International Journal of Web Information Systems*, 5(3):271–304, 2009.

[Amabile, 1998] Teresa M Amabile. How to kill creativity. In *Harvard Business Review 76*, pages 76–87, 1998.

[Axelrod, 1980] Robert Axelrod. Effective Choice in the Prisoner's Dilemma. *The Journal of Conflict Resolution*, 24(1):3–25, 1980.

[Axelrod, 1993] Robert Axelrod. *The Evolution of Cooperation*. Basic Books, 1993.

[Azuma et al., 2001] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, Nov 2001.

[Azuma, 1997] Ronald T Azuma. A survey of augmented reality. *Presence*, 6(4):355–385, 1997.

[Bae et al., 2013] Hyeoungho Bae, Charless C Fowlkes, and Pai H Chou. Accurate Motion Deblurring Using Camera Motion Tracking and Scene Depth. In *Proceedings of the 2013 IEEE Workshop on Applications of Computer Vision (WACV)*, WACV '13, pages 148–153, Washington, DC, USA, 2013. IEEE Computer Society.

[Bartoli et al., 2012] A Bartoli, Y Gérard, F Chadebecq, and T Collins. On Template-Based Reconstruction from a Single View: Analytical Solutions and Proofs of Well-Posedness for Developable, Isometric and Conformal Surfaces. In *CVPR*, pages 2026–2033, 2012.

[Bau et al., 2010] Olivier Bau, Ivan Poupyrev, Ali Israr, and Chris Harrison. Teslatouch: Electrovibration for touch surfaces. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 283–292, New York, NY, USA, 2010. ACM.

[Bay et al., 2006] H Bay, T Tuytelaars, and L Van˜Gool. {SURF}: Speeded Up Robust Features. In *ECCV*, pages 404–417, 2006.

[Beck, 1993] E E Beck. A Survey of Experiences of Collaborative Writing. In *Computer Supported Collaborative Writing*, pages 87–112. Springer London, London, 1993.

[Bermano et al., 2014] Amit H Bermano, Derek Bradley, Thabo Beeler, Fabio Zünd, Derek Nowrouzezahrai, Ilya Baran, Olga Sorkine-Hornung, Hanspeter Pfister, Robert W Sumner, Bernd Bickel, and Markus Gross. Facial Performance Enhancement

Using Dynamic Shape Space Analysis. *ACM Trans. Graph.*, 33(2):13:1—-13:12, apr 2014.

[Berry et al., 2003] Rodney Berry, Mao Makino, Naoto Hikawa, and Masumi Suzuki. The Augmented Composer Project: The Music Table. In *Proceedings of the 2Nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '03, pages 338—-, Washington, DC, USA, 2003. IEEE Computer Society.

[Billinghurst et al., 2001] M Billinghurst, H Kato, and I Poupyrev. The MagicBook - moving seamlessly between reality and virtuality. *Computer Graphics and Applications, IEEE*, 21(3):6–8, 2001.

[Bimber et al., 2001] O. Bimber, B. Frohlich, D. Schmalsteig, and L. M. Encarnacao. The virtual showcase. *IEEE Computer Graphics and Applications*, 21(6):48–55, Nov 2001.

[Blanz and Vetter, 1999] V Blanz and T Vetter. A Morphable Model for the Synthesis of {3D} Faces. In *SIGGRAPH*, pages 187–194, aug 1999.

[Bouguet, 2001] Jean-Yves Bouguet. Pyramidal Implementation of the Affine {Lucas Kanade} Feature Tracker Description of the Algorithm. *Intel Corporation*, 5:1–10, 2001.

[Brooke, 1996] J Brooke. SUS: A quick and dirty usability scale. In P W Jordan, B Weerdmeester, A Thomas, and I L Mclelland, editors, *Usability evaluation in industry*. Taylor and Francis, London, 1996.

[Brunet et al., 2010] F Brunet, R Hartley, A Bartoli, N Navab, and R Malgouyres. Monocular Template-Based Reconstruction of Smooth and Inextensible Surfaces. In *ACCV*, pages 52–66, 2010.

[Brunet et al., 2014] Florent Brunet, Adrien Bartoli, and R I Hartley. Monocular template-based {3D} surface reconstruction: Convex inextensible and nonconvex isometric methods. *Computer Vision and Image Understanding*, 125:138–154, 2014.

[Calonder et al., 2010] M Calonder, V Lepetit, C Strecha, and P Fua. {BRIEF}: Binary Robust Independent Elementary Features. In *ECCV*, pages 778–792, 2010.

[Carmack, 2013] John Carmack. Latency Mitigation Strategies. \url{http://www.altdev.co/2013/02/22/latency-mitigation-strategies/}, 2013.

[Castronova et al., 2009] Edward Castronova, Dmitri Williams, Cuihua Shen, Rabindra Ratan, Li Xiong, Yun Huang, and Brian Keegan. As real as real? Macroeconomic behavior in a large-scale virtual world. *New Media & Society*, 11(5):685–707, 2009.

[Chhatkuli et al., 2014] A Chhatkuli, D Pizarro, and A Bartoli. Stable Template-Based Isometric {3D} Reconstruction in All Imaging Conditions by Linear Least-Squares. In *CVPR*, pages 708–715, 2014.

[Clark and Dunser, 2012] A Clark and A Dunser. An Interactive Augmented Reality Coloring Book. In *IEEE Symposium on 3D User Interfaces*, pages 7–10, 2012.

[Cootes et al., 1998] T F Cootes, G J Edwards, and C J Taylor. Active Appearance Models. In *ECCV*, pages 681–685, 1998.

[Criminisi et al., 2004] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region Filling and Object Removal by Exemplar-based Image Inpainting. *Image Processing, {IEEE} Transactions on*, 13(9):1200–1212, 2004.

[Davis and Williams, 1986] D Davis and A Williams. The effects of rent asymmetries in posted offer market. *Journal Of Economic Behavior & Organization*, 7:303–316, 1986.

[Debeauvais et al., 2012] Thomas Debeauvais, Bonnie A Nardi, Cristina V Lopes, Nicholas Yee, and Nicolas Ducheneaut. 10,000 Gold for 20 Dollars: An Exploratory Study of World of Warcraft Gold Buyers. In *Proceedings of the International Conference on the Foundations of Digital Games*, FDG '12, pages 105–112, New York, NY, USA, 2012. ACM.

[Deguchi et al., 2001] Hiroshi Deguchi, Takao Terano, Koichi Kurumatani, Taro Yuzawa, Shigeji Hashimoto, Hiroyuki Matsui, Akio Sashima, and Toshiyuki Kaneda. *New Frontiers in Artificial Intelligence: Joint JSAI 2001 Workshop Post-Proceedings*, chapter Virtual Ec, pages 218–226. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[Delbue and Bartoli, 2011] A Delbue and A Bartoli. Multiview {3D} Warps. In *ICCV*, pages 675–682, 2011.

[Denning and Pellacini, 2013] Jonathan D Denning and Fabio Pellacini. MeshGit: Diffing and Merging Meshes for Polygonal Modeling. *ACM Trans. Graph.*, 32(4):35:1—-35:10, 2013.

[Dow et al., 2007] Steven Dow, Manish Mehta, Blair MacIntyre, and Michael Mateas. AR fa{ç}ade: an augmented reality interactve drama. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, pages 215–216. ACM, 2007.

[Earle, 2013] Peter C Earle. A Virtual Weimar: Hyperinflation in a Video Game World. https://mises.org/library/virtual-weimar-hyperinflation-video-game-world, 2013.

[Elseberg et al., 2012] Jan Elseberg, Stéphane Magnenat, Roland Siegwart, and Andreas Nüchter. Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration. *Journal of Software Engineering for Robotics*, 3(1):2–12, 2012.

[Fayad et al., 2010] J Fayad, L Agapito, and A Delbue. Piecewise Quadratic Reconstruction of Non-Rigid Surfaces from Monocular Sequences. In *ECCV*, pages 297–310, 2010.

[Fischer et al., 2006] Jan Fischer, Dirk Bartz, and Wolfgang Strasser. Enhanced Visual Realism by Incorporating Camera Image Effects. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '06, pages 205–208, Washington, DC, USA, 2006. IEEE Computer Society.

[Folkestad, 1995] Goran Folkestad. *Computer Based Creative Music Making: Young People's Music in the Digital Age*. Coronet Books, Philadelphia, PA, USA, 1995.

[Friston and Steed, 2014] Sebastian Friston and Anthony Steed. Measuring Latency in Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics*, 20(4):616–625, apr 2014.

[Fullerton, 2008] T Fullerton. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*. Taylor & Francis, 2008.

[Garg et al., 2013] R Garg, A Roussos, and L Agapito. Dense Variational Reconstruction of Non-Rigid Surfaces from Monocular Video. In *CVPR*, pages 1272–1279, 2013.

[Gordon et al., 2004] Andrew Gordon, Michael van Lent, Martin V Velsen, Paul Carpenter, and Arnav Jhala. Branching Storylines in Virtual Reality Environments for Leadership Development. In *AAAI*, pages 844–851, 2004.

[Grasset et al., 2008] Raphaël Grasset, Andreas Dünser, and Mark Billinghurst. Edutainment with a Mixed Reality Book: A Visually Augmented Illustrative Childrens' Book. In *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, ACE '08, pages 292–295, New York, NY, USA, 2008. ACM.

[Guillemot and Le Meur, 2014] C Guillemot and O Le Meur. Image Inpainting : Overview and Recent Advances. *{IEEE} Signal Processing Magazine*, 31(1):127–144, jan 2014.

[Hart, 2013] J Hart. *The Art of the Storyboard: A filmmaker's introduction*. Taylor & Francis, 2013.

[Herling and Broll, 2014] Jan Herling and Wolfgang Broll. High-Quality Real-Time Video Inpaintingwith {PixMix}. *IEEE Transactions on Visualization and Computer Graphics*, 20(6):866–879, 2014.

[Isla, 2005] Damian Isla. Handling Complexity in the Halo 2 AI. *Game Developers Conference*, page 12, 2005.

[Jacobs et al., 1997] Marco C Jacobs, Mark A Livingston, and Andrei State. Managing Latency in Complex Augmented Reality Systems. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, I3D '97, pages 49—-ff., New York, NY, USA, 1997. ACM.

[Jacomy et al., 2014] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PloS one*, 9(6):e98679, 2014.

[Jhala and Young, 2010] Arnav Jhala and R Michael Young. Cinematic {Visual} {Discourse}: {Representation}, {Generation}, and {Evaluation}. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(2):69–81, 2010.

[Joshi et al., 2010] Neel Joshi, Sing Bing Kang, C Lawrence Zitnick, and Richard Szeliski. Image Deblurring Using Inertial Measurement Sensors. In *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, pages 30:1—-30:9, New York, NY, USA, 2010. ACM.

[Kanbara and Yokoya, 2002] Masayuki Kanbara and Naokazu Yokoya. Geometric and Photometric Registration for Real-Time Augmented Reality. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, ISMAR '02, pages 279—-, Washington, DC, USA, 2002. IEEE Computer Society.

[Kapadia et al., 2011a] M Kapadia, S Singh, G Reinman, and P Faloutsos. A Behavior-Authoring Framework for Multiactor Simulations. *IEEE CGA*, 31(6):45–55, 2011.

[Kapadia et al., 2011b] M Kapadia, S Singh, G Reinman, and P Faloutsos. Multi-actor Planning for Directable Simulations. In *Digital Media and Digital Content Management (DMDCM)*, pages 111–116, 2011.

[Kapadia et al., 2013] Mubbasir Kapadia, Alexander Shoulson, Funda Durupinar, and NormanI. Badler. Authoring Multi-actor Behaviors in Crowds with Diverse Personalities. In Saad Ali, Ko Nishino, Dinesh Manocha, and Mubarak Shah, editors, *Mod-*

eling, Simulation and Visual Analysis of Crowds, volume 11, pages 147–180. Springer-Verlag New York, 2013.

[Kapadia et al., 2015a] Mubbasir Kapadia, Jessica Falk, Fabio Zünd, Marcel Marti, Robert W Sumner, and Markus Gross. Computer-assisted Authoring of Interactive Narratives. In *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*, i3D '15, pages 85–92, New York, NY, USA, 2015. ACM.

[Kapadia et al., 2015b] Mubbasir Kapadia, Fabio Zünd, Jessica Falk, Marcel Marti, Robert W Sumner, and Markus Gross. Evaluating the Authoring Complexity of Interactive Narratives with Interactive Behaviour Trees. In *Foundations of Digital Games*, FDG'15, 2015.

[Kapadia et al., 2016a] Mubbasir Kapadia, Alexander Shoulson, Seth Frey, Cyril Steimer, Samuel Oberholzer, Robert W. Sumner, and Markus Gross. CANVAS: Computer-Assisted Narrative Animation Synthesis. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, New York, NY, USA, 2016. ACM.

[Kapadia et al., 2016b] Mubbasir Kapadia, Alexander Shoulson, Cyril Steimer, Samuel Oberholzer, Robert W. Sumner, and Markus Gross. An event-centric approach to authoring stories in crowds. In *Proceedings of the 9th International Conference on Motion in Games*, MIG '16, pages 15–24, New York, NY, USA, 2016. ACM.

[Karamanoli and Tsinakos, 2015] Persefoni Karamanoli and Avgoustos A Tsinakos. Use of Augmented Reality in Terms of Creativity in School Learning. In *IWEC*, 2015.

[Klein and Murray, 2010] Georg Klein and David W Murray. Simulating Low-Cost Cameras for Augmented Reality Compositing. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):369–380, 2010.

[Knez and Niedenthal, 2008] Igor Knez and S Niedenthal. Lighting in digital game world : Effects on affect and play performance. *Cyberpsychology & Behavior*, 11(2):129–137, 2008.

[Kostadinov et al., 2014] Fabian Kostadinov, Stefan Holm, Bernhard Steubing, Oliver Thees, and Renato Lemm. Simulation of a Swiss wood fuel and roundwood market: An explorative study in agent-based modeling. *Forest Policy and Economics*, pages 105–118, 2014.

[Kruijff et al., 2010] E Kruijff, J E Swan, and S Feiner. Perceptual issues in augmented reality revisited. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 3–12, 2010.

[Leutenegger et al., 2011] Stefan Leutenegger, Margarita Chli, and Roland Yves Siegwart. {BRISK}: Binary robust invariant scalable keypoints. In *ICCV*, pages 2548–2555, 2011.

[Li et al., 2005] Lun Li, David Alderson, John C Doyle, and Walter Willinger. Towards a Theory of Scale-Free Graphs: Definition, Properties, and Implications. *Internet Mathematics*, 2(4):431–523, 2005.

[Liu and Caselles, 2013] Yunqiang Liu and V Caselles. Exemplar-Based Image Inpainting Using Multiscale Graph Cuts. *{IEEE} Transactions on Image Processing*, 22(5):1699–1711, 2013.

[Liu et al., 2013a] Ji Liu, P Musialski, P Wonka, and Jieping Ye. Tensor Completion for Estimating Missing Values in Visual Data. *{IEEE} Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

[Liu et al., 2013b] S. Liu, Y. Wu, E. Wei, M. Liu, and Y. Liu. Storyflow: Tracking the evolution of stories. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2436–2445, Dec 2013.

[Lopes and Baudisch, 2013] Pedro Lopes and Patrick Baudisch. Muscle-propelled force feedback: Bringing force feedback to mobile devices. In *Proceedings of the SIGCHI Conference on Human Fac-*

tors in Computing Systems, CHI '13, pages 2577–2580, New York, NY, USA, 2013. ACM.

[Lowe, 2004] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2):91–110, nov 2004.

[Loyall, 1997] Aaron Bryan Loyall. *Believable agents: building interactive personalities*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1997.

[Magerko et al., 2004] Brian Magerko, John E Laird, Mazin Assanie, Alex Kerfoot, and Devvan Stokes. AI Characters and Directors for Interactive Computer Games. *Artificial Intelligence*, 1001:877–883, 2004.

[Magnenat et al., 2014] Stéphane Magnenat, Morderchai Ben-Ari, Severin Klinger, and Robert W Sumner. Enhancing Robot Programming with Visual Feedback and Augmented Reality. In *Proceedings of the Twentieth Annual Conference on Innovation & Technology in Computer Science Education*, Vilnius, Lithuania, 2014.

[Magnenat et al., 2015] Stéphane Magnenat, Dat Tien Ngio, Fabio Zünd, Mattia Ryffel, Gioacchino Nors, Gerhard Rothlin, Alessia Marra, Maurizio Nitti, Pascal Fua, Markus Gross, and Robert W Sumner. Live Texturing of Augmented Reality Characters from Colored Drawings. In *The 14th IEEE International Symposium on Mixed and Augmented Reality*, Fukuaka, Japan, 2015.

[Mateas and Stern, 2003] Michael Mateas and Andrew Stern. Integrating Plot , Character and Natural Language Processing in the Interactive Drama Facade. In *Proceedings of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment TIDSE03*, volume 2. 2003.

[Mateas and Stern, 2004] Michael Mateas and Andrew Stern. A Behavior Language: Joint action and behavioral idioms. In *Life-Like Characters*, pages 135–161. Springer, 2004.

[Mazza, 2009] Riccardo Mazza. *Introduction to Information Visualization*. Springer Publishing Company, Incorporated, 1 edition, 2009.

[McCloud, 1994] S McCloud. *Understanding Comics*. A Kitchen Sink book. HarperCollins, 1994.

[Menou, 2001] Eric Menou. Real-Time Character Animation Using Multi-layered Scripts and Spacetime Optimization. In *Proceedings of ICVS '01*, pages 135–144, London, UK, 2001. Springer-Verlag.

[Millington and Funge, 2009] Ian Millington and John Funge. *Artificial Intelligence for Games, Second Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2009.

[Minton et al., 1994] Steven Minton, John Bresina, and Mark Drummond. Total-Order and Partial-Order Planning: A Comparative Analysis. *Journal of Artificial Intelligence Research*, 2:227–262, 1994.

[Moreno-Noguer et al., 2009] F Moreno-Noguer, M Salzmann, V Lepetit, and P Fua. Capturing {3D} Stretchable Surfaces from Single Images in Closed Form. In *CVPR*, pages 1842–1849, 2009.

[Mu et al., 2015] Mu Mu, Mark Rouncefield, Yehia Elkhatib, Steven Simpson, Jacco Taal, and Nicholas Race. Ethical challenges in collaborative storytelling. In *Proceedings of the 2015 ACM SIGCOMM Workshop on Ethics in Networked Systems Research*, NS Ethics '15, pages 29–32, New York, NY, USA, 2015. ACM.

[Ngo et al., 2015] D Ngo, J Ostlund, and P Fua. Template-based Monocular 3D Shape Recovery using Laplacian Meshes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[Oh and Kim, 2014] Sungchan Oh and Gyeonghwan Kim. Robust Estimation of Motion Blur Kernel Using a Piecewise-Linear Model. *Image Processing, IEEE Transactions on*, 23(3):1394–1407, 2014.

[Oliveira et al., 2014] J P Oliveira, M A T Figueiredo, and J M Bioucas-Dias. Parametric Blur Estimation for Blind Restoration of Natural Images: Linear Motion and Out-of-Focus. *Image Processing, IEEE Transactions on*, 23(1):466–477, jan 2014.

[Ostlund et al., 2012] J Ostlund, A Varol, D Ngo, and P Fua. Laplacian Meshes for Monocular {3D} Shape Recovery. In *ECCV*, pages 412–425, 2012.

[Ozuysal et al., 2010] Mustafa Ozuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua. Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):448–461, 2010.

[Park et al., 2012] Youngmin Park, Vincent Lepetit, and Woontack Woo. Handling Motion-Blur in 3D Tracking and Rendering for Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1449–1459, sep 2012.

[Pawlik and Augsten, 2011] Mateusz Pawlik and Nikolaus Augsten. RTED: A Robust Algorithm for the Tree Edit Distance. *Proc. VLDB Endow.*, 5(4):334–345, 2011.

[Pearl, 1984] Judea Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984.

[Perlin and Goldberg, 1996] Ken Perlin and Athomas Goldberg. Improv: a system for scripting interactive actors in virtual worlds. In *Proceedings of ACM SIGGRAPH*, pages 205–216, New York, NY, USA, 1996. ACM.

[Perriollat et al., 2011] Mathieu Perriollat, Richard Hartley, and Adrien Bartoli. Monocular Template-Based Reconstruction of Inextensible Surfaces. *International Journal of Computer Vision*, 95(2):124–137, 2011.

[Philippsen, 2006] Roland Philippsen. A Light Formulation of the {E*} Interpolated Path Replanner. Technical report, Autonomous Systems Lab, Ecole Polytechnique Federale de Lausanne, 2006.

[Piacenza et al., 2011] Alberto Piacenza, Fabrizio Guerrini, Nicola Adami, Riccardo Leonardi, Julie Porteous, Jonathan Teutenberg, and Marc Cavazza. Generating story variants with constrained video recombination. page 223. ACM Press, 2011.

[Pilet et al., 2008] J Pilet, V Lepetit, and P Fua. Fast Non-Rigid Surface Detection, Registration and Realistic Augmentation. *International Journal of Computer Vision*, 76(2):109–122, 2008.

[Poulakos et al., 2015] Steven Poulakos, Mubbasir Kapadia, Andrea Schüpfer, Fabio Zünd, Robert Sumner, and Markus Gross. Towards an Accessible Interface for Story World Building. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2015.

[Poulakos et al., 2016] Steven Poulakos, Mubbasir Kapadia, Guido M. Maiga, Fabio Zünd, Markus Gross, and Robert Sumner. Evaluating accessible graphical interfaces for building story worlds. In *Interactive Storytelling: 8th International Conference on Interactive Digital Storytelling, ICIDS 2015, Los Angeles, California, November 15 - 18, 2016, Proceedings*. Springer International Publishing, 2016.

[Poupyrev et al., 2000] Ivan Poupyrev, Rodney Berry, Jun Kurumisawa, Keiko Nakao, Mark Billinghurst, Chris Airola, Hirokazu Kato, Tomoko Yonezawa, and Lewis Baldwin. Augmented groove: Collaborative jamming in augmented reality. In *ACM SIGGRAPH 2000 Conference Abstracts and Applications*, SIGGRAPH 2000, 2000.

[Reeder et al., 2008] J Reeder, G Sukthankar, M Georgiopoulos, and G Anagnostopoulos. Intelligent trading agents for massively multi-player game economies. In *AIIDE*, 2008.

[Rekik et al., 2014] Yosra Mefteh Rekik, Wafik Hachicha, and Younes Boujelbene. Agent-based Modeling and Investors' Behavior Explanation of Asset Price Dynamics on Artificial Financial Markets. *Procedia Economics and Finance*, 13:30–46, 2014.

[Riedl and Bulitko, 2013] Mark O Riedl and Vadim Bulitko. Interactive Narrative: An Intelligent Systems Approach. *AI Magazine*, 34(1):67–77, 2013.

[Riedl and Young, 2006] Mark O Riedl and R Michael Young. From linear story generation to branching story graphs. *Computer Graphics and Applications, IEEE*, 26(3):23–31, 2006.

[Robin, 2008] Bernard R. Robin. Digital storytelling: A powerful technology tool for the 21st century classroom. *Theory Into Practice*, 47(3):220–228, 2008.

[Rublee et al., 2011] E Rublee, V Rabaud, K Konolidge, and G Bradski. {ORB}: An Efficient Alternative to {SIFT} or {SURF}. In *ICCV*, pages 2564–2571, 2011.

[Runco, 2004] Mark A Runco. Everyone Has Creative Potential. In *Creativity: From potential to realization*. American Psychological Association, Washington, DC, US, 2004.

[Sacerdoti, 1975] Earl D Sacerdoti. The Nonlinear Nature of Plans. In *Proceedings of the 4th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'75, pages 206–214, San Francisco, CA, USA, 1975. Morgan Kaufmann Publishers Inc.

[Salisbury et al., 2004] Kenneth Salisbury, Francois Conti, and Federico Barbagli. Haptic rendering: Introductory concepts. *IEEE Comput. Graph. Appl.*, 24(2):24–32, March 2004.

[Salzmann and Fua, 2010] M Salzmann and P Fua. *Deformable Surface 3D Reconstruction from Monocular Images*. Morgan-Claypool, 2010.

[Salzmann and Fua, 2011] M Salzmann and P Fua. Linear Local Models for Monocular Reconstruction of Deformable Surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):931–944, 2011.

[Salzmann et al., 2008] M Salzmann, F Moreno-Noguer, V Lepetit, and P Fua. Closed-Form Solution to Non-Rigid {3D} Surface Registration. In *ECCV*, pages 581–594, 2008.

[Scherrer et al., 2008] Camille Scherrer, Julien Pilet, Pascal Fua, and Vincent Lepetit. The Haunted Book. In *ISMAR*, pages 163–164. IEEE Computer Society, 2008.

[Sharan et al., 2013] Lavanya Sharan, Zhe Han Neo, Kenny Mitchell, and Jessica K Hodgins. Simulated Motion Blur Does Not Improve Player Experience in Racing Game. In *Proceedings of Motion on Games*, MIG '13, pages 149:149—-149:154, New York, NY, USA, 2013. ACM.

[Shen et al., 2009] Edward Yu-Te Shen, Henry Lieberman, and Glorianna Davenport. What's next?: emergent storytelling from video collection. page 809. ACM Press, 2009.

[Shoulson et al., 2013a] Alexander Shoulson, Max L Gilbert, Mubbasir Kapadia, and Norman I Badler. An Event-Centric Planning Approach for Dynamic Real-Time Narrative. In *Proceedings of Motion on Games*, MIG '13, pages 99:121—-99:130, New York, NY, USA, 2013. ACM.

[Shoulson et al., 2013b] Alexander Shoulson, Mubbasir Kapadia, and Norman Badler. PAStE: A Platform for Adaptive Storytelling with Events. In *INT VI, AIIDE Workshop*, 2013.

[Shoulson et al., 2014] A Shoulson, N Marshak, M Kapadia, and N I Badler. ADAPT: The Agent Developmentand Prototyping Testbed. *IEEE TVCG*, 20(7):1035–1047, 2014.

[Si et al., 2005] Mei Si, Stacy C Marsella, and David V Pynadath. THESPIAN: An Architecture for Interactive Pedagogical Drama. In *Proceeding of the 2005 Conference on Artificial Intelligence in Education*, pages 595–602. 2005.

[Sielhorst et al., 2007] T Sielhorst, Wu Sa, A Khamene, F Sauer, and N Navab. Measurement of absolute latency for video see through augmented reality. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 215–220, nov 2007.

[Smith, 1994] Vernon Smith. Economics in the Laboratory. *The Journal of Economic Perspectives*, 8(1):113–131, 1994.

[Sodhi et al., 2013] Rajinder Sodhi, Ivan Poupyrev, Matthew Glisson, and Ali Israr. Aireal: Interactive tactile experiences in free air. *ACM Trans. Graph.*, 32(4):134:1–134:10, July 2013.

[Sorkine et al., 2004] O Sorkine, D Cohen-Or, Y Lipman, M Alexa, C Rössl, and H.-P. Seidel. Laplacian Surface Editing. In *Symposium on Geometry Processing*, pages 175–184, 2004.

[Steed, 1993] Anthony Steed. *A survey of virtual reality literature*. University of London. Queen Mary and Westfield College. Department of Computer Science, 1993.

[Sumner and Popović, 2004] Robert W. Sumner and J Popović. Deformation transfer for triangle meshes. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 2004.

[Swan(II) and Gabbard, 2005] J Edward Swan(II) and Joseph L Gabbard. Survey of User-Based Experimentation in Augmented Reality. In *Proceedings of 1st International Conference on Virtual Reality, HCI International 2005*, 2005.

[Tao et al., 2013] Michael W Tao, Jitendra Malik, and Ravi Ramamoorthi. Sharpening Out of Focus Images using High-Frequency Transfer. In *Computer Graphics Forum*, volume 32, pages 489–498. The Eurographics Association and Blackwell Publishing Ltd., 2013.

[Thomas, 2012] Bruce H. Thomas. A survey of visual, mixed, and augmented reality gaming. *Comput. Entertain.*, 10(1):3:1–3:33, December 2012.

[Thue et al., 2007] David Thue, Vadim Bulitko, Marcia Spetch, and Eric Wasylishen. Interactive storytelling: A player modelling approach. In *AIIDE*, 2007.

[Varol et al., 2012] A Varol, M Salzmann, P Fua, and R Urtasun. A Constrained Latent Variable Model. In *CVPR*, pages 2248–2255, 2012.

[Wei et al., 2015] Xiaodong Wei, Dongdong Weng, Yue Liu, and Yongtian Wang. Teaching based on augmented reality for a technical creative design course. *Computers & Education*, 81:221–234, 2015.

[Weyhrauch, 1997] Peter William Weyhrauch. *Guiding interactive drama*. PhD thesis, Pittsburgh, PA, USA, 1997.

[Wood et al., 2004] Richard T A Wood, Mark D Griffiths, Darren Chappell, and Mark N O Davies. The Structural Characteristics of Video Games: A Psycho-Structural Analysis. *Cyberpsy., Behavior, and Soc. Networking*, 7(1):1–10, 2004.

[Yamaguchi, 2004] Hiroshi Yamaguchi. An Analysis of Virtual Currencies in Online Games. *SSRN Electronic Journal*, 2004.

[Zhang and Shasha, 1989] K Zhang and D Shasha. Simple Fast Algorithms for the Editing Distance Between Trees and Related Problems. *SIAM J. Comput.*, 18(6):1245–1262, 1989.

[Zhao et al., 2014] Zhenpeng Zhao, Sriram Karthik Badam, Senthil Chandrasegaran, Deok Gun Park, Niklas L E Elmqvist, Lorraine Kisselburgh, and Karthik Ramani. skWiki: A Multimedia Sketching System for Collaborative Creativity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1235–1244, New York, NY, USA, 2014. ACM.

[Zünd et al., 2013] Fabio Zünd, Yael Pritch, Alexander Sorkine-Hornung, Stefan Mangold, and Thomas Gross. Content-aware compression using saliency-driven image retargeting. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 1845–1849, sep 2013.

[Zünd et al., 2014] Fabio Zünd, Marcel Lancelle, Mattia Ryffel, Robert W Sumner, Kenny Mitchell, and Markus Gross. Influence of Animated Reality Mixing Techniques on User Experience. In *Proceedings of the Seventh International Conference on Motion in Games*, MIG '14, pages 125–132, New York, NY, USA, 2014. ACM.

[Zünd et al., 2015a] Fabio Zünd, Pascal Bérard, Alexandre Chapiro, Stefan Schmid, Mattia Ryffel, Markus Gross, Amit H Bermano, and Robert W Sumner. Unfolding the 8-bit Era. In *Proceedings* *of the 12th European Conference on Visual Media Production*, CVMP '15, pages 9:1—-9:10, New York, NY, USA, 2015. ACM.

[Zünd et al., 2015b] Fabio Zünd, Mattia Ryffel, Stéphane Magnenat, Alessia Marra, Maurizio Nitti, Mubbasir Kapadia, Gioacchino Noris, Kenny Mitchell, Markus Gross, and Robert W Sumner. Augmented Creativity: Bridging the Real and Virtual Worlds to Enhance Creative Play. In *SIGGRAPH Asia 2015 Mobile Graphics and Interactive Applications*, SA '15, page 21, New York, NY, USA, 2015. ACM.

[Zünd et al., 2016] Fabio Zünd, Miriam Tschanen, Marcel Lancelle, Maria Olivares, Mattia Ryffel, Alessia Marra, Milan Bombsch, Markus Gross, and Robert W. Sumner. Gnome Trader: A Location-Based Augmented Reality Trading Game. In *9th International Conference on Game and Entertainment Technologies (GET)*, 2016.