IMPROV REMIX
Video Manipulation Using Whole-Body Interaction To Extend Improvised Theatre

Dustin Freeman

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy,
Graduate Department of Computer Science,
in the University of Toronto

IMPROV REMIX
Video Manipulation Using Whole-Body Interaction To Extend Improvised Theatre

Abstract

This work represents a technical implementation of features seen in modern theatrical improvisation: re-use of previous concepts and moments in performance via recontextualization, coordinated by gestures between the performers on stage. We present a system for manipulating the video content of a stage show, and re-projecting previous moments on the stage. As part of this thesis, we have explored the generic research problems of live video editing and separating for-system interaction (*foreground activity*) from noisy gesticulation (*background activity*). We have workshopped and designed iterations of our system extensively with experienced theatrical improvisation performers. We have exhibited this work in a public showcase, and have observed the impact on interacting with a system, while performance is the primary activity, on both performance and audience members. We provide several use cases discovered during performance, and reflect on thinking of theatrical improv performers as a special group of users.

Acknowledgements

Here, I acknowledge the bevy of fantastic people who have been essential to my personal and academic growth, which, at this moment in time, culminates in this Doctorate of Philosophy in Computer Science.

I would like to thank members of the Dynamic Graphics Project (DGP) lab for putting up with my large, long-term physical setups, especially Haijun Xia for helping adjust it, Ricardo Jota for project and time management, and John Hancock and Ingrid Varga for technical support. Paul Stoesser (technical director, University of Toronto Centre for Drama, Theatre and Performance Studios) and the Luelley Massey Studio Theatre for access to theatre spaces. I would also like to thank the members of the Toronto theatre community for participating in development, especially Xander Williams, Elyse Waugh, Alex James, Justine Cargo, Joe Law and Oliver Georgiou.

I would like to acknowledge those involved in the *Tweetris* project, especially the originator, Derek Reilly. Tweetris stirred my initial interest in interactor/audience relationships, and public art installations.

My supervisor, Ravin Balakrishnan, deserves thanks for instructing me to "do the thesis only you could do", which steered my work towards doing something novel with improvised theatre.

And to Montgomery Martin, my technical director on *Improv Remix*, who knew lots of things I didn't. And to Bruce Barton and Daniel Wigdor for serving on my committee.

And I would like to give an extra-special thanks to Ned Dickens, who, long ago, planted the idea of "repetition, with variation" in my head while we collaborated on *Heterotopia*.

Finally, here is a list of people who I have collaborated with in previous projects, who all owe some degree of thanks: Gibson/Martelli (formerly Igloo), Karan Singh, Daniel Vogel, Steve Engels, Fanny Chevalier, Emma Westecott, Kyle Duffield, Otmar Hilliges, Shahram Izadi, Abigail Sellen, Sriganesh Madhvanath, Hrvoje Benko, Merrie Ringel-Morris, Paul Hoover, Jarrod Lombardo, Julian Lepinski, Eric Akaoka, Jim Davies, Richard Allen, Caroline Baillie and Critical Stage, Liam Karry and Single Thread Theatre Company.

Copyright Notices and Disclaimers

Sections of this document have appeared in other publications.

Association for Computing Machinery

portions of chapter 4

Freeman, D., Santosa, S., Chevalier, F., Balakrishnan, R. and Singh, K. (2014). LACES: Live Authoring through Compositing and Editing of Streaming Video. In *Proceedings of the 32nd international Conference on Human Factors in Computing Systems* (Toronto, ON, Canada, April 26 - May 1, 2014). CHI '15. ACM, New York, NY, 1207-1216.

Contents

List of Figures

1

Introduction & Motivation

This thesis starts from two missing links:

- Tools for manipulating video are not spontaneous
- Interactive technology does not frequently appear in improvised theatre

Video content is typically dead, as it has been rendered or created at some point in the past. Existing tools for manipulating video do not allow us to do so quickly or spontaneously as part of a creative conversation with ourselves or others. When technology is included in theatre, it often appears as merely responding to the performers, not working with them. Other times, while technology may be genuinely responding to performers, the performer's actions are pre-scripted so that the technology's response does not need to be interactive at all.

I aim to examine the use of whole-body gestures for improv theatre performers to control recorded and reprojected video on the stage. In this work, the performer will be directly *using* a novel piece of technology onstage, to extend, while still respecting, the pre-existing art-form of theatrical longform improvisation.

Utilizing depth cameras and bespoke software, I shall enable performers to capture their and other performers' performances in a stage environment. Once captured, these performances can be manipulated and re-projected into the space, enabling the live performers to create scenes not possible before, or to make direct references to previous moments in the performance. Performers will trigger and manipulate the recordings of previous performances using in-air gestures, which need to be distinguishable from the normal movements that arise in natural acting performance.

In the light of choosing to use whole-body interaction as the interaction medium, It is a fair question to ask *why not use buttons?* There are two reasons for this that will be discussed further in the thesis. First, the requirement of gross-scale gestures ensures that interactions will be visible — *exposed* — to other performers and audience members, which encourage transparency and collaboration and is consistent with the themes of Modern Improvised Theatre. Second, if performers are unencumbered and the stage is devoid of buttons, then when it is not being used, the system effectively disappears, so that it serves the performers in their production of entertainment, and not the other way around.

The final output of this work, the system *Improv Remix*, is the product of several iterative stages, including early workshops and experimentation, and ideation through other exploratory research problems.

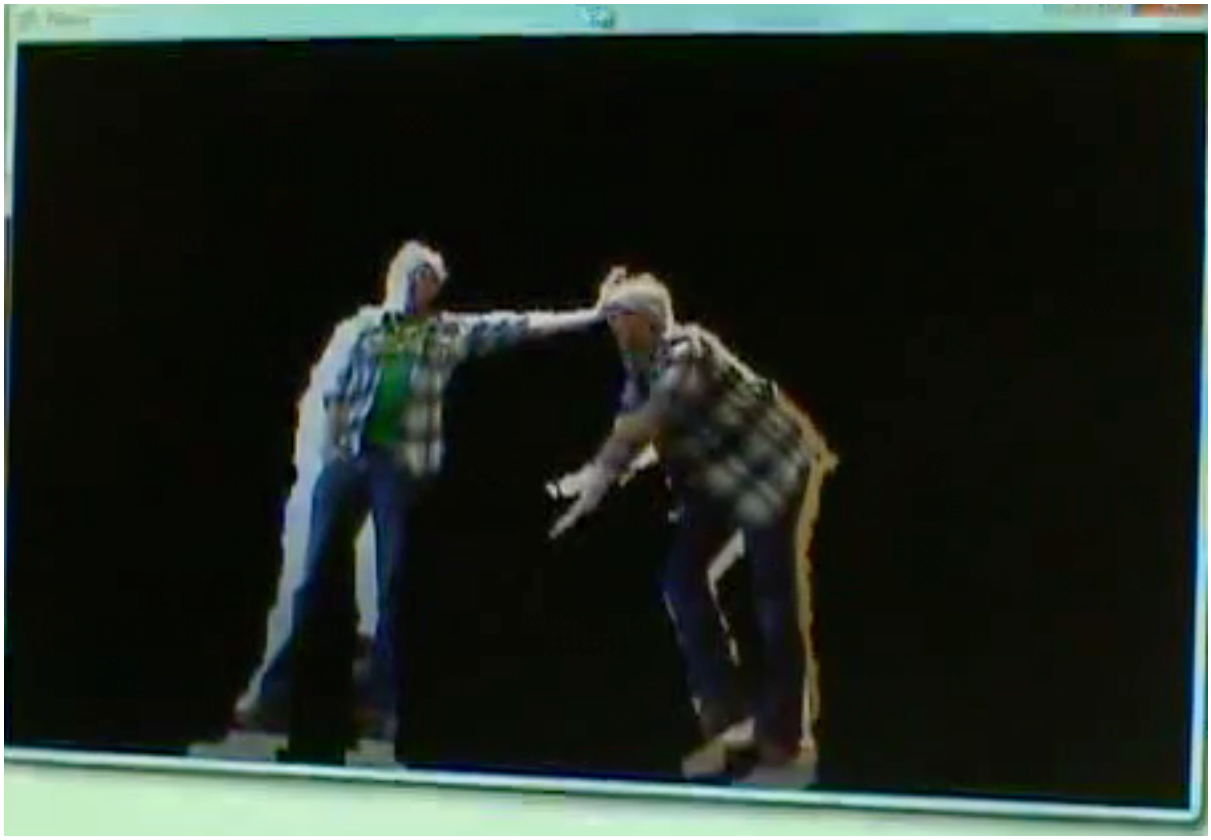

Figure 1.1: A picture of slapstick interaction between a live and playback performer in the first prototype.

1.1 Motivation

Improvisation, in terms of the spontaneous play with other people using meaning and motion, has always been a love of mine. Improv, compared to other forms of theatre, has always been low-tech — bare-bones even. Improvisors are often instructed not to wear distracting clothing, or anything that has a logo. There is a sense that an improvisation performance is a special space, similar to meditation, where we forget all other input or anything we bring with ourselves into that space, and instead respond honestly to what we encounter in it. It seems that the intention of avoiding technology or other outside input is that it brings with it pre-loaded ideas, inhibiting our spontaneity. Obviously, this is a very personal interpretation of my own experiences, but also the result of working and conversing with many directors and performers over my years of experience.

While searching for a suitable research direction, with the instruction that I should "do the doctorate that only you would be able to do", I envisioned and very quickly made a prototype. This prototype recorded Kinect RGB+D (colour+depth) video in a buffer. Output on a computer monitor showed the live version of myself, where I used the depth in the Kinect to cut myself out from the background. I could start playback from different times in the buffer by pressing number keys, and the playback of previous times was merged with the present. In fact, as the merging took place on a monitor, there was no discernible difference between the live and the recorded (Figure 1.1)! There was no audio implementation at this stage.

As I played with this tool/instrument, I found I could satisfyingly do a few slapstick routines, play rock paper scissors with myself, fight myself, or appear to hang out in a group of selves on a street corner. This felt satisfying, but clunky. I noticed immediately that one playback of myself could be interpreted in different ways based on how the other playbacks appeared to react to it. Clearly, I had something.

There was a diversity of directions I could take this system, but I wanted to see how other improv performers that I trusted to be open-minded would use it, and take inspiration from them. Building a system that is just for me to ... play with myself was certainly not appealing. I also knew that I did not want a system where someone offstage mysteriously controlled an onstage performer, as if creepily from the rafters of a dark theatre. One universally-agreed rule about the practice of improvisation that I have always loved is that no one ever exclusively owns their own ideas — they are to be available for others to re-use in ways the originator did not expect, and cannot protest. For this philosophy to work, the control of the videos on stage had to be available to anyone, at any time, and they should not have to go through the authority of any one person.

1.2 Features of Modern Improvisation

Improv is popularly seen in the TV show *Whose Line Is It Anyway?*. In *Whose Line*, like in a lot of improvisation, an inspirational suggestion is given to performers at the beginning. A suggestion can be simple, e.g. "This scene will take place in a living room" to complex, e.g. "In this scene, player A is slowly turning into a pig, while player B feels a strong urge to always be close to the door". How improv scenes progress is unpredictable, exploratory and messy, to the joy of the performers and audience. Ideas are discovered that would have been difficult to do so otherwise. The scenes that appear in *Whose Line* are usually 1-4 minutes long — this is called *short-form* improv. *Longform* improv, with sets of 20-40 minutes long, tends to include many inter-connected scenes. Charna Halpern and Del Close extensively document longform improv practice in *Truth In Comedy* [Halpern et al., 1994]. Actor Bill Murray once referred to longform improvisation as "The most important group work since they built the pyramids".

Extending the genre of longform improvisation was the primary motivation for this work. By extension, we mean providing supplementary expressive abilities to performers that are thematically consistent with their currently practices. The design space of technology usage on stage is very large, and we want performances using our system to appear like an extrapolation of features of the genre of longform theatrical improvisation into territory not possible before.

We will describe three inspiring features of longform improv: *Minimalism*, *Coordinating Gestures*, and *Recontextualization*. For more detail, we refer the reader to the Improv section in our Background, **Section 3.3**.

1.2.1 Minimalism

Improv intentionally minimizes influences on the spontaneous generation of content. Most groups use neutral clothing and few props, as semantic suggestions may constrain the thematic development of scenes. Physical props are rarely used, and if an object is required in a scene it is mimed instead. If a group had to constrain themselves based on the set of props they had on hand, it would restrict the

scene thematically[1].

Many forms of theatre use special effects, including changes in set and lighting, and in the modern era, sound effects and projections. Jerry Grotowski argued in his 1967 essay *Towards a Poor Theatre* [Grotowski et al., 1967], that theatre, in the context of other competing medias such as film, should strip away superfluous elements until only the "actor-spectator relationship" remains. Theatre's liveness is what makes it unique [Dixon, 2007], and the inclusion of technology can be problematic to keeping theatre live and spontaneous. Lighting changes or sound effects are often controlled by technicians from offstage, using a pre-designed list of cues. These cues inhibit improvisation, in the form of customizing the show for the particular audience, or recent events, as may have occurred in centuries past in forms such as *commedia dell'arte*. From our understanding of the value of minimalism in improvisation and theatre, it is important that there are as few constraints and distractions as possible on the performer when they want to generate new content.

In the last century, there have been experiments with controlling technology from *onstage*, but in my opinion this interactivity has not been fully realized. Either the technology intrudes so much on the show that it becomes cyborgized — the show becomes about technology — or the technology merely responds to performers in a vague aesthetic way, and the performers do not have any practical control over the technology. To clarify these two problems we see currently with technology used in stage performance: first, it tends to be so obviously present that its presence takes over the show thematically and second, there is very little exploration of narrative performer's controlled, intentional usage of technology on stage. A seeming exception to the second problem is musical performance, where it is clear, from the perspective of the audience, that the performers are controlling an instrument. However, in a musical performance, maintaining a character is not as important as in narrative performance. Additionally, audiences tend to have a basic literacy for how musical instruments work: it is clear when a performer is tuning a guitar, which is understood to not be part of the musical performance.

1.2.2 Coordinating Gestures

There are several gestures used for coordinating a improv scene between performers. Coordinating are an established property of improv theatre, demonstrating that it is possible for gestures and performance to co-exist. Their primary function is to communicate to other performers, but a secondary stakeholder is the audience, who must be able to understand if an onstage action is part of a performer's acting, or the work of planning the scene. We shall draw inspiration from these gestures for the design of our system. Here we note some of the more pertinent gestures:

Sweep: A performer runs from one side of the stage to the other at the front of the stage. Similar to transitional wipes used in film, this indicates that the scene is over and the group will transition to the next one.

Tag-out: A performer from offstage comes and taps an onstage performer on the shoulder. This indicates that the tagging performer will replace the onstage performer as a new character, in the same scene.

[1] An exception some readers may recall is the game *Props* on *Whose Line Is It Anyway?*. In this case, the props are used as an intentionally-absurd constraint on the scene.

"Cut to that!": A verbal command, suggesting that the scene should transition to a mentioned event. E.g., if a performer describes a birthday cake he ate, when "cut to that!" is called, it is a signal to switch to a depiction of eating the cake.

These and other coordinating actions will be discussed in more detail later.

1.2.3 Recontextualization

Modern improvisation makes frequent usage of recontextualization: taking previous events or themes and juxtaposing them against others in the present for examination and entertainment. A common comedic activity is to take an everyday event and to describe it in a thorough, literal way, thereby exposing its absurdity - this is *observational comedy*.

The most prevalent longform structure, *The Harold* [Halpern et al., 1994], revisits scenes and themes multiple times during a show. The act of referring to scenic material that has not appeared in the show for a long time is referred to as a *callback*. Callbacks are frequent in stand-up and sketch comedy, where an innocuous joke or event at the beginning appears again at the end. The term *Chekhov's Gun* originates from Russian playwright Anton Chekhov, who observed that "If you say in the first chapter that there is a rifle hanging on the wall, in the second or third chapter it absolutely must go off." [Bill, 1987].

Additionally, we have been fascinated with the philosophy behind Augusto Boal's *Theatre of the Oppressed*, where a scene is acted out one way by performers, and then the audience is asked how the scene could have gone differently. Sometimes the audience is encouraged to call out suggestions for action that the performers must follow, but eventually the audience is encourage to jump on the stage and take over from the performer. Boal sees this as empowering the audience: "We destroy the work offered by the artists in order to construct a new work out of it, together"[Boal, 1995]. Our system has the fascinating property that after the formal show, audience members can approach the empty stage and call back previous performances, interacting with them by interpreting them in ways which the original performers did not anticipate, or have even given permission for.

To bring it to another level, recontextualization can appear to be hostile to the source material or source author — this is *co-opting* it for another purpose. As with any artistic reinterpretation, the original author loses control of the material; this is an important decentralization of expressive power. Frost and Yarrow call this *co-creativity* [Frost and Yarrow, 2007].

This activity of observing or producing disconnected material, and then explaining its connection afterwards, is a central feature of longform improv — any orphaned material stands out as not satisfying the property of Chekhov's Gun, and must be connected. In the beginning of a longform improvisational set, performers produce a series of initially disconnected ideas, but during the set, almost unavoidably, they weave them together in amusing and unexpected ways. This structure of improvisation is amenable to additional techniques to call back scenes — in our approach, to enable playback and manipulation of video of the scenes themselves, in new contexts.

1.3 Document Outline

1.3.1 Contributions

These contributions of this thesis are, in chapter order:

1. A technological extension of the improvised theatre art form & documentation of this process (**Whole thesis**),
2. Methods for live editing & re-use of video for consumers on a tablet (**LACES, Chapter 4**),
3. Methods to approach the problem of detecting foreground (for-system) human activity, in the midst of background (any other) human activity (**Background Activity, Chapter 5**),
4. An academic analysis of how interaction with a system co-habits with theatrical performance (**Chapter 6**),
5. A set of novel interaction techniques, designed for improv performers to live-manipulate stage video (**Improv Remix, Chapter 7**),
6. A discussion of performers as a special type of user, client and collaborator (**Chapter 9**).

Immediately after this chapter, in **Chapter 2**, I describe the initial workshops where I prototyped some of these ideas and identified the core problems of this work, as well as some initial experiments in processing and manipulating video of performance (*Actor DJ*). A literature review follows in **Chapter 3**.

Our first contribution chapters following the Background explore related problems to those encountered in the thesis, in a more generalized way. In **Chapter 4**, *LACES: Live Authoring through Compositing and Editing of Streaming Video*, we explore live editing of a input stream of video on a tablet, as a consumer-centric context than improvised theatre. From our workshops, we identified that system detection of gestural interaction, intermixed with a larger volume of noisy movements, was a difficult problem. To this end, in **Chapter 5** we define and study the concept of *Background Activity*, denoting activity of users that is observable by a system, but distinct from users' interaction with the system, which we call *Foreground Activity*.

Given our understanding of interaction and theatrical performance, from our workshops as well as our study of Background Activity, we provide a high-level analysis of the problem of intermixing interaction with performance in **Chapter 6**.

We describe the system we created, *Improv Remix*, in **Chapter 7**, including the physical setup, iterative prototypes and a documentation of the final design. We evaluated Improv Remix in a showcase open to the public, and we provide documentation of our observations, as well as an enumeration of several use cases for the system in **Chapter 8**.

In our final chapter, **Chapter 9**, we discuss our observations about the problem space, and about working with performers. Performers have been involved in this process both as users and designers, and I have found that they are a unique form of user. We close with potential for future work.

1.3.2 Definitions

For the purposes document, we will use a few words as shorthand for more complex concepts. We are not defining these words anew, merely describing our present usage:

Performer

Typically in Human-Computer Interaction literature, we refer to the human being interacting with the system we have created and/or are studying as the *user*. If it is specifically in the context of a rigorous study, they are a *participant*. In this work, we use the term *performer*, as we define their task as creating an interesting theatrical performance first, of which using a system we create is just a part. Additionally, many of the activities we discuss relate directly to performance, not just interaction with a system.

Action versus Interaction

An *action* is anything a performer does, but an *interaction* is an action by a performer to communicate with the technological system that is part of the stage performance.

Performer versus Character

A *performer* is the real person who is part of the theatre performance. As part of the show, they may act as if someone else, a *character*. We may refer to their actions as "in-character" or "out-of-character".

On-Stage versus Off-Stage

For a performer to be *on-stage*, it is not necessarily required that they are on a physical, raised stage. Events that are on-stage are understood to be part of the show, and could include the actors running through the audience, or even sending text messages to audience members. *Off-stage* events are those that are not considered to be part of the show.

Live Performer versus Playback Performer

For the purposes of this work, a *live performer* is performing in the present, whereas a *playback performer* is a performance that was recorded previously, playing back in the present. Note that some other writing uses the term live performer to distinguish from performers whose audio or video is being streamed, in the present, from somewhere else.

2
Workshops & Experiments

To explore the ideas presented in the Introduction I created two prototypes. First, a basic system for capture and re-projection of stage-like video that we used for two workshops with performers, and second, a system for parsing and replaying scenes, called *Actor DJ*.

2.1 Workshop with Prototype

I created a prototype system, and invited personal friends who were improvisors to play with it. While I had many ideas about how the system could be used, I wanted to influence them as little as possible, and instead see how they would use it. The goal I had in mind was to explore the system as something that was useful, that could be used by performers in the midst of performance, but was not any sort of anthropomorphic being, "co-performing" with them, as other systems have been in previous mixings of technology and theatre. Like a good technology, I wanted it to disappear into the background as a tool, and that the performance would instead be interesting as a fusion. With the workshops, I also wanted to explore what sort of scenes would work well in the system; how and when would performances be reusable?

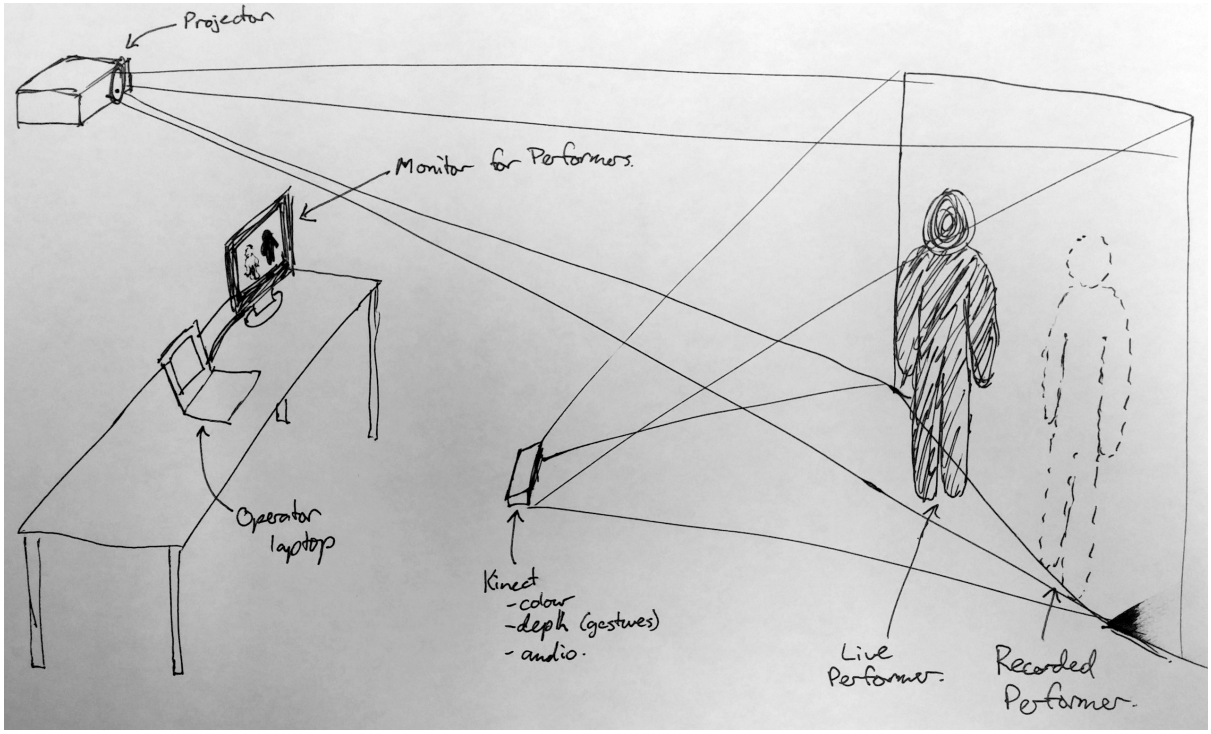

Figure 2.1: Physical setup for the workshops.

2.1.1 Physical Setup

The workshop setup consisted of a projector and a Microsoft Xbox Kinect pointed at the same area of white wall, approximately 2 metres high by 3 metres wide (Figure 2.1). The white wall represented the acting space, the "stage", where live performers would perform and captured scenes would be re-projected, at approximately life-size and in their original position. A large monitor, showing a feedback of the view of the Kinect with some annotations, faced the stage from approximately 4 metres away. The laptop controlling the system is next to the monitor. Performers could wait on the sidelines while considering scenes.

2.1.2 Software and Interaction

For the amount of video we needed to handle during the workshop, we needed to cache it to disk. The system's state was either recording the current input video and audio or not, and could also be playing any number of videos from before. Scenes cached to disk were assigned a scene number, which the system printed out to a command line interface, in this case controlled by myself.

The system recorded a new scene from when the first person appeared on the stage until the last person left it: more precisely, a scene was a continuous series of successive frames with at least one person on the stage. I had a command line interface that reported when a scene finished, and printed the new scene's number. I acted as the operator of this system, able to bring back scenes on request. A section of this interface could look something like the following, where the angle bracket (>) indicates user input.

```
Scene 117 begins.
Scene 117 ends.
```

```
> play 117
Playing Scene 117
Scene 118 begins.
Scene 117 stops.
Scene 118 ends.
> loop 118
Looping Scene 118.
> stop 118
Stopped looping Scene 118.
```

The number 118 is not overly high - after a 4-hour session, we ended up with over 300 scenes.

I wanted to experiment with some gestural interaction. However, the system had a large number of features to control, and I did not want to test gestural interaction at the same time as testing the principle of performers' interaction with video playback. This approach is sort of like a Wizard of Oz system, except the users are aware that there is a Wizard. To lightly test interaction, I implemented a single gesture - a clap, that triggered if the users' hands went within a close proximity of each other. This would start playing the immediately previous scene.

2.1.3 Participants and Recruitment

I will characterize improvisors as participants first, then describe the recruitment process.

I invited participants known through personal relationships via a Facebook event. I advertised that the event was going from from the mid-afternoon until late at night on a single day, and improvisors could show up at any time. The session was advertised as an undirected "play" session, with no set goals in mind, other than "let's see what we can do with the system".

All the attendees were very experienced theatrical improvisors. Some engaged in related activities such as stand-up or filmed sketch comedy. All of them had at least 2 years' longform improv training with the now-defunct Impatient Theatre Company in Toronto. One of the improvisors, Sean Tabares, was the recipient of the Best Male Improviser award at the 2010 Canadian Comedy Awards. [1]

2.1.4 Procedure

The play session lasted for 5 hours, with a total of 15 improvisors attending, dropping in and out when they were able to. There were always at least 3 participants present. At the beginning, I introduced all the features of the system to the improvisors. However, I avoided introducing any use cases I had anticipated unless improvisors were at a loss for what to do. I found that as the creative enthusiasm of the participants built momentum, I had to explain less and less, and only acted to answer questions, and respond when there was obvious confusion. Participants naturally explained how the interface worked to new participants that just arrived.

[1] http://www.canadiancomedy.ca/awardwinners.php?year=2010

2.1.5 Observations

The workshop flowed very smoothly and it required little effort on the researcher's part — this is no doubt due to the fact that almost everyone participating had already worked together in a creative capacity. I had to prompt the improvisors on how the system could be used at the beginning, but once the session starting going, momentum was maintained. Enthusiasm varied between improvisors, some being more assertive about trying out their ideas, while others were content to sit and think more quietly. If you were to sit and watch the session, it would not resemble an entertaining performance as there were long periods of talking where we tried to figure out ways to use the system.

Here we'll list several observations we made during the workshops. In subsequent sections, we will list Discovered Use Cases, and later Proposed Use Cases that were not possible to prototype in the current system.

The Importance of Spontaneity

To execute ideas, performers would have to describe them to the researcher operating the prototype. New ideas are vague and difficult to express, and if the performer was uncertain, they would abandon them. We feel that operation of the system from the stage itself, by performers, is very important.

Camera Proximity

Proximity to the camera is a problem. As we knew before, if you're slightly closer to the camera, you'll be much bigger in the re-projection. Performers stepping too close to the camera happened often and performers had trouble controlling it. Performers ideally would place themselves as close to the back wall as possible, but it was uncomfortable to be physically expressive in close proximity to a wall. Additionally, it was hard to see the projected playback performers while right next to them.

Dealing with Unexpected System Behaviour

Scenes were brought back either by the clapping gesture or the operator's command-line prompts. I had to note down scene numbers with short descriptions as we were playing during the workshop, as performers would only vaguely describe a scene to me when they wanted me to bring it back to play with. There were often bugs, and a scene would start playing unexpectedly due to a clap false positive or I would enter the wrong scene number on the prompt. This elicited laughter, but not good, satisfying laughter as it was a reaction to the system appearing broken and disjoint, rather than profound. My attitude in this sense is clearly biased, as I felt any misbehaviour of the system was *my fault*. My initial urge was to reassure, rather than think high-level about the bug.

Distinguishing Interface Actions from Performance

Performers' interaction techniques were not robust. The clap gesture to play the last scene frequently caused false positives and often had to be disabled. The method to stop and start recording (walking on or off stage) appeared at the beginning and end of every recorded scene and was obtrusive. When

walking on or off, the performers would be "out of character", ruining the suspension of disbelief for that recording.

Performers responded to false positives for "clap" in an interesting way — creative people wear their feelings openly, and they started swearing and blaming themselves for the system making a mistake. Their movements become cautious, tentative, uncomfortable. In their creative, improvisational mode, the scenes they created became about personifying the evil nature of the system. Clearly, gestural interaction in a creative, performance setting has to live by a higher standard than in the typical settings studied in Human-Computer Interaction (HCI) research.

As scenes started and ended when a player entered or left the stage respectively, performers became slightly frustrated that any recorded scene seemed to have junk pieces at the beginning and end. Performers were not in character when they entered or left the stage, unless the act of leaving or entering was part of the character. As audience members, we are willing to accept someone entering the stage as not being part of the performance yet. However, when the entrance onto the stage is played back in a video, it seems like the choice to play that part of the video is imbuing it with significance, and it appears disjoint to watch that part of the video. One performer found a workaround to having walk-on and walk-off as part of the video, by sliding their back along the wall, as if hiding. The Kinect only found the user when he jumped forward, so he could effectively start scenes in the middle of the stage. However, he found that you could not "fade" back into the wall once the Kinect had started tracking you.

Perception of Live and Recorded

Live and recorded performers looked physically very different. In order for the live performer to be captured with sufficient brightness, I had to light the stage area so much it would partially wash out the projection. When watching, performers stated that it did not feel like the live and recorded performers were in the same space. In fact, they preferred to record separate performers and later combine them together, instead of watching a live and recorded performer simultaneously. Clearly, work should be done so that it can feel like live and recorded performers are in the same space.

Location on Stage

The system projected scenes directly back on stage as they were recorded, with no spatial manipulation. Performers became very mindful of space on stage between current and previous scenes. This took a lot of cognitive load and reduced the spontaneity of the play. Performers would often give instructions to each other, either to leave space for hypothetical following scenes, or describing where they were physically in previous scenes. This spatial organization seemed to be a hindrance to the actual creative work.

Since the stage was a special space, where performers knew they'd be performing a scene if they were present, they tended to avoid it unless they had an idea in mind. Performers would often describe the idea they were trying, then step on the stage to perform it, then leave again and describe their thoughts on what they had just done. As the only way to get from one side of the stage to the other was to cross through the stage space, performers tended to stick to one side or the other unless they were in a scene.

Performer Orientation & Sightlines

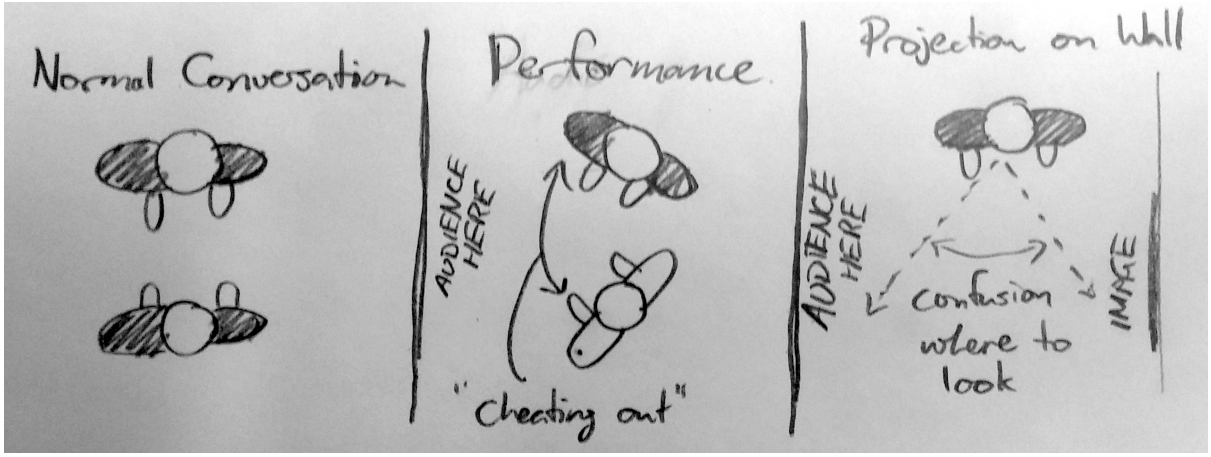

Figure 2.2: Performer Orientation in Different Set-Ups

A more subtle problem was that performers had trouble orienting themselves (see Figure 2.2). In normal conversation (1st frame in Figure 2.2), conversation participants directly face each other. In theatrical performance (2nd frame in Figure 2.2), there is the concept of *cheating out*, where a performer orients their gaze and torso towards the audience to make themselves more interesting to watch and to communicate body language more clearly. In our workshop setup (3rd frame in Figure 2.2), the performer had to look towards the wall to see and provide the feeling that they were acting towards the projection, but they also had to give themselves towards the audience, far in the opposite direction. This led to swinging their neck and torso uncomfortably back and forth, and if they ignored the hypothetical audience and solely faced towards the projection, then the image recorded of them would mostly be their back. When these recordings were brought back, they were ineffective.

2.1.6 Analysis & Use Cases

In this formative evaluation, we asked the question "given this capability, what would you do with it?" We group observations into discovered use cases, performers' suggested use cases and technical issues. Discovered use cases are those acted out during the workshop. Suggested use cases would require coordination or technology not possible during the workshop, but performers requested it enthusiastically. Technical issues are problems related to the system set-up that could possibly be solved in later work.

Half-Dialogues as Template Scenes

Performers were excited by the idea of re-using a video in different contexts. One performer, Sean, said he was going to go onto one side of the stage, and record a scene of him speaking to the other (currently empty) side of the stage. Sean instructed other performers to plug their ears and face away. After he recorded this scene, each of the present performers took a turn acting opposite his video. He intentionally left timed gaps in his performance, which he knew the other performers would fill in with their dialogue. We called this a *half-dialogue*, which acts as a template against which others can perform multiple times. His scene is as follows, where gaps between lines indicate time for implied responses:

[Sean saunters in across the stage]

Sean: You know why I brought you here.

Sean: I expect more from you.

Sean: So tell me one thing you're going to do so that this never happens again.

Sean: That's why you're my favourite.

Sean: Tell me what I want to hear.

Sean: Thanks.

[Sean exits]

Figure 2.3: Three performers improvising opposite the same template scene.

Sean intentionally left his semantics a little vague, so that the other performers could take it in several different directions. Once he finished, the other performers were allowed to uncover their ears and face towards the stage again. He described where he walked, stood, and faced on stage, so that the other performers could start their scene smoothly. Below are two examples of scenes performed opposite the recorded Sean (Figure 2.3). Note that these performers are spontaneously responding to seeing Sean's half-dialogue for the first time.

Chris vs. Sean

[Sean saunters in across the stage]

Sean: You know why I brought you here.

Chris: Damn right I did

Sean: I expect more from you.

Chris: Well I didn't have much to offer

Sean: So tell me one thing you're going to do so that this never happens again.

Chris: I'm going to apologize and then forget about it.

Sean: That's why you're my favourite.

Chris: Thanks

[Chris notices Sean pointing to himself]

Chris: Yeah, I'll kiss you.

Sean: Tell me what I want to hear.

Chris: I'll kiss you!

[Chris kisses Sean]

Sean: Thanks.

[Sean exits]

Deejay vs. Sean

[Sean saunters in across the stage]

Sean: You know why I brought you here.

Deejay: I absolutely do, yes.

Sean: I expect more from you.

Deejay: You really shouldn't, no. I ... [Sean continues, appearing to interrupt Deejay]

Sean: So tell me one thing you're going to do so that this never happens again.

Deejay: Hey! I'm talking, why are not listening to me?

Sean: That's why you're my favourite.

Deejay: [Deejay huffs] You're useless

Sean: Tell me what I want to hear.

Deejay: You're useless!

Sean: Thanks.

[Sean exits]

We later learned that the element of surprise on the performers' part was not necessary — in fact, seeing the scene once before acting opposite helped performers anticipate when the half-dialogue would start again. Features that were accidental or seemed irrelevant at the time of recording Sean's first half dialogue, such as pointing at himself, were given relevancy by the performers as they responded to them naturally, with their current mental context as they were performing. Another event that appears to fit this model is when Sean speaks over Deejay, and then Deejay responds negatively. I would argue that this is undesirable, as then every video that appears to not respond to performers gains a trait of hostility, and I want to avoid characterizing the system as much as possible.

Looping Response Speed Challenge

One performer recorded a short line of dialogue, and requested it be looped repeatedly, with the challenge that the improvisor opposite would have to try to sustain a reasonably lucid scene as long as possible.

To give a little background, this is similar to many improvisation games, where the performance is made difficult by some rules or constraints. One such game is *Questions Only*, where the performers

in the scene are only allowed to speak in questions, and performers are eliminated as soon as they say something that isn't a question or repeat themselves. These are hard rules, but there are also emergent "soft rules", which don't indicate a loss so much as behaviour to avoid since the audience frowns upon it — these are changing the topic drastically by asking a non-sequitur question, or asking about the nature of the game itself. Another such game is *Number of Words*, where all performers are given a specific number of words, and can only speak in lines of dialogue that have that number of words. For example, performer A gets 3, performer B gets 1, and Performer C gets 6. Players are eliminated when they violate the rules similarly.

For our *Looping Response* use case, the challenge is to have a live performer respond to the same motion and line of dialogue continually. There are no hard rules, as in the game examples given above, but the emergent soft rule is to keep the dialogue interesting, and to avoid going meta by questioning why the recorded performer is repeating themselves.

In the example case, the repeated action was:

> [Ryan, hunching his back, his hands claw-like, nearly clasped together]
>
> Ryan: Kill him, he's been holding you back! Kill him now!

Figure 2.4: Looping Response: Oliver and Ryan

One performer, Oliver, did the following scene with this recording (Figure 2.4):

> *Ryan: Kill him, he's been holding you back! Kill him now!*
>
> Oliver: Okay, I can kill him, but what about my Dad?
>
> *Ryan: Kill him, he's been holding you back! Kill him now!*
>
> Oliver: I'm just going kill everyone? What about my cousin?

Ryan: Kill him, he's been holding you back! Kill him now!

Oliver: Well what about my Mom?

Ryan: Kill him, he's been holding you back! Kill him now!

Oliver: He's a him? She's a him?

Ryan: Kill him, he's been holding you back! Kill him now!

Oliver: But I came out of her womb!

Ryan: Kill him, he's been holding you back! Kill him now!

Oliver: Her womb!? I love that thing.

Ryan: Kill him, he's been holding you back! Kill him now!

Oliver: At least I get to keep my teddy bear.

Ryan: Kill him, he's been holding you back! Kill him now!

Oliver: [Whimpering] My headmaster?

Ryan: Kill him, he's been holding you back! Kill him now!

Oliver: [Angrily yelling] I got it! [exits]

The looping response in an enjoyable challenge, as it forces performers to come up with a workable response at a regular timed pace, and does not relent. Performers noted this was enjoyable as an acting exercise. While a live performer repeating the same line may slow down to give the responding performer extra time, the timing with a video looping response does not. Unlike the half-dialogue template, timing is less of a problem, as with repeated iterations, the live performer gets used to the sense of how much timing they have.

Broken Telephone Chain

The name of this use case is inspired by the childhood game *Broken Telephone*, where a circle of people whisper the same message ear-to-ear from one person to the next around a circle. The goal is to have the same message arrive at the end as they started with, but in practice, with very quiet whispering, some of the message is lost. Often players will exacerbate this effect on purpose.

This use case was inspired by the implemented clap gesture to bring back the immediately previous scene and play it through once. In the following, each scene is recorded, then is brought back as recorded by the next live performer. The next live performer after them brings back the previous scene. Since we masked out the projected area brought back with the depth shadow, the stage only ever showed the immediately previous scene. We avoided dialogue, as we did not have an audio filtering system in place. Regular font represents the live performer and italics represent the recording.

[Liz faces the centre of the stage, and bows aggressively]

—

[Liz faces the centre of the stage, and bows aggressively]

[Liz bows aggressively back at her recording, reeling in pain when they knock heads.]

—

[Dustin extends his hand forwards, palm out towards Liz]

[Liz bows aggressively, reeling in pain after hitting Dustin's hand.]

[Dustin flexes both arms triumphantly]

—

[Dustin extends his hand forwards, palm out]

[Deejay starts running towards Dustin, head downwards, but stops moving forwards when his head is held back by Dustin's hand. He continues running in place, growling in frustration.]

[Dustin flexes both arms triumphantly]

[Deejay falls over as Dustin pulls his arms away.]

—

[Deejay starts running in place, head downwards, towards the other side of the stage, growling]

[Dustin waves an imaginary matador cape at Deejay, as if he is a bull.]

[After some seconds, Dustin pokes Deejay.]

[Deejay falls over dramatically.]

This use case also demonstrates the ability to reinterpret parts of a performance in surprising ways. The other performers laughed in surprise at many of the re-uses of the recorded video. Unlike the Half-Dialogue use case, each performance was not intended for re-use, but instead was focused on responding to the immediately previous performance. Despite this, the performances were highly reusable.

Chants and Crowds

There were two use cases that utilized layering of many videos simultaneously.

We had one improvisor sing with himself the round song *Row, Row, Row Your Boat*, with precise timing on the part of the operator. Fortunately, the system supports playing scenes that are still in the process of recording. This indicated the need to maintain this feature going forward.

Another improvisor wanted to create a crowd scene with themselves (Figure 2.5). They crossed the stage many times, with different walking styles. Once each walk was finished, he requested it be brought back and looped. With this, the performer was quickly able to create a sense of a crowded, public, noisy environment just using themselves.

Another case was what a performer called a "thematic collage" of a series of phrase, like several mantras, looped continually. Similar to a crowd scene, these are built up over time, and then looped continually. This resembles the group game opening of the longform improvisation format The Harold, where performers shout out themes or phrases expressing a point of view, building and reflecting on each other, until a common theme emerges. When we workshopped this, we ended up with the following lines, repeated and looping over each other:

Figure 2.5: Crowd from a single person.

> I work for a living.
>
> Why's it always got to be about politics?
>
> If you don't like it, change it — it's up to you.

Similar to the themes throughout the workshop, an apparent theme can emerge from when different pieces of performance are played next to each other, whether intentional or not. One performer expressed his thoughts as:

> For me when you repeat ideas, it's neat to repeat ideas next to each other that weren't next to each other before to see what the new implied meaning is and this is a way to kind of, you know, get that to happen automatically, like perfect repetition, to see what your new juxtaposition means.

2.1.7 Suggested Use Cases

Scene Mashup

One improvisor observed that if you recorded several scene templates where the gaps in the dialogue were the same lengths, you could play any two scenes together. From n half-dialogues on one side of

the stage, and m half-dialogues on the other side of the stage, you could create a show with a total of $m \times n$ scenes. If you are able to flip and position the video so that a half-dialogue could face either side of the stage, you could create $(m+n)^2$ scenes.

During the workshop, we tried recording scenes with similar-length gaps in dialogue, but found that they frequently become out of sync, and the resultant combined scene seemed more like noise rather than anything with semantic values. A solution suggested is that performers are cued, by a visual feedback monitor for example, to speak or not speak at certain times. However, this may be hard to follow.

Additionally, we discussed taking a recording of a scene between two performers and splitting it up. Let's say we have a dialogue between two performers A and B:

A B A B A B A

If we could separate these two dialogues through processing,

```
  B     B     B
A   A     A     A
```

we could use either of these parts of these scenes again in a new context, say with a live performer, C, acting opposite recorded performer A.

A C A C A C A

Let's have an example:

> A: Nice to see you! It's been so long.
>
> B: Well, I wasn't happy after our last conversation.
>
> A: Let's forget that, I got us some cake!
>
> B: No thank you, I expect an apology from you.
>
> A: Oh, look, I also got us some wine.
>
> B: This is what you always do! You pretend everything's fine when you're just going to be an ass again!
>
> A: [Heavy sigh] If you aren't going to accept my hospitality, I just, I just feel like I'm a failure as a host
>
> B: [Wince, pause] You're...not a failure.

Here's a combination of A's same dialogue with a new character, C:

> A: Nice to see you! It's been so long.
>
> C: Heh, you know. Rehab makes you a little antisocial.
>
> A: Let's forget that, I got us some cake!
>
> C: Whoah, haha. I'm just on veggies and a little meat these days. I'm not even supposed to get processed sugar.

A: Oh, look, I also got us some wine.

C: [Bites lip] Uh...I guess I could have some. I really shouldn't though. I just got out.

A: [Heavy sigh] If you aren't going to accept my hospitality, I just, I just feel like I'm a failure as a host

C: Uh, look, uh, I'm sure the wine and cake are great, and I appreciate you throwing this party for me, but it's really not the best idea. Let's, I don't know, go for a walk in the woods or something.

The thought of taking C's half-dialogue and re-contextualizing it is also interesting, ad nauseam.

The above examples utilizing half-dialogues illustrate that we not only could benefit from a method for easily retrieving videos from a library, we also could benefit from way to process and combine video that is aware of the performers in it, including features of their performance.

Unknown Gestures Triggering Random Replay

One improvisor suggested that gestures unknown to the live performer could trigger video from the pre-recorded scene. The performer would have to react to this semi-random behaviour of the system. The idea presented is that the performer would eventually figure out what gesture or behaviour triggered what video, and as the scene progressed the live performer would go from having very little control, to having much more control over the scene.

This use case is similar to improv games with hidden rules where one performer has to guess the rules to "win" the game. One such example is *Party Quirks*, where one improvisor is hosting a party, and all the guests have a secret quirk. The goal of the host is to guess each quirk. Examples of quirks are:

- is slowly turning into a goat
- is a desperate vacuum salesman
- has a condition where food makes them drunk.

From the host's perspective, they will observe eccentric behaviour and try to figure out a pattern from it. The goal of such improv games is never to "win" as quickly as possible, but to be entertaining. As such, once the host understands the secret rules of the performer, they will play with them for a bit before formally guessing what the secret behaviour is. For example, once the host understands that one of the guests desperately wants to sell them a vacuum, they may point out dirty spots on the floor, but then insist that "it isn't a big deal" or suggest that they already have other coping mechanisms, such as moving furniture over the dirty spot.

Intelligent Playback Performers

As discussion built towards the end of the session, some performers suggested that I create an "intelligent playback performer". This seemed like a ridiculous end goal for this project, but it was interesting to listen to their requests. They hoped for very far-fetched features, such as artificial intelligence or

understanding and processing speech and responding appropriately. I found it interesting, as the system we were using was very "dead", in the sense that it was not intelligently responsive, just playing back video. However, it felt like we were actively engaged in sense-making, and the juxtaposition of two videos would more often than not feel intentional, often comically. I think there is much more to be accomplished without requiring any sort of "intelligence". It seems the system itself is more of a Chinese Room — offering semantic tokens that we interpret as fitting together nicely, without any "understanding" itself of their underlying meaning.

Beyond a regular performance, the performers suggested a few uses for the intelligent playback performer. One was having someone to practice a script with, especially when you are memorizing lines. Another was to practice acting out against or confronting someone, for therapeutic purposes.

2.1.8 Discussion

We have found that the core idea of a system for remixing stage video is interesting, though there are many problems that need solving. The performers identified a diversity of exciting use cases, as well as a few desired use cases that should be possible with with supporting interactions and software.

There are a few high-level thoughts and concerns:

Often I and performers would compare this system to a musician's loop pedal, where a musician can record, start and stop loops of their own music, building from a simple melody to the semblance of a complex orchestra. However, the content in our system is different, being linguistic, and either disconnected like a chant, or narrative like a scene. Once participant noted:

> I'm not sure if watching someone build a video/physical scene is interesting in the same way that watching a musician build a riff with a loop pedal is interesting.

I feel like performers can learn how to make the build-up interesting to an audience - it would seem to be a failure of the system to have the first 30% [2] of the show, the build up, to be boring while the remainder is actually interesting. It is unclear at this stage what fraction of the final show will be live performers and what fraction will be recorded performers. Is it mostly live performers with a smattering of recorded performers, like a flashback? Or, are the live performers like a seed at the beginning, and after that it is mostly recordings? To have any generalizability, I need to consider both cases.

It was very interesting that scenes played back against each other, as a juxtaposition, seemed to either create (or, depending on your perspective) unearth new meaning. We quickly discovered that, by juxtaposing any two random videos, we as audience members would find meaning and comedy in any little coincidences we'd find between them. Even a video played back by itself, immediately after a performer made it, would seem to have a different meaning.

This feeling that something familiar has taken on new meaning is not new. This can appear both with *synthesis*, by putting together two separate things, but also by *isolation*, by removing a part of something. Synthesis is certainly well known in mashup art, but isolation is not as commonly used. A good example of isolation is *Garfield Minus Garfield*, where an artist takes comic strips of *Garfield* and removes

[2] http://knowyourmeme.com/memes/the-wadsworth-constant

Figure 2.6: A comic from *Garfield minus Garfield* [Walsh].

the main character, yielding a surreal reading of a man in his own apartment, speaking to himself (Figure 2.6).

Finally, working with performers was an interesting and revealing experience. Most of my HCI work thus far had been typical user studies, and in one case an interview study on the participants' usage of notes. I had had past experience with fixing bugs in software that I was working on for someone else, but it was very bewildering to work with performers. They would alternate between making requests that they believed were very hard, but were trivial, and requests that were nearly impossible, but they believed were easy. Also, it was difficult to tell the difference between a request for something that the participants felt absolutely should be implemented, and a cool idea with no necessary pressure to implement. When a bug was encountered in the system, performers rarely treated it as the system doing something wrong, but instead treated it with all seriousness, playing with it. During these times, I became very apologetic, promising that the bug that they did not even know they had experienced was possible to fix, as if I needed to reassure them of my own competency. A more suitable approach would have been to watch and listen. Clearly performers are a special case of user, something which deserves further comment.

2.1.9 Conclusion

I have described a workshop I ran with a prototype of Improv Remix software, allowing a stage area to be remixed with recently-recorded video. We found the system to have interesting possibilities, but also many limitations. I identified several interesting use cases, as well as use cases that required some further technical work to be possible. I identified some core technical issues going forward.

2.2 Experiment: Actor DJ

Given my observations during the Workshop, I wanted to explore ways to manipulate and remix scenes. I describe this as an experiment, since it documents some prototyping I did, with the understanding that this was to determine if such an approach was feasible technologically, and whether the results were entertaining.

2.2.1 Introduction

My core objective is, given a corpus of source video, to have a system to facilitate creating the appearance of new semantic content from that video. As the objective is theatrical, not photo-realistic, we are not interested in fooling the audience into thinking the resultant video they are seeing was recorded literally as it is displayed to them. Instead, the design priority is to make the video appear to construct a reasonable, or at least entertaining, narrative through verbalization and action.

From our workshop we found that combining different source videos, or source videos with a live performer, could yield enjoyable results. The sensation was that new meaning had been created that was not found in the original sequences at the time of recording. However, the sequences became unwatchable when:

- Recorded video interrupted the other performer
- Long gaps of no content were left in constructed videos
- The video made a reference to something meaningless to the opposite performer
- The video ended suddenly, before the opposite performer was done.

To characterize the video input to our system, we have 30 frames/second colour (RGB) video from the front of performers, then 30 frames/second depth (D) video from behind performers. With careful calibration, we treat the colour and depth frames as representing the same space. Additionally, we have an audio stream from our Kinect, which also provides audio direction, meaning that we can get a somewhat reliable indication of who is speaking on the stage.

To characterize the type of performance we are expecting a little further, we assume performance consists of 1-2 people on the stage at a time. To allow for audio processing, these performers will undergo training with the system so that they learn not to interrupt each other. Additionally, we hope that audience noise will be minimal; perhaps we must notify the audience at the beginning of the show.

Coming out of what we learned from previous work in this project, and our capabilities with our recording system, our objectives for this part of the work are to create a system for RGB-D Video + Audio of Performance, to partition it into pieces that are meaningful for recombination into other narrative pieces.

In this section, we shall first review desired use cases, then discuss the nature of recombining semantic tokens, and discuss the currently-implemented Actor DJ system. We close with a continued work plan to be completed after this research proposal, then describe an example of expected use, followed by a discussion and conclusion.

2.2.2 Desired Use Cases

We found three suggested use cases from the workshop, and add one more. The table below lists these use cases by how they are controlled.

Suggested Use Case	Control Method
Scene Mashup	No live control; after being edited together, resulting mashup plays back.
Unknown Gestures Triggering Random Replay	Performer on stage; they are unaware of what gestures control the scene but slowly discover them.
Intelligent Playback Performers	System logic determines how the performer behaves.
Puppet Performers	Control by an offstage or hidden performer.

All of these require some understanding of the scene content. To mash together two separate scenes, one must get a sense of the scenes so that players are not talking over each other. When the performers in the workshop asked for "replay from random parts", they certainly did not imagine replay starting in the middle of a line or a word. The desired "Intelligent Playback Performer" is a further step that must incorporate some degree of minimal logic on the system's part.

2.2.3 Recombining Semantic Tokens

This section is devoted to investigating the nature of recombining pieces of scenes. While we reviewed re-using a half-dialogue in the workshop chapter, we shall go a little deeper here and look at taking scenes further away from their original context and ordering. We shall cover a few motivating examples to ground the rest of the work in this chapter.

Let us start with an example of re-using a line with the same tone in different contexts:

> A dialogue where A and B are construction workers. A is a worker who is not performing up to standard, and B is their boss. A keeps listing things going wrong, and B appears to be listening patiently. The scene ends with B saying "I just can't deal with this anymore - I'm disappointed and frustrated".
>
> A dialogue where A and B are on a date. It is going extremely well. A and B are surprised at how into each other they are, laughing at each other's jokes. At a lull in the conversation, B's expression changes, and he says the same line above, with the same tone "I just can't deal with this anymore - I'm disappointed and frustrated".

In modern improvisation, if the above two scenes occurred one after the other, this would cue a heightening series of scenes, taking the same line and tone and putting it in even more extreme contexts. We could expect the following to appear next:

> B wins a cruise, and is forced into a comfortable chair, and waited on by several servants, foisting increasingly expensive drinks, food, and entertainment on her. Exasperated, B waves all the attendants away, yelling "I just can't deal with this anymore - I'm disappointed and frustrated".

The above is very common in modern improvisation, and the increasing absurdity of the cases where the line appears prompts increasingly uproarious laughter (this is unlikely here because I'm explaining the joke to death).

It is common in theatrical practice to treat the script of a play, coming from a writer, as constant, where a series of performance by a single theatre company represents an interpretation of that script. I am going to compare this to filmmaking, as that process is better-known, even though it was developed much later. In film-making, usually the mapping from script to movie is one-to-one. It is rare to have the same movie made multiple times from the same script[3]. This is separate from a remake of a film, where it is common for the script to be very different. In theatre, script-writing is most often a separate process from play-making. A rough computational metaphor we may use is the instantiation of a class in an object-oriented system. A script is a class recipe, and each mounting of it is a different instantiation.

The diversity of different mountings of a given script is highly celebrated in theatre and that a local community theatre company can mount a script with their own, locally relevant, interpretation, is treated as a strength of theatre over other, more widely-disseminated forms. It is a fundamental part of theatre acting and directing practice to "find the meaning" in a script. While modern scripts have acting instructions ("stage directions") outside the text meant to be spoken, it is notable that Shakespearean scripts have almost none, leaving the mechanics of what is to happen, other than what is spoken, entirely up to the performers and directors.

Let us look at an example of taking a relatively neutral dialogue and how it may be emphasized for different interpretations.

[3]The remake of Alfred Hitchcock's Psycho (1960) by Gus van Sant in 1998 is one exception.

Characters	Neutral	Angry Boss	Disrespectful Employee	Secret Lovers
Employee:	Can I come in?	[Peeks into office] Can I come in?	Can I come in? [Enters without waiting for a response]	[Seductively] Can I come in?
Boss:	Yes. How do you think your presentation went?	[Impatient] Yes. [Disappointed] How do you think your presentation went?	[Pauses] Yes. [Stammering] How do you think your presentation went?	[Grins widely] Yes. [Speaking so the rest of the office can hear, then closing the door] How do you think your presentation went?
Employee:	Fine. How do you think it went?	[Questioning] Fine. [Dejected] How do you think it went?	[Bored] Fine. [Mocking] How do you think it went?	[Grabbing the boss by the collar] Fine. [Flirting] How do you think it went?
Boss:	Fine.	[Dismissive] Fine. [Boss struts out of the office.]	[Apologetic] Fine.	Fine. [They kiss.]

By changing their tone and action, the Boss and Employee performers can significantly change the meaning of the scene. However, I argue that to reinterpret a scene, you do not even need to re-perform it with different tone. The context of a scene, or even an atomic piece, a semantic token itself, can affect its interpreted meaning. In film, this was discovered by Lev Kuleshov, who observed that the audience will interpret short shots in a film differently based on the preceding or following shots [Kulešov, 1974]. Let us look at a few examples of this effect relevant to our work.

Let's take the following semantic token:

> [confused, distressed]
> I don't know what you mean.

This could be said in response, identically, to the following questions, in all cases saying something very different about the replier:

> What's the inverse square root function graphed onto a Klein Bottle?

Where is your husband?

What is your name?

What is love?

We do not need to use speech to create a semantic token, a simple shrug is sufficient:

How do you feel about the breakup?

[shrug]

What are you going to do next?

[shrug]

I found out you won the lottery!

[shrug]

Do you still have that itch on your shoulder?

[shrug]

I have shown above some minor cases of changing emphasis of semantic tokens between performances, or changing the context of individual tokens. However, once we have a system where we can rearrange and re-contextualize semantic tokens, we can rearrange entire dialogues for different meanings. The following example is the *Crab Canon* from Douglas Hofstadter's *Gödel Escher Bach*, wherein he is noted for saying "meaningless symbols acquire meaning despite themselves" [Hofstadter, 1979]. The dialogue is palindromic, the same backwards and forwards. We present the same section from the beginning, and the end, omitting the middle:

Tortoise: So nice to run into you.

Achilles: That echoes my thoughts.

Tortoise: And it's a perfect day for a walk. I think I'll be walking home soon.

Achilles: Oh, really? I guess there's nothing better for you than walking.

Tortoise: Incidentally, you're looking in fine fettle these days, I must say.

Achilles: Thank you very much.

Tortoise: Not at all. Here, care for one of my cigars?

...

Achilles: Not at all. Here, care for one of my cigars?

Tortoise: Thank you very much.

Achilles: Incidentally, you're looking in fine fettle these days, I must say.

Tortoise: Oh, really? I guess there's nothing better for you than walking.

Achilles: And it's a perfect day for a walk. I think I'll be walking home soon.

Tortoise: That echoes my thoughts.

Achilles: So nice to run into you.

In this example, the speakers are switched, but the point is that the same lines can be used to create different scenes is made.

2.2.4 Processing & Remixing a Performance

The suggested use cases all encompassed the same need: to be able to play a portion of a performance at a specific time. However, if you play a 1-second chunk of video, then return to black, the stage is empty, a jarring effect for the audience. An alternative is to pause the video chunk while it is waiting to be played, and then pause it again at the end, but video of a character frozen in place on stage is also jarring. In this section, we solve this problem by instantiating a playback performer on the stage. By default, the performer is in a state of quietly listening, unless they are cued to speak a specific semantic token.

We accomplish this in a mouse-controlled interface, by taking a scene and parsing it into neutral and non-neutral sequences. Non-neutral sequences of sufficient length are labelled as utterances, in line with speech processing literature [Clark, 1996], which may be played by clicking on the corresponding button on the UI.

Parsing a Video into Utterances

Given an input scene from a performer, defined as the time they enter the stage until they leave it, we seek to parse this scene into utterances that could be meaningful for replay, as well as find a suitable neutral, non-speaking section of video to use for appearance of "listening".

Our scene parsing consisted of two passes:

1. Label frames as neutral or non-neutral
2. Merge regions of non-neutral frames into utterances

We defined frames where something was happened, that could be used for an utterance, as "non-neutral", while frames where nothing was happening, as "neutral". There are many different features that may be used to partition a scene into "neutral" and "non-neutral" periods of time (Figure 2.7). For simplicity, we focused on movement and sound. For movement, we set a threshold based on the number of changed pixels in the binary silhouette user image of the Kinect. For sound, we set a threshold based on the average amplitude in 100 ms windows. After we marked every frame in a scene as either "neutral" or "non-neutral", we grew each "non-neutral" region by 0.5 seconds backwards and forwards. Each non-neutral region was declared an utterance if it was longer than 2 seconds. The longest neutral region was declared as the "listening" utterance. This approach is somewhat similar to the algorithm used in DemoCut [Chi et al., 2013].

We validated this method and tuned the thresholds and times based on the corpus of scenes collected from the workshops, so that from each scene we would get sub-scenes with meaningful utterances.

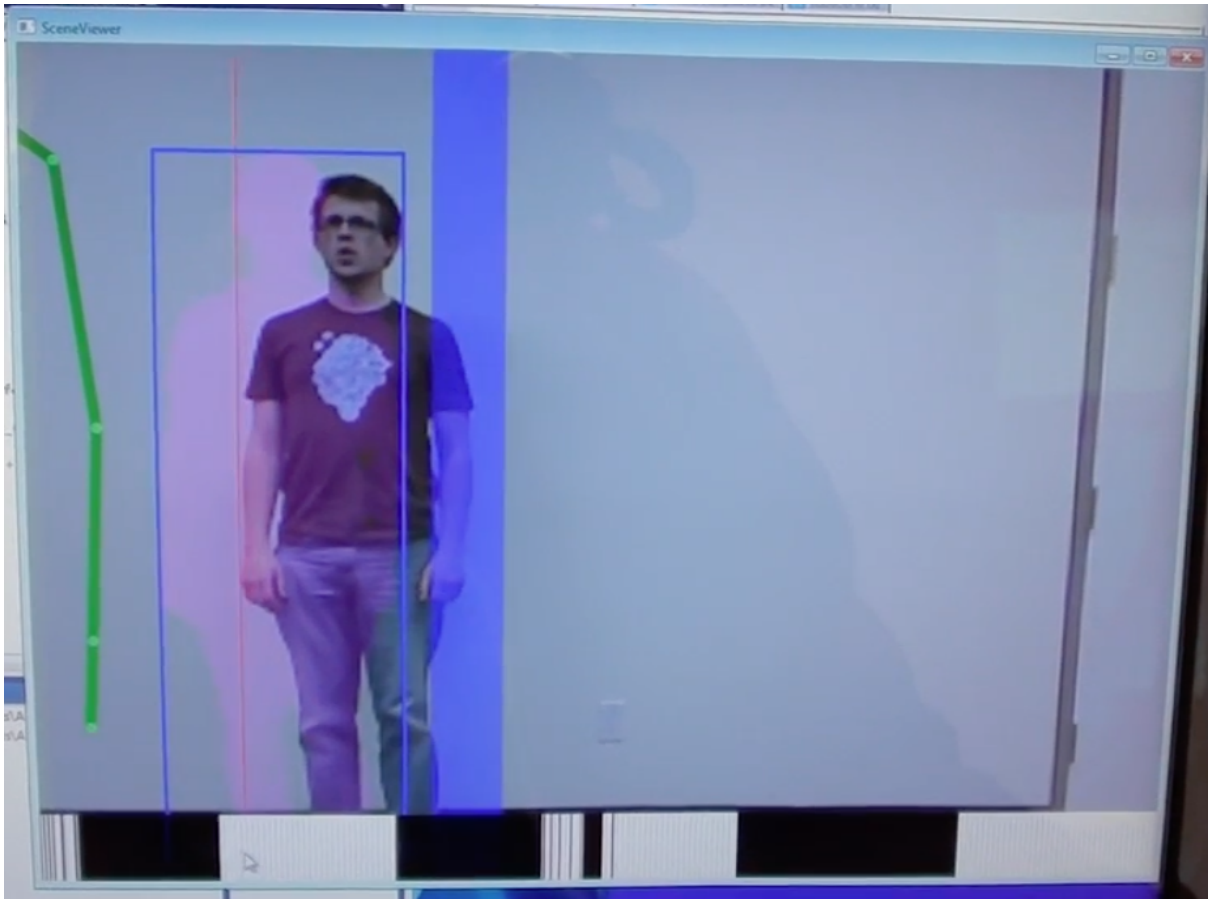

Figure 2.7: Parsing a scene into neutral and non-neutral segments, as indicated by white and black lines along the bottom of the interface. The labelling of frames is shown before any merging pass.

Playback Behaviour

Here, I shall demonstrate playback behaviour of a playback performer. First, Figure 2.8 shows regular playback of a pre-recorded scene. In the scene, the performer says "Yes!", "No!", "Maybe!" in that order. In playback this is identical. I use this Figure, despite it being trivial, because it establishes the notation I use to describe a playback performer.

Figure 2.9 shows playback of a playback performer, where the source is the same captured scene as in Figure 2.8. Initially, the playback performer loops the longest neutral part of the scene, to appear to be listening. Next, the operator of the playback performer triggers the "Maybe!" utterance. When it finishes, the playback performer returns to looping the listening sub-scene. Next, the operator of the playback performer triggers the "No!", then the "Yes!" utterance.

Performer Control Interface

The interface consists of a live view of the stage space, with a thumbnail view of instantiable scenes along the right-hand side (Figure 2.10). Clicking a thumbnail puts the playback performer on the stage, physically projecting them on the stage space. In the UI, a box appears around the playback performer's known position, with buttons alongside.

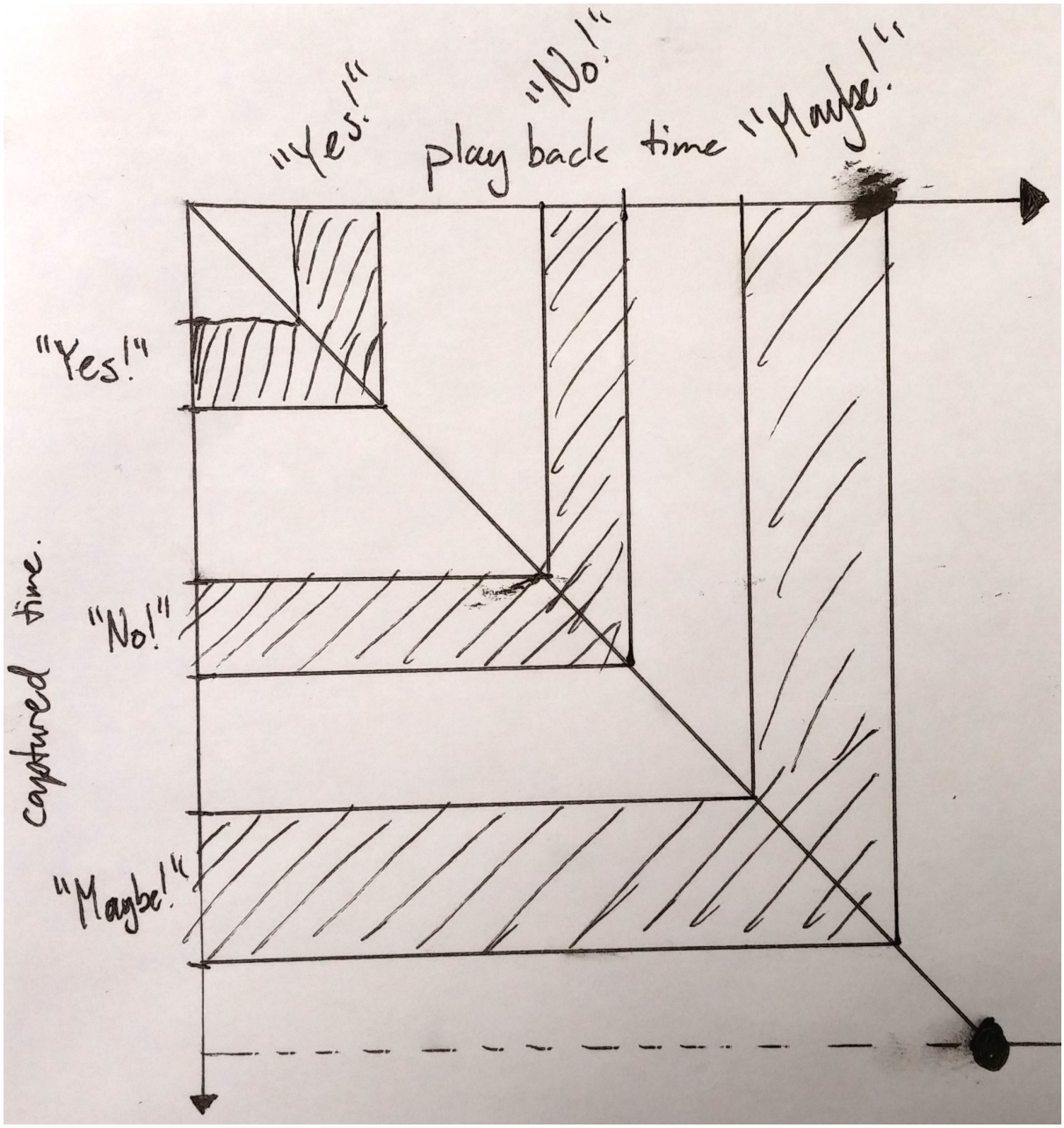

Figure 2.8: A diagram of normal playback of a capture scene. The vertical axis is the captured time; whereas the horizontal axis is the playback time. The timing of the utterance is identical in the playback as in the recording.

Flip - Horizontally flips the playback performer, so the operator can change the direction they are facing.
Play - Toggles the pause state of the playback performer.
Close - Removes the playback performer from the stage.

The remaining numbered buttons are each of the parsed utterances. Clicking a button immediately starts playing that utterance. "0" denotes the listening utterance, so the operator can click that button to stop the current button and return to the playback performer's default listening state.

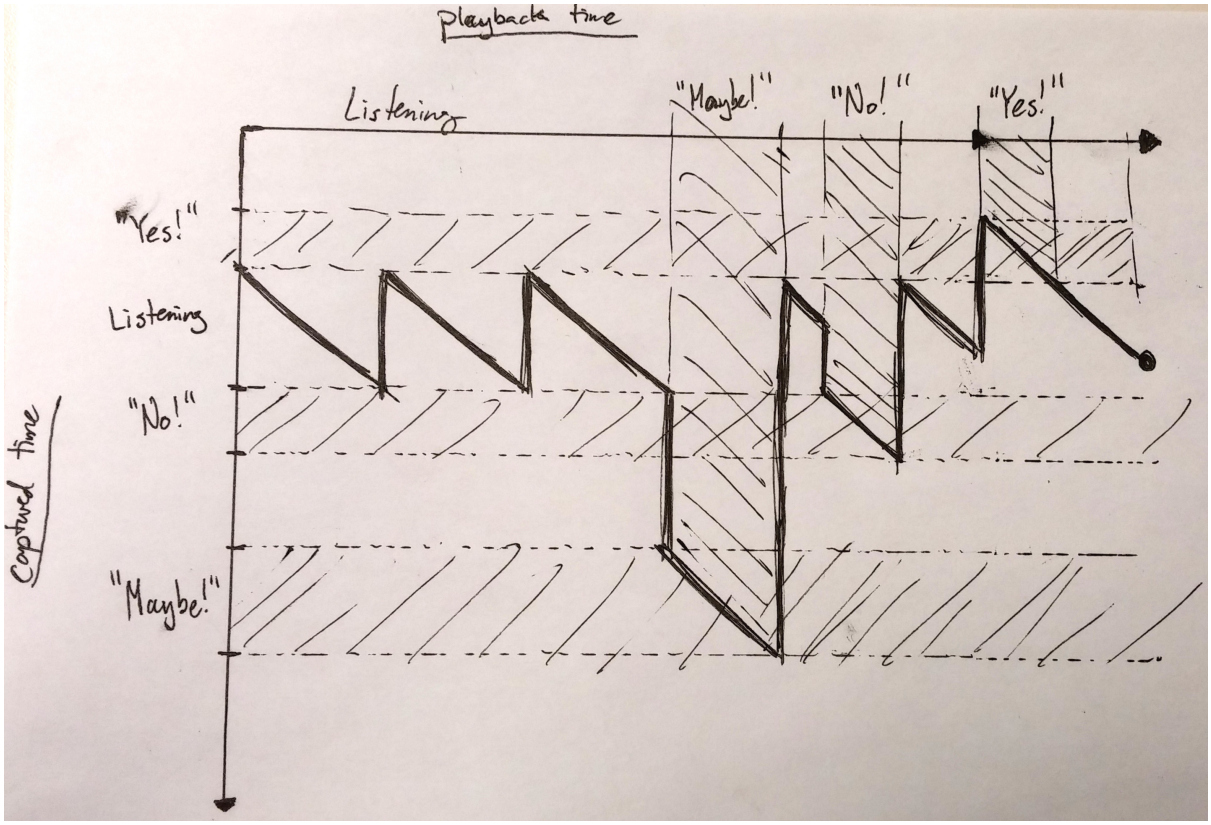

Figure 2.9: A diagram of playback of a playback performer. The playback performer loops the listening segment by default. Then, the utterances "Maybe!", "No!", and "Yes!" are triggered, with short times in between, where the playback performer returns to the listening animation.

Playback Example

In this example, we are using the following half-dialogue as source:

> You know why I brought you here.
>
> I expect more from you.
>
> So tell me one thing you're going to do so that this never happens again.
>
> That's why you're my favourite.
>
> Tell me what I want to hear.
>
> Thanks.

After instantiating one playback performer on the stage, we can instantiate another, and flip it. The operator can trigger different utterances, to give the appearance that the performer is talking to himself.

> *Left:* I expect more from you.
>
> *Right:* Thanks!
>
> *Left:* Tell me what I want to hear.
>
> *Right:* Thanks!

Figure 2.10: Actor DJ, showing a live performer (left) with a playback performer (right). Live performers are instantiated from stored scenes along the right side of the UI. The numbered buttons along the side are sub-utterances that may be played by clicking.

Figure 2.11: Actor DJ, showing the same playback performer cloned on either side of the stage. The operator can use the interface to make the performer appear to talk to himself. In this shot, the playback performer on the left is listening, while the playback performer on the right is in the midst of a gestural utterance.

Left: I expect more from you.

Right: Tell me what I want to hear.

Right: That's why you're my favourite.

Left: Thanks!

2.2.5 Discussion and Future Work

As it stands, the work in this chapter represents a pair of prototypes to validate the approach, which motivates further work in this thesis. To close, I shall cover some perspectives on the realism of the playback performer's behaviour, as well as allowances for creating playback performers when the source video contains multiple performers.

Realism

It is very unclear, to the observer, that the playback performer is not a live, unaltered video feed, but rather video feeds that are unapologetically stitched together. In the examples we have chosen, the position of the performers on the stage does not change significantly, so the difference between different segments of video is not jarring. We could probably be clever, such as trying to adjust for performer position, or we could make the looping of the listening segment more seamless by cleverly choosing start and end frames based on similarity. However, as it stands the system is theatrically fully effective, and manages to avoid landing in the Uncanny Valley.

Multiple Performers

While the system assumes only a single performer, we did some preparatory work for parsing two-person scenes. If performers are well-behaved, and stay on their respective sides of the stage, we can determine who is speaking with the Kinect directional microphone. To parse a two-person scene into two, separate, playback performers, performers in frames would be labelled as neutral or non-neutral based on their individual behaviour. Special care would have to be taken so that a period of time where both performers are speaking is not parsed into an utterance for one person. Part of this could be accomplished by performer training.

3
Background

3.1 Overview

Janet Murray said in her book *Hamlet on the Holodeck* that many of the new theatre techniques appear to be somehow technologically inspired. Even though they may be nothing more than humans wearing normal clothes, they are "holodeck experiences without the machinery." [Murray, 1997, p. 43]. In this literature review, we treat technology as a means to an end. Alternatively, many of the artistic works in this review treat technology as a magical idol that they seek to make a commentary upon. We are primarily interested in new techniques for expression, which, of course, includes new technology. It often seems the only difference between a "new technique" and "technology" is that the details of how a technology "works" is not likely understood by the average audience member.

This work specifically examines improvised theatre, and the potential for the usage of technology in it. However, as improv theatre is relatively young and poorly-documented, we will contextualize our discussion in the larger context of all theatre, the vast bulk of which is scripted.

It is the suggestion of the author that the addition of stagecraft (technology) to theatre often reduces the ability of theatre to be spontaneous and respond to the audience. In stagecraft, the field pertaining to the technical aspects of theatre production, the goal is to accurately implement the director or designer's artistic vision. This necessitates that there is an *a priori* vision of the performance, and that reactive technical aspects are a noisy hindrance to implementing that vision. By contrast, the original improvised *commedia dell'arte* productions in the 16th through 18th centuries were travelling troupes that could adapt to any stage they could find [Brockett and Hildy, 1998].

The history of theatre from Wagner's Ring Cycle in the mid-1800s to the mid-1900s reads like a series of obsessively escalating attempts in shock the audience with audiovisual spectacle. The Italian Futurist movement is particularly guilty, demanding worship of the new technology [Kirby and Kirby, 1971, Salter, 2010]. Wagner and Gropius' thinking on the work of theatre escalated towards the idea of "total theatre", combining all elements of experience, including the architecture of the theatre itself [Dixon, 2007, Salter, 2010]. I feel like this gets away from the wonderful live, spontaneousness of theatre. Once we move past the eager fetishization of technology, I feel like the technology should disappear into the background. The technology should not be a part of the story being told, just a way of telling it.

Jerry Grotowski wrote a series of essays in 1967 titled *Towards a Poor Theatre* [Grotowski et al., 1967]. He argues that artists should seek to eliminate superfluous elements of theatre, so that it reduces to "the actor-spectator relationship of perceptual, direct, 'live' communion". Grotowski desires a "poverty" in theatre. Since theatre cannot compete with the spectacle of film, it should not, and should renounce all "outward" technique, strip away everything to reveal the essence of the performer. Grotowski distilled theatre as follows: "theatre is found neither in the narration of an event, nor in the discussion of a hypothesis with an audience, nor in the representation of life as it appears from outside, nor even in a vision — ... the theatre is an act carried out *here and now* in the actors' organisms, in front of other men..." [Grotowski et al., 1967, p. 118]. Grotowski referred to theatre containing extra, "synthetic" elements as "rich theatre".

Can our theatre be digital, but still "poor"? Phelan makes an emotional argument that theatre should stay ephemeral, poor, because that is what makes it unique [Phelan, 1993]. It is possible that we must simply wait until the audience has seen enough technology that they are not immediately dazzled by it. Elizabeth LeCompte, director of The Wooster Group, said that people in the 1960s were afraid of using a TV in the theatre, because everyone would want to look at it instead of the actors. For her, the television did not seem novel, as she grew up with one [LeCompte, 1987].

While our focus is theatre, there is much cross-pollination with installation art and other research work, and we reference those works as necessary. I will not attempt to draw a line between theatre, performance art, installation art and other research projects.

We present a review of related work in the following sections:

- Story-Making
- Modern Improvisation
- Capturing and Projecting Images
- Video Manipulation
- Whole-Body Interaction

3.2 Story-Making

As new technologies arise, so do new techniques inspired by them. While there is sometimes the fear that technology will replace an older form of art, in reality it creates something new that could not have

been expressed with the previous technology. In this section I describe story-making, a term I use in lieu of story-telling as the cases I am interested in do not have a pre-arranged script at the outset, but involve live collaboration between performers, and possibly the audience or technology.

While this section is called story-making, I am not too concerned with the output being a finished story, nor am I concerned with defining what a story is. The story being made is experienced live - we are not preparing a story for presentation at some later event, as in collaborative authoring. I use the term story to distinguish from other forms of narrative or plot that does not change over time - to be a story, ideas presented earlier must build towards a conclusion, even if it is only temporary.

In this section, I will cover *Sampling*, the building of narratives from smaller components, and *Managing Story-Making*, techniques for the live creation and experience of stories in the context I am concerned with. As I have special experience with the genre of Modern Theatrical Improvisation, I leave that for a later section.

3.2.1 Sampling

In her 1997 book on digital theatre, *Hamlet on the Holodeck*, Janet Murray determined four properties of computers as a medium for art [Murray, 1997]:

- procedural (mechanistic properties)
- participatory (interactive)
- encyclopaedic (large databases)
- spatial (ability to represent space)

I am most interested in the *encyclopaedic* property. As proposed in Vannevar Bush's MEMEX, a computer can create a hyper-linked model of previous scenes and references, similar to how our mind is structured [Bush, 1945]. Nyman wrote this on the musician John Cage: "Form thus becomes an assemblage, growth, an accumulation of things that have piled-up in the time-space of the piece. (Non- or omni-directional) succession is the ruling procedure as against the (directional) progression of other forms of post-Renaissance art music" [Nyman, 1999, p. 26]. This concept sounds similar to the concept of building up a "library" of previous videos during a live event, as in my proposal.

Prior to the digital revolution, several works in the French Dada movement experimented with collages and cut-ups. Hugo Ball's Cabaret Voltaire (1916) mixed pieces of performances, music and other content [Dixon, 2007]. Many of the film and live theatre examples shown in Dixon's *Digital Performance* emphasize the use of film excerpts or archives as moments of the past, re-assembled together to augment the meaning of the events onstage. Robert Edmond Jones was, in particular, interested in using projected film sequences to show the performers' inner thoughts, a "fusion of theatre and cinema" - Jones describes the potential interaction of live performers and a pre-filmed version of themselves as the "unembodied part" (inner emotions or thoughts) meeting the embodied part (the performer's physical body on stage), parallel representations of the same being [Dixon, 2007, p. 80].

A performance by digital theatre company Builders Association explicitly describes their performances as combining early-twentieth-century experiments with the "sampling" found in nightclubs' "drum and

bass" genre. A a system of MIDI triggers was activated by "onstage performers and offstage technicians to prompt video samples and sound loops in real time" [Dixon, 2007]. "XTRAVAGANZA samples fragments of the theatrical past through the language of contemporary DJ and VJ culture" [The Builders Association, 2002].

There can be a perceived significance to semantic tokens assembled in a random order that is just as valid as tokens assembled in a carefully-authored order. Zimmerman's randomly-ordered garden of Eden story *Creating a Meaning-Machine: The Deck of Stories Called Life in the Garden* is a deck of cards with one to several sentences on each. When shuffled and read in any order, they appear to tell a story [Zimmerman, 2007]. The *Kuleshov Effect* is an observation by Russian filmmaker Lev Kuleshov that audiences will interpret the expression on an actor's face differently depending on the preceding scenes - for example, a bowl of soup, a dead woman, or a little girl playing with a teddy bear. While these juxtapositions appear random, the audience actively creates a meaningful interpretation - they are actively making a story to explain what they observe [Kulešov, 1974].

Of interest in sampling from a source media is how it may be parsed into smaller components. McKee refers to a "beat" as the smallest unit of dramatic action [McKee, 1997]. Mataes and Stern created the interactive drama game Façade, where the player character can converse with a couple on the verge of a break-up. They speak of how their program included the notion of beats: "In dramatic writing, a beat tends to consist of just a few lines of dialogue that convey a single narrative action/reaction pair. A Facade beat is composed of anywhere from 10 to 100 joint dialogue behaviours (JDLs)...Only one beat is active at a time." [Mateas and Stern, 2003, Michael Mateas, 2007].

3.2.2 Managing Story-Making

There are three groups of people directly involved in a performance: the onstage performers, the offstage spectators, and those who aid in the production of the show — stage managers, lighting technicians and ushers. An important concept for this section is that there is a "magic circle" separating the people on the stage from the people off of it, and implicitly also separating the actions of the performers in the special space of the theatre from their behaviours in everyday life.

Augusto Boal, the creator of *Theatre of the Oppressed*, calls spectator "a bad word!" [Babbage, 2004], claiming that spectators should be able to reclaim the stage from the actors. Dixon identifies multiple levels of an audience interaction with a performance: Navigation, Participation, Conversation, Collaboration. However, he acknowledges that "play...a childlike fascination for the pleasure of cause and effect" is missing from this taxonomy [Dixon, 2007]. Barton argues that, in particular, intermedial performances should strive for *intimacy* between the audience and the performers, which is made more difficult by the inclusion of non-live elements [Bay-Cheng et al., 2010, p. 46].

Janet Murray discusses the fickle nature of allowing the audience to control some aspect of the live art: "Whether or not it is destructive to art, audience participation is also very awkward...When we enter the enchanted world as our actual selves, we risk draining it of its delicious otherness." [Murray, 1997, p. 101]. However, she clarifies that it can be helpful to define the role of the audience that is temporary interacting with the live performance: "[Despite our desire for the spectators to share an author-like role], there is a distinction between playing a creative role in an authored environment and having

authorship of the environment itself...This is not authorship, but agency." [Murray, 1997, p. 152-153].

Out-of-Character Activities

When a story is being created spontaneously, the creators need to delineate when they are speaking in character and when they are communicating to coordinate the story. The scripts of traditional plays are primarily composed of text to be spoken aloud by performers, but also contain *stage directions* to suggest action.

Out of live theatre, in Multi-User Dungeons (MUDs) and other chatrooms which are entirely text-based, special syntax and escape characters are used to indicate whether one of the chatroom members is saying something in character, out of character (as the person logged into the chatroom) or narrating the actions of the character [Murray, 1997]. When orchestrating improvised drama as a group, it is very important to be explicit about which actions are in-character and which are not, and are instead part of the orchestration. Modern improvisation has several special gestures that help orchestrate improvised shows among several performers, and I will discuss those in that section. There is a metaphorical similarity between the notion of in-character and out-of-character activities and foreground and background gestural activity. I will re-visit this in the section on Whole-Body Interaction During Performance.

Technician-Performer Relationships

I argued previously that the increase of technicians offstage decreases the ability for a performance to respond to its audience — the performance's sense of liveness. As the goal of these technicians is the same as a technologist, not at artist, they are typically not given the same license as those onstage to express themselves during the performance.

There are a few interesting cases where the technician/performer divide has been specifically examined. In one case, a CG projection on two screens on either side of the live performers was placed onstage [Shiba et al., 2010]. The CG projections were controlled live, able to load and move different landscapes and animals to complement the performance. Instead of performers or technicians in this case, the controllers of the CG projections are referred to as "operators". More interestingly, Kirk Woolford, Michael Klein and Bruno Martelli are specialist programmers who work on interactive technology for dancer performers and are referred to not as technicians but "co-dancers" [Dixon, 2007, p. 199].

Audience Directing Scenes and Content

Some works have opted to give partial control of the performance to the audience. While the performers still have autonomy in their actions, the audience may direct the show at a higher level. For works where the audience directly controls the performers, see the next section.

Augusto Boal is a strong proponent for re-imagining the performer-spectator relationship. As a child, he staged shows where "no individual 'owned' their character, whoever was available to take on a role at the critical moment would do so, interpreting it as they saw fit." As an adult, he stated that "all people have both the ability and the right to be active makers of art." Boal draws inspiration from Marxism, noting that the "means of production" of art have been taken away from the spectator, and instead he

coins a new term the " 'spect-actor' one who observes but is also able to act." Boal synthesized these ideas into a new form of theatre - the *Theatre of the Oppressed*. One technique he used was termed *Simultaneous Dramaturgy*, where "the spectators call out suggestions for action which are immediately improvised by those performing" [Babbage, 2004]. Boal describes his hopes for the audience in his theatre: "We [intentionally] desecrate the stage, that altar over which usually the artist presides alone. We destroy the work offered by the artists in order to construct a new work out of it, together" [Boal, 1995].

Keith Johnstone, one of the founders of improvised theatre, described a theatre game where members of the audience would raise their hand when they were bored with the current scene. When enough audience members raised their hand (based on some previously-agreed threshold), the scene was cut off and the next set of performers started. Sometimes the audience members and the next set of performers overlapped, but if the ongoing scene was interesting, the next set of performers would be content to watch it [Johnstone, 1999].

Jeffrey Shaw's *Points of View* (1983) was a computer-generated 3D projected virtual world. The audience sat in front of a large screen to view it, while one audience member could control the point of view of the camera using a joystick placed at their seat. In this way, "the particular audio visual journey made by a spectator who operates the joystick which constitutes a 'performance' of this work. For the other, non-interacting, spectators, that performance becomes 'theatre' " [Dinkla, 1994].

Susan Kozel laid on a bed for Paul Sermon's *Telematic Dreaming* (1992). A live video was taken of her and projected onto an identical bed elsewhere, and vice versa, so a gallery visitor could come and lie on the identical bed and they could interact with each others' projections. The bed as a setting implies intimacy, and Kozel wrote at length about her experience, both the friendly manipulations and abuse her virtual body suffered at the hands of gallery visitors [Dixon, 2007, p. 216-220]. While it is true that in this case the audience does not "control" he performer directly, the mediatized performer, particularly in the context of the bed, betrays an intimacy to the audience that feels similar to other work in this section.

Roca's *Epizoo* (1994) attached pneumatic and robotic devices to the bodies of the performers. These could be actuated by a remote-control panel in the audience. Roca described this as "probably the first performance to feature a remote control device enabling the spectator to control elements including the artist's body" [Dixon, 2007].

Acting upon or controlling digital avatars can have a strong psychological effect on the avatars' supposed controller. Julian Dibbell describes an instance of a "rape" in a chatroom in 1998, where one anonymous hacker managed to falsely attribute actions to several characters. Many of the those who experienced these actions felt these were equivalent to real-life sexual assault [Dibbell, 1993].

Chameleons 3: Net Congestion (2000), directed by Steve Dixon, is a theatre performance with a projected backdrop, where anonymous participants from the internet could type in suggestions via IRC. The result was very chaotic, and interesting in terms of how roles and power were represented. The director observed that performers "frequently over-compensated, since they were working in a theatrical vacuum unable to adequately gauge audience reaction." It is interesting how lack of "presence" of the audience, which usually gives subtle feedback, affects the performer.

Single Thread Theatre Company's *The Loyalists* (2012) was set in a public park with a set meant to resemble a microcosm of Toronto during the American occupation in 1812. The cast spoke directly to the audience, who were treated as citizens of Toronto under American control. After a brief introduction, the cast gave the audience roles but they were free to wander the park, encountering small scenes and contributing to the furthering of the story, such as by helping in an interrogation of a prisoner, joining the American military, or helping purchase rations for the soldiers. The performance represented an interesting tension between a "scripted" performances and actors responding truthfully, in character, to the actions of the audience. Single Thread's work is heavily inspired by video games with a story element, and their work represents a fusion of lessons learned from interactive storytelling in different media [Single Thread Theatre Company].

3.3 Improvisational Theatre

This section will be a description of modern "improv theory", in particular describing how modern organically improvised theatre is unique from other forms. Theatre, as a medium, has the ability to be incredibly reactive to its audience, subtly tailoring each performance, regardless of script, to suit the occasion. Theatre was certainly the first form of art, in the form of rituals at the origin of civilization. Liveness appears to be the only thing still relevant about theatre, in the presence of other, mediated, art forms, such as film. We do not have precise descriptions of theatrical performances before the introduction of scripts, in Ancient Greece [Brockett and Hildy, 1998].

I have learned a great deal of the content in this chapter first-hand from my 14 years (since 2001) of hands-on experience of performing comedy. Unfortunately, most of the knowledge of improvised comedy in terms of "what is funny and why" is transmitted orally, whether through casual word-of-mouth or paid workshops. As Shakespeare said, "brevity is the soul of wit", and the value of a joke is lost when it has to be explained [Shakespeare, 1603, II,2,92]. A search for a discussion of these topics has not been very fruitful — yielding at best an online glossary of terms at a stand-up comedian's website. Jeffrey Scott, whose 2014 PhD thesis is on the history of improvisation in theatre, reinforces my observations:

> "In spite of being such a continual presence in the theatre, no comprehensive account of the role of improvisation in theatre history exists ...
> The vast majority of texts available on the subject of improvisation are much more of the "How To" variety, giving the would-be improviser a list of games with tips on how to improvise well. Considering the persistence of improvisation in performance, the lack of scholarly and theoretical discourse on the subject seems to be a significant gap in the current body of theatre research ...
> improv is, by nature, difficult to study since it leaves behind no artifact, like a written script." [Scott, 2014]

Performances using a large amount of explicit improvisation (i.e., not relying on a script) began to appear in 16th century Italy, with *commedia dell'arte*. Despite the performance being mostly improvised, every actor would play a stock character "type" (such as *Pantalone* or *Harlequin*). Instead of scripts representing an entire performance, commedia would rely on *lazzi*, loose patterns of dialogue and/or

movement that could be initiated by any actor, and the other actors would follow [Brockett and Hildy, 1998, Scott, 2014].

Note that the term improvisation is also often applied to fields of music (particularly jazz) and dance (i.e. contact improv). Across all fields, there is a sense that the act of spontaneous improvisational creativity and expression is important, in some ways a political statement about the power of self-expression [Fischlin et al., 2013]. Arguably, theatrical improvisation is qualitatively different in that it produces a narrative (see Story-Making) instead of a series of aesthetically pleasing patterns. We shall move on quickly without worrying whether the previous sentence offends anyone.

3.3.1 Statement of Personal Experience

I should state here my personal experience with Modern Theatrical Improvisation, as it motivates my work and acts as a first-hand source for much of my experience with longform collaborative story-making and improv gestures. This section serves as evidence for my authority on the topic of Modern Improvisation.

I joined my high school's improv team when I was 16 (2001), part of the Canada-wide *Canadian Improv Games*[1] tournament. The tournament consists of four shortform "events", each four minutes long, that could be one of the following five structures: *Life*, *Theme*, *Story*, *Character* and *Style*. After graduating high school, I became a trainer at my local tournament (Kingston) for 5 years, the last of which I was head trainer, coordinating other trainers and curriculum for 16 high-school teams.

Also after graduating high school, I joined the Kingston-based collective *The Improv Show* (website now defunct). The Improv Show performed a weekly 60-minute show for most of the 5 years I lived in Kingston, and I accumulated significant stage time and experience in different practices and managing audiences (>200 shows). While we had a core cast (which I was a member of) we had rotating guest performers through bringing their own practices, from Atlanta (Dad's Garage), Edmonton (Rapid Fire Theatre) and Montreal (Uncalled For).

Members of The Improv Show have gone on to be artistic directors in Ottawa (Insensitivity Training) and Oxford (Oxford Imps). The Improv Show's style was mostly short-form scenes, but experimented more with longform towards the end of my tenure there. At one point, we took around 30 suggestions and wrote them in chalk all over the stage floor and back wall, and did a 40-minute scene, occasionally looking on the wall or floor for inspiration. In my final two years in Kingston, I was the artistic director of The Improv Show.

I moved to Toronto in September 2008 and started taking classes with Impatient Theatre Company (ITC) [2], focusing specifically on the longform, unstructured format, *The Harold*. I completed ITC's curriculum, approximately 125 hours of class time. I have since performed with ITC Harold teams weekly over a cumulative period of 2 years. During 4-month internships away from Toronto, I spent time in India working with stand-up comedians to teach improv, and was part of the players in the Cambridge Improv Factory in Cambridge, UK.

[1] `http://improv.ca`

[2] `http://www.impatient.ca/`

In 2006, I briefly established an improv team in the virtual world *Second Life* with Dan Zellner [3]. However, we found the interface too difficult to use for physicalized acting, and it was not possible to create spontaneous virtual sets quickly enough. At this time in Second Life's development, microphones were not common for most participants, so performers and audience members would have to wait in anticipation while others typed.

3.3.2 Modern Improvisation

Modern improvisation can be divided into two categories: shortform [Johnstone, 1999, Spolin, 1983] and longform [Halpern, 2006, Halpern et al., 1994]. Short-form improv sets are up to 5 minutes, and tend to take the form of semi-structured games (e.g., "You can speak in a certain number of words") while long-form improv sets are between 10-40 minutes, and tend to be more open-ended.

While an entire short-form set tends to take place in one scene, indicating a continuous moment on stage, long-form sets tend to transition between scenes through several pre-arranged techniques. One basic scene transition technique is inspired by film: the "sweep", where a member of the cast runs from one side of the stage to the other in front of the improvisors. This indicates to both the improvisors and the audience that the scene is over and they will (as quickly as possible) transition to the next. It is interesting to note that if the improvisors were to leave the stage to start another scene without such a clear cue, it would be difficult to distinguish if the motion was part of the character's behaviour still in the context of the scene. Of course, there is a large diversity of such transition techniques and many subversions of each, often to comic effect. More examples of such activities are described later in the "Gestures" section.

The form that most long-form improv takes is the whimsically named **"Harold"**, invented by Del Close and Charna Halpern and originally described in their book *Truth in Comedy* [Halpern et al., 1994]. Fotis also provides an up-to-date, in-depth analysis of the Harold [Fotis, 2005]. The form is a series of 3 sets of 3 naturalistic scenes, with less naturalistic "group games" in between each set. Each set is supposed to revisit the same scenes, but possibly at a different time period for the characters, or with new characters but somehow thematically linked. Close and Halpern were very clear that the structure of the Harold is a guideline, a series of rules meant to be broken, and it is very rare to make it to the "end" of a full Harold before a natural ending is found. An artist's diagram of the structure of The Harold is shown in 3.1.

Most modern improvisation is *minimalist* - it eschews props, costumes, stage sets and changes in lighting except for the lights down at the end of the set. The popular short-form improv TV show *Whose Line Is It Anyway?* makes use of props and recorded video on a green screen, but the comedy of these segments tends to derive simply from the sense that the inclusion of the prop or video is absurd, rather than working with the improvisors to create the scene. Drew Carey, the host of the US version of *Whose Line Is It Anyway?*, followed up with a single season of a show called *Drew Carey's Green Screen Show*, where all the improv scenes were recorded in front of a green screen background, with animation, music and sound effects inserted in post-production.

The Neutrino Video Project is a "live improvised film": audience members are invited into a theatre as

[3] `http://studioz.org/about/`

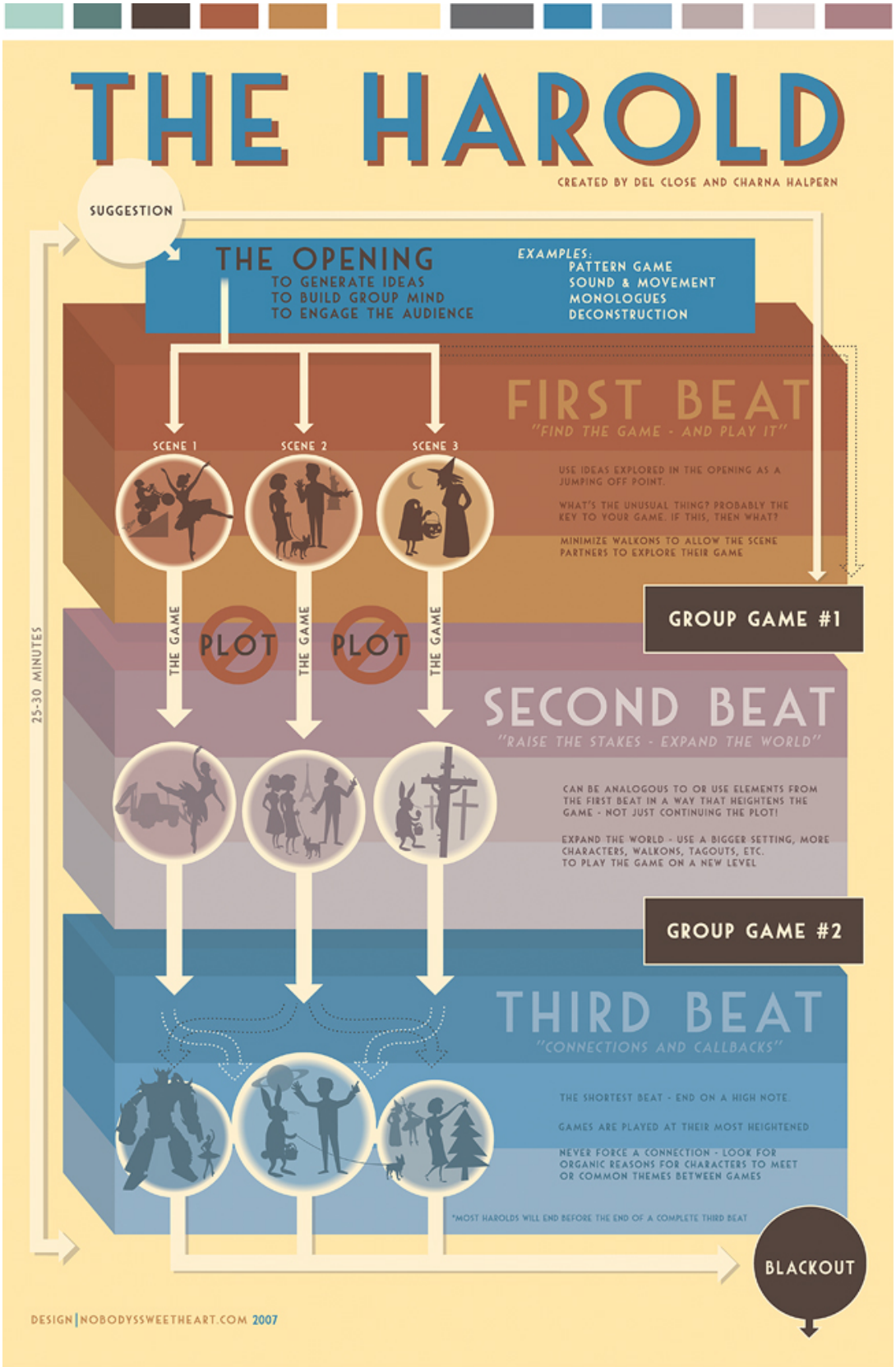

Figure 3.1: Dyna Moe's diagram of the longform improv structure *The Harold*. [Moe, 2007].

usual. Upon receiving a suggestion from the audience, film crews run outside into the city and begin filming scenes. As scenes are finished, "runners" (in the era before fast mobile internet) bring the tapes back to the theatre to be played. There are gaps in time where the audience is waiting for the next tape, but these are usually filled with a host discussing the film made so far, or performing some stand-up comedy. The audience experience of The Neutrino Video Project is of watching a film that appears to be non-live, but with the nagging conscious awareness that the scenes are being filmed live outside the theatre [The Neutrino Video Project].

3.3.3 Structure of Improvised Sets

We will not try to describe all of modern theatrical improvisation and comedy here, but we will focus on core features we find inspiring for the present work. All comedy, unscripted or not, has several intriguing features: callbacks, subversion, reinterpretation and recontextualization. A callback is a feature of pattern in comedy where something is referenced near the beginning of a set, and then set aside and referenced again much later. The use of a callback brings laughs to the audience due to the sudden remembrance of the moment, or the surprise that the previous moment and current moment are linked. Subversion, reinterpretation and recontextualization are all different terms for the same thing: the ability to take a familiar situation and reinterpret it out of context, whether the social patterns around getting seated at a restaurant, the phrasing of a politician's speech, or how cultures deal with death. In Eric Idle's novel *The Road To Mars*, comedy is treated as the most powerful force in the universe, as anything can be the butt of a joke, empowering the jokers [Idle, 1999].

The structure of a story, in common knowledge, is a plot with a beginning, middle, and end, with rising tension in the action and a climax just before the end. It is a common mistake that the output of theatrical improvisation is meant to be similar to a story. In my experience, there are some who treat improvisation work as only a non-performance warm-up or ideation tool, and do not think it is appropriate for actual performance. The theatre company Second City, with training and performance centres in Chicago, Toronto and elsewhere, are noted for saying that they use improv techniques to develop ideas for their scripted comedy shows, which are treated as the primary output.

Improv performance, on the other hand, looks less like a plot and more like the performers playing with each other on stage in a way that is entertaining to watch. A term initially used by the Upright Citizens Brigade in New York is "finding the Game of the Scene". Tim Uren describes the process of structureless improvisation well: "...the stories are created by playing a game. As each story is unique, so is the game that creates that story. The rules of such a game do not exist until the story begins." Uren describes how skilled improvisors fixate on aspects of each other's utterances in order to create patterns of the game, which lead to the "story" made during the improv session [Uren, 2007]. So, the progression of improv is the performers creating and playing with patterns and exploring those patterns in new contexts. The dubious distinction between "finding" or "creating" a pattern is as old as reflecting on art-making itself.

Zaunbrecher covers some "rules" of improvisation that practitioners observe when making decisions during performance [Zaunbrecher, 2011]. In this work, we are concerned with flexible re-use of content in the context of improv. This structure of improvisation is amenable to additional techniques to call back scenes — in our approach, to enable playback and manipulation of video of the scenes themselves,

in new contexts.

3.3.4 Coordinating Gestures in Modern Improvisation

I use the term gesture here to describe actions of the performers that are perceived as not part of the behaviour of the characters they are playing. These gestures are external to the action of the characters on stage, and are useful for the players to coordinate the action between themselves. It is usually easy for the audience to tell the difference between actions that are meant to be attributed to the fictional characters, and gesture actions. Similar to scripted theatre, it is obvious to even novice audience members that the people who come out in all-black clothing between acts are, in fact, stage managers, and not actors. By contrast, performing a narrative, i.e. directly telling a story to an audience, can include gestures, but in this case the performer is not as worried about staying in character [Mostafapour and Hancock, 2014].

I will give a listing of some of the gestures used in improvisation I have witnessed since 2001. Usage of these gestures vary from group-to-group, and have a regional variation similar to accents in any language. I have observed different usage while performing regularly with groups in Canada (The Improv Show, Kingston; Impatient Theatre Company, Toronto) and the United Kingdom (The Cambridge Improv Factory, Cambridge).

Sweep: A member of the cast runs from one side of the stage to the other in front of the improvisors. Similar to transitional wipes used in film, this indicates that the scene is over and the group will transition to the next one, such as a clock wipe as used in Akira Kurosawa's *Hidden Fortress* and extensively in George Lucas' *Star Wars* series. Often the initiator for the sweep has an idea for the following scene and initiates it, but some times they are euthanizing a scene that has gone stale.

Tag-out: When a scene with more than one performer is ongoing onstage, a performer from offstage comes and taps one of them on the shoulder. This indicates that the tagging performer will replace the onstage performer as a new character. The other performers will remain on stage and their characters will remain the same, but the scene may now be in a new location. For example, two characters are in a pet shop, one man trying to get a refund from a shop owner for a dove that he killed. A performer comes from offstage and tags out the shop owner and then establishes that the scene is in an orphanage. The other man now starts trying to return a child that he didn't like to the orphanage (an example of *recontextualization* in comedy).

"Cut to that!": A verbal command from offstage that sounds like the call of a film editor. This indicates that the onstage scene should transition to an event just mentioned. For example:
A: I see that your dog only has three legs.
B: Yeah, he lost the one while serving in the Canine Squad in Afghanistan.
(offstage performer): Cut to that!
[A set of new performers come on stage and perform a short scene where a human drill sergeant instructs several dog recruits about the dangers of the hardened Afghan breed they will encounter.]

(any performer): Cut back!
[The performers in the Canine Squad scene clear the stage and performers A and B return to their original positions]
A: Sounds awful.

Most *"Cut to that!"* calls have a corresponding *"Cut back!"* call, though it is not strictly followed. An alternative is *"Let's see that!"*

Scene Painting: When establishing a location on an empty stage or describing a physical feature of an ongoing scene, a performer will either come from offstage or step out of character and verbally describe a feature. For example, on an empty stage, a performer could come on and say "We see an old, rusty grandfather clock that has stopped keeping time" while miming its location. Other performers could add more details to the clock, or the scene in general. Sometimes, a performer from offstage will interrupt a scene to describe a visual component that would not come up in conversation. For example, a character appears to be eating very messily; a performer comes from off stage and says "We see spaghetti sauce covering this woman's neck, shirt, and dripping all the way down to their pants" all the while indicating the grotesque dripping of the spaghetti sauce with their hands. Saying "We see..." at the beginning of scene painting is a convention that varies between different regions.

Soliloquy: Soliloquies occur in Shakespearean plays, where one character steps or turns to one side of the stage and speak their thoughts out loud, with the conceit that only the audience can hear them. It is made clear that this is a soliloquy instead of conventional dialogue because the performer steps forward and/or turns aside very clearly and perhaps changes their tone of voice. This is often used in theatrical improvisation in well. Improvised "mystery" shows are notoriously difficult to perform as it is difficult to have a compelling mystery unless all the performers share the same backstory. Soliloquy's are one way for performers to share their characters' thoughts unambiguously with other performers while on stage.

While not directly related to improvisation, there has been work on classifying gestures used during freeform narration and conversation [Okada et al., 2013, Ponce-López et al., 2013].

3.3.5 Improvisation as a Cognitive Task

The practice of theatrical improvisation has served as inspiration for some work in the field of artificial intelligence. However, we cannot find any in-depth academic work on interfaces to be used during theatrical improvisation. The relevance of the following work is how it explores the decision-making process of improvised narrative work.

In 2004, Owsley et. al. created an "Association Engine" to associate between pairs of words, explicitly inspired by theatrical improvisation games [Owsley et al., 2004].

Brian Magerko and Daniel Fuller at Georgia Institute of Technology's *Digital Improv Project* have done a great deal of work exploring improvisation as an artificial intelligence problem [Baumer and Magerko, 2010, Fuller and Magerko, 2011, Magerko and Riedl, 2008, Magerko et al., 2009, 2010].

There has also been an exploration of building computational agents capable of theatrical improvisation-

like behaviour [Hayes-Roth and Van Gent, 1997, Zook et al., 2011]. There is a short paper describing "improvisational narrative agents" controlled by full-body gestural interaction, but it appears to be marionette-like, where the controller is directly controlling movements instead of co-improvising with an intelligent agent [Piplica et al., 2012].

3.4 Capturing and Projecting Images

This section is concerned with artificial images used in theatre. Dixon and Salter provide a good coverage of the use of shadows and other projected forms in art up until the mid-20th century, starting with Plato's conceptual Cave, which has interesting philosophical implications in its own right [Dixon, 2007, Salter, 2010]. Janet Murray has said that the purpose of technology for spectacle is less about the audience suspending disbelief and more about "creating belief" [Murray, 1997, p. 110]. We will not be concerned about trying to render spatially-realistic or photo-realistic images, but rather about creating an effect in the mind of the audience and performers that creates an enjoyable experience.

In the 1970s, Phaedre Bell enumerated the relationship between film and theatre as follows: primary, where the film is the primary source of the performance, and live, theatrical elements merely augment the film; secondary, where the theatre is the primary source of the performance and the film augments; or "dialogic" - the film and theatre are an equal balance between the live and recorded, as opposed to either recorded or live performance serving one or the other. Such performances are "dialogic media productions" [Bell, 2000]. Dixon noted that the semiotic relationship between screen image and stage action can be either dialogic (i.e. A versus/in relation to B) or additive (A + B = C, something new) [Dixon, 2007, p. 335]. An interesting feature of projected images in the theatre environment is whether they are treated as **separate** or **conjoined** - this is discussion by Dixon in the context of The Builders Association's *Jump Cut (Faust)* (1997) [Dixon, 2007, p. 343-348].

In this section, we will cover the concept of *Liveness*, of importance to theatre and other interactive experiences, *Capturing and Using Images* in the context of a performance, *Interacting with Artificial Bodies*, where captured images are instantiated on stage in the form of a body, and *Projection Technology*, where we review technology to create the effect of a live image on stage.

3.4.1 Liveness

The topic of *liveness*, meaning the dramatic sensation that what the audience is witnessing is urgent, happening here and now, has been of much concern to theatre and other performance arts, especially given the success of non-live, mediatized art. Many argue that liveness is what makes theatre unique, and theatre should not be tainted with non-live elements so as to maintain what is special about it [Grotowski et al., 1967].

Dixon defines liveness as the feeling of sharing the air with the performer versus the performer somehow being mediated from elsewhere in time and space, or simulated - he also argues that the 20th and early 21st century enjoys and revels in fakeness, treating every digital image as likely false. There is a concern that theatre should be a pure live form, and un-live elements detract from it [Dixon, 2007].

Peggy Phelan and Philip Auslander have conflicting perspectives about the use of non-live, reproducible components in performance. Phelan makes a passionate, phenomenological argument: "Performance's independence from mass reproduction...is its greatest strength..Performance cannot...participate in the circulation of representations of representations: once it does, it becomes something other than performance" [Phelan, 1993]. Auslander makes an argument that it is not tenable to distinguish between live and "mediatized" theatre, and that much of modern "live" theatre is actually mediatized in some way [Auslander, 1999]. He describes the combination of mediated and non-mediation as a "fusion, not a con-fusion" and perhaps that is because practitioners have become better at the fusion, as opposed to fetishizing the separation, either optimistically or pessimistically. Auslander thinks the dominant force is the digital, mediated, which the live is incorporated into, i.e. "Live Dance + Virtual = Virtual". Note that this seems to change the definition of live and mediate in a way that is slightly confused. Auslander states: "If the mediatized image can be re-created in a live setting, it must have been 'real' to begin with" [Dixon, 2007, p. 124].

It seems too difficult to define "liveness" in terms of the technology used, so perhaps a phenomenological definition is easier. What gives you the feeling of "being right there" is live, particularly the temporality of it. Dixon notes that expectations of audience behaviour varies between the cinema and the theatre, but also between different types of theatre - more "live" performances seem to demand more respect from the audience. "Live performance always carries with it the possibility that the unexpected may happen. [whether or not it does, from the audience or the performer's perspective]" [Dixon, 2007, p. 130]. Peggy Phelan argued that "presence" seems to be more graspable term than "liveness". Dixon claims that "presence is about interest and command of attention, not space or liveness". A live performer and their adjacent, recorded, two-dimensional projection of themselves will have different sensations of liveness, but if they start having different behaviour they will pull focus from one or the other. Art that feels present is the art that demands attention from the audience. Theatre, as opposed to other forms of media, is traditionally confined in one time and space. Digital theatre, especially with the use of projection, allows artists to "fragment" time and space (a term used by the Italian Futurists), the final product being a "bombardment of images from different times and spaces" [Dixon, 2007, p. 335].

Jonathan Hook [Hook et al., 2012] discusses the definition of "liveness" in the context of Human-Computer Interaction, though there is less vexing over the significance of liveness in HCI than in the world of theatre.

We shall discuss a few performance works that specifically play with the concept of liveness. In 1914, Winsor McCay performed with *Gertie the Dinosaur* in 1914, using precision timing to coordinate his live action with an animated film, including prop transactions, where a real prop would join the animation and become virtual, or vice versa [McCay, 1909]. This is an example of using close timing to "cheat" a sense of "liveness", in this case *dialogic* interactivity [Dixon, 2007].

In John Jesurun's *Deep Sleep* (1985), characters onstage and onscreen argue about who is more real. As in the Gertie the Dinosaur performance, Jerusun treated onstage and onscreen as separate spaces that characters could move between [Dixon, 2007].

In Robert LePage's *Needles and Opium*, a screen with a background of water projected on it is positioned to cover the majority of the performer's body. Only the live performer's head is "above" the water, playing a trumpet, while his projected body below constantly treads water [Dixon, 2007, p. 356].

The Wooster Group's *Poor Theatre* (2004) was a (re)production of Jerzy Grotowski's *Akropolis* (1960) production, where monitors displaying a video of the 1960 production were visible during the performance to the Wooster Group's performers, who were instructed to mimic the gestures and movements of the original as precisely as possible [Salter, 2010, p. 130]. While the live actors and recorded actors are not projected in the same space, the live actors trying to align their behaviours to the non-reactive video creates an interesting effect.

In this thesis, we are concerned with a recently-recorded performer projected on stage next to a live performer, as in a performer who is bodily present. Since they will have witnessed it, it will be clear to the audience that the recent recording is not live, but we are not trying to create that illusion. While the topic of liveness has been discussed heavily, and thus we must call attention to it as we did above, we are primarily concerned in this thesis with making a performances that are interesting to watch. One component of creating interest in improvisation, as the audience or performer, is not knowing what will happen — there is a sense of risk. This sense of vulnerability is similar to what the authors above have identified is important about the difficult-to-define concept of liveness.

3.4.2 Capturing and Using Images

Salter provides an overview of the history of people fascinated with capturing the movement of the body. In 1872, with the use of chronophotography, Eadweard Muybridge captured the record of a horse galloping. In 1884, Čtienne-Jules Marey captured human subjects walking while wearing full-black costumes with white stick figures drawn on top [Salter, 2010, p. 223-224]. Much of the discussion of body capture and movement is in the context of dance performances, which are of a different, special character than we are concerned with [Salter, 2010, p. 225-227]. However, Laban's *Labanotation*, a precise system of notation for dance movement, deserves mention as an early symbolic attempt to abstractly describe body movement [Hutchinson, 1955].

In David Foster Wallace's novel *Infinite Jest*, Hal Incandenza and Mario Incandenza exhibit a "film" consisting of a live video of the audience, noting that the people who stayed the longest were the academics, paralyzed by the meaning of the piece of art they were involved in [Wallace, 2009].

Bruce Nauman and Dan Graham separately created video art installations that gallery visitors could interact with through 1969 to 1986. Live video of visitors, which the artwork turned into performers, was fed back to them, distorted, from different viewpoints, and sometimes mixed with pre-recorded content. In principle, these installations had no higher aspiration than a technologically-enhanced version of the Hall of Mirrors found at a circus, but their effect was extremely strong. One critic, Margaret Morse, described the experience feeling like her body "had come unglued from my own image" [Salter, 2010, p. 124-125].

Svoboda's *Intolleranza* in 1965 used lived CCTV technology for various purposes, including recording and projecting a live view of the audience on the stage. One of the themes explored in *Intolleranza* was that of racism, and at one point a negative image of the audience was projected on the stage, the white audience members appearing black and vice versa [Salter, 2010, p. 126-127]. CCTV was used again by Allen Ginsburg in *Kaddish* by projecting a recording of the live scene on stage, sometimes slowing it down or speeding it up, altering the audience's perception of time [Salter, 2010, p. 128]. *Prune Flat*, in

1965, had a previously-recorded film of the performer projected onto the live performer, in exact scale. The live performer wore white and synchronized her movement exactly with the superimposed film projection. The filmed performer removed her clothing piece-by-piece, while the live performer mimed the same actions without undressing. The performance ended with the filmed and live figure standing still, the naked filmed body projected onto the live performer's dress. Work in this area continued with Gertrude Stein Repertory Theatre's interpretation of Alfred Jarry's *King Ubu* (2000). In this case, the performers were live, but video feeds arrived from different locations using video-conferencing software. The live performer's costume could be referred to as a "neutral costume", able to be projected upon.

There have been a few relevant examples of work exploring representations of live performance. Jimenez et al. create a mirror which shows a distortion of the current image in space and time, both in the video and the audio [Jimenez et al., 2005a]. Miwa et al. explore the idea of representing performers as shadows or silhouettes on a slit screen between them and the audience. These shadows may be displaced in time, and the audience can also interact with them [Miwa et al., 2011]. There are two recent works which present video from a constant time interval in the past. In both cases, the users have no control over the video playback. First is Piper and Agamanolis' Palimpsest [Piper and Agamanolis] and next is Bartneck et. al.'s Interactive Visual Canon [Bartneck et al., 2009b].

Uninvited Guests' performance *Film* (2000) is an interesting example of the mixing of live and non-live components. During the performance, a photographer (present on stage) takes stills of a performance and these are projected on the back wall, frozen in time, while the performance is still ongoing. The taking of the photo captures the present, which immediately becomes the past [Dixon, 2007, p. 525-531].

Peter Petralia's *Virtuoso (working title)* has several actors go through the process of filming a 1950s-era American sitcom, the actors themselves positioning the cameras and preparing the scenes. The entire process of the actors preparing to produce particular scenes, which are displayed on 3 large monitors facing the audience, is visible to the audience. The perspective of the camera is sometimes lacking in information, such as when one character moves their lips close to it, showing intention to kiss another character. This forced perspective, from the camera's point of view, means the audience changes where they are focusing as they watch the show - while all that appears on camera and on the monitors is considered part of the produced show-within-a-show, the live actors' actions on stage are ambiguously included in the show-within-a-show. All video is shown live - there is no delay or buffering [Petralia, 2010].

Dan Graham's *Present Continuous Past* (1974) was an art installation with video of participants delayed by 8 seconds [Dixon, 2007]. In Blast Theory's *10 Backwards* (1999), the main character videotapes themselves exaggeratedly eating breakfast, then plays it back, mimicking themselves again and again, fixating on tiny features, becoming grotesque and unnatural [Dixon, 2007, p. 247].

Technology exists not just to capture raw video of the body, but to create a model of it — Xu et al. present a case where new 3D characters can be created from source video of live performers performing certain activities in place, e.g. kicking or punching [Xu et al., 2011].

3.4.3 Interacting with Artificial Bodies

Dixon discusses what he calls the *Digital Double*, beings that appear on stage with ourselves, in detail, identifying four types [Dixon, 2007, p. 241-270]:

- Reflection
- Alter-ego
- Spiritual Emanation
- Manipulable Mannequin

A *reflection* is a double that is reflected back to the performer, somehow changed. This is the usage in Blast Theory's *10 Backwards* (1999), described in the previous section. The installation artist George Khut uses the term "transforming mirrors" to describe a project that reflects your sense of self back, but in some transformed way [Khut, 2006].

An *alter ego* is an alternative self to converse with. Pre-recorded video that live performers time themselves to, as in the example given above, fall into this category.

A *spiritual emanation* is an abstract, impressionist representation of self. For our purposes, we are concerned with direct, realistic reference, so this is out of our scope.

A *manipulable mannequin* is arguably the most advanced form, an image whose behaviour is not predetermined, but may be controlled in some part by the performers, and in some part by pre-programmed behaviour. Many of Dixon's examples of this are avatars in a 3D virtual world that speak to the performers.

Dance simulation software is one interesting example of manipulable mannequins. Credo Software Products' *Life Forms Dance Software*, with its first version made in 1989, has been used to make a number of simulated dance sequences. Sometimes this simulation software can be manipulated to create "humanly impossible" sequences, which are interesting as challenges to performers to replicate [Dixon, 2007, p. 184-187]. It was even used distributively among several artists to make an *exquisite corpse* dance sequence [Hilton, 1998]. Others have used this dance software to play with the bounds of human possibility - "figuring out what a body on two legs can do" [Dixon, 2007, p. 188].

3.4.4 Projection Technology

Theatre performances that involve projecting an image seemingly suspended in the middle of a stage use a scrim - a special piece of fabric that is transparent unless light is projected directly on it. With careful arrangement of stage lighting, a performer can be standing near the scrim and appear well-lit, and the scrim only emits light where a projected image touches it. It is convenient to have the live, present performer "behind" the scrim, so they can look forwards to both the projected performer and present the forward side of their body to the audience [Dixon, 2007, p. 190]. Riverbed's *BIPED* (1999) combined live dancers and figures projected on a front scrim.

Tsuchida et al. present an interesting technological alternative (see Figure 3.2) [Tsuchida et al., 2013]. To support a single dancer practicing dance choreography with multiple dancers, a human-sized screen is

Figure 3.2: The work of Tsuchida et al., showing a live dancer alongside a self-propelled robot with a projection screen on top. A calibrated projector projects the video of a pre-recorded dancer on the screen [Tsuchida et al., 2013].

mounted on top of a small robot. The video and movement in space of a dancing human are recorded, and the robot may exhibit that dance, in video and movement in space, as output. While a moving screen is certainly interesting, and necessary to re-project performer positions faithfully in space, we are concerned that the audience will perceive it as distractingly novel.

It is in our interest to record a performers movements and project it back so it appears in roughly the same place. The impression of appearing the same place and moving in the same way is only for effect — it does not need to be physically accurate. Lee et al. present a useful technique for calibrating multiple projectors by using a fast dark/bright projector pattern and embedded sensors in the object to be projected on [Lee et al., 2004].

3.5 Video Manipulation

This section describes the state-of-the-art in video editing that is either casual, or done with a sense of urgency. Captured video must be *navigated*, which is often aided by *summarization or processing*. Then video can either be *automatically edited and presented*, as appears in many casual consumer interfaces. We also review two categories of manual video manipulation: *editing* and *compositing*. We also cover video editing as a *live interface* during a performance.

There is a vast amount of commercial software for video editing where the interaction technique is mouse and keyboard, and the intended output is a rectangle of conventional video to play on a screen. In the intended performance this literature is for, we aim to explore how performers can control video where the control of the video is part of the performance, and the video is projected or represented some way on stage, and even on the performers' bodies. To this end, we use the term *manipulation*, which implies a bit of a more ad-hoc, in-the-moment, playful attitude than *editing*, which seems to imply careful, off-line consideration. We shall use to the term editing when talking about video manipulation in terms of timing.

Another goal of this exploration is to find techniques for users to manipulate video quickly. In order to aid this activity, the video often can be pre-processed to find points-of-interest, meaningful sub-chunks and other features.

Goldman et al. and Borgo et al. both provide good surveys of relevant work [Borgo et al., 2012, Goldman et al., 2007]. Borgo et al. present a very thorough review in 2012 of *video-based graphics* and *video visualization*; the former is manipulation of video for artistic or entertainment purposes and the latter is the act of creating a visual representation from input video to reveal summarization or overview information. We are primarily interested in ad-hoc manipulation of videos, and are thus interested in compositing, editing and retargeting techniques, as well as visualizations of video that afford understanding or further decision making about the video at a glance.

There does not seem to be much work studying how time-sensitive video is prepared in professional settings, but [Bergstrand and Landgren, 2011] is one such case.

3.5.1 Video Navigation

A large body of research has explored video navigation. Scrubbing provides a real-time update of the current frame of the video as the user moves the slider along the timeline [Li et al., 2000, Matejka et al., 2012]. ZoomSlider [Hürst, 2006] and PVSlider [Ramos and Balakrishnan, 2003] explore different dynamics of playback sliders for finer control while scrubbing. Victor demonstrates a comic strip view of the video that may be scratched [Victor]. We will later describe how we apply scratching and scrubbing within in live contexts. Several projects have explored video navigation through direct manipulation, where the user directly drags objects in the video rather than relying on the timeline [Dragicevic et al., 2008, Goldman et al., 2006, Karrer et al., 2012].

Another approach to facilitate browsing exploits visual cues for video content. Most common are thumbnail visualizations, which provide an overview of the entire video stream [Hürst and Darzentas, 2012], or summaries, where duplicate content has automatically been discarded [Ma et al., 2002, Truong and Venkatesh, 2007]. Enhanced timelines, which augment the seeking bar to make it content-aware, have also been explored to aid quick retrieval of content of interest [Alexander et al., 2009, Pongnumkul et al., 2010]. There has also been work on visualizing navigation history to aid future navigation [Al-Hajri et al., 2014]. Videotater [Diakopoulos and Essa, 2006] and Swifter [Matejka et al., 2013] are two examples of seamless combination of interaction and visualization, displaying near-context thumbnails during navigation.

All of the above navigation techniques have the common goal of facilitating quick access to segments of interest through enhanced interaction or visualizations that spare the user from passively watching the entire stream at normal speed. While they have primarily been designed for offline manipulation, similar approaches are suitable in the context of live manipulation.

3.5.2 Video Summarization and Processing

Video can be summarized for ease of use in some later activity. The goal may be an overview, to display points of interest or ideal parts for clipping, segmenting or cutting. There may be a desire to "chunk" a temporally long video into smaller components, at different levels: Video (Whole), Scenes, Shots (contiguous frames), Frames (single). The output of a video summarization system may be static, like a thumbnail, dynamic (e.g. a cinema trailer), or interactive for quick browsing.

Borgo says that "key frame selection is typically the first step in image-based video visualization", where a keyframe is chosen such that it "optimally represents the contents of the video according to a specified criterion". Keyframes may be chosen according to content change (inter-frame difference), maximum frame coverage (similar to the most frames), feature space analysis (keyframes represent clusters), minimum correlation (between keyframes) or some heuristic of "interesting" (high "information content") [Borgo et al., 2012].

We explore non-interactive and interactive representations of single videos, followed by a discussion of relevant video processing techniques.

Figure 3.3: Shinichi Maruyama's visualization of a nude dancer [Maruyama, 2013].

Non-Interactive Summarization

Painters have long tried to capture a sense of dynamic movement in static form. Artist Shinichi Maruyama presents the movement of nude dancers summarized in a single "shot" in 3.3. Forsythe and Tanzarchiv's dance instruction book includes a CD that shows dancers' movement against a background of several human-chosen silhouettes meant to summarize the dance [Forsythe and Tanzarchiv, 1999][4]. Caspi et al. present a system for summarizing video into a single frame, one interesting feature of which is awareness of occlusion from foreground objects at different times [Caspi et al., 2006]. Slit-Tears uses a technique where users draw a line on a video, and the output is a summarization of activity on that line throughout the video [Tang et al., 2008]. Cliplets is a user-assisted system to "juxtapose still and dynamic imagery" from a single video clip, the output being a mostly-static frame, with a user-chosen component smoothly looping in time [Joshi et al., 2012].

Interactive Exploration Interfaces

Li et al. study advanced playback controls for a conventional video-browsing interface: time compression preserving audio pitch, speaking pause removal, and navigation between shot boundaries [Li et al., 2000]. It is increasingly common for videos to be browsed, rather than watched, by scrubbing through a timeline — Pongnumkul et al. enhance the timeline to make it content-aware [Pongnumkul et al., 2010]. Ramos and Balakrishnan allow browsing a video as a storyboard (strip of frames), with sub-selecting allowing them to "drill down" hierarchical levels of detail in video segments, and flicks right or left to minimize or expand the video's display in the freeform "workspace" [Ramos and Balakrishnan, 2003]. There are several projects that explore video browsing by direct manipulation of the video, treating the video as content to manipulate, rather than a stream of frames [Dragicevic et al., 2008, Goldman et al., 2008, Karrer et al., 2012, Kimber et al., 2007]. A commercial for the Pro X Fade, a cross-fader aimed at DJs, features a fanciful scenario where a DJ reaches down from the sky and "scratches" several pedestrians and cars near a roundabout[5]. Peker and Divakaran "measure...spatio-temporal activity or visual complexity of a video segment" for the purposes of varying the speed of high-speed playback based on content [Peker and Divakaran, 2004]. Cheng et al. take a similar approach but with pre-defined types of "semantic events" [Cheng et al., 2009]. Some systems explore hierarchical browsing of video, where segments can be further browsed into sub-segments, often by keyframe, where selecting a keyframe shows several sub-keyframes [Lee et al., 2000, Ramos and Balakrishnan, 2003, Sull et al., 2001, Zhang et al., 1995].

Video Processing

Foreground extraction is a classical image processing problem, the usefulness of which here is to flexibly use foreground and background during video manipulation. Wang et al. had users indicate a foreground object by a novel painting interface, and that objected is tracked through space and time[Wang et al., 2005b]. Bai et al. achieve robust foreground extraction in video using local classifiers without interactive input [Bai et al., 2009]. Ballan et al. attempt to make a 3D rendering of a scene from several

[4] http://vimeo.com/2904371

[5] http://www.YouTube.com/watch?v=23Sd4eJBJgY

simultaneous "casually" captured video, and in doing so discuss foreground extraction in detail [Ballan et al., 2010].

Kang et al. condense moments across a long video temporally as much as possible through temporal and spatial rearrangement, yielding a "video montage" — they diagram the resultant video montage as a composite "video volume" [Kang et al., 2006]. Others do the same, except without spatial rearrangement [Pritch et al., 2008, Rav-Acha et al., 2006]. Teodosia and Bender summarize a video into the same dimensions as a single frame [Teodosio and Bender, 2005], while Assa et al. layer frames with slight displacement so they appear like a panorama [Assa et al., 2005]. Goldman et al. uses automatic arrows to indicate motion in the video, inspired by traditional storyboarding techniques [Goldman et al., 2006]. Truong and Venkatesh represent a good review work on the topic of video skimming, which is distinct from video montage and synopsis as it seeks to discard uninteresting parts of the video [Truong and Venkatesh, 2007]. Correa and Ma create interactive *Video Narratives* using foreground extraction, among other techniques [Correa and Ma, 2010].

3.5.3 Automation of Editing and Presentation of Video

Some systems take a video as input, and automatically manipulate it for the purpose of external presentation, with no or minimal user fine-tuning. We explore *automatic editing* from a larger source video to a smaller edited video that is supposed to represent the source video. We follow with *automated presentation*, where a body of video is used to create something more. We finish with a short summary of techniques that use a hybrid of automatic and manual editing.

Automatic Editing

Zsombori et al. generate video narratives from UGC (User Generated Content), using the notion of a "library" of video and a novel notion called "Narrative Structure Language" [Zsombori et al., 2011]. Yip et al. create a system to automatically edit video, focusing specifically on home videos, noting that "home videos tend to be very long and boring to watch" and "the average home videographer does not have the time, or the editing skills to edit their home videos" [Yip et al., 2003]. Wang et al. present a system to create an automatic video output in the genre of "sports music video", using audio, video and text feature analysis [Wang et al., 2005a]. Bocconi examines "semantic-aware" video editing, providing a technique to extract sequences about a user-selected topic from a documentary film [Bocconi, 2004].

In the domain of combining multiple videos, Shrestha et al. present a technique for combining multiple (typically poorly-shot) amateur videos from a music concert [Shrestha et al., 2010]. Tompkin et al. demonstrate a system called "Videoscape, a graph whose edges are video clips and whose nodes are portals between clips" [Tompkin et al., 2012]. Videoscape takes, as input, a series of geographically-located videos in an urban setting. A user can indicate a path throughout the urban setting, and the system will create a "video tour", smoothly warping from one video to the next.

Arvid Engström has done research on combining several mobile novice-recorded videos together, including building interfaces and studies of their use. In one case, a human "director" is designated to manage all the input streams from various people [Engström et al., 2008, 2012a,b]. MoViMash is another

such interface, focusing on recordings of live performance events [Saini et al., 2012].

There are a few consumer products for mobile devices that do "automatic" video editing. Vyclone is marketed as: "Now you can mix film taken on your iPhone with footage taken by other people filming the same events. Just shoot something with your friends; Vyclone does the rest. In a few moments it synchronizes and edits everyone's clips to create one movie with all the angles cut together" [Vyclone]. Vyclone appears to automatically choose the best view among the various inputs, though the user may manually override it. Magisto markets itself as "I hate editing! That's why we made it automatic!"[Magisto]. To use Magisto, one uploads a few minutes of video and selects a soundtrack. Magisto uses "artificial intelligence" and selects the "best" parts of your video to align with the soundtrack.

Automatic Presentation

Many systems use a video as "source" material to synthesize something new, distinct from a simple edited summarizations of the video.

Video Textures explored the ability for a system to play input video frames out of the recorded order, to create the perception of infinitely varying movement, in one example with a swimming fish that did not appear to loop. They used a heuristic to measure perceptual frame similarity so as to "play frames out of the original order only at places where it is unnoticeable for the viewer" [Schödl et al., 2000]. Notions of similarity, under the terms "saliency maps" and "attention models", have been used to choose which areas of a video to cull when retargeting (i.e., resizing appropriately), also including a knowledge of frame-to-frame relationships [Rubinstein et al., 2009, Wang et al., 2011].

Xu et al. create a 3D "Video-based character", from input of a colour multiple camera view video of a real human's movement [Xu et al., 2011]. This input is blended together and mapped to an animated character skin, able to create the appearance of a "real" character exhibiting new movement.

For our work, we are particularly interested in capturing a performers' movements in 3D space with both 2D colour and some other tool (such as a depth camera), and re-projecting it into the space that appears to be somehow faithful to the original, from the perspective of an observer in a theatrical setting. We have not found any work that focuses on that specifically. Zitnick et al. take input from several video cameras and can synthesize video that is apparently from a new 3D viewpoint [Zitnick et al., 2004].

Semi-Automatic Techniques for Video Editing

We are interested in quick and messy video editing, and so systems that automate some of the activities found in traditional editing, where the output is a finished video, are useful. LazyCut is a system for "content-aware, semi-automatic video authoring", with an emphasis on speed [Hua et al., 2005]. Videotater uses a "veridical representation" of the input video to suggest where video segments should be placed, as well as supporting automatic video segmentation (in time, not space) [Diakopoulos and Essa, 2006].

DemoCut is a system for generating instructional videos, where the source video is a single take from the same angle [Chi et al., 2013]. To grab a section of video, users place a marker in the video, and

the system analyzes the video using a "decision pipeline" to segment video into meaningful regions. Segments are detected by:

1. Frame similarity, relative to the marked frame
2. Non-silent sections (using adaptive loudness threshold)
3. Segment growing/merging algorithm (based on an audio analysis to find speaking/non-speaking regions)

3.5.4 Video Manipulation: Editing

We refer to video editing as the act of cutting and joining pieces of one or more sources together in time to make one edited movie [Okun and Zwerman, 2010]. Effectively collecting video segments of interest is made difficult by the current crude nature of the play/pause status of cameras — they can either be recording, or not. Vine [Vine] allows for a more fluid time control, where the video only records when the user is touching their finger on the screen, functioning as a quasimode [Raskin, 2000]. In the status quo and with Vine, the captured videos are immutable in that recording applications usually do not support editing, which is therefore performed off-line.

Fong et al. aim to support "casual" video authoring, by having the system make more prominent sections of video that have been viewed in the past [Fong et al., 2014].

Several techniques support fully automated video editing based on content analysis of the footage, such as the sound track [Magisto, Shrestha et al., 2010, Vyclone], or by leveraging meta-data captured during recording, such as geographical location [Tompkin et al., 2012]. All of these works reinforce the idea that video editing is usually tedious and painstaking, especially when performed separately from and long after the actual capturing. Fully automating the editing process eliminates this problem but is lacking in flexibility and control.

In contrast to prior work, we propose a seamless integration of manual editing capabilities with video capture, enabling quick cutting and slicing of segments of footage on set, in the midst of recording the scene.

3.5.5 Video Manipulation: Compositing

In contrast to editing, video compositing refers to the assembling of video segments together in space to make one composite video, mainly by combining different areas of each source frame [Okun and Zwerman, 2010].

Live compositing has been addressed for still photographs, with projects such as Group Shot [Group Shot], that allow the combination of several pictures of the same scene by rubbing to remove parts of the photo that are undesired. In a similar vein, Cinemagram [Cinemagram] allows users to create hybrid photo and video, instilling dynamics to still images, similar in the spirit of Cliplets [Joshi et al., 2012], but more instantaneous.

Live compositing of several video inputs, however, remains an open problem. Some works have explored compositing the present and the near-past together, where a person can directly compose themselves with their own shadow, played with a few seconds' delay [Snibbe, 2003]. Similar systems include Dancing with myself [Bartneck et al., 2009a], DELEM [Jimenez et al., 2005b] and Social Comic [Lapides et al., 2011]. These approaches involve specific settings, including a video projector or a green screen, and are limited to a single, pre-determined compositing style. We propose to extend such approaches to any video stream, by supporting simple yet rich compositing capabilities of the live stream with recently recorded videos, while providing instant feed-back of the result.

3.5.6 Live Interfaces

We discuss several relevant semi-live video management interfaces. There have been few studies of video editor use beyond usability evaluations. One such study is of the use of hand gestures (particularly indexical gestures) while co-ordinating the production of a live televised sport [Perry et al., 2009]. In this work, we aspire to have basic video editing controlled by free-hand whole body gestures. There has been at least one case of complex in-air gestures designed for video editing, TAMPER [Oblong Industries], though there we could find no formal documentation apart from demonstration videos [6]. The system and its gestures are similar to those appearing in g-stalt, the real-world implementation of the in-air gesture system seen in the film *Minority Report* [Zigelbaum et al., 2010]. Whether it is the intention of the performer of these gestures in the demonstration videos or not, the gestures appear theatrical, grandiose, as if attempting to exude cool. We want observers of these systems to be interested in the results of these interactions, not blown away by the cool-ness of the person performing them.

Progress in digital technologies has yielded a new form of movie pre-production: virtual production enables filmmakers to interactively visualize and explore digital scenes using CGI pre-visualizations [Autodesk, 2009]. Such techniques afford a visually dynamic, non-linear workflow, blurring the barriers between planning, capturing and editing. We wish to achieve something similar to this process, but with live streaming video.

Consumer-Oriented Interfaces

There are several relevant casual-oriented interfaces to help with image and video production that have arisen due to the mass ownership of camera-equipped smartphones.

For still photographs, Group Shot for iOS takes, as input, several photos of the same scene from the same camera, such as a family photograph. A composite photo can be made by rubbing to remove parts of photos that are undesired, such as family members blinking [Group Shot].

For video on the order of seconds, Cinemagram creates a hybrid photo and video, where small portions of the photo are moving [Cinemagram], similar to but more casual than Cliplets [Joshi et al., 2012]. Vine is an app for 6-second videos released by Twitter, known for small-size content - video recording is toggled by holding a finger on the screen of the recording phone [Vine].

For video on the order of minutes, YouTube Capture is an interface designed by Google to quickly

[6] `http://www.YouTube.com/watch?v=ULDEDwAJDlE`

upload videos to YouTube. Upload is asynchronous, only after the video is fully recorded. Basic editing features are offered [YouTube Capture]. Game Your Video is an iOS app that allows real time editing during playback; after a session of live editing is recording, this can be exported as a modified video. The layout of Game Your Video has a comic strip storyboard view underneath a "live" view of the video being recorded [Global Delight Technologies].

In terms of professional interfaces for theatre or installation, Isadora appears to be the most widely used [Coniglio]. Isadora has a visual programming language interface similar to Max/MSP, which is easy to use for the novice programmer but lacks in customization. Its interface is composed of a series of "scene" cues which an offstage technician may advance during the show, similar to lighting cues.

VJ Input and Control

In this section, we will talk about interfaces tailored towards VJs (an acronym for "video jockey"). Jonathan Hook is a notable expert in studying the expression of and designing for VJs [Hook and Olivier, 2010, Hook et al., 2009, 2011]. Salter provides a high-level description of the VJing that occurs in lucrative music clubs, outside the context of art galleries [Salter, 2010, p.173-179]. We are primarily interested in the low-level interactions between multiple videos or video and a live person mixed together. We have not done a full survey of the corpus of video work that has been labelled "VJing", but so far it seems interested in distortions and filters of a single rectangular video frame, rather than combining multiple videos in some content-aware way. Turco does have a discussion of two VJing performances in a club environment, and their relevance towards the notion of Intermediality [Turco, 2010]. Cyriak is one video artist who does interesting work by taking several video clips and looping them into one [Cyriak].

Several consumer-level VJ applications exist. Vjay is a live video editing app for iPad, using all pre-existing media on the device. Recording new video clips from the camera is also supported. Users can start recording output and then start editing; they can choose between several clips, apply filters and transitions, as well as scratch back and forth through the clips. Scratch pads can even be used as input to control playback [Vjay]. In Colorcode VJ, the user can load, mix and output VJ files. New videos and images can be added while playing, though the app has two separate modes: Play and Edit [Teknika]. LiVES mixes realtime video performance and non-linear editing in one professional quality application. It will let you start editing and making video right away, without having to worry about formats, frame sizes, or framerates. It is a very flexible tool which is used by both professional VJs and video editors — you can mix and switch clips from the keyboard, use dozens of realtime effects, trim and edit clips in the clip editor, and bring them together using the multitrack timeline. You can even record your performance in real time, and then edit it further or render it straight away [LiVES]. Another VJ tool is Resolume [Graphics].

There are a few research-level projects creating VJ interfaces: [Taylor et al., 2009] and [Hook and Olivier, 2010].

Playful Expressive Video Interfaces

A survey of work on the use of live captured and displayed video interfaces does not reveal much sophistication. Many of these instances are very compelling and playful, but do not give the performers or spectators much control over the output.

Social Comics is presented as a "casual game", allowing players to act in short comic strips they create. The authors argue that the game combines elements of sociability, physicality and authoring. The "players" have 20 seconds to pose for each frame, while being able to see a mirror image of themselves. Players can be inspired by a variety of physical props and two speech bubbles shown on each frame, with the text prepared a priori by the game designer. The final result is a multi-frame comic strip [Lapides et al., 2011].

A few interfaces capture short segments of user performance and use them as loops. Snibbe captures the silhouette of a person in front of a screen, and presents it back as a short loop, contemplating the separation of the shadow from the body, reminiscent of Peter Pan [Snibbe, 2003]. Bartneck et al. present *Dancing with myself*, a system for creating what they call an *Interactive Visual Canon, IVisualCanon*. A camera and projector are pointed so they view the same white wall. If a live body is captured standing in front of the wall by the camera, a life-size version of it can be easily projected back on the wall. The Interactive Visual Canon Platform works by re-projecting what it sees with a time delay, in this case time-delayed by a few seconds. Performances can be easily created by a single performer this way[7]. DELEM - Delayed Mirror explores the aesthetic effect of an 8-second delay in what is otherwise a conventional mirror. [Jimenez et al., 2005b]. The Looking Glass records moments using the colour and depth camera in the Microsoft Kinect, and visually merges them with the present moment, where the z-depth determines whether the present or the past is shown [Aseniero and Sharlin, 2011].

Vaucelle and Ishii demonstrate a video storytelling interface for children, where several toys have attached cameras [Vaucelle and Ishii, 2009]. Video recording is toggled by physical gestures. The authors delineate separate video-making activities of rehearsing, recording, and playback. Ryokai et al. explored instantaneous re-use of captured video for artistic purposes in I/O Brush [Ryokai et al., 2005, 2007]. A physical brush records images of real-world texture, and allows users to "paint" these textures onto a special canvas. These are targeted for production of abstract artworks and do not support video authoring in the traditional way.

3.6 Whole-Body Interaction During Performance

Whole-Body Interaction is widely studied, but it still is not an everyday experience for most people. In our case, Whole-Body Interaction during a theatre performance has an interesting set of features:

- all movements are part of a definite performance
- the gesture-ers are (initially) experienced theatre performers
- the primary activity of the performers is to *act*, not interact with the interface.

[7] `http://www.YouTube.com/watch?v=1UcDiPv2pF4`

We must recognize that interacting with an interface in the *magic circle* of theatre, in a non-public space, in the context of a performance, even if one is not a performer, is of a different character than interacting with a system in an everyday public space where one can be seen. Designing gestures to interact with a system during a performance is certainly interesting in terms of human perception — there are several solutions discussed previously in the *Coordinating Gestures Used in Modern Improvisation* section. It is also an interesting problem to detect these gestures in the noisy context of physical performance. For our purposes when examining detection, we will take the system's point-of-view, and refer to gestures intended for the system as foreground activity and everything else as background activity.

We describe prior work on *Performativity and Audience Perception* of interaction, then discuss the problem of intermixing interaction with performance in *Foreground versus Background Activity*, and we explore prior work on *Explicit Input with Gestures*.

3.6.1 Performativity and Audience Perception

HCI literature refers to anyone interacting with a system as a "performer", and anyone watching them, whether intentionally or accidentally, as a "spectator". The study of interaction while being watched is covered under the term *performative interaction*. These roles are much more specific during a theatrical performance — Reeves et al. refers to these as "staged" performances, as opposed to any performance we participate in when we are conscious of our own behaviours [Reeves et al., 2005]. It is common in the study of performative interaction to question the social acceptability of the users' actions, but in this case it does not apply as it is normally formulated [Rico, 2010]. While we anticipate that performers may feel slightly odd performing the gestures, their experience will be significantly different than the typical user experience.

Our goals are to have usable whole-body interaction without the spectators being distracted by the novelty of the system, but rather enjoying what it produces. A good metaphor would be the invention and display of a new paintbrush - we do not want observers to be distracted by the magic of the paintbrush itself, but rather be able to observe its use, and understand it for use themselves. Brenda Laurel has a different desire, preferring that the computer be a form, not simply a tool, for theatre [Laurel, 1991]. There have been a few theoretical examinations of the spectator's perception of novelty [Dix et al., 2006, Dixon, 2007, Reeves et al., 2005], but we could not find any that examined it ethnographically except in one case [Friederichs-Büttner et al., 2012]. Dix et al. note that the traditional value in Human-Computer Interaction of efficiency does not necessarily apply in interfaces that are meant to be performative [Dix et al., 2006].

From the pure arts perspective, one topic that is relevant is the role of *virtuosity* in the performing arts, discussed extensively by Royce: "Mastery of technique such that performance is effortless is something that audiences recognize even if they are unable to articulate it ... the essential ingredient is nonchalance" [Royce, 2004]. By Royce's definition, virtuosity is a high level of mastery of skill. If we briefly return to the language used in Human-Computer Interaction studies, someone who is highly skilled at operating a system is a user of high virtuosity. If that system is the user's body, and they are doing a ballet performance, then they have a high degree of virtuosity in ballet. Interestingly, virtuoso performance appears often in theatrical improvisation, where the audience is explicitly aware of the difficulty

of the performer's task, and their struggle to do it without "screwing up" is part of the performance. Two such games are *Questions Only* and *Number of Words*.

For our case, we may seek to understand the perception of skill by the audience — a seemingly highly skill use of the system may become the performance if it is impressive enough, and this is, in fact, when we want to avoid in this work. In Royce's survey of artists' perspectives on virtuosity, it is generally agreed that focusing solely on being high skilled at techniques in your own artistic field does not make for good art. Virtuosity and artistry are different. Royce says "The aesthetics of the performing arts is composed of two parts: virtuosity and artistry." Virtuosity is technique and artistry is style. Technique has a well-defined codified vocabulary, and is conservative, whereas style has a metaphorical vocabulary, and is innovative. The purpose of the interface in this work is to allow the performer augment the performance.

Reeves et al. ask *"How should a spectator experience a user's interaction with a computer?"* and conceptualizes performative interaction in terms of *manipulations* and *effects* [Reeves et al., 2005]. Manipulations may be exaggerated or subdued by the performer, due to the knowledge that spectators are observing them. Manipulations may reveal information about the system to the spectator, whether they are detectable by the system or not, such as dramatically donning a brainwave-sensing helmet. Effects are the results of performer manipulations. [Reeves et al., 2005] notes 4 design strategies for performative interaction:

- secretive (manipulation and effects hidden),
- expressive (manipulation and effects visible)
- magical (manipulation hidden, effect visible)
- suspenseful (manipulation visible, effect hidden)

If the secretive design pattern is used as part of an art piece, one can imagine that part of the spectator experience of the art is to eventually recognize that some of the effects are being caused by the performer. The magical or suspenseful design patterns seem to desire to create a sense of wonder, or a commentary on technology. For the purposes of this project, we are interested in the expressive design pattern. However, there are values beyond allowing the spectator to recognize the link between cause and effect. In Reeves et al.'s writing, they treat the interaction between the system and the performer as the primary focus of the performance, as opposed to the means to an end. Dixon also discusses the problematic issue of "invisibility" of interaction when the link between performer manipulation and effect is not clear - to the spectator, it may appear that the system is not interactive at all [Dixon, 2007]. Reeves et al. note that there is a special moment in systems where the spectators may approach and interact with the system, becoming temporary performers. One technique to introduce temporary performers to the system is what Reeves et al. call a "ritual" - a theatricalization of the procedure of introducing any new user to a system. Some of the other performativity literature is concerned with novice users of a system approaching an interface in a public space and being concerned with how they are perceived [Hansen et al., 2011].

The 80 minute interdisciplinary play *Parcival XX-XI* features digital media and six dancers [Friederichs-Büttner et al., 2012]. After an introduction of the play, audience involvement is required to advance the

show: *"It is designed to require active involvement on stage of several of the visitors...In this context, we define our play to be (dramaturgically) unfinished if the interaction we designed for does not unfold."* The authors interviewed the audience after the show and found four motivations for participating (or not) in the play: *fun*, *frustration*, *fear*, and *schadenfreude*[8]. Interestingly, some audience members viewed other audience members' participation in the play as a disruption to the performers' activity.

Evidently, care must be taken to make it clear to the audience that the interactive system is live, not pre-recorded, and the audience must be carefully encouraged to participate, and feel that they are improving the performance by doing so.

3.6.2 Foreground vs. Background Activity

To study intermixing interaction (foreground activity) with noisy behaviour (background activity), we will end up collecting a large amount of body movement data. This coverage of related work shall reflect that.

Separating natural movements from explicit input actions is analogous to separating background from foreground in computer vision; thus, we call naturally occurring movement *whole-body background activity*. A common strategy to separate explicit actions is to use exaggerated gestures unlikely to occur unintentionally [Cohn et al., 2012, Song et al., 2012]. However, this assumes a deep understanding of background activity in order to design the gestures and evaluate their performance. Indeed, a common approach in computer vision is to model the background, and suspect that anything not fitting the model is foreground [Parks and Fels, 2008]. For device gestures, researchers have logged device-sensor background activity in real environments when carrying a smartphone [Ruiz and Li, 2011] and writing with a pen [Grossman et al., 2006]. Although private and purpose-built for each project, these logs demonstrate the potential for building a shared dataset of whole-body background activity.

Large datasets of natural occurrences are useful for conducting post hoc observational enquiries, modelling phenomena, motivating technique designs, training algorithms, testing individual techniques, and comparing multiple techniques with a common baseline. Examples of well established public datasets include the MNIST handwritten digit database [LeCun et al., 1998] for handwriting recognition, the MacKenzie Phrase Set [MacKenzie and Soukoreff, 2003] to evaluate text entry techniques, and datasets of static objects captured by depth cameras [Janoch et al., 2013, Lai et al., 2011] for computer graphics algorithms. In the field of gesture recognition, algorithms are trained and tested using datasets like Marcel's compilation of hand gesture and posture images [Marcel], and the Cambridge Gesture Database of image sequences showing various hand motions [Kim et al., 2007]. The Chalearn dataset of hand motion and postures, captured by Kinect RGB and depth cameras, was even the basis for a gesture recognition competition [Guyon et al., 2012].

There are several examples of whole-body capture datasets, but these focus primarily on short sequences of high-energy actions performed by actors in a motion capture studio. The CMU Graphics Lab maintains a collection of human activity motion capture [CMU Graphics Lab Motion Capture Database]. These are short segments of predominantly active motions, such as locomotion and sports, but also segments of "common behaviours and expressions" closer to what we consider background ac-

[8] pleasure derived by someone from another person's misfortune.

tivity. However, these short sequences, performed by actors in full motion capture suits in a cavernous capture studio, are typically less than a minute in duration. A similar example is the Berkeley MHAD [Ofli et al., 2013], a database of high-energy motions performed by actors, which includes several modes of capture including the Kinect. Shotton et al. initially used the CMU database to train the Microsoft Kinect SDK pose estimation algorithm, but later created their own unpublished database of game-like activities such as "driving, dancing, kicking, running, navigating menus, etc." [Shotton et al., 2013].

More recently, the CMU Quality of Life Technology Centre created a multimodal capture database of people cooking in a simulated kitchen [CMU Kitchen Motion Capture Database]. The motivation for this dataset is to improve activity recognition using limited sensors. This dataset contains sequences longer than those described above (but typically less than 5 minutes), and were performed by an actor in a full motion capture suit wearing a bulky tethered backpack to support multimodal sensors. Currently, motion capture data is only available for 4 out of 200 sequences. To achieve such a diverse set of data, concessions were made with regard to intrusiveness and comfort. As background data, these activities are still too short and too focused on a specific task of a cooking sequence.

In contrast to these pre-existing datasets, our emphasis is on obtaining much longer sequences with minimally invasive capture equipment and encouraging a high degree of social interaction and comfort. Rather than clean, segmented sequences of distinct actions, we want realistic, noisy, everyday actions. Unlike previous datasets, we require a controlled mixture of background activity and explicit input sequences for baseline testing.

Detecting gestures in a continuous stream of input is known as the Gesture Spotting Problem. A common approach is to model each gesture type as a Hidden Markov Model (HMM) and detect gestures when their likelihood exceeds that of a synthesized threshold HMM [Lee and Kim, 1999]. The threshold HMM is simply a fully connected combination of all states taken from the trained gesture HMMs. In principle, it represents all non-gesture movements and is sometimes referred to as a background model [Fourney, 2009]. The problem is that this "background model" is not only synthetic, but composed of foreground states. Lee and Kim note: "it is not easy...to obtain the set of non-gesture training patterns because an almost infinite number of meaningless motions can be obtained."

An alternate approach is to design a gesture delimiter, which rarely occurs naturally. For pen input, Grossman et al. logged naturally occurring pen hover motion to design distinct pen hover gestures [Grossman et al., 2006]. For device motion gestures, Ruiz and Li gathered naturally occurring motion data to design and test the distinct DoubleFlip motion gesture delimiter [Ruiz and Li, 2011]. These projects demonstrate the power of using background activity data, but neither project released the dataset, or offered a generalizable methodology to capture and distribute the data.

While there are existing approaches to gesture recognition with always-available input, there is certainly no dataset that can be used to evaluate these approaches in the context of naturally-occurring whole-body background activity.

3.6.3 Explicit Input with Gestures

Using body gestures for explicit input has been extensively studied [Aggarwal and Ryoo, 2011, Poppe, 2010, Turaga et al., 2008, Weinland et al., 2011]. With always-available body input, the difference be-

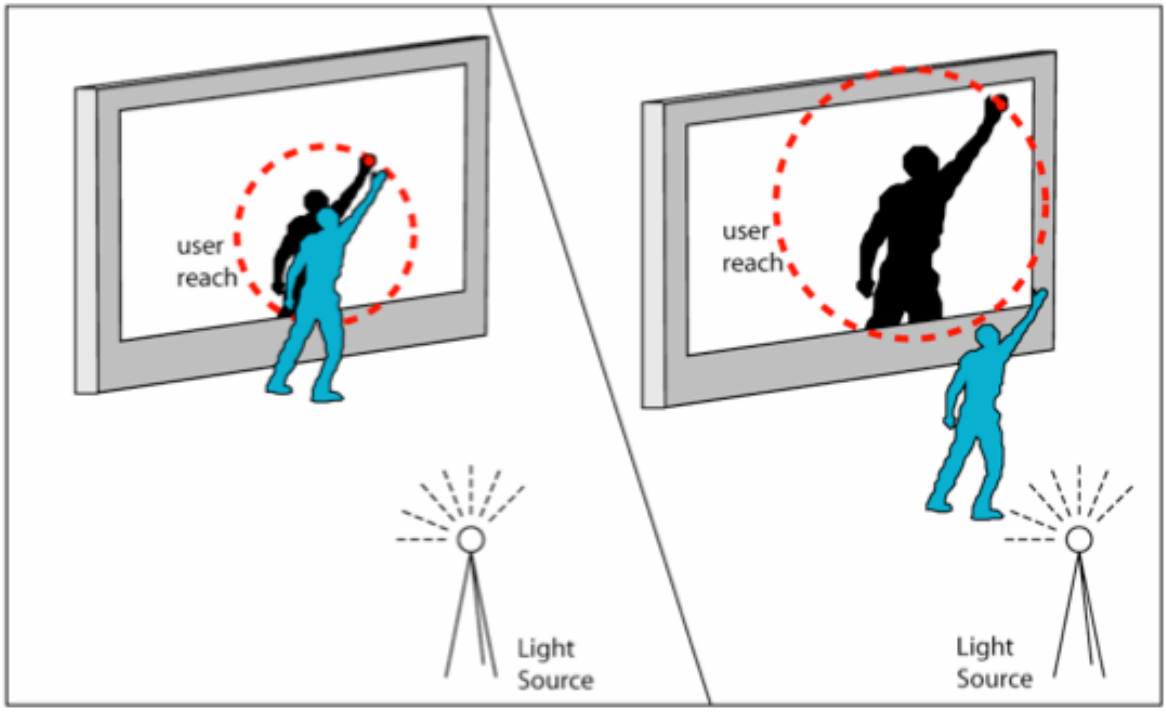

Figure 3.4: Shadow Reaching: where distance from the screen is used to define the scale of the interacting silhouette.

tween gesture and non-gesture can be subtle, introducing false positives [Lei et al., 2012, Panger, 2012].

Shadow Reaching explored used a real-world shadow as feedback, with a Polhemus position tracker held in the active hand as cursor input [Shoemaker et al., 2007] (see Figure 3.4).

We have considered using an ad-hoc user-defined gesture set during the performance to define references to scenes. It has been found that User-Defined gesture sets may be more memorable than those defined in advance [Nacenta et al., 2013].

Shoemaker et al. attach widgets to different parts of the user's silhouette [Shoemaker et al., 2010]. They argue that it is easy for people to process notions of space around their shadow, as their sense of personal space extends to it, and thus it is equivalent to interacting with parts of one's own body. Notably, a camera is not used to calculate the silhouette, and instead the shadow shown is from a generic 3D model. This means that virtual light sources that cast the shadow may be individually adjusted, though this would add unnecessary complexity in our case.

YouMove [Anderson et al., 2013] has a similar setup to ours, a screen with life-size projection, but the aim is for body movement teaching. Similar to our system, recording start and stop is accomplished by a dwell button. Unlike our system, editing appears to be accomplished using a WIMP interface. To provide a score on how much the user's pose matches the training pose, a skeleton-scaled spatial alignment algorithm is used, followed by Euclidean distance.

4

LACES: Live Authoring through Compositing and Editing of Streaming Video

Video authoring activity typically consists of three phases: planning (pre-production), capture (production) and processing (post-production). The status quo is that these phases occur separately, with the latter two having a significant amount of "slack time", where the camera operator is watching the scene unfold during capture, and the editor is re-watching and navigating through recorded footage during post-production. While this process is well suited to creating polished or professional video, video clips produced by casual video makers as seen in online forums could benefit from some editing without the overhead of current authoring tools. This chapter introduces LACES, a tablet-based system enabling simple video manipulations in the midst of filming. Seamless in-situ integration of video capture and manipulation forms a novel workflow, allowing for greater spontaneity and exploration in video creation.

Ever since consumer camcorders were introduced to the mass market, video has become a powerful way of capturing and sharing personal experiences. The current pervasiveness of recording devices and social media sites such as YouTube and Facebook has further increased casual video creation and distribution. However, unlike still photography, for which there are reasonable tools for quick, convenient editing, video manipulation typically requires a tedious and cumbersome offline process with tools that are often too complex or unsuited for casual video. This mismatch between the ease with

which video can be captured and the difficulty of making edits is evident in the rawness of much of the videos posted in online forums today.

Whether documenting an event, performing surveillance, or recording takes for a scene of a film, the ratio of captured video to useful content is always high. During editing, a great deal of off-line time is devoted to sorting through the corpus of video content to recall, analyze and make cuts. On the other hand, there is often a fair amount of down-time for the camera operator during capture while passively watching the scene unfold. We refer to these two periods of low user activity as "slack time", and explore how overlapping production with pre- and post-production can make video authoring a more spontaneous and less time-consuming experience. Applying this strategy, we propose the seamless integration of video capture and manipulation operations into the same phase of a fluid workflow to make the authoring of videos more spontaneous and accessible.

We start with a few motivating scenarios of modern casual video production which we feel are underserved by current techniques, and we follow with a review of related work. We next examine limitations and issues in the traditional video production workflow, and discuss how the live workflow with LACES addresses these shortcomings, as well as the associated new challenges and opportunities it introduces. We then describe the LACES system and present several use cases brought forth by users in an informal evaluation.

4.1 Motivating Scenarios

Below are three scenarios where we feel traditional video tools are lacking. These examples target casual video makers collecting footage to be post-processed off-line. This footage can have a range of uses from reviewing and extracting information to producing video creations, which extend beyond short single-clip segments. We aim to demonstrate that the realization of such scenarios should be possible with low overhead.

4.1.1 Scenario 1. Curating Content

Isa is a parkour artist visiting Toronto. At High Park, she encounters another parkour group with a unique style. Isa has a social media following, and wants to upload a highlights compilation of their stunts from her phone. She starts recording the group and collects a series of clips, each focusing on different members as they make numerous attempts at each trick. Even after catching a good stunt, Isa continues recording through breaks and failed attempts to avoid missing anything. She ends up with half an hour of footage which she will review and patch together offline.

4.1.2 Scenario 2. Annotating Content

Jane is a primatologist; Taz is her assistant. They are examining how chimpanzees grasp objects while completing puzzles. Jane is standing behind a camera, and Taz sits across from a chimpanzee. Taz presents the puzzles to the chimpanzee, and helps prompt them when they get stuck or distracted. Jane's goal is to get the timestamp of the beginning and end of each grasping activity, for further analysis

by herself as well as others. This is difficult to do precisely — she writes down the timestamp that she thinks is nearest to the start of a grasping activity. She will have to review the video later to refine these timestamps.

4.1.3 Scenario 3. Coordinating Content

Jill and Pradeep are high school students working on a film. The film will have a shot of Jill, dressed as a gorilla, climbing Toronto's CN Tower. Pradeep goes out and films the tower from an arbitrary vantage point on a windy day. Later, they film Jill making climbing motions in a studio. These two videos get passed to an editor, Cheryl, and while compositing, she finds it too hard to line up Jill's motions with the tower. She also points out that it would be nice to have a shot of Jill grabbing the top of the tower from just the right angle. Pradeep is sent out to film the tower from a different angle, and Jill records a few more takes, hoping that Cheryl can make it work in editing.

In the above scenarios, we see opportunities to improve workflows by integrating video capture and post-processing. This would provide a means for on-the-fly adjustments and spontaneity in the video making process.

4.2 Traditional Video Production Workflow

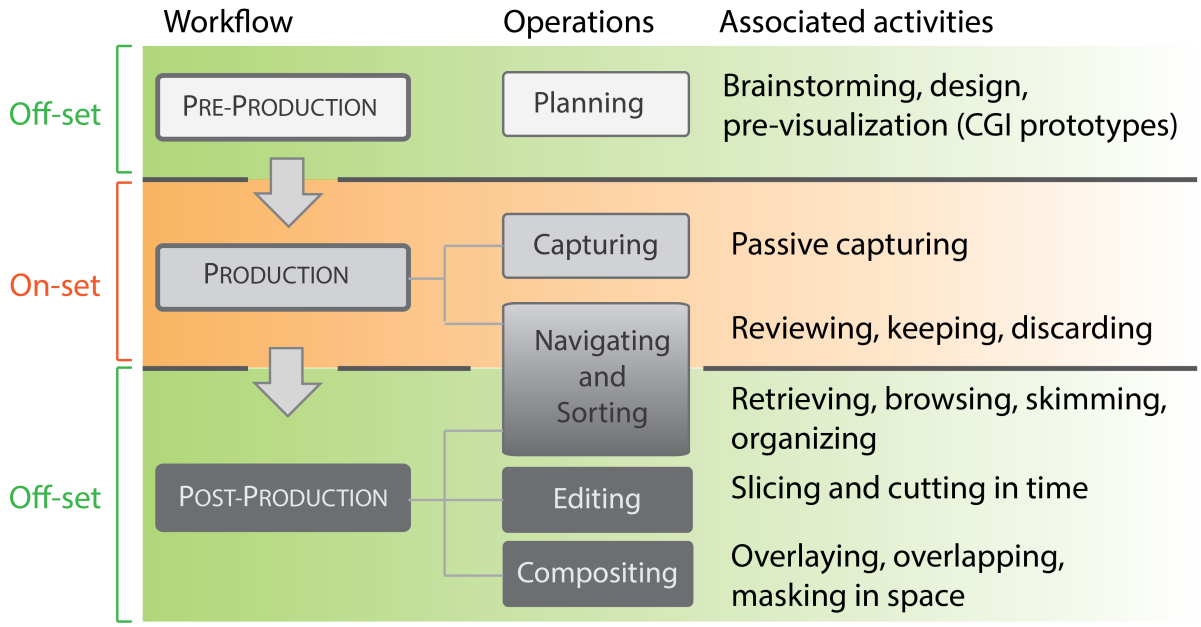

Figure 4.1: Traditional video production workflow.

We first describe the traditional video production workflow and discuss the issues and challenges associated with it. We provide a broad workflow description here, encompassing both casual and professional video creation. We identify several limitations, and thus opportunities for exploration. In the subsequent section, we will introduce our proposed workflow within this unexplored space.

The traditional workflow for video production, shown in Figure 4.1, consists of 3 stages: pre-production, production, and post-production [Autodesk, 2009, Okun and Zwerman, 2010]. Pre- and post-production typically take place separately and with different tools, enforcing a linear, forwards-only process and leaving few opportunities for back-and-forth iteration between the different stages.

Pre-production concerns the planning of the video. In professional environments, it includes activities such as brainstorming, designing, and pre-visualizing. The outcomes can be storyboards and scripts, to be used for the controlled, planned capturing of the video segments. In the context of casual or spontaneous usage, planning is not as thorough as in professional filming. An example of this is illustrated in Scenario 3 above, where desired shots are roughly planned out prior to execution.

Production concerns capturing the raw files, which are either informed by planning, as in Scenario 3, or ad hoc recording of live moments, as in Scenario 1. User interaction during capture is mainly passive, where the sole task of the operator is simply to observe the video as it is being shot. If a specific shot is needed, there may be many takes of a single scene before an adequate one is captured, possibly involving re-watching for verification.

Post-production concerns manipulation of the raw files to generate ready-to-share movies. As with planning, this stage may be bypassed completely for casual users that consider the raw camera data the final product. The level of interaction can then range from minimal clip alteration to extensive, professional production. Actions during this stage include navigation, editing, and compositing.

Video navigation is performed to review footage and locate key moments, in support of editing and compositing. We refer to video editing as manipulating video segments *temporally*. This involves slicing and removal of clips to isolate segments of interest, and assembling clips together. In contrast, video compositing concerns the manipulation of video segments *spatially*, where all or portions of the image from one clip are mixed or layered with those of other clips.

Editing and compositing are typically off-line processes, in both casual and professional cases. Tool support for video editing can vary in complexity from simple trimming of the start and end of each clip to joining multiple clips together with different transition effects. Video compositing tends to be an advanced practice, and tools supporting such actions are not generally targeted towards casual users. However, as shown in all three motivating scenarios, there can be a wide range of situations which require off-line processing.

4.2.1 Limitations and Opportunities

There are several limitations we perceive with the current workflow that highlight problem areas we seek to address in the design of our workflow.

Slack Time: In the traditional workflow, we observe that both production and post-production have "slack time" - periods of low-intensity user activity. Overlapping operations from production and post-production stages allows slack time during capture to be used more efficiently for performing editing and compositing. This makes video authoring a more spontaneous, flexible, and less time-consuming experience. For example, we see in Scenario 1 that there is opportunity for Isa to cut out footage of failed stunts while the group is not doing anything of interest.

Precision Timing: Desired editing operations are often reactive to precise events in video. For example, a user may want to add a bookmark to a clip when an exciting event occurs, but would not know to do this until the event has passed. Jane from Scenario 2 needs to review footage in order to achieve precision in event time markers. In another case, an interesting event may start before the user is able to turn the camera on - the operator intending to be parsimonious about recording to avoid having to sort through excess video content later.

Workflow Phase Separation: Since the toolsets for each phase of the traditional workflow are separate, a clear choice must be made when transitioning from one phase to another. In Scenario 3, for example, all footage must be captured on site first, and an off-line compositing process follows elsewhere. Since going back to shoot more footage when you have moved on to the post-production phase can be frustrating and costly [Okun and Zwerman, 2010], the ability to visualize raw video with some rough editing and compositing effects would be beneficial.

Coordination: In the final outcome of a traditional workflow, several different clips may contribute to a scene, whether alternating in time or composited together as with the CN tower scene in Scenario 3. While large production studios can afford to use multiple cameras, with one camera a scene must be recorded twice, inevitably with different timing due to real-world variation. The changes in timing must be later fixed by an editor.

4.3 LACES: A Fluid Workflow

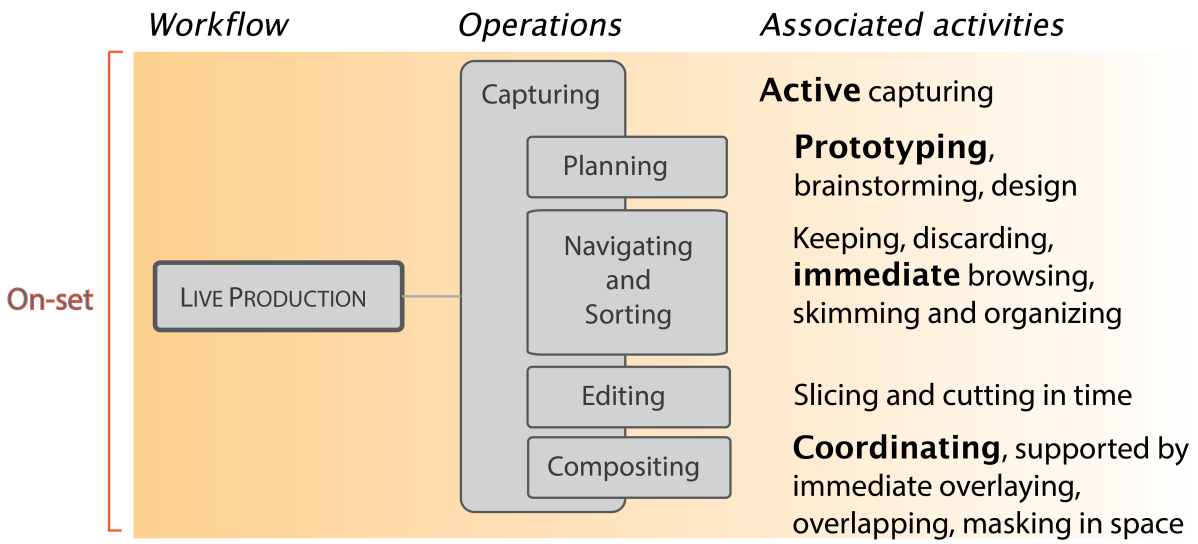

Figure 4.2: Our proposed live authoring workflow.

We propose a workflow which blurs the boundaries of the traditional phases of video production, by allowing the co-located, simultaneous and seamless operation of planning, capturing, navigating, and manipulating in the same, fluid, flexible workflow (Figure 4.2). Our fundamental question is: How can a user interact with a live video stream? The live workflow we propose addresses many issues we find with traditional tools. We will now present the new challenges it presents, as well as design goals for a system supporting it.

4.3.1 Challenges in Working with a Live Stream

Performing manipulation on a live stream while new video content is coming in presents several challenges.

Conflicting operations: Capturing and manipulating the same video are interfering operations a priori, and traditional video manipulation tools have expected video to be immutably stored in a file. While manipulating requires freeform temporal navigation, capturing will continually extend the file of interest — the meaning of "present time" will be constantly changing.

Catching up: Bezerianos et al. discuss the difficulty of interacting with changing content, and techniques to "catch up" when the user did not observe a visual change [Bezerianos et al., 2006]. Silva et al. present the Hold and Speed Up technique so users may apply annotations to the current frame of a live video [Silva et al., 2012]. For a user to be able to edit while also paying attention to capture, the system must allow these two activities to share focus effectively—neither one can consume the full attention of the user.

No preview mode: While the saved video output from the system could conceivably be sent through a traditional post-production workflow, for the purposes of our exploration, the output is live. This means that there will be decreased opportunity to fine-tune a manipulation before it is applied to the live video. However, we see this as a trade-off: the traditional workflow applies video manipulations separate and off-line, whereas our workflow aims to apply these manipulations at the time of filming.

4.4 The LACES System

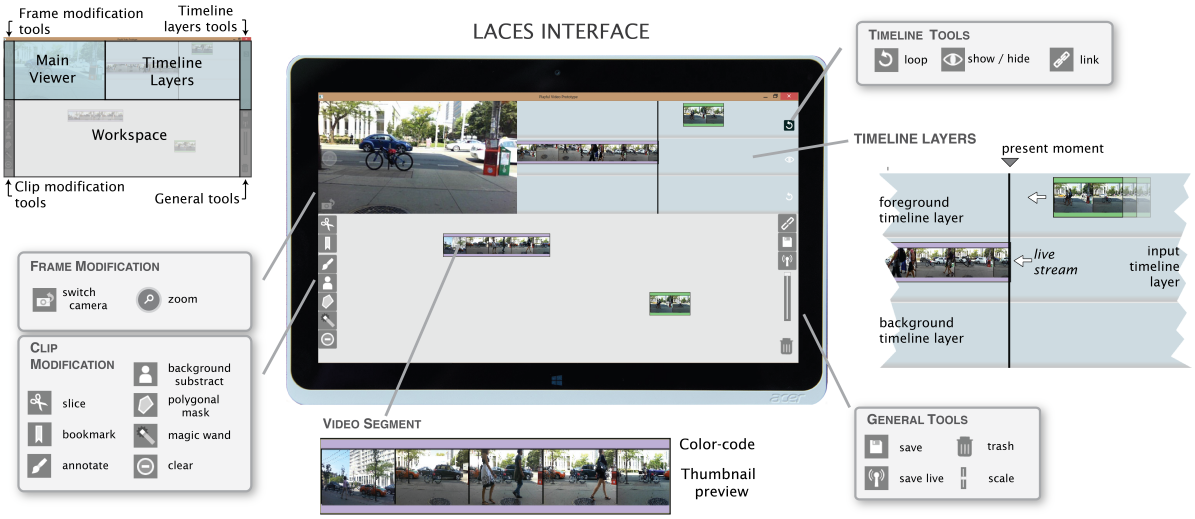

Figure 4.3: The LACES user interface, comprising the interactive main viewer (top left panel), timeline layers (top right) and workspace (bottom). Clip modifications can be performed through bi-manual interaction on the layers and side toolbar.

We implemented LACES, a live video editing and compositing system supporting our fluid, live workflow (Figure 4.3). The interface contains three main interactive components: the main viewer, displaying the current video frame, the timeline layers, a stack of timelines featuring the captured live stream

and additional layers for compositing, and the workspace, a library area where the user can save and arrange assorted video clips. We provide a set of tool palettes on both sides of the interface, for easy access while holding the tablet with two hands [Wagner et al., 2012].

4.4.1 Overview

LACES is characterized by the continuous recording of the input video from the tablet. As the user launches the application, the recording of the live stream automatically starts. As time progresses, the associated movie strip progressively builds up in the input timeline layer (Figure 4.4). Video clips on LACES timelines move right to left. In the middle of the timelines is a black vertical marker indicating the "present". The main viewer shows the real-time view of the camera. When passively capturing, the user can use LACES as a traditional recording device.

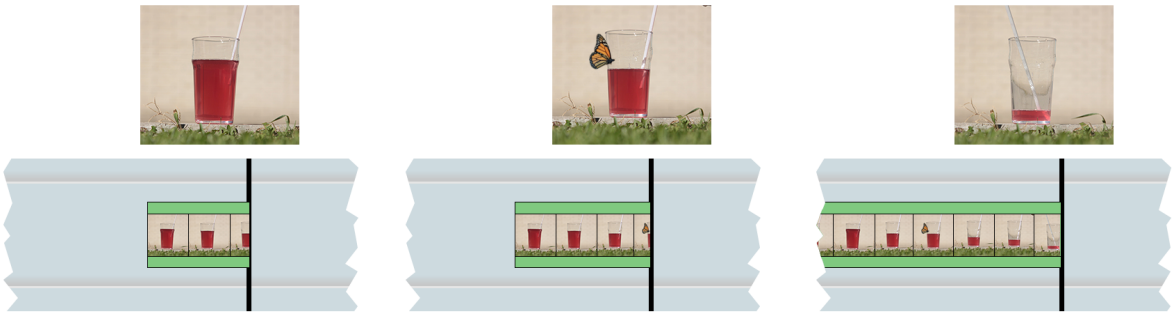

Figure 4.4: Capturing the live stream as seen in LACES. The main viewer (top picture) shows the real-time view of the camera. The recorded clip is visualized as a comic strip, progressively building up as time passes.

At any moment, the user can interact with the live stream and assorted clips in the workspace. Video clips can be dragged to and from the workspace or the timeline layers. There are 3 timeline layers, ordered from top-to-bottom as *foreground*, *input*, and *background* (see Figure 4.3). The foreground and background timeline layers start empty, while the system continually adds new frames from the camera to a clip on the input timeline layer.

Video can be sliced at the input and dragged to the workspace or to other layers, in which case the viewer outputs the blended view of all clips including the input overlapping at the present marker. The real-time input stream can be blended with other camera footage, displaced into the past, or even hidden from view to place focus on a pre-recorded video clip in one of the other layers.

LACES provides several features to users, with interactions that can be performed while capturing: clip control, frame editing, clip transform control, layer control and saving.

4.4.2 Clip Control

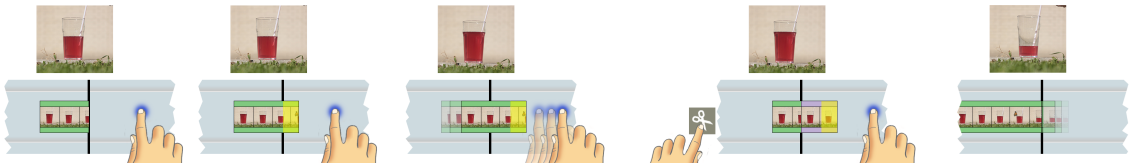

Figure 4.5: Manipulating clips on the input timeline layer. The user places a finger on the input timeline layer, which freezes the current frame (a); portions of the clip the user has not seen yet are shaded yellow (b). She scratches the clip into the past (c) then slices it at the desired frame (d). When she releases her finger, the input clip plays back to the present at an accelerated rate.

Slicing and Bookmarking

A typical feature of video editing is the chunking of video into clips that may be sliced and arranged into sequence. We provide *slice* and *bookmark* buttons. The slice button separates the current clip into two clips on either side of the slice, and the bookmark button adds a marker to that frame within the clip. By default, if the user presses the slice or bookmark button, the corresponding operation is applied to the current real-time frame in the input layer clip — cutting it into two segments, or placing a frame mark for future reference. Slices are coloured to provide additional visual cues for recall and organization.

Sorting

Clips can be dragged from the input to the workspace. We think of the input timeline layer as a queue of clips to be sorted. If not dealt with by dragging to the workspace, they move off the timeline to the left. However, removing a clip from the input closes the gap between its neighbouring clips, so all clips may eventually be retrieved.

Re-using

Clips can be dragged from the workspace, or even directly from the input timeline layer, to the foreground or background layer. The display of several clips underneath the present time marker is, by default, an alpha blend in the viewer. The meaning of foreground or background only comes into play with different blending styles, which we will discuss later. We support snapping when the ends or bookmarks of clips are dragged near those of other clips.

Navigating

Inspired by "scratching" as it appears in disc jockey (DJ) performance, the user can move both the background and foreground timeline layers left or right using their finger. If the user simply holds their finger still on a layer, it stops that layer from playing forwards and freezes it on a specific frame (Figure 4.5a-b). We refer to scratching as moving the medium while the play marker stays fixed, as in our interface or with a vinyl record and a needle. The term "scrubbing" is used when a user moves a

player marker while the medium remains fixed, such as with most digital video player interfaces. This is a subtle but important difference — especially if multiple media are playing and one marker indicates the present-time position in all of them, as with our interface.

We feel that it is important to support the typical video editing activities of slicing and bookmarking during capture. However, it is very difficult to anticipate the correct time to slice or bookmark a live video stream until after the key event has past. To this end, we support *scratching the input*.

To apply a slice or bookmark operation to a specific time in the recent past, the user places their finger on the input layer. By holding it still, the viewer's display will be frozen at the exact time the user began the scratch. The user can scratch the input layer right to left to tune the exact timing of the desired slice or bookmark, and then press the desired operation's button to apply it to the frame underneath the present time marker.

Since LACES is a live interface, the camera is still recording while the user is performing this operation. We provide a visualization of how new, un-viewed video builds up during this operation (Figure 4.5b). When the user releases their hold on the input timeline layer at the end of any operations they want to perform, the input layer plays at an accelerated rate to catch up to the real time display of video (Figure 5e). The high-speed play is a useful alternative to a sudden transition to real time, giving the user a summary of the video they missed while focusing on their editing operation [Bezerianos et al., 2006]. The user can slice or bookmark previously-recorded clips using a similar technique, with the operation button applying to whatever layer the user is currently touching, or to the input if none are currently touched. We find it possible to perform this bimanual operation without an obvious disruption in the filmed video [Goldman et al., 2006].

When a user drags a clip from the workspace to a layer, it is possible to lose that clip when it progresses off the timeline. Thus, references to each clip are maintained in the workspace. Excess clips can be removed from the workspace by dragging them to the recycle bin. Frames in each clip are visualized as a comic strip, with a default frequency of one frame every three seconds. With longer clips, this scale can get cumbersome, so a scale slider is provided to increase the amount of time each displayed frame represents.

Examining our first two motivating scenarios, these simple clip controls would allow Isa to discard uninteresting parts and, during slack times, recombine shots of cool stunts, for which she can add bookmarks, into a final highlights reel. Similarly, Jane would be able to review footage during capture by scrubbing back to determine precise timestamps corresponding the chimp grasping activities, eliminating the need to review all footage offline.

4.4.3 Frame Editing

Here we discuss options to control the input coming from the camera. First, note that we have a *camera flip* button that controls whether we use the tablet's back or front camera. Other frame edit controls are applied to frames as they come in from the input: *ink annotations*, and a variety of methods for creating. Annotations persist on the frame they are drawn over and on subsequent frames. *Masking operations* are also supported. The results from multiple different masks are merged into a final mask. If a clip has a mask and is blended with another layer, the mask replaces the default alpha blend with a direct pixel

overwrite.

To perform *background subtraction*, the user must ensure the camera's scene will be stable and clear of any foreground objects. As the user taps the button, LACES captures a background frame and immediately begins masking out the background from any foreground. We used a basic hue-based discriminant on a Gaussian-blurred image, followed by an erosion and dilation pass. While this occasionally worked sufficiently in a carefully prepared environment, it did not in real world examples.

To create a user-defined *polygonal mask*, the user taps the polygonal mask button, and then sketches the mask outline directly on the main viewer. All pixels not belonging to the mask are hidden. This is useful to grab a specific portion of a video that is not likely to move significantly.

The *magic wand* removes all pixels that closely match a given hue. To do this, the user taps the magic wand button and then taps a location on the viewer. The hue in a small region around the tap is averaged, and subsequent pixels that closely match that hue are masked out. This worked well with solid colours. In contrast to the user-defined polygonal mask, this suits objects that will change shape significantly, such as moving limbs and hands.

Multiple masks can be combined — for instance, a magic wand to remove all elements in the frame with a certain colour, and then a user-defined polygonal mask to remove the remainder. All annotations and masks can be removed from subsequent frames using the *clear* button.

4.4.4 Clip Transform Control

We provide traditional *panning and zooming* capabilities. This is particularly useful when we have multiple video clips playing simultaneously, and want to create a spatial relationship between them for compositing.

Pan and zoom operations are performed directly on the viewer. Similar to the slice and bookmark operations, pan and zoom transformations are applied to the input timeline layer by default, or any currently scratched timeline layer. These operations are keyed to the frame they are performed on, allowing the user to "act out" a pan and zoom sequence as they scratch through a video clip, and then replay the transformations at regular speed as many times as desired.

The user applies a pan by dragging their finger in either dimension on the viewer. While zoom is typically performed with two fingers on touch interfaces, in most of our scenarios the user is holding the interface with both hands and can only operate it with their thumbs [Wagner et al., 2012]. Thus, we provide a zoom handle on the left side of the viewer. Pulling this handle upwards or downwards increases or decreases the zoom of the current video clip. A tap on the handle resets the pan and zoom.

4.4.5 Layer Control

Each timeline layer can hold a collection of clips. In the input layer, incoming frames will always be added to the last clip. We provide a few simple layer controls.

The foreground and background layers have a *loop* toggle button. When the button is activated and the present marker reaches the end of a clip on that timeline layer, the timeline layer will shift back to the

start of that clip.

The input timeline layer has a *hide* toggle button. The user can use this to view previously-recorded video on the foreground or background layers, without it blending with the input layer. Input is still recorded even if it is hidden.

We provide a layer *link* toggle button. By default, scratching any layer moves it independently of the other layers. When the layer link button is activated, a scratch on any of the layers scratches all layers together, preserve intra-layer timing relationships.

In Scenario 3, Frame Editing, Clip Transforms and Layer control as described above could be used to isolate Jill as she performs her climbing motions, scale her as required while filming the CN Tower, and layer her climbing clip over this background footage.

4.4.6 Saving

We provide two modes to save the resulting video: *saving out*, and *live saving*. Saving out renders all video clips on the timeline, similar to a traditional nonlinear video editor. Live saving records the stream as seen on the viewer. In contrast to saving out, this preserves any live performance components, such as scratching back and forth.

4.4.7 Device and Platform Information

We implemented LACES on a Microsoft Surface Pro tablet, which provides high performance and memory capacity in a tablet that could be comfortably held in two hands, or one for short periods of time. For video processing, we use Emgu, a C# wrapper of OpenCV. We capture frames from the camera at 30 fps with 424x240 resolution.

4.5 Informal Evaluation

Designing a suitable evaluation was difficult - we could either assess quality of video outcomes from traditional versus LACES workflows, or we could perform a usability study of specific areas of the system (e.g. simultaneous scratching and cutting). The former is difficult, as our motivation for LACES was to enable editing where it did not occur before, and the latter would miss evaluation of the concept of capturing and editing in one interface. We gave the LACES system to 4 people to use in a self-determined manner for extended periods (2 hours to several days each) in their own environments. Our goal was to observe their use cases and creations without imposing specific use scenarios on them, and to observe whether the interaction techniques we presented would be effective in fostering creativity in opening up new options for expression with video. We found that our informal evaluation participants invented many interesting uses. One collected all clips from a comedy show of a performer laughing, intending to create a long track of her continually laughing. Another prototyped a two-view guitar lesson one could imagine broadcasted on social media. Another notable use case we observed is tracing; tracing is typically done over a still image, and if done over a moving image with a longer duration, it is called rotoscoping. However, our participant noted that tracing over a short, looping clip of a few

seconds could lead to sketches that capture the movement sequence. Participants also saw the value of LACES as an improvement over traditional video capture — one referred to it as *"organized taping"* and noted that "[what] I hate most about [video capture] is getting rid of the chaff because at the end of taping I have all this video I have to go through."

4.6 Discussion

We have presented an instance of a fluid video workflow that seamlessly integrates video authoring tasks usually performed at separate phases with different tools. We discuss the performance of this relative to our motivating scenarios, the challenges we identified in working with a live stream, and the potential for issues with cognitive load.

Our scenarios motivated a tool that enabled in-situ *Curating*, *Annotating* and *Coordinating* of video content. The LACES workspace, being adjacent to the live timeline layers, affords collecting and arranging video clips without having them disappear, and without committing them to a location on the timeline. LACES' compositing on multiple layers supports coordinating.

The input timeline layer clip was effective in providing access to live footage. One of the subjects described the movement as "initially stressful", but relaxed after recognizing he could collect clips in the workspace instead of having to use them immediately. In terms of the anticipated design challenges of *conflicting operations* and *catching up*, we have addressed these with scratching and accelerated playback. For the design challenge of *no output preview*, the lack of an additional viewer to preview changes is a trade-off to maintaining focus on the live action.

There is potential for the cognitive load of performing capture and editing together to be overwhelming if the user is required to divide focus on both operations at once. LACES accounts for this by providing flexibility in the timing of editing. Interactions such as scratching input and workspace clip pooling allow for editing to be performed when appropriate. We found users naturally edited during slack time — moments of low interest, during or immediately following capture: ideal opportunities where low cognitive load can be capitalized on to perform quick edits.

4.7 Use Cases

We present a few use cases, some developed by users in our informal evaluation, to illustrate the design principles and features in this interface, starting with the simple case of collecting interesting sub-clips for later use during capture, and building to complex compositing scenarios. Our accompanying video demonstrates some of these use cases, as well as other interesting examples.

4.7.1 Editing during Capture

Karim turns on music at home and his two children start dancing to it enthusiastically. He wants to film this to share with friends. Traditionally, he would take out his smartphone and record the entire moment. This results in a large and lengthy video file. Karim could edit this video down to

key moments, but this is very time consuming, as after transferring it to his computer he will need to re-watch it to retrieve the interesting segments. If he shares the entire unedited video, it can be boring to watch and cumbersome to upload. Karim needs to be always recording video, but have an ability to mark key moments, and remove uninteresting sections. This ability to edit while recording is what our interface is designed to support.

Karim holds the tablet running LACES with both hands and films his children. He is able to see the live view in the viewer, but he can also see a comic strip view of previous frames. As the music changes in style, Karim can press the slice button to segment the input video into separate clips for each style. He drags these to the workspace and arranges them based on the different styles of music.

At one point, Karim turns to talk to his wife, and shortly after, both of his children sit down to take a break. Karim notices his children have stopped dancing and wants to discard this part of the video. He scratches the input timeline layer back to the beginning of the break and slices it, and then scratches to the end of the break and slices again. He drags the sliced clip containing the break from the input timeline layer to the recycle bin. As the input video quickly plays back the live events that occurred while he was editing, he sees that one of his children jumps in a funny way. He scratches the input again to this exact point and sets a bookmark — this would be a good still to send in an email.

4.7.2 Storytelling with Props

Figure 4.6: Storytelling with props, mimicking a speeder run from Star Wars VI. Demonstrates blending a live and recently-recorded video. The user scratches the previously-recorded video so it runs at a higher speed.

Derek is a big fan of the Star Wars movies. His favourite sequence is the floating air speeders navigating through the forest moon Endor in *Star Wars VI*. This was filmed by compositing a green-screen scene of vehicle models in a studio with footage of a camera moving through a forest.

Derek takes a tablet with LACES and films a "flying" view by walking through wooded area in his local park. He slices and drags the scene clip to the workspace. He then drags the forest clip to the background layer, which now is blended with the live view of the tablet's rear camera. He takes out the speeder toy he brought and puts it in front of the rear camera (Figure 4.6).

He then moves it around so it appears to be steering to avoid trees. Since the forest clip was filmed at a walking pace, he scratches the layer to increase the playback speed, making the flight appear much faster. As Derek is moving the speeder toy around live, blended with the forest clip, the movement of the speeder toy is itself being recorded. He can slice this clip and overlay it on the live view for yet another walkthrough of the forest. The ability to keep one source of a compositing constant while adjusting the other is very compelling as a prototyping tool.

4.7.3 Overlaying Faces

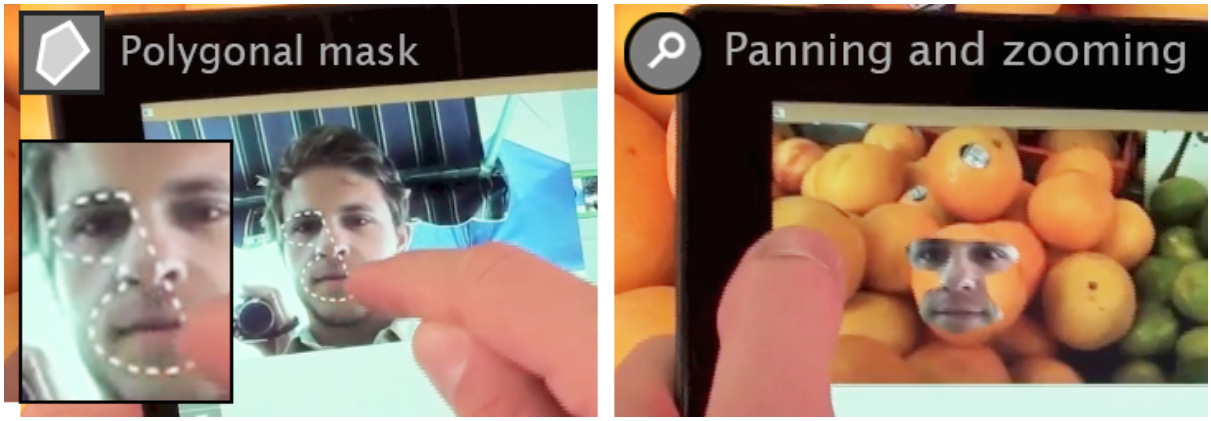

Figure 4.7: Overlaying face on objects. Demonstrates the use of a user-defined polygonal mask.

With a user-sketched polygonal mask, a user can overlay parts of their face on other objects or people in fun and interesting ways. This is seen in popular online videos as *The Annoying Orange* or the Quebeçois *Têtes à claques*.

When Flint is at a fruit market and comes across a stall of oranges, he is reminded of the funny Annoying Orange videos, and is inspired to use LACES to put his face on an orange. He starts off by using the front camera and makes a few funny faces in anticipation of placing them on a fruit.

Flint presses the polygonal mask button and draws a mask carefully around the boundaries of his eyes and mouth (Figure 4.7). Once finished, everything outside the mask is blacked out, and he only sees his eyes and mouth. He makes funny faces for a few seconds, looking left and right and wiggling his tongue. He needs to capture a good section of his performance to use as an overlay on the fruit, so he scratches the input layer to the start of his performance, slices it, then scratches it to the end and slices it again. He drags the face-making video clip onto the workspace, then removes the polygonal mask from the input.

Flint presses the "flip camera" button so the viewer shows the stall of oranges in front of him. Flint drags the face-making clip from the workspace to the foreground layer. He presses the loop button on the foreground layer so that this short clip will loop continuously. The masked part of his funny faces

clip is overlaid on to the live view. However, it is not aligned with any fruit in particular, so Flint uses pan and zoom controls to align it spatially. Flint holds his finger on the foreground layer to indicate operations will be applied to it. First, he uses the zoom handle to scale his face down so it will fit inside a fruit. Next, he pans his face so it appears on a fruit by direct dragging on the viewer (Figure 4.7).

4.7.4 Fighting with Yourself

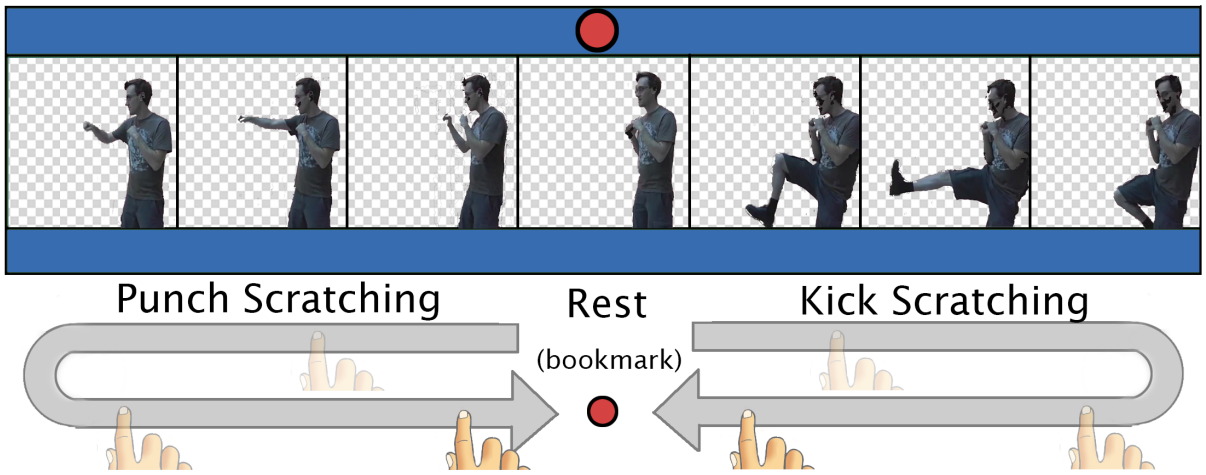

Figure 4.8: Self-fighting scenario: From a neutral frame in the centre, scratching right and back produces a kick; scratching left and back produces punch.

Patrick and his friend Felicity want to play a game where Patrick fights a video version of himself. They find a large wall with a uniform pink colour and Patrick stands in front of it. While Felicity films Patrick, she selects the magic wand and taps a pink part on the wall in the viewer. This masks out any pink pixels in subsequent frames. Patrick turns to the side and makes a punch, pauses, and then makes a kick. Felicity uses our input scratching technique to isolate this clip from the input layer, and drags it to the workspace then clears the wand. The clip of Patrick now consists of a first part where he punches, a brief neutral rest in the middle, and a second part where he kicks. Next, Felicity turns the tablet around, reversing the camera at the same time so Patrick can see himself. She drags the punch and kick clip to the foreground layer and holds it so the middle of the clip, where Patrick is neutral, straddles the present-time marker. This freezes Patrick's previous video in time. Real-time Patrick takes up a position opposite his pre-recorded self. Felicity can choose to play a punch or a kick by scratching the video in one direction or the other - forwards to play the punch section, or backwards to play the kick section, always returning to the middle (Figure 4.8).

4.8 Conclusion and Future Work

We have presented LACES, a system that combines video capture, editing and compositing to make video production more accessible to everyone. We have presented techniques to perform operations on live, moving media, methods for compositing live action with recently-recorded video, and several compelling use cases for LACES. We believe that that ability to operate on and annotate live, incoming data is important, whether video or not, as it closes the gap between capture and usage of media for any

application. While we have presented several short video authoring use cases, we have not explored scenarios of extended video activities, such as using the tool to help document a live event or produce a finished film, both of which expect a traditional video file as output. Our interactions can be extended for such cases, and we leave this to future work, and limit our exploration of live video manipulation to a smaller scale here.

5

Background Activity

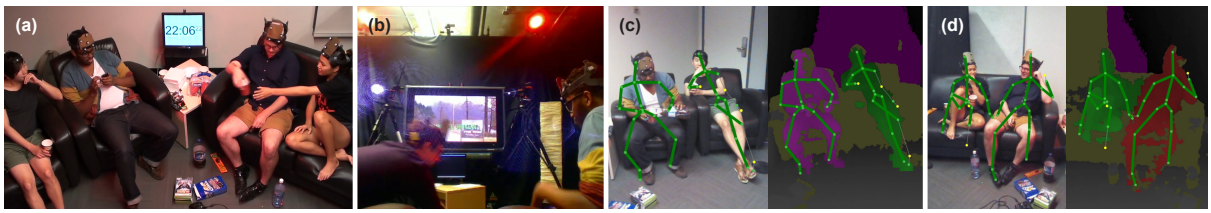

Figure 5.1: Example living room background activity dataset captured using our tools and methodology: (a) front HD video; (b) rear HD video; (c) Kinect facing chairs; (d) Kinect facing couch. All data is time stamped for synchronization. Kinect steams include colour, depth, skeleton, and spatial audio. Vicon motion capture of head positions (note "tracking hats") was included in 7 sessions.

In this chapter, we will study the general problem of embedded interaction (foreground activity) in noisy, natural everyday movement (background activity). This is a generalization of the problem specific to Improv Remix — interaction by performers in the midst of performance. In this chapter, we will define and test few strategies for solving the problem.

We define whole-body background activity as naturally occurring body movements that are not explicit input actions. We argue that understanding background activity is crucial to the success of always-available whole-body input in the real world. To operationalize this argument, we contribute a reusable study methodology and software tools to generate standardized background activity datasets composed of data from multiple Kinect cameras, a Vicon tracker, and two high-definition video cameras. Using our methodology, we create an example background activity dataset for a television-oriented

living room setting and use it to demonstrate how background datasets are useful for qualitative observation, quantitative evaluation, and generating design implications. The supporting software tools and example living room dataset will be made publicly available.

Whole-body input translates tracked body part positioning into explicit input actions. This has obvious advantages: there is no device to hold and people can interact when their hands are dirty (e.g., when cooking) or when their hands must stay clean (e.g., when performing surgery). Whole-body input has been demonstrated in various scenarios including public places [Müller et al., 2012], classrooms [Bolt, 1980], meeting rooms [Aggarwal and Ryoo, 2011], and kitchens [Panger, 2012]. However, unreliable body tracking and gesture recognition can make always-available whole-body input less robust. One factor is the "Midas touch" problem [Hilliges et al., 2012], where naturally occurring body movements are mistakenly interpreted as explicit input.

Separating explicit input actions from natural body movement is analogous to separating background from foreground in computer vision; thus, we call naturally occurring movement *whole-body background activity*. We argue that capturing background activity for observation and design testing is crucial to improving always-available whole-body input. For example, a common strategy to distinguish gestures is to use exaggerated movements unlikely to occur in background activity [Aggarwal and Ryoo, 2011, Song et al., 2012]. This reduces the space of available input gestures, and exaggerated gestures can be tiresome. By analyzing realistic background activity data, it should be possible to design and test more comfortable gestures that remain clearly distinguishable.

We contribute a reusable methodology and supporting software tools to generate standardized background activity datasets with 3-D motion tracking, depth cameras, spatial audio, and high-definition video (Figure 5.1). Our data gathering protocol also requires participants to perform explicit input gestures at regular intervals, so that datasets contain controlled foreground activity. To validate our methodology, we captured a dataset with 52 person-hours of background activity in a television-oriented living room setting, which we also make available to the community.

To demonstrate the utility background activity datasets, we use our example living room dataset for multiple purposes:

- Observing body postures of people sitting in comfortable positions, and using these observations to classify postures according to explicit input potential
- Qualitative evaluation of Microsoft Kinect SDK tracking with these more relaxed body postures
- Quantitative evaluation of a traditional Hidden Markov Model (HMM) gesture recognizer, finding a 37% recognition rate and 29,390 false-positives due to background activity
- Designing two new features: gaze vector and correlated hand movement, allowing us to reasonably eliminate 20% of false positives
- Identifying and testing three gestures, circle, slash, and 'L', that are less likely to occur in background activity
- Proposing *design implications* for gestures and gesture recognizers

These are only a small sample of potential dataset applications. We believe that building and sharing a corpus of background activity is critical to improving gestural interaction in real environments.

5.1 Defining Background Activity

Background activity is interleaved with all interface input, but some input techniques explicitly differentiate between input and non-input actions using an explicit control signal. As a simple example, consider that hand movement is only used for cursor control when a mouse is manipulated — all other movements away from the mouse are easily ignored. Similarly, touch screens use finger contact as the control signal to register input, but this also creates a "palm rejection" problem when a resting hand causes errant input. In this example, the background activity is mistakenly interpreted as an explicit control signal.

When whole-body input systems constantly track the movements of body parts, they essentially "reach out" into the real environment and become confounded by the ambiguity between background activity and explicit control. The reason is that typical background activity movements are often used for control signals and can be highly interleaved [Panger, 2012]. An outstretched arm with a pointed index finger could be a gesture to select a location on a computer display or gesticulation to support human communication. The problem is compounded in active environments where multiple people are multi-tasking with others, or where the physical environment is not conducive to careful, explicit gesturing.

In computer vision, background subtraction is a common method to separate objects of interest using a model of the image background [Stauffer and Grimson, 1999]. The separation of foreground objects (explicit input) is achieved by a deep understanding of the background scene (i.e., background activity). We argue that the whole-body input research can use an analogous approach. Current gesture and motion training datasets [CMU Graphics Lab Motion Capture Database, Shotton et al., 2013] are not applicable; a corpus of background activity datasets in realistic environments is needed, as are a methodology and tools to enable collection of additional datasets, taking the context of use into account.

5.1.1 Approaches to Managing Background Activity

There are a few different approaches used to further distinguish foreground activity in the midst of noisy background activity from the point of view of a gesture detector. This is, of course, more relevant in interaction mediums where the line between activity types is fine. Readers should note that we believe understanding background activity is more important than just improving detection rates; with a broad understanding, an interaction designer can get a sense of what sort of interactions users are able to do in the chosen interaction medium, what other activities they are engaged in, and of other features could be used for detection.

Here is an incomplete listing of possible approaches:

Always-On. The system always responds to gestures. If the context of the system has a great deal of Background Activity, gestures must be chosen carefully, based on the rarity of occurring in Background

Activity. Gestures are chosen based on their rarity in a collected dataset of Background Activity.

Explicit Clutch. The system only responds when in a specific user-determined state, i.e. Hold-To-Talk, as appearing in audio communication.

Delimiter or Framing Gestures. Similar to an escape character, a gesture is chosen as a delimiter, and indicates interaction is about to begin. The delimiter gesture is designed to be especially unlikely in background activity, while any other gestures may be more relaxed. There are varying ways to indicate the end of this interaction sequence. One strategy is that it ends after the first gesture recognized [Ruiz and Li, 2011], or after a period of inactivity, as seen in the gestural interface for the Kinect on the Xbox 360. Hudson et. al. use the term framing gesture to when the gesture used to explicitly declare the end of an interaction sequence is the same as the first [Hudson et al., 2010].

Implicit Clutching Using features outside those directly related to gesturing to estimate probability that interaction with the system is intended [Schwarz et al., 2014].

Baudel and Beaudouin-Lafon call Always-On systems, that interpret every gesture of the user as possible meaning, as having "immersion syndrome", ignoring that interacting with the system is not the user's only ongoing activity [Baudel and Beaudouin-Lafon, 1993]. An Explicit Clutch should be the simplest and most reliable approach, but requires user attention on the clutch to be maintained during the entire interaction. Delimiters do not have to be maintained during the interaction, but requires extra time at the beginning of the interaction sequence. Delimiters or Clutches may be multi-modal, such as pushing a switch or using a voice command [Bolt, 1980], but when pure whole-body input is desired, a unique gesture can be used [Walter et al., 2013].

In this work, we exploit our collection of a dataset to explore improving gesture detection rate with both Rarity by Design, and Implicit Clutching. For the final system in this thesis, Improv Remix, we use an Explicit Clutch (covered in a later chapter).

5.1.2 Establishing a Methodology for Dataset Building

Our objective is to establish a repeatable methodology for capturing an ecologically valid recording of whole-body background activity in a form suitable for distribution. In this section we establish a study protocol that includes occasional prompted foreground activity segments for baseline comparison, provide format specifications for a public domain dataset, and describe our logging and analysis software. We use our methodology to capture background activity in a television-oriented living room, a plausible context for whole-body interaction. We demonstrate different uses for this initial living room dataset, but the real strength is that our dataset, methodology, and tools are available to the community, who can use them to capture additional background activity datasets.

5.1.3 Eliciting Background Activity

Unlike typical methodologies where people are instructed to perform specific motions, asking people to "act out" background activity would not produce realistic results. We therefore advocate creating a physical and social environment that allows background activity to emerge naturally. Recording people in a real environment without their knowledge would be ideal, but ethical issues and cumbersome

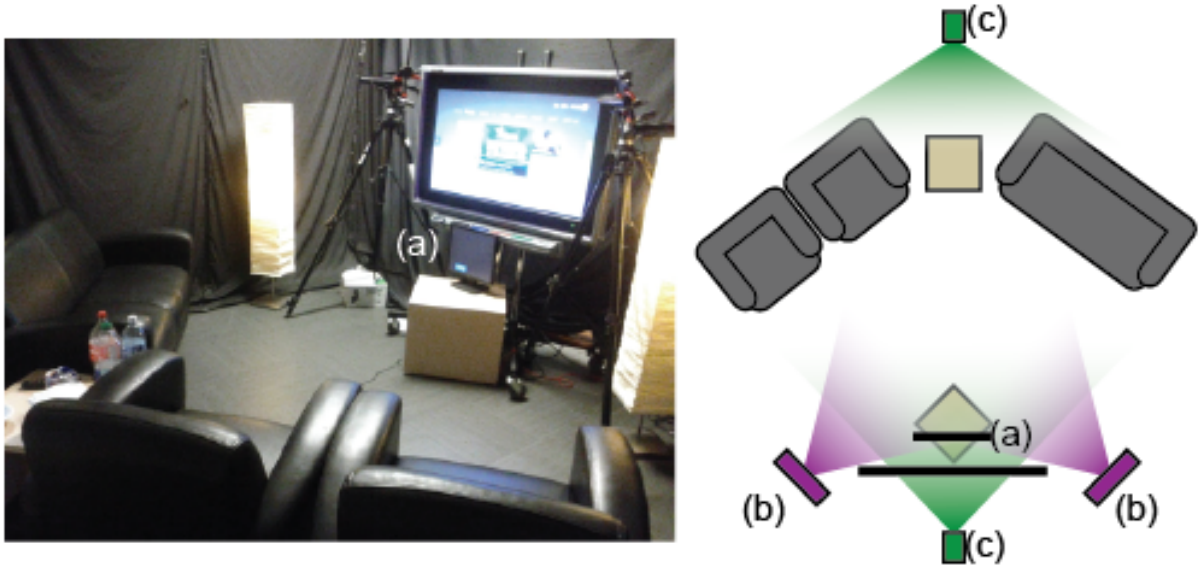

Figure 5.2: Living room environment with seating and large screen television. (a) small display for prompted foreground activity gestures; (b) Kinect cameras; (c) HD cameras.

capturing equipment make this impractical. A simulated environment in a lab is a more practical alternative. For our sample dataset, we simulated a living room setting with a game console and television (Figure 5.2). To increase social interaction, we ensured our participants had existing social relationships. To encourage object manipulation background activity, we provided snacks and drinks.

This provided a somewhat ecologically valid environment, where our example dataset is derived. The key methodological takeaway is that great pains should be taken to come as close to *in situ* data collection as possible. As shall be apparent when describing our results, the environment played a significant, though generalizable, role in shaping the data we collected during our study.

To gain full benefit from a dataset, the inclusion of typical *foreground* gesture activity is essential. This functions as a comparison baseline, much like the inclusion of foreground objects in the background removal datasets in computer vision. We achieve this by occasionally prompting participants in a subset of groups to perform one of four common gestures. Our methodology may be extended for testing a particular gesture language, by adding those gestures to the experimental protocol. This can even be done before recognizers have been built for those gestures.

5.2 Study Protocol

Our protocol includes physical environment setup, capturing apparatus, recruitment, and procedure.

5.2.1 Physical Environment Setup

We simulated a 3.8 m by 4 m living room with comfortable furniture likely to be found in a home and used soft incandescent lighting and curtains to hide the institutional walls (Figure 5.2). We placed two armchairs and a two-person sofa in front of a 54″ television with external speakers approximately

2m away. Participants could watch Netflix programs or play video games, controlled using a single wired Xbox controller. We intentionally provided a single controller to increase background activity: controller usage had to be socially negotiated and transferred. Similarly, background activity was encouraged when selecting a video game from a stack on the floor and inserting it into the Xbox.

To maintain an unobstructed view of participants, we placed a small coffee table between the couch and nearest armchair, rather than in front. This table held food and other personal belongings within arm's reach of the two nearest participants. This was another intentional choice to encourage background activity, as items on the table were requested by the outer two participants and passed back and forth.

5.2.2 Capturing Apparatus

We used minimally invasive capture equipment. A wide-angle HD video camera captured audio and video of the entire scene from the front (Figure 1a) and a second HD camera captured from behind, including the gesture prompt screen and television content (Figure 1b). One Kinect faced two people on the sofa (Figure 1d), and the other faced the two people in the armchairs (Figure 1c). Each Kinect recorded 13 bit, 640 by 480 px depth with 3 bits of "player id" masks (pixels classified as part of a human body), 640 by 480 px RGB video, 20 segment skeleton tracking (when possible), and spatially separated sound using Microsoft Kinect SDK version 1.5. When used, a six-camera Vicon system placed high in the ceiling tracked head position and orientation of all four participants using four lightweight hats. We were concerned that the Vicon "tracking hats" would affect behaviour, so we used them with a subset of groups in order to increase the breadth of our sample dataset. When practising this methodology, sensors might be reasonably limited to those available in the target platform.

We found that the built-in Kinect SDK recorder produced extremely large files (typically 1.5 GB-per-min-per-Kinect). To keep data manageable, we designed a more efficient Kinect data capture format (typically 0.3 GB-per-min). We used RIFF as a generic container to house all time-indexed depth, RGB, and skeleton frames in one file. RGB frames were compressed with lossy JPEG compression and depth frames lossless LZF compression. Since the Kinect SDK does not output depth, RGB, and skeletal frames at a consistent rate, each frame is time stamped. We provide custom Windows C# software to capture and play back Kinect data in this format, as well as Python software for gesture detection and other analyses. We are planning to update the file format and tools when the high resolution Kinect 2 is available. A detailed file format description is included with the dataset to enable other language and operating system implementations.

5.2.3 Participants

A large amount of background activity is socially motivated (e.g., conversational gesticulation), so we recruited participants in groups instead of individuals. Online posting and word-of-mouth yielded 13 groups of four participants, 52 participants in total. The mean age was 26 years (ranging from 19 to 59). Overall 67% of our participants were male, but gender distribution within groups varied: one all-female, four all-male, and the remaining mixed. Seven groups used Vicon motion tracking, seven groups included prompted foreground gestures, and five groups had both.

In three groups, one participant was meeting the others for the first time, but all others had existing social relationships. Pairs of participants who had closer relationships would often rush to the sofa to remain physically close, often touching and cuddling. We encountered some unexpected behaviour. In one group, a participant was frustrated with the other members and avoided social interaction - he spent most of his time reading a newspaper. This too represents an interesting example of background activity.

5.2.4 Procedure

The procedure emphasizes putting participants into a mood suitable for the simulated environment. In the case of our living room simulation, this meant getting participants comfortable and minimizing the feeling of being in a lab. The researcher always met participants outside the building and guided them to the study room on a route to minimize office spaces. During the walk, the group was engaged in small talk to help everyone relax. We wanted participants to act as if at home — shouting, cheering, joking — without worrying about disturbing others working in the building. Study times also reflected this social situation, with most group captures occurring in the evenings and on weekends.Ê

To make everyone comfortable, the researchers purchased requested snack food and non-alcoholic drink in advance (typically less than $20 per group). Food and drinks were placed on the coffee table in the study environment, along with disposable plates, cups, and napkins with a garbage can in the corner. This increased realism, but as explained earlier, eating and drinking also elicit interesting and externally valid background activity.

After signing informed-consent forms, the formal study introduction began. In instances where prompted gestures were collected, the researcher gave instructions on performing them (details below). Then, he provided instructions on the use of the Xbox media device. Participants were encouraged to relax and enjoy whatever they wished on the television, or to just talk, as long as they remained in the simulated living room space and in the same order on the furniture.

The capture apparatus was switched on, and the study ran for 60 minutes. During this time, the researcher remained out-of-sight in a nearby location monitoring the capture streams in case there were any problems, and then gave a five-minute warning before the study ended.

Prompted Foreground Gestures

Seven groups were regularly prompted to perform gestures to capture foreground activity in the context of background activity. We chose four common gestures: horizontal swipe, whole-hand AirTap [Vogel and Balakrishnan, 2005], wave [Vogel and Balakrishnan, 2004], and point [Grossman et al., 2006], all performed with the dominant hand. *Horizontal swipe* is a left or right motion (~60cm) with the palm perpendicular to the large display, arm extended away from the body, and elbow relaxed. *AirTap* is a forward and back movement (~25cm) with palm facing the large display. *Wave* is a left and right periodic motion (~25cm) with the elbow roughly fixed in space. *Point* extends the arm and index finger towards the television. The required duration of wave and point are 800ms. These gestures were chosen since they have been used for explicit input, with demonstrated successful detection, but we believed

they were also likely to occur in background activity. We kept the set of gestures small to reduce cognitive load on our participants and avoid interference with our primary goal of observing background activity.

A 17-inch display below the television (Figure 2a) prompted people to perform a gesture using an iconic representation and audio cue. The prompt was shown until the gesture was "recognized" by the researcher monitoring the HD camera feed, a Wizard-of-Oz recognition technique. Each gesture was prompted five times during the 60-minute session, resulting in a foreground gesture sequence approximately every three minutes. Participants were prompted by number (1-4), so each performed each gesture at least once.

Before the study began, the researcher demonstrated each gesture to the group twice. The researcher left the room so that each participant could practice following the small display prompt to perform one gesture. All gesture-training demonstrations are included in the dataset.

5.3 Results

We captured 1 hour of data per group of 4 people, totalling 52 person-hours of background activity and 750 GB of data.

5.3.1 Participant Behaviour

Most groups played a game or watched television while also talking, eating, and using mobile devices. While the television display was the primary focus, participants were almost always multi-tasking. Participants assumed a wide variety of comfortable positions on the furniture that suggest we were successful at simulating a realistic living room setting.

Intensity of background activity varied. Aggressive gesticulation was common, especially for boisterous groups. One group of hip-hop dancers was very expressive with a high level of dynamic gesticulation. Another group had two of its members playfully compete to be the centre of attention, successively outdoing each other in speaking volume and gesticulation intensity. There were also quieter groups, such as a married couple and one set of parents. This group quietly watched a movie and ate snacks, speaking occasionally.

5.3.2 Prompted Gestures

For groups with prompted gestures, we captured a total of 140 gesture sequences (7 groups x 4 gestures x 5 prompts). We noticed that well-intending participants reminded others to perform a gesture. This usually involved some communicative gesticulation similar to the required gesture, similar to what might happen in a living room where one user is teaching or reminding another how to use the UI. Nonetheless, because this appeared to be an artefact of our setting, we asked participants not to engage in this behaviour.

5.3.3 Capture Quality

The Kinect captured data between 15-30 fps. The quality of body tracking and skeletal tracking varied, largely due to unexpected postures. For groups with Vicon motion tracking, 6 DOF data for each hat was captured at between 60-120 fps. At first the tracking hats seemed conspicuous to the participants, but after about 10 minutes, they settled into seemingly relaxed behaviour.

5.4 Example Dataset Applications

We demonstrate the utility of a background activity corpus by using our initial living room dataset for observation, qualitative evaluation, quantitative evaluation, finding useful features, and evaluating proposed gestures.

5.4.1 Observation: Body Postures

A corpus of background data can be used to classify natural postures in a given setting. Here, our goal is to classify body postures that occur in a comfortable environment like the living room. These can be individual postures or combined to include multiple bodies. Our results are relevant to understanding the availability of a person's specific body parts to provide explicit input for a computer system, which could aid in off-line gesture design, as well which type of controls the system offers in-the-moment. It is also possible that this could motivate a model of typical movements, given a certain body posture - this would allow a system to better distinguish unusual movement (a candidate for foreground activity) from background activity. In addition, this provides motivation for improving body and skeletal tracking for this kind of environment.

To find static postures, we used a script to extract depth and RGB frames from the data where the depth frames had inter-frame differences below a threshold for five seconds or more. This resulted in 2014 frames from the two scenes (couch and chairs). The frame samples are reasonably uniform across studies, with a median of 51 samples for the two scenes across 13 groups. Using these frames, we classified postures according to two characteristics: torso lean and arm position. We also observed interesting multi-person body postures.

Torso Lean

We found that the degree of torso lean is a useful way to gauge how available someone is for performing explicit input. We categorized leans into three levels. In decreasing level of availability: forward, neutral, and back (Figure 5.3).

- A *forward lean* is when the head and shoulders are in front of the hips; arms have less contact with furniture, and attention focus is forward. This often resulted from handling food, mobile devices, or the Xbox controller.
- A *neutral lean* when the torso is near vertical; arms on armrests with one arm often supporting the head. In this case, one arm typically remains available for interaction.

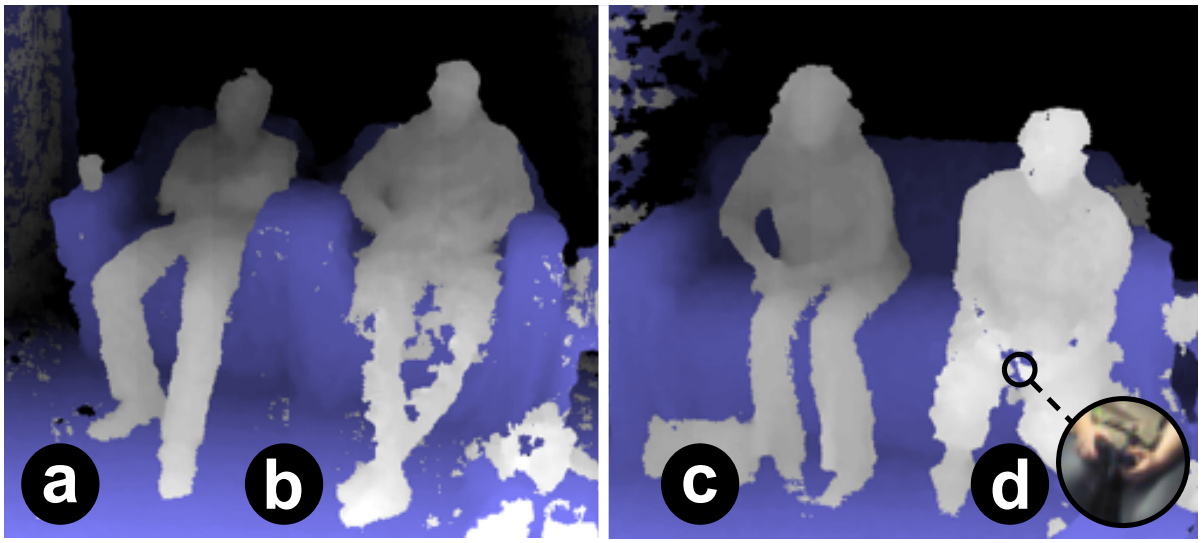

Figure 5.3: Torso lean degrees: (a,b) backward lean (least active); (c) neutral lean; (d) forward lean (most active).

- A *backward lean* is characterized by the body appearing relaxed, with the torso fully supported by the backrest, often adopting asymmetrical poses with crossed arms and legs. This is the least probable torso lean for interaction.

Arms

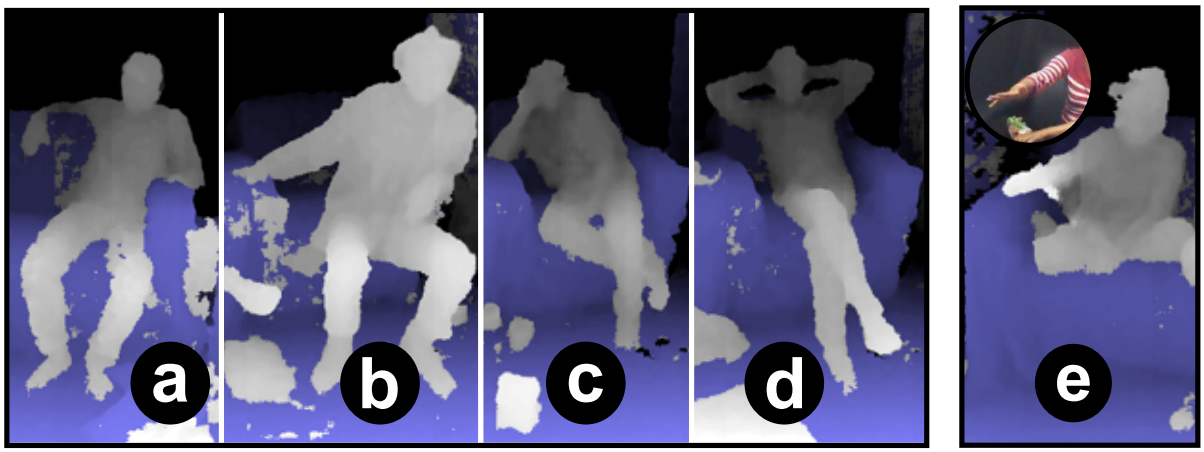

Figure 5.4: (a) - (d) Examples of arm unavailability: (b) Participant gesturing with the available hand. Note, in the RGB overlay, the other hand is occupied with a bag of chips.

We observed a variety of different arm postures, ranging from extended arms far away from the torso, to crossed arms, and arms kept close to the body. Body symmetry is indicative of which limbs are available for performing explicit input motions. Any limb supporting the body, head, or other objects is unavailable for immediate explicit input. Even when resting, relaxed extended arms, aimed towards the system were indicative of availability (Figure 5.4).

Combined Body Postures

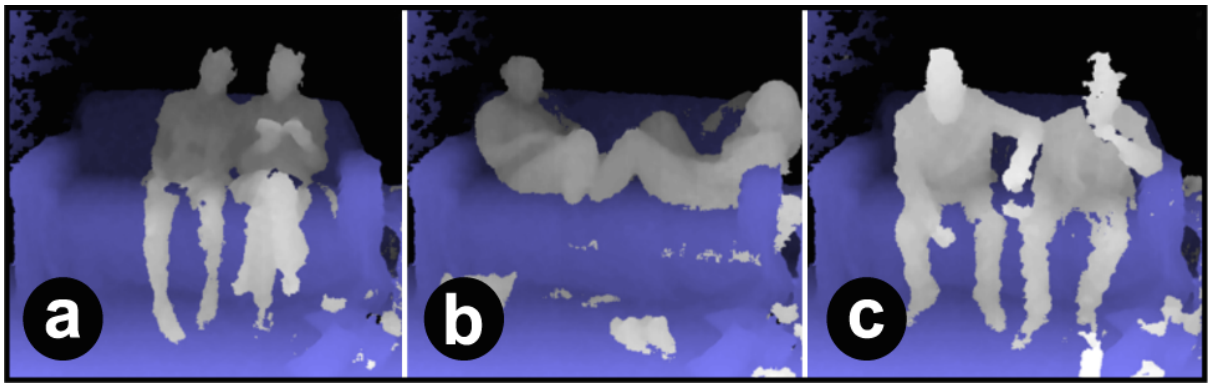

Figure 5.5: Examples of combined body postures: (a) pressing torsos together; (b) interweaving legs; (c) sharing food.

We observed combined body postures where two people sat very close. This happened when sharing food, viewing another person's mobile device, expressing intimacy. In these cases, there was a breakdown of each participant's body limits, and skeletal and gesture recognizers' effectiveness was very low. Gesture designers could specifically consider close postures, for example, designing two-person gestures (Figure 5.5).

5.4.2 Qualitative Evaluation: Body and Skeleton Tracking

We used our dataset to evaluate Kinect SDK tracking. We found that the tracker performs well when people sit upright and make large movements, but performs poorly when people are seated with legs crossed, leaning, touching other people, or holding objects. To investigate methodically, we reviewed the 140 prompted gesture sequences. We found 62 (44%) of these sequences have properly tracked skeletons. Due to issues with low or uneven depth frame rates or lack of skeleton recognizer output, 41 sequences (29%) have no skeletal data. However, the depth data quality in 33 of these sequences should be adequate for post-capture skeletal detection using other libraries. The remaining 37 (26%) of the sequences represent interesting failure cases. In five sequences (4%), the participant was sitting in a position that makes skeleton detection difficult, such as having their legs crossed or arms folded tightly (see body posture observations above). In 15 cases (11%), the skeleton was generally correct, but another object was erroneously tracked as the dominant hand (often the participant's torso, leg, or parts of the furniture). This failure was likely due to the arms being held close to the body or hands occupying a small area when extended directly towards the camera. In 11 cases (8%), a skeleton was detected away from the two primary participants in the scene, such as on some of the items in front of the participants, or another participant leaning into frame. Since the Microsoft Kinect SDK supports a maximum of two skeletons simultaneously the addition of this new skeleton resulted in an inability to track the participant performing the prompted gesture. For six cases (4%), person-tracking merged two people sitting close together, creating aberrant skeletons. This was most pronounced in one session where a couple sat close together on the couch. Two of the sessions without prompted gestures also have sequences where body tracking merges people sitting close together. Identifying and correcting these failure cases has the potential to improve tracking.

5.4.3 Quantitative Evaluation: Gesture Recognizer

Background activity datasets can be used to test different gestures and recognizers. As an example, we use our initial dataset to evaluate the performance of a HMM Gesture Spotting Network (GSN) using the four prompted gestures in our dataset: swipe, point, wave, and AirTap. These results are dependent on skeletal tracking quality for hand position, a realistic limitation when using current skeletal trackers, especially in comfortable environments, like a living room where poor tracking seems more common.

HMM GSN Design, Implementation and Training

Our design is based on Fourney [Fourney, 2009] and Lee and Kim [Lee and Kim, 1999]. A GSN is a meta-HMM containing multiple HMMs connected in parallel. There are left-to-right gesture HMMs for each variation of the gesture to be detected and a special threshold HMM representing non-gesture movements. A gesture is detected (or "spotted") when the final state of one of the gesture HMMs has a higher likelihood than every state in the threshold HMM. Like Fourney, we discretize hand position and velocity into features, albeit ours are in 3D. We measure the depth of the hand relative to the shoulder and its horizontal and vertical position relative to the elbow. These continuous measurements are discretized into bins: 3 horizontal and 2 for depth and vertical. Velocity is discretized by finding the nearest unit vector of form [-1,0,+1,-1,0,+1,-1,0,+1].

We found that many participants performed two of the prompted gestures slightly differently, and a single HMM cannot easily describe all variations. Instead, we trained one gesture HMM for each gesture variant. Swipe has two variants: elbow straight and elbow bent. AirTap motions all began with a fast forward motion, but three ending variants: relaxing the arm, dropping the arm, or pulling the arm back quickly. Wave and point had a single variant. For training data, six volunteers, who did not participate in the study, performed each gesture variant 3 to 15 times while sitting on the study couch. For handedness, we generated training sequences using mirrored body positions. After discarding approximately 20% of cases with poor tracking or unusual motions, there were 80 training examples per gesture variant. We trained the gesture HMMs using Baum-Welch, with 10% of the training examples as held-out test data. Running all gesture HMMs on the test data using a partial GSN without the threshold HMM achieved 97% accuracy. Adding the threshold HMM to make a full GSN reduced accuracy to 72%, primarily due to low point gesture HMM likelihoods (the GSN achieved 86% accuracy without point). This demonstrates shortfalls of commonly used synthetic threshold HMMs.

Performance

We first evaluate the true-positive rate using the 140 prompted gesture sequences. As we noted before, only 44% of these have good skeletal tracking. Using the partial GSN without the threshold HMM, detection accuracy was 37% and with the full GSN, accuracy was 20%. When conditioned over 44% (good tracking sequences) detection accuracy is 84% and 45%. The low accuracy is partially due to gesture performance changing over time and the researcher's liberal Wizard-of-Oz recognition. Gesture performance became subtler, more individualized, and with more oscillation when recognition was not immediate. Oscillation profoundly changed the performance of swipe and AirTap. To evaluate false-

positive rates, we ran the GSN over each tracked skeleton in all background activity sequences. We found 29,390 false positives: 18,886 for swipe, 5,307 for wave, 3,492 for point, and 1,705 for AirTap. In total, this is one false positive every 6.5 seconds per-participant. We examined 20 false positives for each gesture and found many cases where poor skeleton tracking was the cause. Focusing on false-positives with good skeletal tracking, we identified five common causes: reaching or manipulating objects, gesticulating, touching, repositioning, and stretching. *Reaching* or *manipulating an object* created motions similar to a point or swipe. *Gesticulation* led to expressive hand movements that could look like any of the gestures. When participants *touched* themselves, such as scratching, a wave gesture was often recognized. When participants *repositioned* their body, such as leaning back and extending their arms forward on the armrest, this appeared as a forward-extended point gesture. Finally, *stretching*, often with both arms, triggered an AirTap or forward point gesture. In the next section we discuss design implications based on these causes to reduce these false positives. This is only an initial examination of false-positive causes; the dataset provides the means to complete a more formal analysis.

5.4.4 Recognizer Design: Discriminating Features

We explore two new features to help distinguish foreground activity, gaze vector and correlated hand movement, as *Implicit Clutches*. We examine each feature separately, during prompted gestures and during false positives. If there is a significant difference in the feature's value between prompted gestures and false positives, we can effectively use it to reduce false positives without unduly increasing false negatives.

Gaze Vector

There were many false-positive cases where participants were facing each other while conversing, which we would expect to be rare when performing explicit input. From examining every true positive, we know participants always looked at the television at some point. In most cases, participants looked at the television TV for the duration of their the gesture, but in other cases they would look briefly at the television at the beginning, then turn back to other participants while performing the prompted gesture.

We examined the gaze vector coming from each of the participants' hats and compared it to the vector between the hat and the TV. We projected each of these vectors onto the floor and measured the angle between them, giving us an approximate measure of how close the participants' gaze was to the TV. We averaged this measure over the period when the participant was gesturing.

Correlated Hand Movement

Based on observations during the study and an examination of false positives, background gesticulation often involves correlated movement of both hands. We rarely saw significant motion in the non-gesturing hand during our single-handed prompted gestures. To analyze this feature, we measured average motion velocity of the other hand during a time interval near a prompted gesture.

Feature Performance

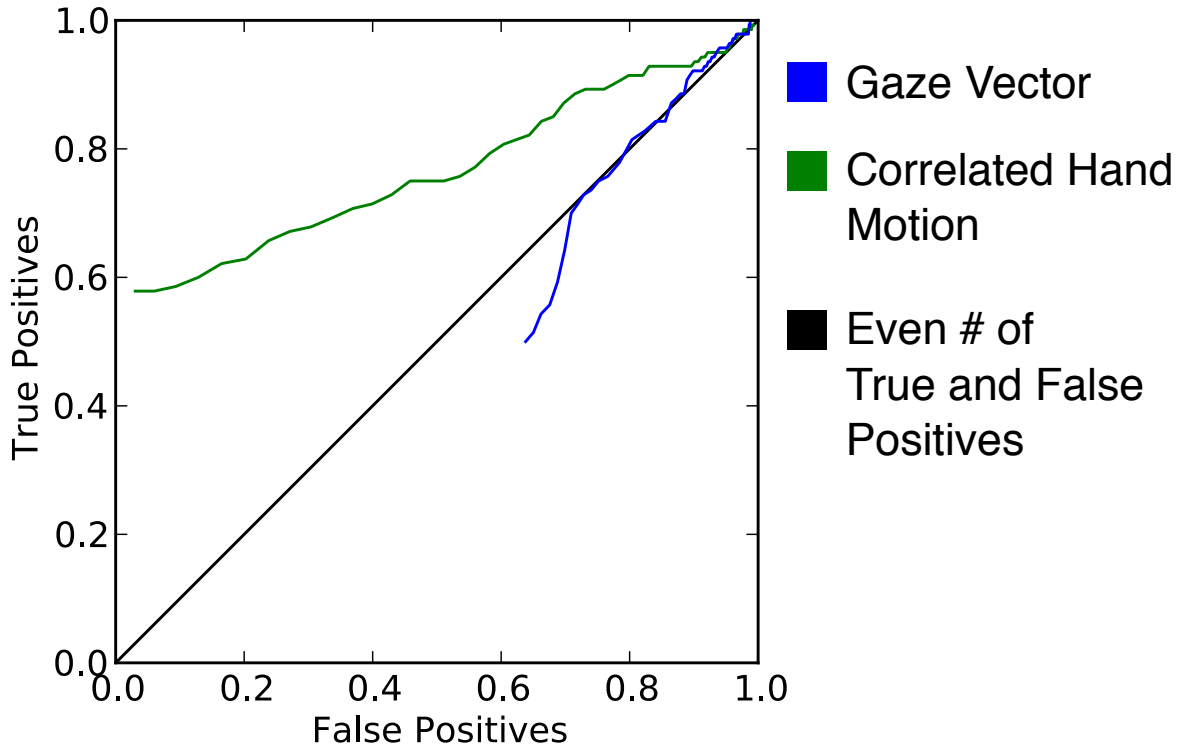

Figure 5.6: ROC for correlated hand motion and gaze vector.

When we graphed the histograms of gaze vector and correlated hand movement, the distribution of false positives and true positives appeared distinguishable. To visualize the false positives rejected at the cost of true positives rejected, we use an ROC (receiver operating characteristic) curve (Figure 5.6). A diagonal line indicates no difference between the rejected false and true positives, which is the case for gaze. However, correlated hand movement rejects more false positives than true positives. Using correlated hand movement as a feature to distinguish foreground from background activity, we could eliminate 20% of false positives with a false negative rate of 10%.

5.4.5 Application: Proposing New Gestures

The prompted gestures we chose produced far too many false positives to be useful in a real scenario. While recognition may be improved with a better recognizer, this will provide diminishing returns. We demonstrate the utility of background activity datasets by using our living room dataset to redesign our gesture set to be more robust to the real-world activity, without any changes to the design of our gesture recognizer. To test the utility of a given gesture in a certain background activity context, we can simply train a detector to recognize the gesture, then run it through our data and count the number of false positives, where fewer false positives is better. This is an extension of previous procedures used in different sensing domains [Ruiz and Li, 2011]. We created a set of proposed gestures that semantically correspond to each gesture in our prompted gesture set 5.7. Instead of left and right swipe, we create

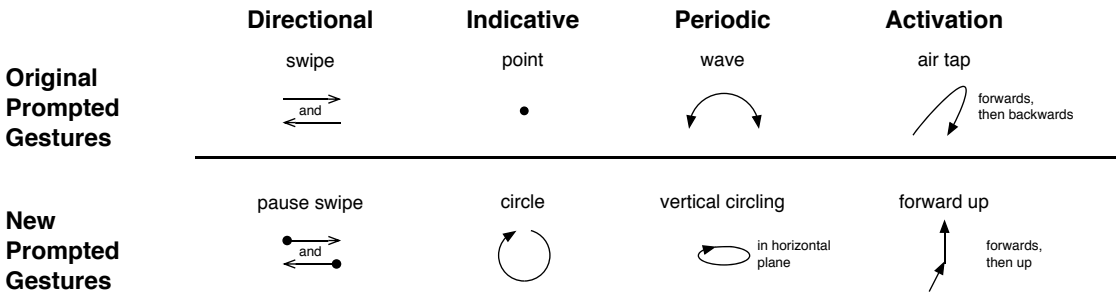

Figure 5.7: Diagrammatic representations of our original prompted gestures, followed by the corresponding proposed gestures, which are semantically similar but produce substantially less false positives.

Pause Swipe, a swipe that is preceded by a short pause; this preserves the swipes' directional property. Instead of point, we create *Circle*, meant to be a single circle motion of the ex- tended arm parallel to the torso of at least 30 cm in radius; this preserves the point gestures' ability to indicate an object by circling around it, as if with a cursor. Instead of wave, we create *Vertical Circling* a continuous circling motion in the horizontal plane with the arm extended upwards from the elbow; this preserves the periodic property of wave, providing a gesture that could be performed until a system response is given. Instead of AirTap, we implement *Forward Up*, a push forward towards the interface, then an upward flick. This preserves AirTap's semantic sense that a specific location on the surface is being activated or approved, similar to a click. We trained our gesture recognizer on 10 examples of each of these proposed gestures. We ran our same GSN HMM recognizer through the dataset to look for these gestures, and consistently found fewer false positives. For Pause Swipe, we found 2,494 false positives (15.2 times less than Swipe); for Circle, we found 5,409 false positives (3.5 times less than Point); for Vertical Circling, we found 5,172 false positives (3 times less than Wave); and for Forward Up, we found 268 false positives (3.3 times less than AirTap). Overall, we reduced the false positive rate by a factor of 5.5.

We have successfully produced gestures that are *natural* in the sense that they are comfortable to perform, but *unnatural* and *unique* in the sense that they are less common in background activity. While we have only created a tested a single alternative to each original gesture here, this methodology could be fused with other gesture design techniques.

5.5 Design Implications

Informed by the observations and evaluations above, we discuss implications for the design of gestures and gesture recognizers supported by evidence in the dataset.

5.5.1 More representative HMM background model

Our HMM implementation uses a standard strategy to build a synthetic background model for a gesture spotting network [18]. The background model is a fully connected network with all states taken

from the gesture recognition models. Essentially, this strategy considers background movements to be out-of-sequence foreground movements. This naïve method is simple but does not model true background motion, nor does it contain additional features needed to distinguish background activity from foreground gestures.

The dataset enables the construction of a more representative gesture spotting network background model. New features can be added to model background motions, and the background motion recognizers can be trained. Background recognition features could also improve the foreground gesture recognizers and reduce false-positives.

5.5.2 Gesture-specific spatial zones

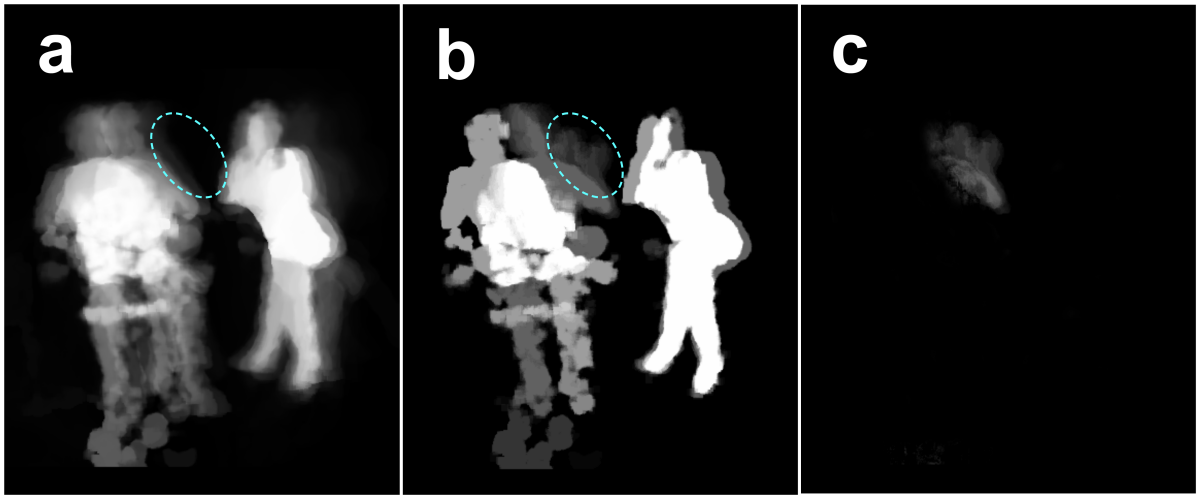

Figure 5.8: Proposed gesture-specific spatial zones visualized using average depth occupancy: (a) background sequences; (b) AirTap gesture sequences; (c) subtraction revealing spatial gesture zone.

We observed participants performing gestures at greater distances from their body than typical background motions. To operationalize this, we calculated the average body depth during background sequences (Figure 5.8a) and average body depth occupancy during prompted gesture sequences for each type of gesture (for AirTap, Figure 5.8b). Subtracting the average background occupancy from average gesture occupancy reveals a spatial zone where that gesture was performed. Although they appear similar, early results indicate that gestures may populate spaces not common to background activity.

5.6 Conclusions and Future Work

We described a methodology to capture whole-body background activity and used it to capture a television-oriented living room dataset. To demonstrate the utility of this approach, we used this example dataset for multiple purposes. By observing comfortable body postures in the dataset, we highlight how torso lean, arm position, and combined body postures could aid explicit input detection. We quantified skeletal tracking in the dataset and showed that tracking relaxed body postures is challenging.

We used the dataset to quantitatively tune and test a HMM GSN gesture recognizer and to uncover new contextual features that can reduce false positives by 20%. The dataset also enabled us to test three gestures, circle, slash, and 'L', to see which are less likely to occur in background activity. Finally, we use the dataset to justify design implications for a more representative HMM background model and the potential for gesture discrimination using gesture-specific spatial zones.

These practical findings are encouraging, but it is important to note that our living room dataset and example dataset applications are primarily intended to illustrate and validate our reusable dataset capture methodology. The living room dataset and supporting capture and analysis tools are available to the research community.

Our intention is that these methods, tools, and techniques will assist in the research and design of whole body gestural interactive systems by motivating the capture and sharing of many background activity datasets. In addition to this community benefit, our work provides evidence supporting our argument that understanding background activity is crucial to bringing always-available whole-body input into the real world.

For the purposes of Improv Remix, we have found that the state-of-the-art gesture recognizers are not sufficient to distinguish foreground from background activity. While in this chapter we showed promise by examining some discriminating features (e.g. correlated hand motion), the improvements are not sufficient without further in-depth research. A reliable gestural interaction technique will have to be well-designed — indeed, the observation about possible gesture-specific spatial zones partially inspired the *Vitruvian Menu*, to be introduced in a subsequent chapter.

6

Theory of Gestural Interaction during Performance

In this Chapter, we will outline considerations when designing interaction for improv theatre performers as a general problem. In the next Chapter, *Improv Remix*, we will present our specific design problem and solution, of manipulating stage video. The problem space we define here may be foundational for other work that seeks to implement performer interaction with a responsive system onstage. Our specific focus is on improvised theatre; as noted before, if the onstage actions are heavily scripted, it may be more practical to have an offstage technician control elements of the show.

The goal of the understanding we seek is to answer the question "What makes for an effective coexistence of performance and interaction?" The design space we are interested in is performances that include primarily performance *as a character*, with some interaction with a system. However; we are not expecting the interaction to be "always on", indeed, negotiating when a performer is performing, or interaction, or both, is a problem we seek to understand from multiple perspectives.

This chapter starts with an overview of *Terminology* used in the rest of this thesis. Next, with a discussion of *Stakeholders* when designing for interaction during performance. Next, a listing of *Design Principles* that are important considerations when working in this space. Then, we will describe an exercise we undertook in *Interaction Mapping* as preparation for the design of Improv Remix.

6.1 Terminology

We will define several terms necessary for understanding the work in this thesis. This terminology is ad-hoc and unique to this thesis — it is tedious for the reader to parse the phrase "intentional interaction with a digital computer system", when we can use the short-hand "interaction".

To contextualize the terminology to come, we will describe an improvised set I observed in the Savannah Room in Toronto in Fall 2008, as part of the Impatient Theatre Company's *Harold Night*:

A group of 5 performers are in the middle of a longform improv set. Two of them (A,B) are performing onstage, as an injury lawyer speaking to a lumberjack. Three others (C,D,E) watch from the sides.
C sees an opportunity to bring the show in a new direction. C steps on stage, tapping B on the shoulder (the standard Tag-Out coordinating gesture). B leaves the stage, and A retains his character (the injury lawyer), while C assumes a new character (a potter) and a scene between A and C begins.
After a short period of time, D perceives that A and C's scene has become stale, and performs a Sweep (another standard coordinating gesture). A and C step off the stage. D steps on stage and begins speaking "I have gathered you all here in the town square...", implying that she is beginning a group scene. All the performers step on stage, to support the scene, except B. D begins a serious speech about workplace safety, while B acts as a heckling dissenter (a different character from the injury lawyer she played before).
At this time, a real mouse runs across the stage, where all the performers are able to see it. Hilarity ensues, and the performers' reactions differ significantly:

- *A and D panic, losing their character and running offstage.*
- *C and E pretend not to see the mouse, and hold fast to their characters, attempting to continue the scene.*
- *B, in character, starts complaining about the lack of cleanliness in the town square.*

Seeing that the scene has gone off the rails, C steps forward and signals to the technician to cut the lights, ending the group's performance.

The above example contains many components common to improvised performances. Let us define some terms:

Action. An atomic act of a performer. e.g. waving, sitting down, scratching their neck, coughing. A performer may be engaged in multiple actions at once, i.e. speaking to another performer while turning off a light switch.
Interaction. For the purposes of this thesis, when we say *interaction*, we are referring to actions of a performer to intentionally produce a response by an observing system. Activities between performers or the audience are not termed interaction in this case.
Interaction Technique. A means by which a user conveys a command to a system.

The relationship between characters, actors, and performers, is covered in detail in the *Audience Perception* section later.
Character An entity expressed by the actions of the performer that is understood to be separate from the performer themselves.
Performer While the term "actor" typically applies to one acting out a character, we use performer,

which is does not have that strict requirement. A performer is someone engaged in putting on a performance, designed to be observed or interacted with from an audiences' perspective.
Performance. A series of actions by a performer that is intended to be observed by an audience, with the understanding the the actions of the performer are for the purpose of observation and distinct from their normal behaviour.
In-Character. An adjective that may be applied to actions of a performer that are meant to express or reinforce the audiences' understanding of a character under development.
Out-of-Character. An adjective that may be applied to actions of a performer that do not express or reinforce a character that the performer is in the midst of expressing.
Onstage. For the purposes of this thesis, when a performer is "onstage", their actions are meant to be observed by audience members to serve the ongoing performance. Performers do not literally have to be in an area that is defined as a stage in order to be "onstage". The performer acting from the side in the example above is still onstage in this sense. The term often used to describe where performers are when they are performing is "magic circle". This is not necessarily physically well-defined. In the document, the opposite of onstage is offstage.

Live Performer. For the purposes of this thesis, a physically (as in, a human body) present performer in the theatre space, able to respond in real time.
Playback Performer. In this thesis, projected video and audio of a single performer's previous performance in that session. We emphasize that we treat this instantiation as a "performer", instead of a video feed we are observing. It is meant to appear to occupy the same space as the live performers.

6.2 Stakeholders

We see three stakeholders when designing interaction for theatre:

- system (detection)
- performer (experience)
- audience (perception)

Similarly, Loke and Robertson describe the three perspectives on movement as the mover, observer and the machine [Loke and Robertson, 2013], though they keep their notion of observer very general, whereas we are specifically concerned with a theatrical audience member. We have covered the system's perspective generically in the Background Activity chapter, and from now on will focus more specifically on the types of movements possible during a standing, as opposed to seated, performance.

We shall now cover each of these stakeholders' perspectives on interaction during performance.

6.2.1 System Detection

We have an advantage over the typical walk-up-and-use case in that we can offer our performers some training so they can be careful to avoid false positives. However, we have a disadvantage in that our performers will want to express themselves physically more unpredictably and frequently than the standard user. The system must detect candidate gesture movements in a continuous stream of movement as part of performance — gestures intended for the system will be the exception to the rule, the signal to the bulk of noise. Even if our strategy is to use an unusual delimiter, it must still be detected reliably. In literature, this has been termed the *Gesture Spotting Problem*.

We have covered much of the problem of intermixing system interaction with background movement in the Background Activity chapter. We know that, if we are not careful, false positives will be unacceptably frequent. There appears to be a *tension*, along a spectrum, of how distinct gestural interaction appears from a system perspective. On one end, the designer could make gestural interaction movements so unusual that they would never appear in background activity, and thus are immediately recognizable as foreground activity. These sort of movements are perhaps more effortful, take longer to perform, and are unusual, perhaps requiring a larger cognitive load to perform. On the other end of the spectrum, the gestural interaction movements are less distinct, but much more prone to detection errors. When designing the Background Activity study, we chose gestures that were intentionally not very distinct. Furthermore, we intentionally did not give participants any training or feedback on how to reduce false positives. Given that the present system will be used by performers who have an opportunity to practice with it, we should be able to mitigate some of the issues found during the background activity study.

6.2.2 Performer Experience

While the gestural interaction in our workshops was light, we found that performers were very unaccustomed to it. Lacking confidence in the system's response, their body stopped being expressive and they would fixate on the visual feedback (a monitor facing them), moving very carefully until they were certain their gesture was correctly received. It is clear that it is very important, for the performer, that the actions be fluid, and it is easy to alternate between interaction and performing. Continuous feedback may, in fact, be a poor choice as it distracts and entrances performers.

From the performer's perspective, the relationship of their character to the interaction is interesting. How much of the performer stays in character while they interact? In the extreme case, the performer drops their character entirely, focuses their entire attention on the interaction, and then returns to their character. This seems undesirable. On other other end of the spectrum, the interaction may be performed in-character, the manner of the interaction adjusted so that it appears like something the character would do. To be clear, the audience may still understand that the interaction is for the system, but it is done in such a way that it does not seem out of character for the performer. A dissonant interaction would be if a violent diagonal slash was required to interact, when the performer is trying to play an introverted character who avoids occupying too much space. A middle ground between the two ends of the spectrum may be preferable, where only a limb may leave the character, to return once the interaction has finished.

We expect that the primary cognitive activity of the performer will be performance, while the system usage is usually occasional. One possible exception could be where a performer takes direct control of a playback performer for a period of time. Unlike a music concert, where the primary point of the performance is the performer interacting with the instrument, we expect our system to be useful equipment to be used during the show. Continuing the music concert analogy, our system is a volume knob on an amplifier that the musician must adjust, or the string-tuning apparatus on a guitar.

It is important to note that the performer is already fully engaged in a cognitively-intense task, even before we are adding our system as another tool available for use. With improvisation, this is even more true, so interactions with the system must be as cognitively lightweight as possible. We do not intend to do an in-depth cognitive analysis of the task of using our system, but simply intend to keep these limitations in mind. The cognitive task of improvisation has been studied in detail by others [Fuller and Magerko, 2011, Magerko et al., 2010].

Conveniently, designing gestures for use by performers may be easier in some sense than for typical users. One issue commonly noted in including whole-body gestural interaction in day-to-day life is a negative novelty effect; that users may feel self-conscious performing gestures in front of onlookers, who may not be aware of the system at all [Rico, 2010]. We suspect that this will not be an issue with performers, who are familiar with such feelings and aware that the onlookers are consenting to watch something novel.

6.2.3 Audience Perception

While we did not evaluate audience perception during the workshop, this has been a concern. The problem of observer perception of system interaction has been studied briefly in general [Zigelbaum, 2008]. In our reading of a large amount of background on the usage of technology in theatre, the technology becomes the focus and the topic show is a critical deconstruction of technology. Our desire is not that the technology is the point of the show, but merely that it is a useful tool. There are a few approaches to exposing the interaction to the audience: on one end it could be entirely hidden, so the stage appears to respond as if by magic; on the other hand, it could be entirely clear, so much so that the audience members could walk up and use the interface immediately after the show — the latter is preferred.

A good illustrating example of onstage interaction in the context of performance is tuning a guitar at the beginning, or in the middle, of a performance. The audience, unless they are extremely naive, understand that this is not directly part of the performance. However, the guitar tuning moment is not so exposed that an audience member could pick up the guitar themselves and perform the same interaction.

Some perspective here comes from Brenda Laurel: she recalls Aristotle's six elements of structure in drama. *Thought* is one of these - the actor's portrayal of a character means they must imply a series of empathizable thoughts, inferred internal choices of cognition, emotion and reason [Laurel, 1991]. In our guitar tuning example, from the audience's perspective, whether the guitarist is playing or tuning, both contribute to the performance. For the audience observing an actor, all of their actions are perceived to be part of a construction of a character. If a performer does a non-character activity, and it is unclear

to the performer whether this is so, this can be very confusing. A common example is when actors' eyes briefly look at the teleprompter in live shows, such as *Saturday Night Live*. Typically, when theatre performers do non-actor business, such as bringing a prop onto the stage, lighting is intentionally set to distinguish these activities as out-of-character.

To contrast these concerns, gestures to coordinate the performance between performers are very common in modern improvisation. These are discussed in detail in the Background. Typically, modern improvisation is much less formal, and there is less of an attempt to seriously suspend the notion that the audience is looking at performers on the stage.

The audience's awareness of the system is one of the most interesting problems we face. It is an explicit goal of this work that we do not want the technology to be confusing or mysterious to onlookers. The technology onstage is to be a useful tool that performers are using to make art. Again, to draw an analogy to a guitar performance, it is clear to the audience that a guitarist is using the guitar to make the music they hear, but this does not require that the audience members be talented musicians themselves.

One interesting case is a scene in Blast Theory's *10 Backwards* where the actress uses a remote control to record and play back video of herself eating a bowl of cereal [Theory, 1999]. A camera on a tripod is facing her, and behind the camera is a large projection screen, where she and the audience can see the output from the camera. She records herself using the camera, and then plays herself back, trying to imitate her actions with slight exaggeration. She then records this exaggerated performance, and then repeats the same self-imitating procedure, eventually using the remote to go forwards and backwards frame-by-frame to imitate exact facial expressions (Figure 6.1). This is certainly a novel use of technology on stage. However, the technology itself is familiar to the audience, and so will be viewed differently than in our case.

The performance-going audience is engaged in a sort of cognitive activity not often studied in Human-Computer Interaction literature. They are not a casual onlooker, who happens to see an interaction by chance, but rather they have explicitly come to witness the interacting performers. Unlike a typical business presentation, it is much more accepted that the information being transmitted is not to be immediately clear, but is *open to interpretation*. We take this as the audience's primary task: interpretation of the performer's actions on stage. As such, the nature of system interaction overseen must be carefully designed (and enacted) to either provide the intended interpretation, or not to muddle the interpretation that the performer is trying to emit with their non-interaction activities.

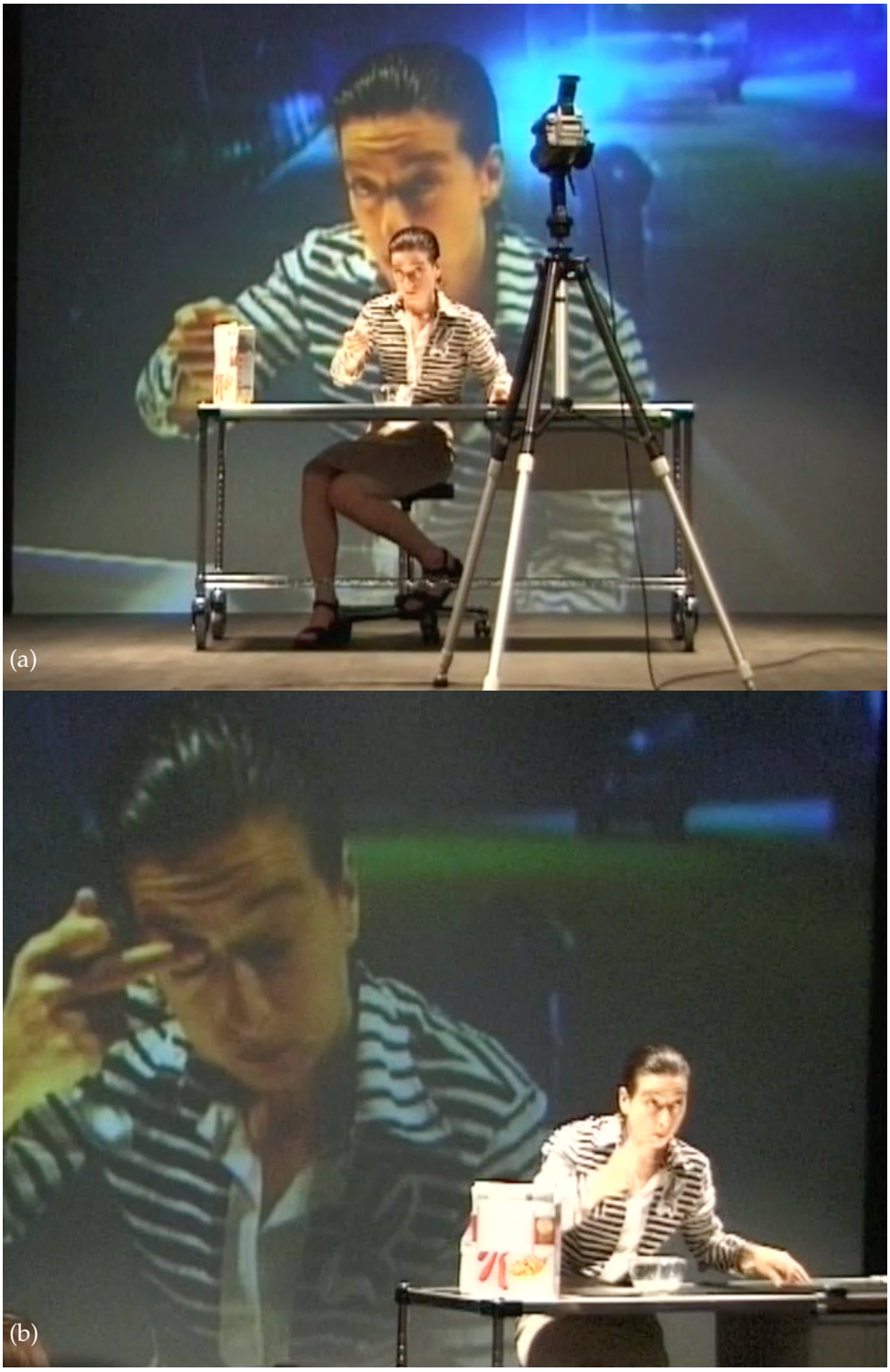

Figure 6.1: A scene in Blast Theory's *10 Backwards*, where a performer uses a standard remote control (a) to record and re-project herself, to try to imitate her actions eating breakfast (b).

6.3 Design Principles

During this work, we created several design principles to understand interaction design in our specific context. Each of these design principles represent a spectrum along which a specific instance of interaction design for theatre may be placed. It so happens that each of these principles strongly aligns with the philosophical purpose of Improv Remix. For example, one of our design principles is *Exposure*, and we wanted Improv Remix to have a high degree of exposure, for reasons we shall give below. However, there are reasons why a theatrical interaction designer would wish to have a low degree of exposure. One contribution of this work is the design principles as useful language when thinking about the type of theatrical interface that one wants to build.

The design principles presented were developed during iterative work on Improv Remix, as we developed a language to describe why we had to make certain design choices. Essentially, we had to make meta-choices about these design principles so that they most successfully separated interaction design choices that felt like they aligned with our goals, and those that did not.

The genesis of the design principles was when analyzing the coordinating gestures that appear in theatrical improvisation. The design principles arose from trying to describe what was successfully functional about the coordinating gestures, and thus what would (hopefully) be functional in Improv Remix.

At this stage, it would be good to remind ourselves of the context and purpose of Improv Remix: improvisational theatre can go anywhere, and a general improvisation tool should restrict improvisors as little as possible. There is a temptation when thinking of a specific scenic moment to imagination an interaction technique that would suit it particularly well. While this may be suited for a highly-rehearsed moment in scripted theatre, this is not the case for improvisation.

6.3.1 Exposure

Performers are engaged simultaneously with performing and interaction with a system — interaction with the system is not the sole action in their show. If an interaction is said to be more exposed, then it is more clear that it is an interaction with the system, as opposed to part of the regular actions of performing. The relevant observers may be audience members, or possibly other performers, who may need to anticipate the interacting performers' actions as part of the story-making activity. For example, consider a waving gesture: this could be a wave to an imaginary character offstage, or it could be an interaction with the system. The degree of exposure is how clear it is whether an action is an interaction, or a a character action. When the audience is viewing a performance featuring interaction with the system, they may ask *"Was that an action of the performer, or of the character?"*. If it is difficult to answer this question, then interaction is not exposed.

The degree of exposure is independent of whether observers understand the purpose, or referent, of the interaction. In the Background Chapter, we cover Reeves et al.'s framework of performative interaction, which discussed whether manipulations were hidden or visible, and whether effects were hidden or visible [Reeves et al., 2005]. However, that framework does not include an understanding of how well system interactions distinguish themselves from performance.

In Improv Remix, as with improvisation's coordinating gestures, we aim for a high degree of exposure

— interactions with the system should be distinguished from the performance. This assists in the coordinating act of story-making on stage between the performers, and, for our case at least, we do not wish that audience members are confused about the intention of performers' actions. There are cases where low exposure may be useful for a show, where performers are perhaps competing with each other for control of the system, or audience members are meant to be left slightly in the dark as part of the show's theme.

It is possible to interpret this definition of exposure as containing the assumption that an action by the performer is either an interaction with the system, or in-character acting, but not both. However, earlier, we presented examples of coordinating gestures that feature acting, such as an in-character sweep. This will be discussed further in *Semantic Capacity* below.

6.3.2 Neutrality

An interaction with a system has *Neutrality* when it has little semantic value.

For Improv Remix, we desire our interaction techniques to be highly neutral. An improvisor who wants to play an office worker, a homeless person, a cold-hearted person or a warm, loving person must be able to perform the gesture without it seeming out of character for them. This is best illustrated by negative examples: a violent, quick diagonal slash with an open palm would be out of character for a benignly cheerful character. So would raising both arms above the head for a decrepit character.

Of course, in scripted performance it may make sense to design a specific interaction so that it carries a semantic value. This is even more sensible if the interaction is intended to be done in-character. However, the focus of this work is on improvised performance.

Even if an action is highly exposed, as in the audience has easily tell whether it is an interaction or in-character, neutrality is still relevant. If a performer is playing a loud, aggressive character on stage, then has to jump to the side to a small interface to trigger the next lighting cue, this contrast while leave an impression in the audience's mind, likely a comic one.

6.3.3 Semantic Capacity

This design principle is similar, yet subtly different to neutrality. While neutrality describes how much innate semantic value is in an interaction, semantic capacity describes how much semantic value may be applied to an interaction at the time of a single performance of it. Interactions that may be performed in a variety of ways and still be recognized by the system have high semantic capacity.

If an interaction has high semantic capacity, it is easier for the performer to perform it in-character, as they are not forced to perform it in a specific, constrained way, but may customize their interaction, to make it yet another way the character is expressed to the audience.

For Improv Remix, we desire interactions that have a high degree of semantic capacity. This was a common deciding factor when designing interactions — we would invent a possible interaction, but then realized that the requirements on its performance lacked flexibility. The best way to characterize this requirement was not simply flexibility for flexibility's sake, but when we sought deeper, it was so

that performers were given an option to act while interacting.

It is difficult to come up with examples where low semantic capacity is desired. Perhaps an interaction designer wants a specific interaction to always appear to be out of character; by making its performance requirements so specific, that the performers appear rigid to the observers, and thus implicitly informing them that this is a transitional, out-of-character moment. It is more likely that low semantic capacity would result out of difficulty of design — for example, perhaps the gesture detector simply requires a strictly-defined gesture.

It seems that in general high semantic capacity is desired, but difficult when also hoping to ensure system interactions are consistent enough to detectable. A careful balance must be found.

6.3.4 Graceful Error Recovery

Any system will not detect gestures perfectly — this principle concerns what happens when errors occur. This principle is concerned with whether error recovery is not merely possible, but if it is graceful from the point of view of the performer and the audience. The term graceful here is slightly different than how it is normally used in the field of interaction design in general. Graceful Error Recovery in the context of interaction for performance means that inevitable errors must do not derail the show and it should be clear to the audience that an error occurred. If a system is designed to allow grace, then a performer may re-attempt the interaction while still maintaining a semblance of controlled performance.

In Improv Remix, we do desire graceful error recovery, our justification being that we aim for Improv Remix to be a tool that fades in the background when not being used. It is possible and common to have a performance where awkward error recovery is desired; perhaps where the performer is struggling to perform a difficult interaction and this struggle is an explicit part of the performance. An excellent example of this is many Japanese Game Shows, where failure on the performer's part is designed to be humorously catastrophic.

6.4 Interaction Mapping

For a given set of features required by the system, the interaction designer must map interaction gestures to features. Of course, during the iterative design process, features may be added or eliminated based on how easy they are to implement, either technically, or within a functional interaction framework.

Whole-body interaction is necessarily more imprecise than keyboard, mouse, or even touch input. During the design process, we found it useful to write the list of features, and apply categories. One such categorization is if a feature's invocation is *time-sensitive*, i.e., must be done with precise timing, and if it is *value sensitive*, i.e., the user must have fine control over the feature. These categories were the most helpful for determining how to design invocation of these features. The listing created for *Improv Remix* is shown in Table below.

Feature	Time-Sensitive	Value-Sensitive
A method to indicate recording	—	—
start recording		
stop recording	X	
A method for displaying an overview of videos and selecting a video to be instantiated as on the stage		
If a video contains multiple performers, a method to choose to instantiate a subset of them		
A method to control playback performers	—	—
play/pause	X	
playback position control (i.e., scrubbing)		X
looping or not		
triggering a specific utterance	X	
A method to remove playback performers from the stage		
A method to control the position and horizontal flip of videos on the scrim in front of the stage		X
A method to crop scenes (in time)		X

7

Improv Remix

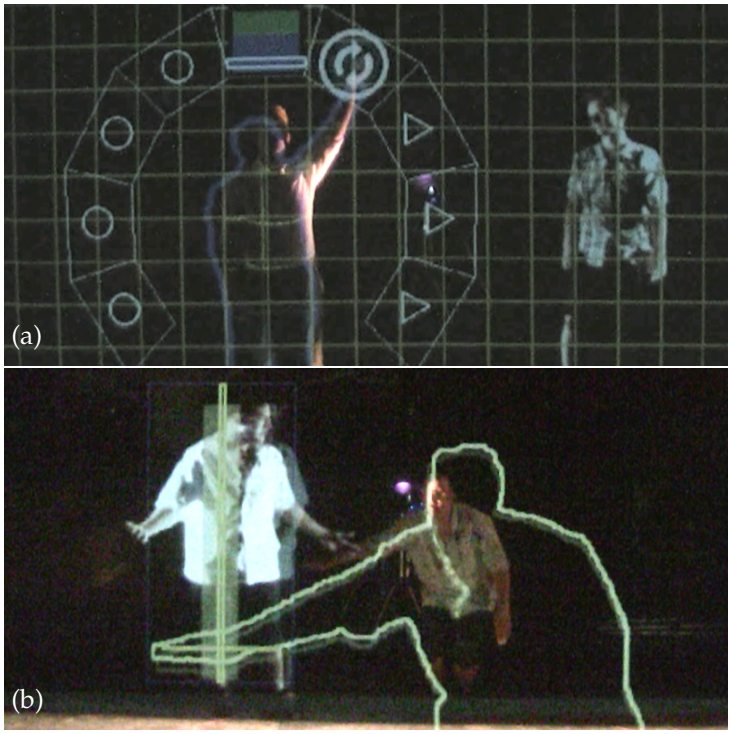

Figure 7.1: An overview of Improv Remix. In the first frame, a live performer (left) accessing scenes to load our novel Vitruvian Menu, and a video of a playback performer (right) is paused before playback. In the second frame, a live performer (right) scrubs his previous performance (left).

7.1 Core Physical Setup

Our objective is to solve the problem of co-locating video projections and live performers for interaction during performance. There are several sub-problems here:

- performers must have convenient **sight lines**
- live and projected performers must have a **similar appearance**
- live performers must have a **capturable depth image**
- live performers must have **visibility control**

These requirements come out of our workshops, and other observations. We shall discuss how we have implemented a system that solves all of these problems. This part of the work was done in heavy collaboration with PhD drama student Montgomery Martin. We will discuss the setup requirements in more detail, as well as Martin's prior work, then give a description of our chosen setup, followed by implications of the chosen setup. We close with limitations and possible extensions of this setup for other scenarios.

7.1.1 Setup Requirements

Sight Lines

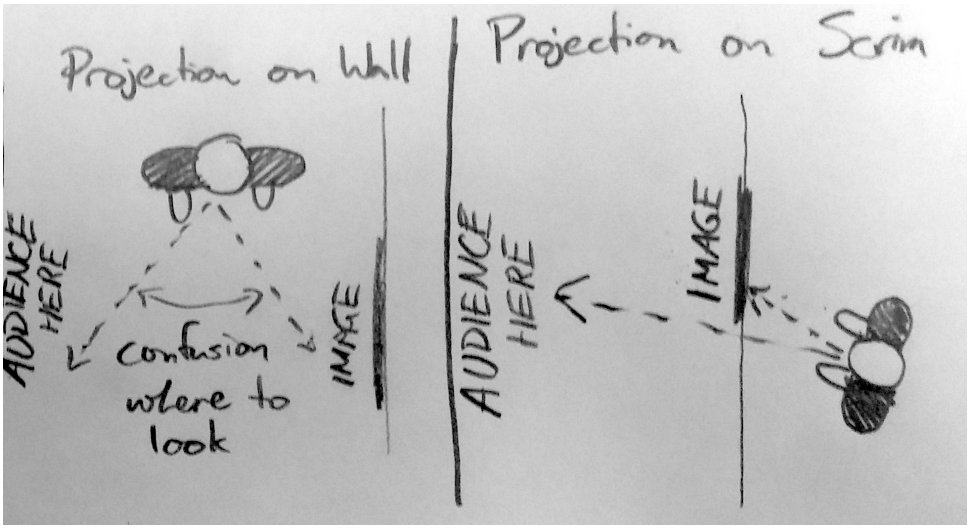

Figure 7.2: Performer orientation issues with the projected performer in different positions. On the left side, the live performer is between the audience and the projected performer. On the right side, the projected performer is between the audience and the live performer.

We discussed this issue as "Performer Orientation" in the Workshops chapter — the performers must be able to orient themselves so they can comfortably see the projections and the audience. On the left side of Figure 7.2, we see how the problem appears for performers. The right side shows how Martin solves it in his setup, which we used for ours.

Similar Appearance

The live performer and projected performer must have a similar-enough physical appearance that they are not jarringly different. The projected performer's features, particularly their facial features, must be clear enough to be visible to audience members far away. Additionally, the live and projected performer must appear to occupy the same physical space. Their co-occupancy does not need to be so tight that they can touch each other, but they should appear to be near enough together that it feels like they could be in an intimate relationship.

Capturable Depth Image

We need to be able to capture the performer's depth image for a few reasons. First, it is the easiest way to get a clean silhouette to isolate the performer in colour video. Second, depth will be used for interaction with the system. Unfortunately, we have experimented and found that infrared structured-light depth cameras (e.g., the Microsoft Kinect) do not transmit well through a scrim, so depth capture from the front will not work.

Visibility Control

During the workshops, we defined scenes as starting from when a performer appeared on stage until they left it. Unfortunately, this meant any recorded scene had junk at the beginning and end, where the performer walked on and off the stage, often out of character. While we could have a gesture for controlling recording more precisely, or a series of techniques for clipping video, it would be nice for the performers to have some way of appearing in the middle of the stage.

7.2 Prior Work

Martin previously experimented with a system for merging a live and a remote performer in the same space on the stage (Figure 7.3). He used a setup where a black fabric scrim hung vertically in the middle of the performance space. Stage lighting was carefully set up so the stage was lit everywhere *except* the scrim. The consequence of this lighting setup is that the areas of the scrim where an image is projected are opaque, whereas the rest of the scrim is transparent. Lighting was such that the live performer could walk both in front and behind the scrim, and she could turn to face the perceived position of the projected performer in space. This technique is a variant on the widely-used stage technique of *Pepper's Ghost* [Steinmeyer, 1999].

Figure 7.3: Scrim with a projected performer (left), and a live performer (right) behind.

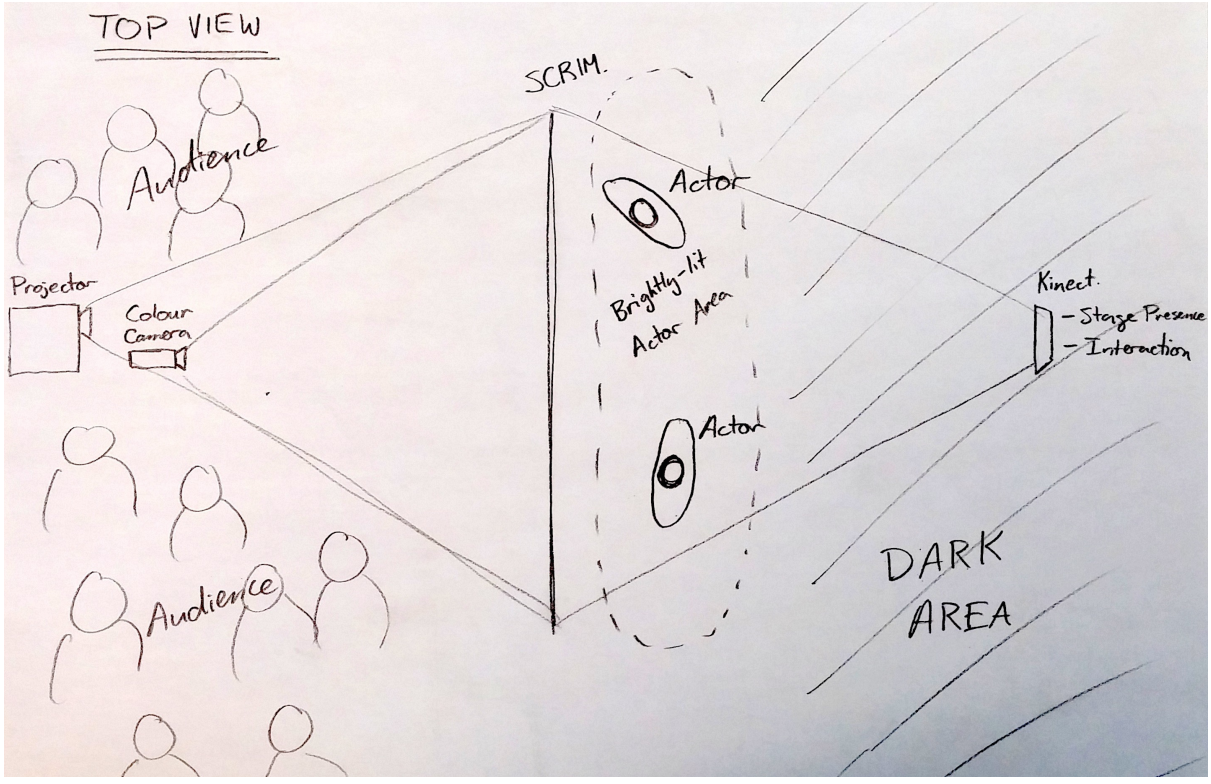

Figure 7.4: Final physical setup design.

7.2.1 Physical Set-Up Description

Figure 7.4 shows our final physical setup. This solves all the requirements given above sufficiently, but creates a few implications we will cover in the next section.

By having the scrim between the performer area and the audience, we have solved the sightline problem, so that performers can always present out to the audience and see the projected performers.

By adjusting the projector and stage lighting brightness, it is also possible to have a similar appearance between the live and projected performers. Martin has noted that while the scrim is barely visible to the audience, it changes the appear of a live performer slightly when they step behind it, giving them a slightly blurred appearance. This apparent reduction of resolution of the live performer makes them appear less real, and more similar to the projected performer.

In what felt like an initially odd choice, we have placed the Kinect *behind* the performers. With careful calibration, this will allow us to mask out the live performers projected back on the stage.

When stage lighting is done properly, the transition between the well-lit and dark areas is sudden. We have labelled the area immediately behind the performer area as dark. When we made this test setup in a controlled theatre environment, we found performers were effectively invisible in this area. This allows a performer to start and end their appearance on stage anywhere, by simply moving between the dark and light areas.

7.2.2 Setup Implications

There are two interesting implications that come out of choosing this set-up, both relating to placing the Kinect behind the scrim.

The first is that we have *interaction from behind*. Most depth-based interaction in the literature assumes a depth camera in front of the user. We shall be exploring interaction techniques that don't require a view of the front of the user's body. As such, subtle interactions in front of the torso will not be possible. The interaction techniques we explore may have applications elsewhere in similar setups, where whole-body interaction is desired while the user is physically close to an object, such as a billboard. This has been somewhat been explored in Shadow Reaching [Shoemaker et al., 2007].

The second is *invisible interaction* — by backing up out of the well-lit performer area, but still remaining in view of the depth camera, performers can interact without being seen. In fact, choosing to be visible to the audience or other performers while performer certain interactions can be a powerful artistic choice. For example, choosing not to be seen can indicate shyness or mischievousness.

7.2.3 Limitations and Extensions

One limitation we have found is that, with the bright lighting required for the stage, it is hard for a performer to see any detailed visual feedback on the scrim, particularly if it is vertically higher, forcing them to look more into the lights mounted on the ceiling. Live performers can see projected performers sufficiently, but not small details. This creates an opportunity: if the interaction feedback for a hidden performer in the unlit area is subtle, both the audience and a hidden performer can see the results of

their interactions, but they will not be immediately apparent to the performer in the well-lit area.

The bright stage lighting, which no doubt contains a high infrared component, appears to blow out the structured light pattern of the Kinect for small limbs. This gives a messier depth silhouette than we hoped, but it may be possible to fix with a dilation image processing pass. We shall experiment with whether this same effect happens with the recently-arrived Kinect 2, which apparently uses time-of-flight sensing.

While it is not necessary for our purposes, it is possible to extend this setup, by effectively doubling it as a mirror image, each side containing a depth camera, colour camera, projected and well-lit area near the scrim but not directly on it. We have experimented with two Kinect cameras and found that they can be on opposite sides of the scrim and detect performer silhouettes sufficiently without too much interference. A setup like this would allow live performers to walk around a projected performer without too much visual disruption, by software carefully choosing which video source (front or back) to project video from given the performer's position.

7.3 RGB + D Video Buffer Backend

We shall give a listing of any software developed especially for the system here. As this is a proposal at this stage, it will be incomplete. The major pieces of software developed are the video data serialization back-end, a system for gesture detection, as well as the design of any actual interaction. Interaction techniques will be described fully in later sections.

7.3.1 Video Data Serialization

We have completed multiple projects by now which require a video and audio buffer that is simultaneously readable and writeable, for up to 1-2 hours of video. The bulk of video requires that it be compressed in some way. Additionally, with the uneven way that the Kinect delivers colour and depth frames, we am unable to anticipate a consistent framerate. After the first few iterations, we gravitated to an approach of a large binary file for video, where each frame is in the RIFF format. Frames are added to this file sequentially, each frame with a timestamp. During playback, frames are displayed when their timestamp is passed by a playback marker. In our implementation, colour, depth and skeleton are held in their own frames; they are not kept in sync.

This approach has been used in LACES and Background Activity with a C# codebase. With more recent work on the final prototype described in this chapter, we have rebuilt this system in a (backwards-compatible) C++ codebase to get higher performance.

We have published, low-level components of the codebase online on GitHub. Riffer is a framework for storing flexible data types into chunks, and reading and writing large files with high-performance in the RIFF format, including seamless read/write access. A sample usage is as follows:

```
//creating a chunk and storing it in a capture session
rfr::CaptureSession cs("./capture.dat");

rfr::Chunk chunk("colour frame");
int width = 640; int height = 480;
chunk.add_parameter("width", width);
chunk.add_parameter("height", height);
int64_t timestamp = 1234567891011;
chunk.add_parameter("timestamp", 1234567891011);
cs.add(chunk);

cs.close();

//re-opening the capture session and retrieving a chunk
rfr::CaptureSession cs_opened("./capture.dat", false);
cs_opened.index_by("timestamp");
cs_opened.run_index();
rfr::Chunk opened_chunk_by_timestamp = cs_opened.get_at_index("timestamp", timestamp);
```

Kriffer wraps Riffer, and manages data formatting specific to the Kinect SDK version 1.8. Kriffer's KProcessor wraps a Riffer CaptureSession for a single Kinect, as shown below:

```
int num_kinects = kfr::get_num_kinects();
if (num_kinects < 1) {
        std::cout << "No Kinects found. \n";
        return;
}

kfr::KProcessor kprocessor(0, "./capture.knt");

//kprocessor is now capturing from kinect 0.

kprocessor.close();
```

We will most likely transition to using the Kinect 2, and kriffer will be upgraded for it.
Riffer can be found at: `https://github.com/dustinfreeman/riffer/`
Kriffer can be found at: `https://github.com/dustinfreeman/kriffer/`

7.4 Software Development Process & Early Prototypes

I have documented the development process, since developing software to be used by performers in an artistic context is a major contribution of this work. While the development process was continually iterative, there were a few major milestones that involved formal invitations of performers or audience, with installation in a theatre space outside of a lab. The milestones encouraged us to create a somewhat self-consistent set of interaction techniques. These interaction techniques were taught to performers who used the system at that time.

FOOT 2014 - February 2014. A one-hour demonstration of the system as part of the Festival of Original Theatre 2014 Conference[1]. We recruited two improv performers and one dancer to meet for a few rehearsals before the presentation to rehearse some use cases for the demonstration. Audience size was approximately 40 people.

Luelley Massey - May 2014. A five-day-long installation in the Luelley Massey Theatre, which was an opportunity to test lighting and interface layout. Some performers were invited in during the last two days for testing. There was no formal audience presentation.

Final Showcases - Storefront Theatre - late July & Early August 2014. A series of three showcases of the final interfaces in front of a live audience, described in entirety in Chapter 8.

When we refer to the development process in the rest of this section, we will refer to these different milestones. The primary interaction approach for FOOT was to use dwell buttons, but this was found to have a host of problems we shall enunciate later. In between FOOT and Luelley Massey, we (Monty and myself) invented the Vitruvian Menu.

7.4.1 Interaction Medium of Whole-Body Interaction

While there are a few options for interaction, we have chosen to focus on *whole-body gestural interaction*, instead of any sensors or wearables, or any other controllers held on the body or placed on the floor — we acknowledge this as a design choice. This choice is partially justified by our feeling that an extra physical device may provide a feel of encumbrance that makes the performer feel like what they are doing is different from regular theatre. The trade-off we are willing to accept is that the performer must perform gestures for the system. Voice has been mentioned as a possibility, but we am not as familiar with it and in our experience it has been less reliable, and we feel like vocal interaction is more disruptive to a performance than gestural interaction. In a theatre environment, it is easier to isolate gestures than voice, as we do not have control over the behaviour of the audience.

From our experience of uninhibited background activity, we know that, in noisy environments with multiple users who are using their body expressively, skeleton tracking may be unreliable. There is also a lack of flexibility when tying interactions to certain body parts — if a performer decides to play a character with a limp left arm, or with their right and left hands that have been super-glued together, they may only realize too late that they are unable to perform an interaction they desire. As such, we should free interaction from any skeletal limitations, instead looking for movements that appear significantly different from background activity in performance.

[1] http://foot2014.wordpress.com/

Figure 7.5: The FOOT system UI. The red circles on either side are the record buttons, and the lower grey squares are the library buttons. The green rectangle overhead is the stage occupancy indicator, showing that the current user is on the right side of the stage.

7.4.2 FOOT System Overview

We will describe the first major iteration of our system, made for the FOOT conference. From this, we learned some lessons about our problem space that aided the design of the final system.

Figure 7.5 shows the standard view of the FOOT interface, as it appears most of the time during a performance. Throughout this section, photographs will show a performer silhouette for illustration, but this would be disabled during performance. During the majority of the performance, the only interface elements visible are:

- the stage occupancy indicator
- the record buttons
- the library buttons

The *stage occupancy indicator* was an idea to simplify remixing of scenes, treating the left and right sides of a stage as separate pieces of a puzzle that can be fit together. When the performer is on stage right, the stage occupancy indicator is green. When on stage left, the stage occupancy indicator is red. When the performer overlaps the sides, the stage occupancy indicator is white.

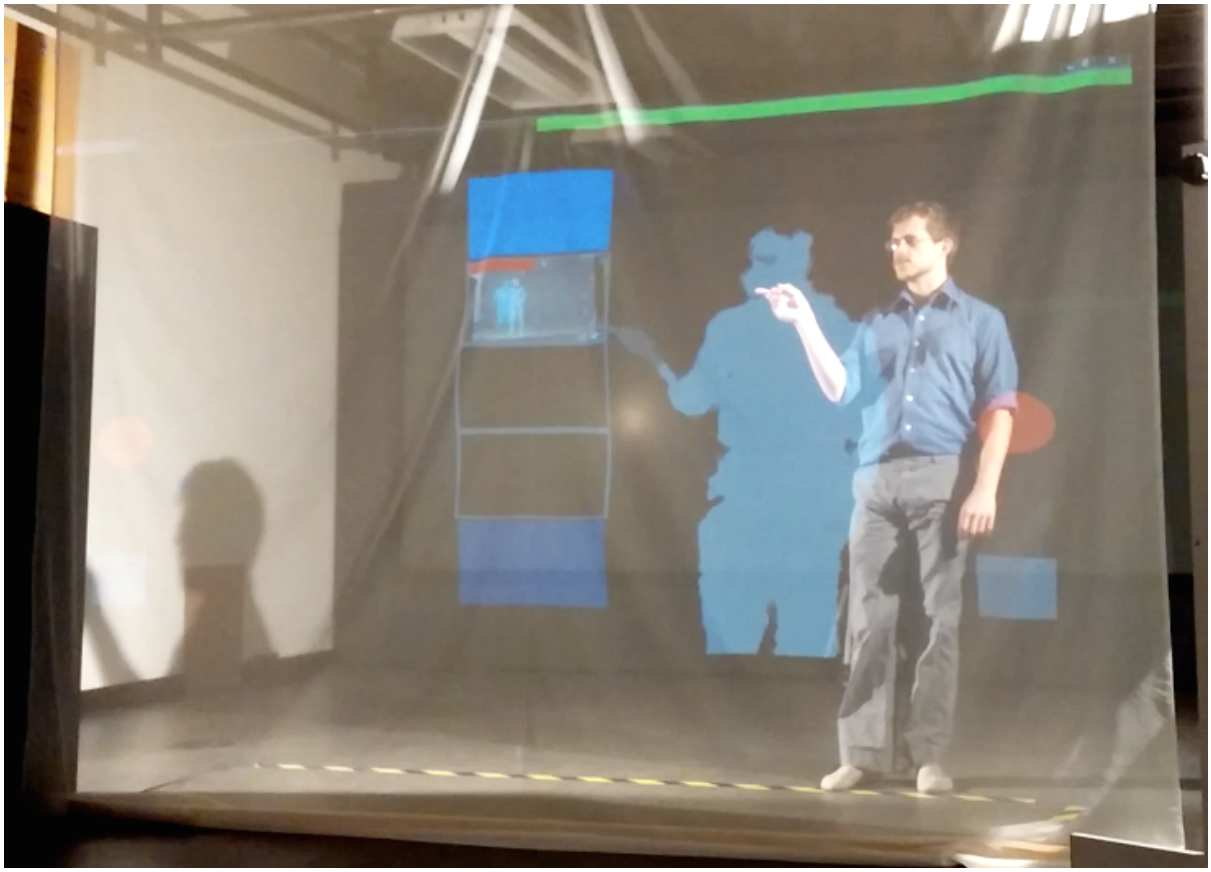

Figure 7.6: When the user invokes the library, the list of recent scenes appears in the middle of the stage. Three are shown at a time, and buttons above and below the list move the list up and down. Scenes have stage occupancy indicators themselves. In this case, the scene in the library was recorded on the left side of the stage (as visible from the red stage occupancy indicator), and the live performer is currently on the right side of the stage (as visible from the green stage occupancy indicator).

When a record button is invoked, a 3-second countdown is shown, and then a new scene is recorded of the performers. The record is stopped by the button being invoked again.

When the library is invoked by the library button, it appears in the centre of the stage as seen in Figure 7.6. Previous scenes are shown as a series of thumbnails. The scene thumbnail itself is a button, and invoking it instantiates the scene as a playback performer on the stage. The stage occupancy indicator hints to the performer which scenes are cleanest to instantiate, either with themselves or with each other.

Figure 7.7: One performer scrubs a playback performer by inserting her hand into the scrubbing area. A yellow line indicates the current play marker.

When the scene is instantiated on stage, as seen in Figure 7.7, it plays initially. The performer can scrub through the scene from start to finish by inserting an appendage into the scrubbing area, and raising it up (towards the beginning) or down (towards the end). Retracting the limb makes the scene continue to play from its last position. An orange button is provided above to delete the playback performer.

Dwell Buttons

Almost all elements in the FOOT interface are dwell buttons, which activate if the user's silhouette occupies their space for a period of time. This is not ideal for some of our interactions, as it does not satisfy our time-sensitive requirements. The dwell buttons are fixed in the stage space, which means that performers must get to the buttons to interact with them. While inconvenient, this may be beneficial — we have chosen to place the standard dwell buttons at the extreme edges of the stage, and since most performer activity is towards the middle of the stage, false positives during performance were extremely rare.

Once the pixel area of a button is filled by a user's silhouette above a threshold, these buttons count down time until a pre-designated value. The value we used was 600 ms, which seems like a long time, but as it is much slower to move a limb than a mouse, this felt fast. Dwell buttons use a filling-pie visualization to indicate how close they are to activation, and fill and unfill at the same rate.

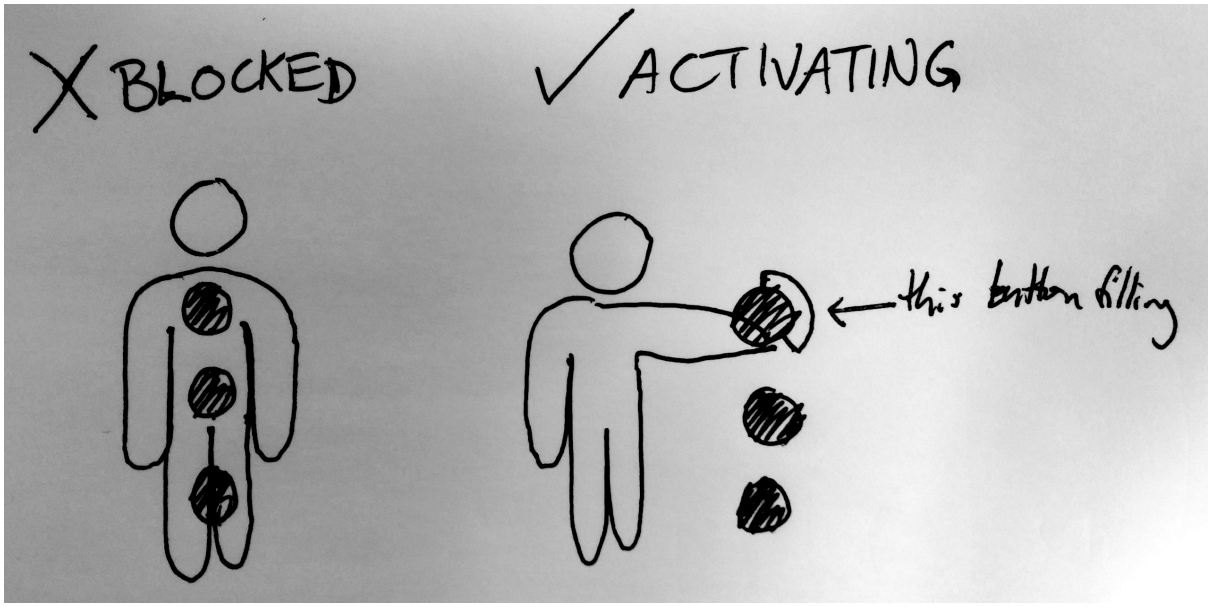

Figure 7.8: A demonstration of the blocking feature of vertically-aligned dwell buttons. On the left, the user is standing over top of multiple buttons, but they are not filling and will not activate. On the right, the user stands to the side, and hovers a limb over a single dwell button, which will fill until it activates.

One innovation that came out of the FOOT version of the interface is that only one dwell button can fill at a time. If multiple dwell buttons are filled by the user, the filling of each of them is frozen. This is equivalent to blocking input if multiple keys on a keyboard are mashed at once. This solved a lot of problems, such as when the user walks off stage and touches both the recording and library invocation buttons at the same time. Designing these buttons with a blocking behaviour led to us designing the interface as vertical columns of UI elements — this orientation of elements is safe from accidental activation (see Figure 7.8).

It feels like an odd design choice that the button to invoke the library (on the lower extremes of the stage) is far away from where the library appears on the stage interface (in the centre). This is, in fact, intentional. Ensuring the user goes to the side to invoke the button means that they will be out of the way from the library when it appears, and can then turn to activate their desired scene in the library. As opposed to typical cursor-based interaction, in our case it would be a problem to have several UI elements near each other. In the case where one performer goes to the side to invoke the library, and another performer happens to be standing where the library appears, the blocking nature of our dwell buttons prevents any accidental activation.

There is one special circumstance where we allow *fast invocation* — the activation of a dwell button instantaneous on contact (0 ms instead of 600ms). When a scene is being recorded, any part of the user touching the record button instantaneously stops the recording. It is important to be able to stop recordings quickly, otherwise every recording ends with an awkward period of the performer hovering over the stop record button. We found that this reduction of dwell time did not create any additional errors, since when the users were in a performing mindset, they were particularly attentive to the recording button and would not accidentally invoke it.

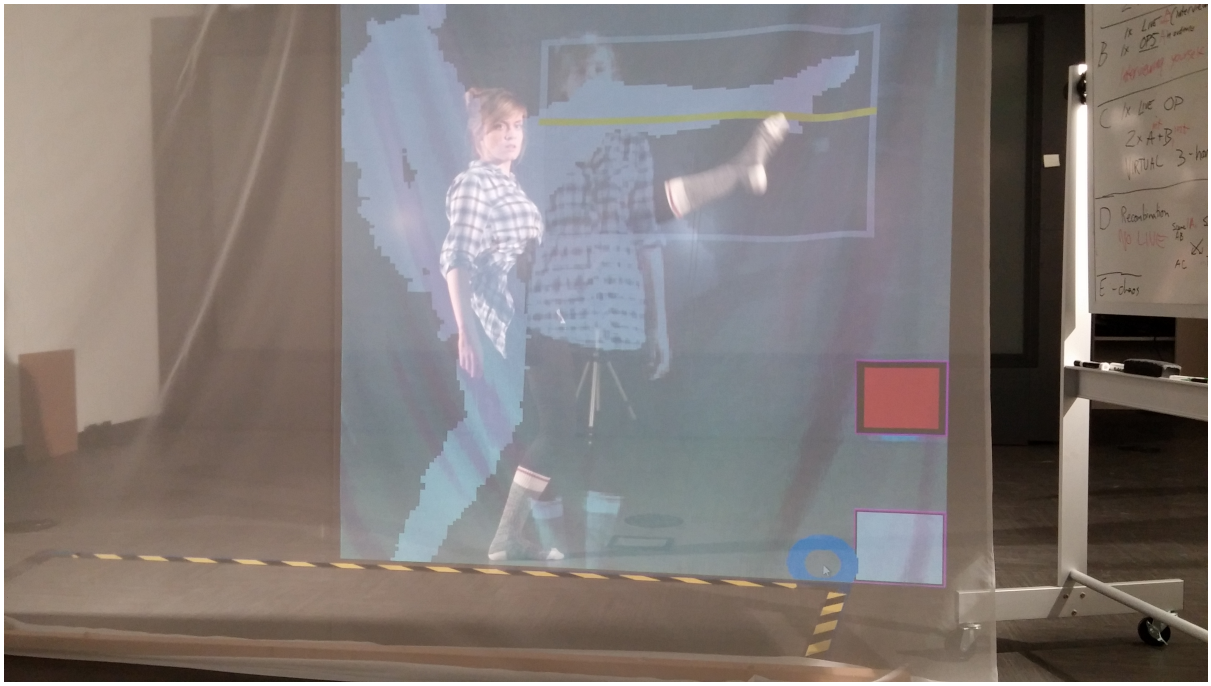

Figure 7.9: A performer uses her foot to scrub a playback performer.

Initial Observations

We workshopped the FOOT iteration of the interface over two one-hour sessions with three performers, making minor changes after observing performers' usage of it. We then did a 45-minute presentation at the FOOT (Festival of Original Theatre) 2014 conference in Toronto. The presentation took the format of a demo session, where the performers ran through a few skits showcasing the usage of the system, while the two researchers described how the system worked to an audience of about 30. Audience reaction as a whole was mildly enthusiastic, but in an academic sense, as it was very clear that the system was too immature to be put in a show.

We observed that the vertical alignment of interface elements gave opportunities to interact with both arms and legs (Figure 7.9). Even though arms tended to be used more, we found that performers enjoyed the flexibility of choosing to use their arms or legs to activate a button or scrub. We defined the scrub-to point in the playback performer video as the centroid of the user's body in the scrubbing area — one cool consequence of this is one performer, who was a trained dancer, could step into the scrub space and squat up and down, adjusting the play point of the video by a, literally, whole-body interaction.

We have shown the Kinect silhouette as feedback in the figures above. We showed this to performers as they practised finding the location of interface elements until they could hit them roughly consistently, and then hid the silhouette. Unfortunately, the performers still had occasional trouble hitting the buttons. Some of this targeting difficulty was caused as the silhouette of the performer scaled as they moved towards or away from the Kinect camera.

We also observed how the movements of system interaction and performance appeared from the perspective of the audience; it was jarringly clear when the performer was finished performing, and the intention of their body transitioned to interacting with the system. To illustrate this in a similar sense,

imagine an performer has completed a wondrous, uplifting monologue on the stage, and then turns to exit the stage, and appears to concentrate very hard on manipulating the door handle, in painful contrast to the performance the audience just witnessed. The interface the performer is using becomes conspicuous, and a distraction to the audience's thoughts. This problem may be solvable with a mix of interaction design, interaction improvements (polish) and performer training and familiarity with the system. It is interesting that the goal of interaction design differs strongly between the audience and system perspectives for the system, interaction should appear as different as possible from the motions of acting; whereas for the audience, this interaction should appear different, but not jarringly so. Figure 7.10 shows a performer trying to find a button during the FOOT 2014 demonstration.

Figure 7.10: A performer tries to find a button during the FOOT 2014 performance. The apparent location of the performer's hand and the dwell button appear different due to parallax, which is also a problem for the performer.

7.4.3 Using and Invoking the Vitruvian Menu

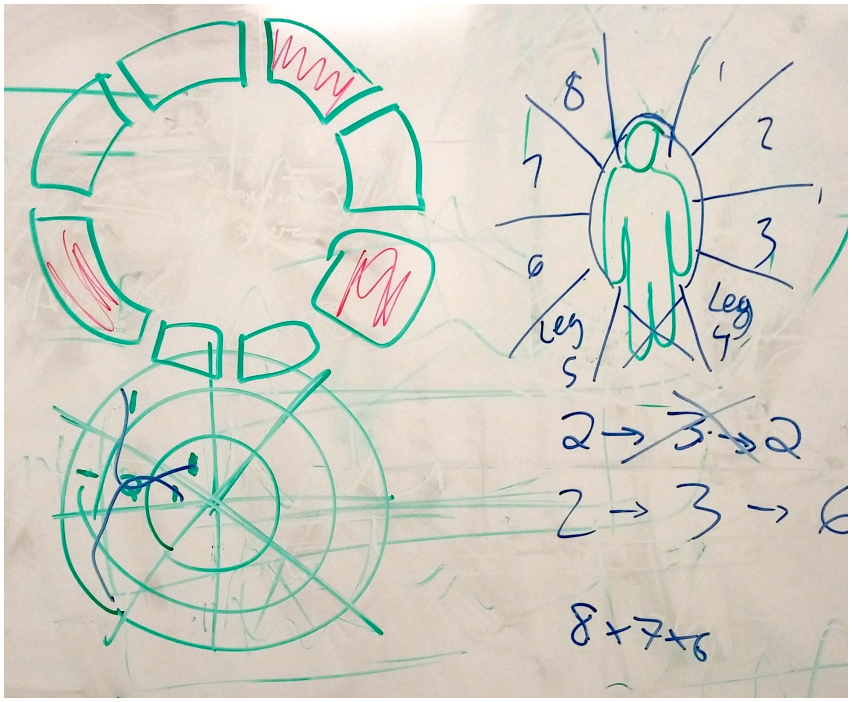

Figure 7.11: Original sketches of the Vitruvian Menu.

We determined that absolutely-positioned interface elements on stage were ineffective, as much performer effort was required to get to those positions without feedback. So, we figured performer body-relative interface elements would be more effective. This resulted in the first sketch resembling the form of the scene menu we used in our system: the *Vitruvian Menu* (Figure 7.11).

However, these interface elements could not be active and available at all times and also within reach, so we needed a way to turn them on and off. We made a prototype where the Vitruvian Menu only appeared when the performer was standing still for a short period of time, approximately 2 seconds. However, this led to many false positives, as pauses in motion are normal in acting. We played with increasing the period of time but it had to be made so high it made performers impatient.

During the Luelley Massey prototyping installing, we observed that we could light the stage very sharply with a harsh back edge, so that performers could "back out" of the stage if they wanted to cancel an action and disappear from sight. This inspired using a step-forward to trigger the Vitruvian Menu. We taped the ground of the lit area of the stage to define a line that performers had to step over in order to invoke the menu.

Referring to our discussion of ways to distinguish foreground from background activity in Chapter 5, the performer staying still was an ineffective delimiter, while stepping forwards or backwards was an effective explicit clutch for our menu. We will discuss our specific implementation of this strategy as *Interaction Depth Zones* in the subsequent section.

The Vitruvian Menu was also designed with some knowledge gained from our examination of background activity — the distance to the buttons in the Vitruvian Menu is tuned to be just at the edge of reach for most people. This design is similar to our proposed gesture-specific spatial zones (Figure 5.8).

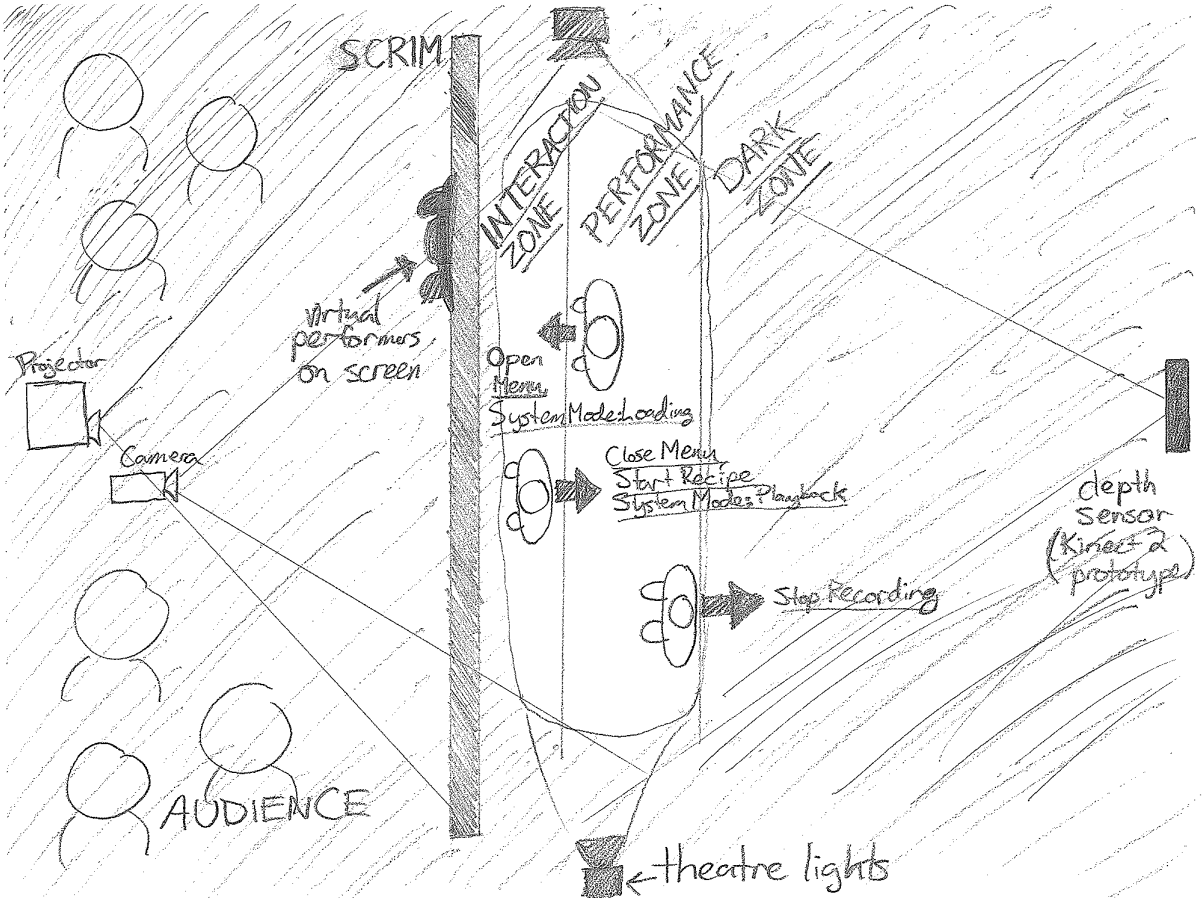

Figure 7.12: Our physical setup. In the middle is the projection scrim. To the left, the projector and camera, in the audience. To the right is the performance space, 1 m wide, lit by theatre lights. To the far right is the depth sensor for interaction. We use zones in depth for different functionality: the Interaction Zone closest to the scrim, and the Performance Zone farther away. Past the edge of the light is the Dark Zone. We have labelled the function of performers' transitions between depth zones.

7.5 Improv Remix Design and Implementation

Improv Remix is a tool for performers to control recording and playback of stage video over a performance session, with three global modes: *Loading*, *Playback*, and *Library*. In Loading mode, performers may quickly instantiate scenes from our novel *Vitruvian Menu* (Figure 7.13, described later). In Playback mode, scenes play onstage and may be directly manipulated. In Library mode, a performer may browse all recorded scenes, and load them into the Vitruvian Menu for quicker access. We have designed interactions carefully to be robust in an inherently noisy and ambiguous environment.

7.5.1 Physical Setup

We designed our setup so we could place projected ("virtual") performers adjacent to live performers (Figure 7.12). Our scrim, a special projection screen for theatre, is invisible unless lit, so playback performers and UI elements appear to hover in mid-air. For live performers, having playback performers

between themselves and the audience makes sightlines significantly better than in our prototype.

We discovered that the bright lighting from the side of the stage inhibits performers' ability to see UI elements in two ways: range and colour. Performers' practical range of sight appears to be restricted to 1.5 metres — we replaced global visual feedback for audio cues. The lighting also prevents performers from distinguishing colour — we adjusted our initial design to not use colour as an information channel.

7.5.2 Interaction Depth Zones

Our solution to distinguishing performer foreground activity from background activity is to use an Explicit Clutch — as defined in the Chapter on Background Activity. Our Clutch is based on different zones on the stage (Figure 7.4): the *Interaction Zone* — for loading scenes, *Performance Zone* — where performers will be primarily and the *Dark Zone*. Direct scene interaction is possible in the Performance Zone and the Dark Zone. Making these zones explicit signals to the audience what activity on-stage performers are engaged in (exposure), and also reduces performer nervousness about trigging the system incorrectly, which was present in many of our previous iterations. As the clutch is with the whole body, it is very robust to accidental activation. Additionally, the Dark Zone gives the ability for performers to completely disappear when desired, which can indicate a break in the scene.

We determine a performer's zone based on distance from the rear depth camera, with hysteresis applied to the zone boundaries to prevent debouncing[2]. We use audio feedback to inform users of zone transitions — escalating in pitch going forward, and de-escalating going backwards. When a performer transitions zones, it affects system mode and scene recording (Figure 7.4). Stepping forward from the Performance Zone to the Interaction Zone sets the mode to Loading, and the performer's Vitruvian Menu (described later) appears. The performer can choose to record a scene, or instantiate playback of previous scenes — we call this the *scene recipe*. However, this recipe does not execute until the performer steps back from to the Performance Zone, and only then after a 2.5 second tonal countdown to give the performer time to prepare. To stop a new scene's recording in progress, the performer steps back from the Performance Zone to the Dark Zone.

7.5.3 The Vitruvian Menu

Our novel *Vitruvian Menu* (Figure 7.13 a&b), is designed to be used by a standing human, with buttons arranged radially around them. The name is inspired by Leonardo Da Vinci's *Vitruvian Man*, as it is usable by arms and legs. We chose this user-relative design after finding that widgets with fixed absolute position took a long time for performers to acquire. We refer to the button directly above the user as the *Keystone*.

The Vitruvian Menu had to support quick, spontaneous access of scenes, yet also be robust to the noisy nature of whole-body interaction. We opted for a 300 ms dwell time, which we tuned to minimize accidental activation, without inhibiting speed of use excessively. When the user is touching the button, it is "filling", with an expanding circle for feedback. When a button is untouched, it slowly un-fills —

[2]Debouncing is a term from electrical engineering for strategies to prevent a switch rapidly alternating between states, by making a state "sticky", tending to preserve itself.

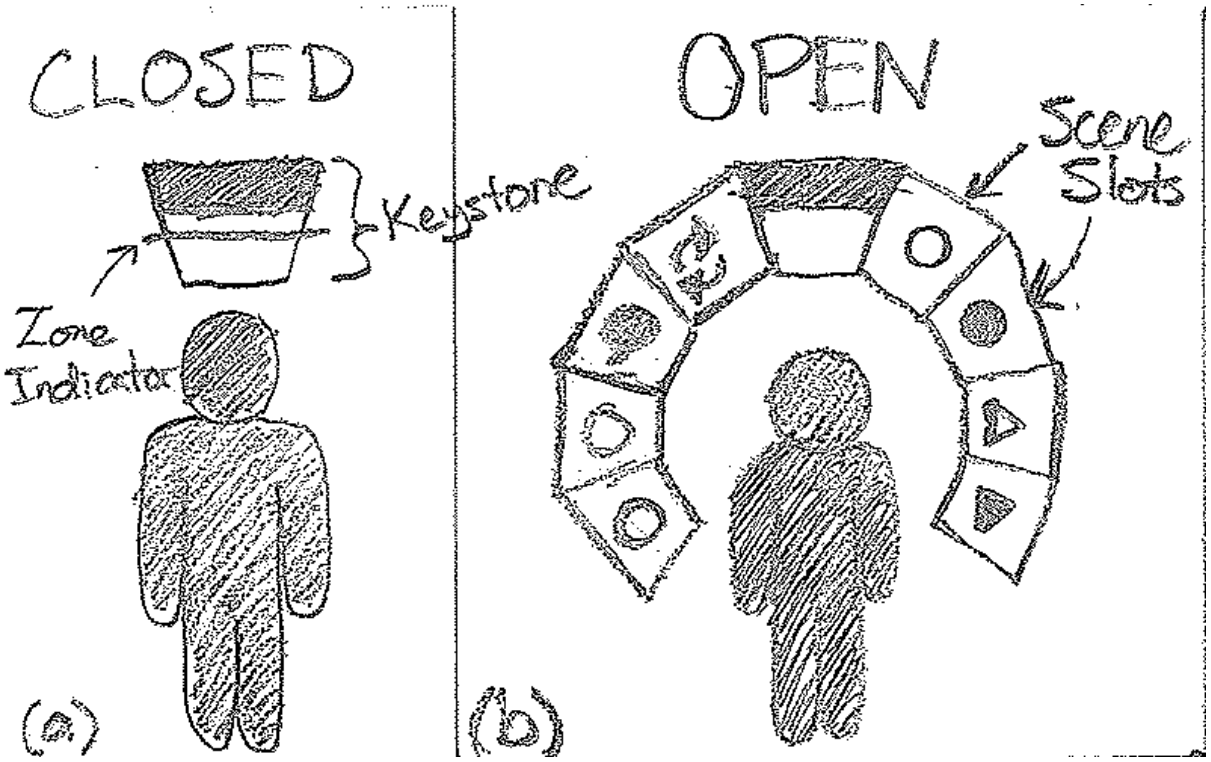

Figure 7.13: The Vitruvian Menu. (a) Closed, with just the Keystone visible. In the Keystone, we provide depth zone feedback. (b) Open, showing all scene slots, accessible by arms and legs. Icons in each slot indicate whether it has a scene, and its playback type in the Scene Recipe (described in text).

the filling mechanic ensures the button does not activate without a degree of certainty, but also makes it still possible to activate if the user has trouble maintaining overlap with the button.

When the performer is in the Performance Zone, their menu is closed, with just the Keystone visible; when the performer is in the Interaction Zone, the menu is open, with all buttons are visible and Improv Remix is in Loading mode. We use the Keystone to display feedback of the performer's current depth zone, while every other button is a slot for a scene. Slots start empty, with Empty Circle icons resembling record buttons, whereas slots with scenes have Empty Triangle icons, resembling play buttons. When the user activates an icon it becomes solid, indicating that it is part of the scene recipe. An activated play icon means the system will start playback of that scene, and an activated record icon means that the system will record a new scene, and save it in that slot. While activating an empty slot again toggles its state, a slot with a scene it in already will cycle through possible playback behaviours: *no playback*, *play once*, *loop* and *polite* (described below).

7.5.4 Direct Interaction with Scenes

Performers may *scrub* or *delete* scenes instantiated on stage (Figure 7.14). These both depend on the live performer being within interact-able distance of the playback performer in the scene — we tuned this so it was closer than natural acting speaking distance to prevent accidental interaction.

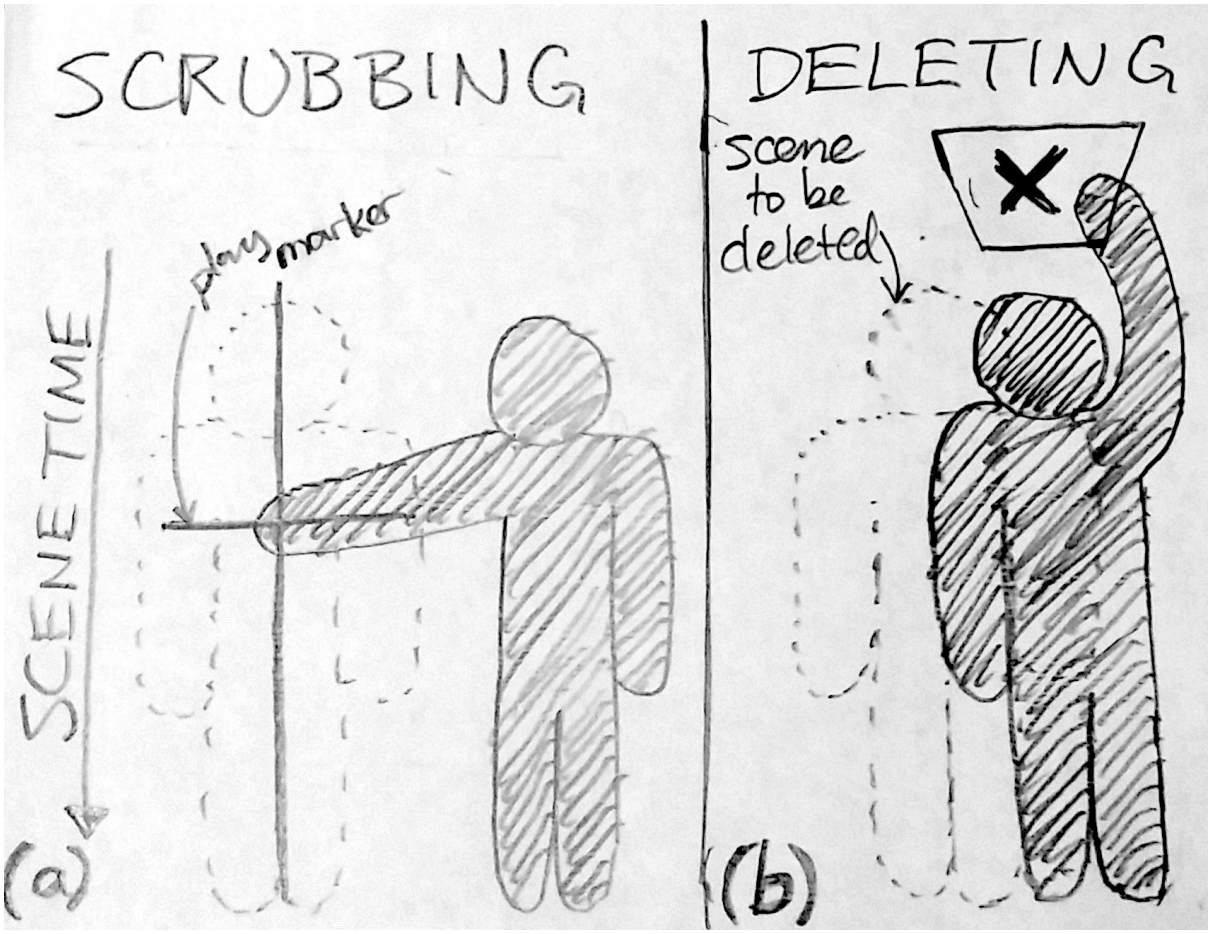

Figure 7.14: Interaction with Scenes: (a) Deleting a scene by standing over its performer, which shows a delete button in the Keystone. Invoking the Keystone deletes the scene. (b) Scrubbing a scene by reaching into its centre. A timeline appears, and the play marker of the scene is set to the centroid of the user's overlap with the timeline.

Scrubbing

In Improv Remix, scrubbing is the ability to freely move a video backwards or forwards in time, where a release resumes normal play. Our scrubbing interface is composed of a vertical timeline, as well as a horizontal line representing the current play marker. To scrub the scene, the live performer touches the timeline, which sets the scene's play marker to that time. To avoid accidental scrubs when a performer walks through a scene, we ignore silhouette overlaps with the scrubbing area above a threshold.

Scrubbing enables playful re-imagining of scenes, a powerful control which has not been possible before onstage. Scrubbing was found to be very intuitive by almost all who used it. In informal workshops, scrubbing was used innovatively by performers outside our primary target group. Dancers would finely control previous versions of themselves co-dancing with their arms, legs, elbows, knees, and even head. Vocal musicians set previous tracks to play at points that were easy to memorize, exploiting proprioception.

In practice, we found performers have very fine control over the scrub position. Scrubbing is possible both from the Performance Zone, but also the Dark Zone. When performing with an audience, these

take on drastically different characters; hidden interaction from the Dark Zone feels like puppeteering. Due to perspective projection, scrubbing from the Performance Zone is best done with an ankle or a wrist, while from the Dark Zone, scrubbing may be done with the finger.

Deletion

Scenes with *looping* or *polite* behaviour persist onstage until deleted manually. When a performer is within interact-able distance of a scene, a delete icon appears in their Keystone. Tapping the Keystone deletes any scene they overlap. The stage can be quickly cleared by holding one's arm in the keystone while walking across, much like the *stage wipe* gesture that appears in improv.

7.5.5 Politeness: Playback Performers with Manners

When a scene's playback mode is *polite*, its behaviour is to play random excepts from the scene so as to create the appearance of an actor delivering new, often surprising, material. We accomplish this by parsing the scene into suitable utterances and when the playback performer is onstage, controlling their behaviour carefully so they appear to have manners with respect to the other stage occupants.

Parsing a Scene into Utterances

After a scene is finished recording, we split it into suitable utterances using an algorithm that finds speaking and non-speaking sections in the audio track of the scene. We turned our algorithm to create separate utterances if there was a longer pause than what would be expected between one sentence and the next in normal conversation.

Polite Playback Mechanics

When a scene is in *polite mode* its default behaviour is to loop the longest non-utterance or "idle" section of the scene. This creates the appearance that the playback performer is listening on stage, without the jarring effect of freezing the video completely. Our system is careful when to trigger playback of utterances so as to not cause scenes to talk over each other or live performers on stage. We call the logic of this process the *Manners Module*. The manners module keeps a running check of the last time the ambient volume of the stage exceeded a threshold representing a stage occupant speaking. If the onstage volume is silent for a given duration (we used 1.8 seconds) the manners module is free to randomly choose an utterance from a scene in polite mode and to play it. If there are multiple onstage scenes in polite mode, the manners module maintains a circular queue, so all scenes take turns speaking. When the Manners Module chooses to play another utterance, it plays through and then returns to looping the idle section again. If a live performer continually speaks, the polite playback performer will never play an utterance; it is effectively being filibustered.

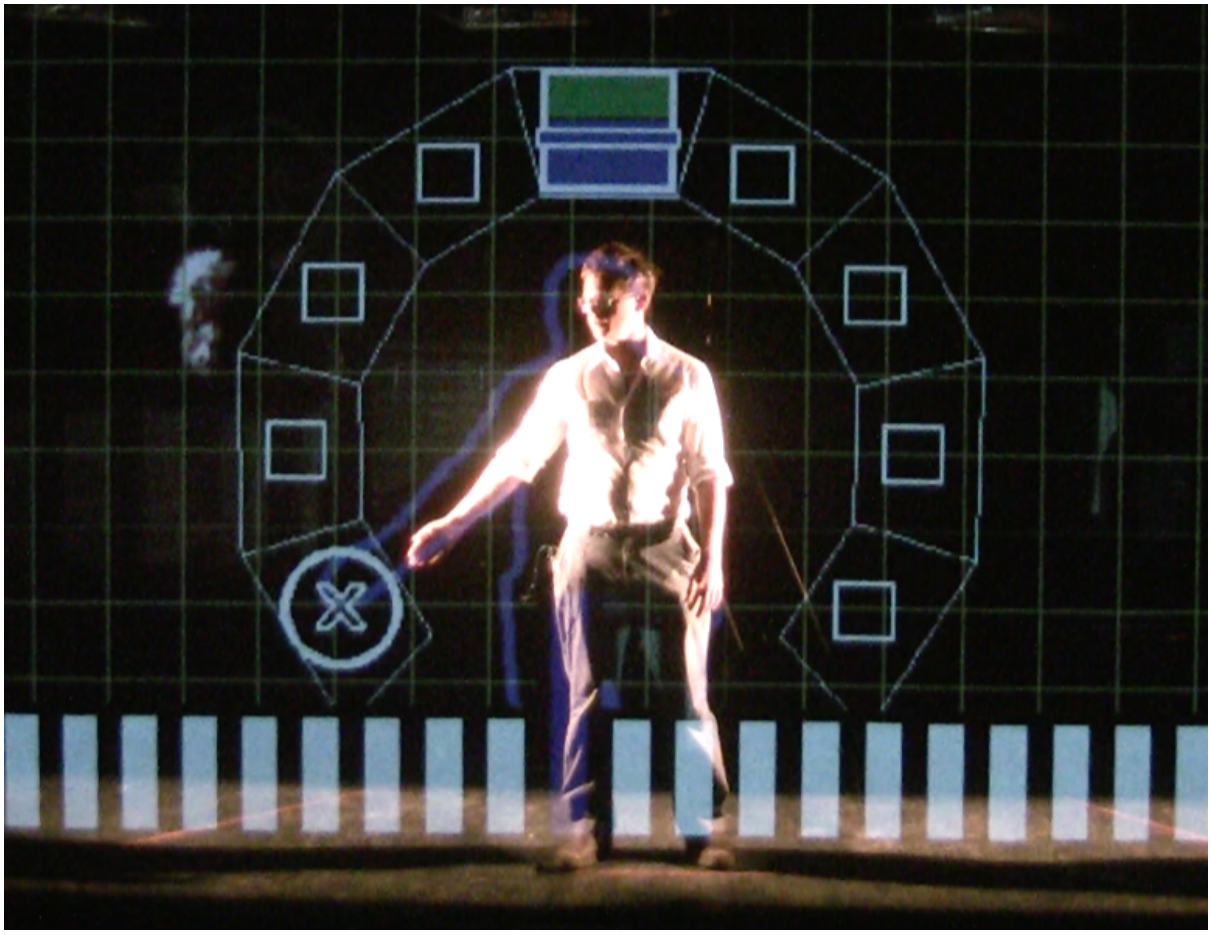

Figure 7.15: A performer using the Scene Library. To his left is the projected image of a puppet from his currently selected scene.

7.5.6 The Scene Library

The Vitruvian Menu is meant for quick access, and only has 8 slots for scenes. To fetch scenes for these slots from the large quantity of scenes that may be generated over a 2-hour session (up to 50), we implemented Library mode. To access the Library, the performer taps the Keystone when in the Interaction Zone (Figure 7.15). In Library Mode, a slot with a scene in it displays an X — if the performer activates this slot, they empty it. An empty slot has a square icon, indicating it can be filled with a scene, fetched from the Scene Library. In the Library interface, scenes are represented by white rectangles on stage. The user walks left and right on stage to select a scene by standing over it, and the first frame of the selected scene is displayed onstage. To fetch that scene into a specific empty slot, the user activates it.

8

Final Showcases & Use Cases

8.1 Evaluation: Showcases

This work makes arguments about the fluidity of video manipulation, and the complexities of interaction with a system in the context of a performance. It is ineffective to examine these in isolation, in a laboratory setting. We determined the best way to do so was *holistically*, in a real theatre space, with real, non-captive or paid audience members who were free to walk out if bored.

We scheduled three showcases during a week and installed Improv Remix in a theatre space: Friday and Saturday nights, with a Sunday matinee. For the showcase, we recruited 3 performers to create a quick demonstration of system features and rehearse some use cases. We advertised the showcases over social media, and had approximately 50 attendees in total. Attendees of the showcases were both experienced improv performers and regular audience members.

The structured section of the showcase took approximately an hour, followed by an unstructured hour where audience members could come up and use Improv Remix, including bringing back video of performers from much earlier in the night. This was treated as a freeform session, where the showcase was "already over", and audience members were free to leave if they wanted. Performers would leave the stage and fetch refreshments from the foyer, or casually sit in the audience. This level of informality was designed to ensure audience members were not too intimidated to use the system. If an audience member appeared confused, a performer would approach them and attempt to discern what they wanted to do, and then suggest how to use the system to accomplish it.

Going into the showcases, we sought to answer the following high-level questions:

Q1. What is the relationship between live and playback performers?

While the design of our setup places live and playback performers in the same, flattened 2D space, they are still very separate. There is really only one dimension performers can use to control their visual position: left or right. When a previous scene is instantiated and the playback performer is "on top" of the live performer, this is not immediately clear from their perspective, even though we use a silhouette to suggest it, and it takes a bit of presence of mind to step to the side.

When two live performers are co-present, they are likely to physically interact, from a handshake to a full-body hug. However, during workshops and showcases our performers consistently found that these sort of actions were not satisfying with playback performers. During design of Improv Remix, this meant it was safe to assume any interactions near a playback performer were intended for the system.

Q2. How do interaction and theatrical performance inter-mix?

Our design literally separated interaction and performance into separate zones, but scrubbing was still possible during "performance". However, we found that performers rarely scrubbed or delete scenes in character. It may be that these operations require too much cognitive load, or that the amount of time performers had to familiarize themselves with the system was not enough to be relaxed.

A use case that is possible, but did not arise, is scrubbing another playback performer while you are speaking to it, much like a marionette. It appears that the acquisition of scrubbing is not fluid enough to allow this at this stage.

Q3. Does Improv Remix facilitate novel creative work?

There was a great deal of creative use cases and we consider it a success. A listing of all the uses cases follows in the next section. One difficulty of our setup that inhibited spontaneity slightly is that it is hard to view a composition from behind the stage, due to the lighting. Thus, there is a strict audience versus performer side, and in practice any performer not performing would return to the audience side to watch. This is in contrast to typical group improv or theatre workshops, which are usually done in a circle of people, and any member may initiate action.

Q4. Do users find features of Improv Remix useful, and can they apply them effectively?

All features in Improv Remix received some degree of use. Note the contribution of this thesis was not to create polished interaction techniques — an in-depth usability of the novel interaction techniques created in this thesis (i.e. Interaction Depth Zones and the Vitruvian Menu) is out of scope of this work. During the showcases, we observed two major errors by performers and novice audience members who used Improv Remix:

The hover-based interaction for the Vitruvian Menu made it more robust, but due to the small range of vision of the user, it was possible to be hovering over a button without realizing it. This occurred most often with a button on the opposite side of the one a performer was reaching for.

When performers transitioned to the Dark Zone, the performer's silhouette would expand, due to the

perspective projection of the rear camera. If they were standing over a playback performer, their head would touch the Keystone unintentionally, deleting the scene. This could be fixed by scaling the silhouette or Vitruvian Menu based on the user's average depth.

8.2 Use Cases

Figure 8.1: Physical interaction: A live puppet climbing virtual columns. The puppeteer places only the puppet in the lit area so his body is mostly invisible.

8.2.1 Physical Interaction

In one example, we had a performer create a chain of videos high-fiving himself, with the audience clapping at each virtual impact. In a more complex example, a performer captured video of a pillar at three positions, and then instantiated them as looping scenes. He controlled a puppet climbing and jumping between the pillars (Figure 8.1). Precise physical interaction between a live performer and video is difficult, as to keep the interface minimalist from the audience perspective, we do not provide alignment feedback. Additionally, if a live person and video overlap, there is no ability to appear in front or behind the video. As the audience's viewpoints are all at slightly different positions, the alignment between live and playback performers will always vary due to parallax.

Figure 8.2: Physical interaction: Collage: A dance party of several playback performers.

8.2.2 Collages

Performers will often layer several looping videos onstage depicting a sound and/or physical action. In one example, performers recorded themselves individually dancing silently, intending to create a "dance party" when combined (Figure 8.2). One of the performers searched for a song [1] and played it on his smartphone next to his lapel mic. The resulting collage has him standing on the side of the stage, awkwardly moving to the music relative to the dancers. While the end result of a collage may be interesting, the audience can feel bored watching the build-up. It may be possible to make this payoff more exciting with performer practice, or it may be fine to bore the audience slightly before the, hopefully surprising, final payoff.

[1] with intentional irony, Robyn's *Dancing On My Own*

Figure 8.3: Music: A beat-boxer layering one other instance of himself beatboxing and dancing.

8.2.3 Music

We treat music as a special case of chants. Something can appear musical unintentionally, either through apparent harmonization, or rhythmic alignment. Over time, a repeated phrase with no tonality can appear to become musical as well.

We had performers who were musically skilled harmonize with themselves. In one example, the audience watches in somewhat-bored anticipation as the performer records three videos, each a single, but different, note. The payoff was surprising when they appear to be perfectly harmonized.

One of our performers was an accomplished beatboxer and singer. He recorded himself beatboxing while dancing to create a layer, and then would begin to rap on top of that (Figure 8.3). He constructed another example by singing the same song 3 times, and coordinated pointing to himself between the different video tracks. Later a different performer brought back the beat-boxing track so he could dance alongside it.

In many of these musical examples, the performer would construct the composition on the performer side of the scrim, then cue the scenes to play together, and then excitedly run around to the audience side of the scrim to observe, with everyone else, the quality of their result.

Musical samples from highly-skilled performers would often be brought back to accompany performers

with less skill, and they would lip sync or react to the music.

Figure 8.4: Constructed Scene: Part 1, a beckoning man beckons a duck.

8.2.4 Constructed Scenes

The ability to record a scene while playing a previous one allows single performers to *construct* complex scenes, but exploiting timing and alignment. We will describe one example where a performer recorded a series of scenes to that were interesting to build, yet surprising in combination.

Scene 1: *Acting like a duck, the performer waddles from stage right to stage left, occasionally looking behind itself. Finally, it turns around and waddles back slightly faster, to stand up and kiss an empty spot in the air.*

The audience watches with curiosity. What is the duck doing?

Scene 2: (Figure 8.4) [Recorded with Scene 1 playing] *The performer stands on far stage right, repeating, as endearingly as possible, "Come here duck!" and beckoning as the duck, from Scene 1, walks away. Finally, the performer loses his patience and loudly yells "Hey duck!", at which point the duck in the video turns around and starts coming back. The performer non-verbally encourages it, and picks it up and kisses it, saying "You're so cute!".*

The audience laughs as the glances of the duck in Scene 1 are now explained.

Scene 3: (Figure 8.5) [Recorded with Scene 2 playing] *A dishevelled man stands on stage left reading a newspaper. The beckoning man from Scene 2 repeats "Come here duck!", distracting the dishevelled man from*

Figure 8.5: Constructed Scene: Part 2, a dishevelled man responding to the beckoning of the virtual man.

reading the newspaper. Initially, the dishevelled man looks around, discerns that the beckoning must be not be speaking to him, and then looks back to his paper. As the beckoning man continues, the dishevelled man looks at him more angrily. Finally, the beckoning man from Scene 2 yells "Hey duck!" and the dishevelled man drops to the ground, ducking from possible danger. Seeing there is none, he charges the beckoning man, saying "Hey, buddy, what's the big idea!?". The beckoning man kisses him and the dishevelled man slaps him in response.

8.2.5 Responsive Scenes

Both scrubbing and polite playback allow the creation of novel content which live performers must respond to spontaneously.

Scene Puppetry with Scrubbing

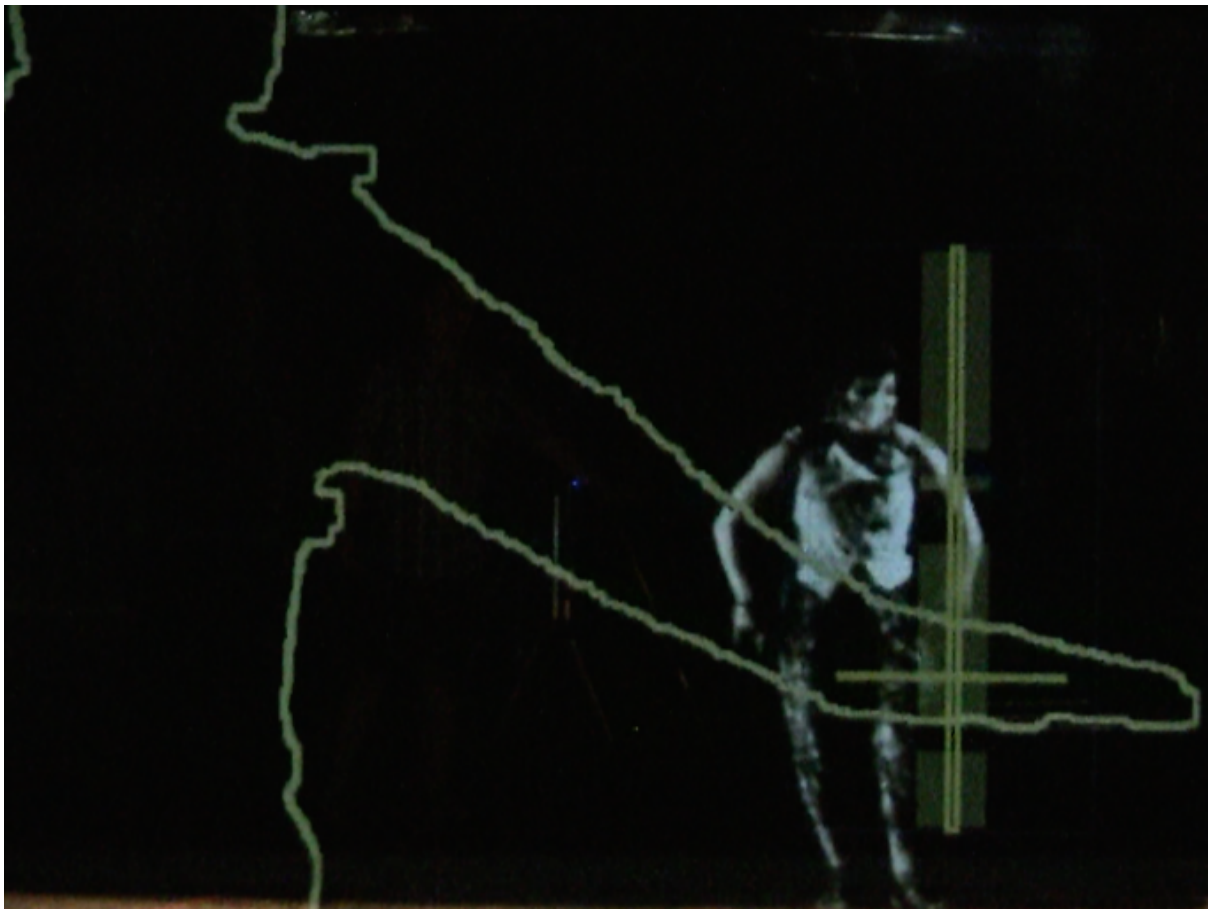

Figure 8.6: Responsive Scene via Scrubbing: a performer controls their own video from the Dark Zone.

Any verbal scenes that are under a couple of minutes lend themselves well to scrubbing. In one example, the Performer asked the audience for 3 simple phrases that could be used in conversation. He got "It's raining", "These pretzels are making me thirsty." and "That's what she said". Performer A recorded himself saying these phrases from stage right, facing (empty) stage left. After, he loaded his own scene and retreated into the Dark Zone. Performer B, who had previously been sent from the room, was called back and stood opposite the playback performer A, while the live performer was in the dark (Figure 8.6). While Performer B eventually learned all 3 phrases, Performer A had control over which one was going to come next, and could do so to comic effect.

Figure 8.7: Responsive Scene using polite playback: A performer records himself saying "true" or "false", then instantiates the playback performer in polite mode and monologues, with his virtual self informing him if he is lying.

Polite Playback

As previously described, a scene in polite playback chooses a random sub-utterance to speak whenever the "manners module" has determined that the stage has been silent for long enough. This allows a live performer to speak as long as they want without fear of interruption, as would occur with basic playback.

One audience member recorded himself saying "True" and "False", and then instantiated that scene in polite playback (Figure 8.7). He then began his experience of the evening thus far, and let his virtual self randomly choose whether the statement he just made was true or a lie. If a lie, he then had to explain the actual truth.

Figure 8.8: Failed Dissonance: The beatboxing scene from before playing alongside the dishevelled man. The dishevelled man's angry looks, originally intended for the beckoning man, appear to be directed at the beatboxer.

8.2.6 Failed Dissonance

When two seemingly random scenes were set next to each other, they would appear to create something unexpected. While a clumsy, meaningless dissonance would be expected of two randomly played videos, in fact the dissonance *fails*. This effect is due to the sense-making activities of the observers of these videos [Kulešov, 1974, Zimmerman, 2007].

We had two good examples of failed dissonance: a) the dishevelled man from our Constructed Scene reloaded alongside our beatboxer, where his angry looks now appear directed at the beatboxer (Figure 8.8) and b) An accidentally recorded scene of the beatboxer discussing an idea offstage, including the phrase "bastardized the equilibrium of harmonized core sounds". This was brought back to amusing effect against multiple scenes, appearing as an overwrought academic commentary on the silly events of the stage.

9
Discussion & Conclusion

We have presented "Improv Remix": a design exercise in extending a specific art form — longform improvised theatre — through creation of a fluid interface for manipulating video. Our primary contribution is our documentation of this process, from an analysis of the art form, to proposing an extension of it (performer control of video from earlier in an improv set), to designing a system, and observing novel use cases. This work has inspired the construction of novel interaction techniques, such as the Vitruvian Menu, depth zones relative to a large display for controlling the type and amount of interaction, and scrubbing through life-size video using any part of your body. While these interaction techniques were produced in response to a specific design case, they may, with deeper examination and tuning, be useful for general applications.

This work was inspired by existing practices; but the goal was not to replicate them in digital form, but to extrapolate from them. While the final discovered use cases are all surprisingly novel, experienced longform improvisors did not find them unusual or alien. We propose that extensions to many other specific art forms and genres are possible using the process in this paper. This work was particularly rewarding for performers who were able to participate in the process from the low-functionality workshop phase until the final form.

To close, we will discuss how our system — *Improv Remix* — measures against the considerations for designing interaction for performers as we described in Chapter 6: *Stakeholder Perspectives* and *Interaction Design*. Then, we will provide high-level *Reflections on Designing Interaction with Performance Artists* and finally close with *Final Thoughts and Future Work*.

9.1 Stakeholder Perspectives

We initially defined three stakeholders for our system: the system itself, performers, and the audience. We will analyze the interaction techniques we designed into Improv Remix from these perspectives here.

9.1.1 System (Detection)

Regrettably, the system cannot speak for itself. It seems best to write this section from the programmers' point of view, in how easy it was to detect system interaction (foreground activity) versus performance (background activity).

Going into the design of Improv Remix, we expected to have a large amount of trouble designing it so that Foreground Activity was sufficiently distinguishable from Background Activity. We used a few different approaches:

For the Vitruvian Menu, we used an *Explicit Clutch* of stepping forward into the interaction zone.

For Deleting a playback performer, the live performer reaches up to the Keystone when standing over top of the playback performer.

For Scrubbing a playback performer, a small overlap of a silhouette does the scrub, whereas a large overlap, representing a live performer walking past the playback performer, is ignored.

For 2D spatial positioning of playback performers, we also tried various techniques, but disambiguating from scrubbing and deleting became over-complicated and eventually this feature had to be abandoned.

To have to rely on an Explicit Clutch felt somewhat frustrating, and from the programmer/designer perspective there is the sense that maybe, somehow, if we kept prototyping we could find interaction techniques or a gesture detector that better separated foreground and background activity. However, the depth zones were highly successful and easily-understandable by performers, so they were kept.

By contrast, Deleting and Scrubbing were contextual. Whenever a part of the live performer's silhouette touched the playback performer, their silhouette would appear, as well as other annotations. Thus, live performers trained themselves to avoid touching playback performers unless there was some interaction intent. This approach was also highly successful.

9.1.2 Performer (Experience)

We made observations of performer usage over the course of iterative development, and can thus compare the performer experience with the final prototype to performer experience with early prototypes. We also conducted brief, informal interviews during the experience with the final prototype. From this, we can roughly assess performer experience.

Again, the close, yet separate, spaces of the Performance and Interaction Depth Zones affected Performer Experience significantly. Interacting while performing was rare, and thus the awkward interaction moments we observed in the workshops and in early prototypes did not happen mid-performance. This is good, and the interaction depth zones afford a very quick, and clear mode switch. The concern

we had before, where part of the performer's body would be in-character while the other parts of it were not, did not occur.

9.1.3 Audience (Perception)

We had informal interviews with audience members during and immediately after the showcases. This gave only vague results, and they were overwhelmed with the novelty effect, meaning it is not possible to extract results from perception of interaction from these. It would be possible to run a study where we played back segments of video of the showcase to audience members, but the utility of such analyses may not be worth the effort.

This problem of uncertainty in evaluating audience perception of the final details of interaction means that the interests we have, in ensuring that interaction does not disrupt the show, are somewhat nebulous. Concerns about not wanting to hide interactions from the audience were successfully satisfied (see comments on Exposure in the next section).

It may be possible to study audience perception of performers' actions or interactions on stage. These effects are subtle, and would seem to depend highly on the observer, e.g., when the actors enter the room in *No Exit*, how does whether or not they glance at a knife on the table affect the audience's reception? It seems like would be highly contingent on the audience members' previous experience with knife-based violence. Perhaps the effect is not so much predictable, but merely that, like the Kuleshov Effect discussed in the Background, there is a strong, highly variable effect. Thus, we should be careful in how interaction appears to the audience members.

9.2 Design Principles

In Chapter 6, we defined several principles of interaction design for performance contexts: exposure, neutrality, semantic capacity and graceful error recovery. We refer the reader to that earlier section for definitions. We will now analyze the qualities of our implementation of Improv Remix relative to these principles, as well as evaluate their utility.

9.2.1 Exposure

A result of the design of Improv Remix is that in-character interactions are rare. However, interaction and in-character actions are highly interleaved. As the interface responds in a menu-like fashion when a performer steps forward into the interaction zone, it is clear that the intention of any actions on the performers' part is to interact with the system. The interactions possible in the performance zone, scrubbing and deletion, both contain visual feedback, so it is clear to the audience and other performers when a performer is interacting with the system. Otherwise, the interface of the stage is blank, except for the Keystone that hovers over each onstage performer, merely indicating acknowledgement that it sees the performer. The Keystone's position does not appear to be something the performer is explicitly controlling to achieve some end goal, and thus it does not appear to be interaction.

Making the system usable to performers was the primary goal, but since visual feedback to performers is visible on the large scrim, and audio feedback is audible to both performers and the audience, then it seems we have achieved exposure by a consequence of our physical setup. Indeed, even when live performers are scrubbing playback performers from the dark zone, the audience can see their silhouette controlling the scrubbed play marker in the playback performer. A, perhaps, harder problem would be creating exposure when performers were interacting with smartphones[1].

A secondary concern is if performers' interaction is legible to the audience. While we did not do a formal survey of audiences' perception of specific interactions on stage, informal discussion with audience members immediately after the showcases indicated they were not baffled by any element of how the system worked. Perhaps one such property that made all our interactions legible is that system response was always immediate and consistent. Reloading a playback performer immediately showed its first frame on stage; scrubbing a playback performer immediately made them move. The link between cause and effect was always immediate, and the Improv Remix interface was otherwise relatively minimal. One exception to this is polite mode, where if the scene was not parsed neatly on audio, it could be hard to diagnose. This could be remedied, if desired, by the inclusion of a live play marker. However, it is unclear if always-on annotations of playback performers would help or be a detriment to the performance.

9.2.2 Neutrality and Semantic Capacity

We pair these two principles here as they overlap considerably, even though we feel it is important to treat them as separate aspects of the same issue. They could perhaps be summarized together as the need for *semantic flexibility* when the performer is interacting with the system.

Evaluating whether interaction is semantically flexible could be highly subjective. During the design process, semantic flexibility was tacitly understood as something to be aware of, but ensuring it did not require significant cognitive effort. The awareness was sufficient to ensure that our interaction techniques we defined were sufficiently neutral, yet had high semantic capacity.

During Improv Remix's use in the showcases, in-character performance and interaction rarely occurred simultaneously. Performers would interact with the Vitruvian Menu when in the Interaction Zone, and clearly out of character, or they would be scrubbing playback performers from the Dark Zone, and not visible to the audience and thus not "on stage". As we observed before, performers rarely scrubbed while in the performance zone and maintaining a character; this felt strange, as it was hard to justify why a character would reach into another character's personal space and control them. Scrubbing a playback performer while also performing yourself may also just be too much cognitive load.

Deleting playback performers was semantically loaded, in an interesting way. This is evidenced as performers would exclaim things like "Get out of here!" or "I am your end!", when dismissing performers from the stage. This was due to the function of the interaction, not the interaction technique itself (hovering over the Keystone).

[1] The way the BBC television show *Sherlock* (2010) achieves this is by not pointing a camera at the phone itself, but by providing an overlay title screen with the relevant content of the phone hovering mid-air, next to the performer.

9.2.3 Graceful Error Recovery

The importance of this principle became clear during development. The most common errors were instantiating the wrong video or set of videos, or instantiating them with the wrong playback behaviour. Both the systems' designers and performers reacted very negatively to unintended life-size video beginning its playback on stage as if barging into a private meeting, unwelcome. So, we needed a quick-cancel technique during the playback countdown. We exploited our interaction depth zones, and at some point, on a whiteboard somewhere, have "Step Backwards == Cancel" written. Stepping into the Dark Zone during a playback performer instantiation countdown cancels it, as a primary purpose. But also, and this is very important from the performer and audience perspective, it makes the performer disappear from the stage. This useful for the performer's ego, where they are given the opportunity to emotionally react to the mistake while hidden from the audience. This discontinuity in the visibility of the performer is also useful, from the audience's understanding of what the performer is doing; this step into the dark is a break in the scene — when the same performer steps forward again into the light, it is understood to be a new scene, and the mistake from before is easily forgotten.

9.3 Interaction Mapping: Time- and Value-Sensitivity

To aid us in the interaction design process, we determined a minimal set of features we desired for the system. We did an analysis where we labelled each feature as time-sensitive and/or value-sensitive. As we were in a relatively new, open-ended space of interaction design possibilities, defining these constraints explicitly was very useful to constrain the possibilities of interface design. Additionally, some features were dropped due to time constraints, infeasibility of implementation, or lack of demand, after we made our listing. However, the listing itself was useful and we recommend others who are designing interaction in an entirely new medium do the same.

9.4 Reflections on Designing Interaction with Performance Artists

Performers, whether in theatre, improv, or music, are an unusual user group to design for, and worth discussing. Improvisational theatre performers were involved in many stages along this process in addition to the discussed workshops and showcases, and one of the researchers has significant improv experience. There does not appear to be much attention given to technology used by performers while on stage. We hope this work has motivated this as an interesting area, and we will give our understanding of what we have learned when designing technology for users who are performers, and wish to use technology in a performance setting.

First, we will cover the process of *Learning Interaction* for performers, then we will emphasize the *Importance of Direct Control* in interaction, followed by how we managed *Feature Elicitation* in a performance context.

9.4.1 Learning Interaction

In the early phases of this project, we hoped that interaction could tightly intermix with performance. The guiding example we used was a musician quickly tuning their guitar in the middle of a song. We have made the specific choice in this project to not encumber the performers with hardware, and instead use coarse whole-body gestures, which are naturally noisy. This project went through a phase where we tried to search for gestures that would reliably not appear in theatrical performance, and could be reserved for system interaction. However, performance was found to be far too unpredictable, and if we banned certain, specific movements for performers, then they became self-conscious and ineffective.

To contrast, performers have a very high degree of control over their bodies when given specific instructions. When they are told they must pose or move in a certain way, they can consistently reproduce it. This ability comes from practice of moving and posing on stages and in front of cameras with a high degree of precision. Performers have experience being cognizant of lighting on themselves and their visibility to the audience. Many performers also have formal movement or dance training.

The lesson is that "Do it exactly this way" works very well, while "Don't do this" works poorly.

9.4.2 Feature Elicitation

Improv is known for its "don't say no, say yes" rule. In truth, this is more nuanced — while saying "no" or rejecting something is certainly bad, much of improv teaching focuses on what to do with new information during scene. Longform improv teaching focuses on the game of the scene [Halpern et al., 1994], in which a pattern of behaviour arises that the performers may explore and heighten (intensify). Through improvisors' training, they become very good at finding what is interesting about the present scene, and exploring it in a playful, generative way. From a system developer perspective, this can run counter to finding a finite set of features that must be implemented.

Achieving a consistent and predictable map between user goal and intention is not a good measure if our desire is, as developers, to produce new tools for performers, which we can think of like new brushes for a painter. Perhaps a better result would be if the tool has properties that are initially surprising, but then become something that the user has some degree of control over. A perfect map between goal and intention is asymptotic eventually, but we should not dismiss a system if the user is initially clumsy.

Performers frequently asked the system developer questions of the form "Is it possible to...[X]" and the answer instinctively given is whether it is possible with the current equipment, and current state-of-the-art algorithms, not whether it is possible to implement and debug within the amount of time that the asker desires. However, the system developer was cautious about telling them "no", as there was always a worry that they will stop asking for interesting ideas. In practice, a better answer is "Yes, but not during this session".

In typical software development, it is important to find and remove bugs. Performers' reactions to bugs was often much more positive than a typical user. When the system incorrectly detected a gesture and started playing one scene or many, users would often respond in surprise and joy, as if the system was talking to them, though if this happened to the same performer multiple times they would eventually

become frustrated. Performers are trained to make every situation interesting, and often would not care whether a bug was fixed, as they could work it into a show. As this means they are highly flexible in what a typical user would consider a stressful situation (unexpected behaviour), it became difficult to work with this playful mindset during the early stages, when the goal of the developer was to end the workshop with a precisely-prioritized list of action-items.

9.4.3 Articulation of Creative Ideas & Direct Control

The most important finding in Improv Remix is that *Articulation of creative ideas flourishes when a user has direct control*. The status quo in theatre, which we observed at the beginning of this thesis, was that performers are rarely engaged in direct usage of technology as part of the theatrical performance onstage — it must be coordinated with offstage collaborators.

A consistent observation since the early stages of this work is that novel theatrical ideas are difficult, even tedious to explain in the middle of creative work. The goal of Improv Remix was to empower performers to have control of technical elements from the stage. This allows performers to execute ideas without having to articulate them, while still half-formed, to people offstage. In the workshops, performers would occasionally abort ideas that were too difficult to articulate.

We are not stating that performers are less committed or persistent in general than the average user; just that performers, especially improvisors, are used to generating a large amount of novel content. If they are in a creative, theatric state, as opposed to, say, a seated discussion, then an impediment preventing them from executing ideas is in danger of removing them from that state. In this case, performers are quick to discard the impeding idea and are eager to move on to someone else's or another one of their own.

We saw that the coordinating gestures used in modern improvisation are quick, unambiguous, and clear to all performers on stage and the audience. The goal for our system was for its interaction to have the same degree of immediacy and clarity. We feel we achieved this, and performers were able to execute complex ideas without having to describe them — indeed, creative ideas flourished. The visibility of interaction in our system also meant that there was no mystery as to how they achieved the result they intended, and others could replicate and build on their work.

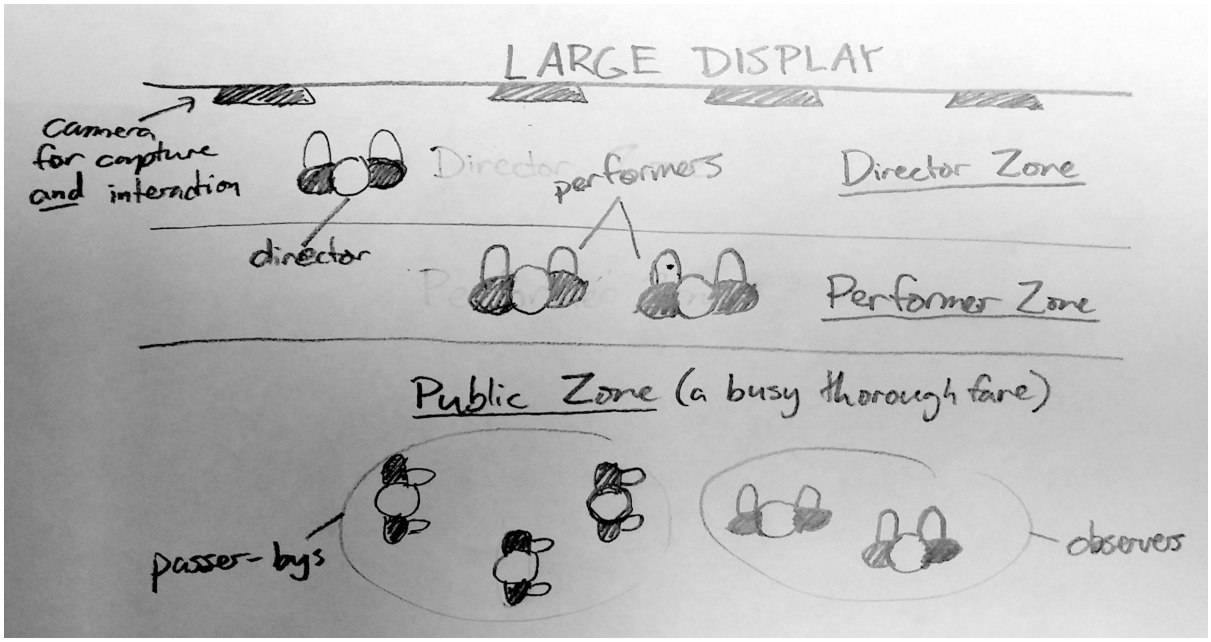

Figure 9.1: A top view of the proposed system. The large display is in a public space. Different levels of proximity to the display indicate different roles with respect to it. Director, for queueing and managing scenes; Performer, for performance, and Public, for observing the interface or passing by.

9.5 Final Thoughts and Future Work

Technology has been incorporated into theatre on many occasions before. In my survey of it, it has usually been for specific purposes, at specific moments, in specific shows. By analogy, imagine if researchers designed a sketching interface, custom-tailored to draw a specific drawing. In my work on this thesis, I had to be more generic, as in theatrical improvisation, there is no script, just structure and mechanics, which inspired much of the design of Improv Remix. It would be interesting to hear about other forms of performance, whether traditionally scripted or not, and imagine generically-useful, technology-powered extensions to them.

One area where this work fell short of the desire was ease of audience involvement. Inspired by a reading of Augusto Boal's work, I desired to enable audience to take video of performers, with or without their consent, and re-imagine them. The design of the physical setup still assumed separate audience/performer roles, a literal large wall (the scrim), separating them. Audience members were seated, requiring them to get up, and walk around the scrim, after which their initiative or inspiration might be gone.

Were I to install this setup again, somewhere else, I would like to explore a different topology — where the performers and audience are on the same side of the display, and the transition between different roles is as simple as stepping forward or backwards. I would call the "Interaction Zone" the "Director Zone" instead, making that role explicit (Figure 9.1).

A few times in our process, as early as the workshops, performers expressed a preference for viewing combined videos, over a virtual and live performer combined. This remains to be evaluated thoroughly, and could highly depend on performer and audience preference. In the proposed setup, live video of

performers would be captured and projected immediately to be combined with playback performers, like a live video mirror. Audience members or passer-bys in the public zone would be shown as silhouettes, with some light method of suggestion to join the performance by stepping forward (Figure 9.2), in alignment with Vogel's work [Vogel and Balakrishnan, 2004].

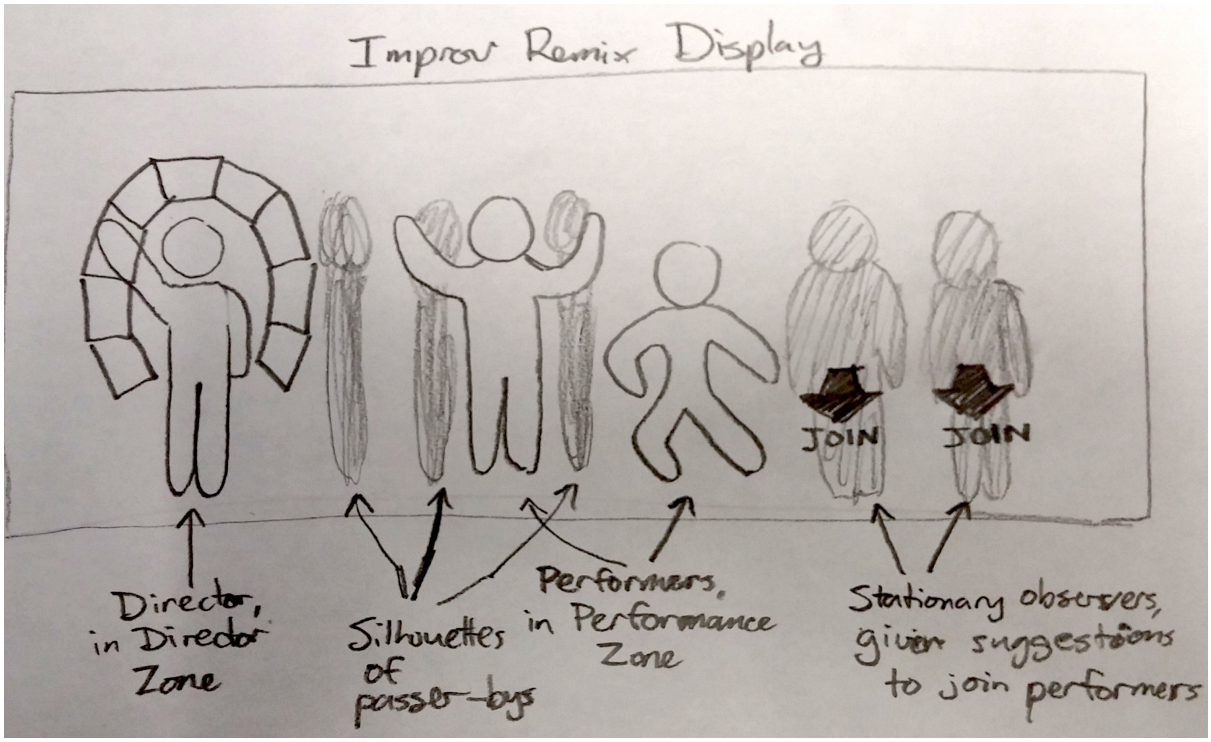

Figure 9.2: A view of the display in the proposed system. Directors and live performers are shown feedback of themselves in full-colour, as a live video mirror. Playback performers on the display thus appear indistinguishable from live performers. People in the public zone are shown as silhouettes and, if they appear to be stationary and observing the display, will be encouraged to step forward and become performers.

We have presented a tool designed for theatrical improvisors to extend the genre of modern theatrical improvisation. I am very satisfied with the final product, and the wealth of use cases supports that the tool achieved its goals, of "creating art not before possible", it is indeed a novel new brush, extended from properties of the genre from which it came.

References

JK Aggarwal and Michael S Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

Abir Al-Hajri, Gregor Miller, Matthew Fong, and Sidney S Fels. Visualization of personal history for video navigation. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 1187–1196. ACM, 2014.

Jason Alexander, Andy Cockburn, Stephen Fitchett, Carl Gutwin, and Saul Greenberg. Revisiting read wear: Analysis, design, and evaluation of a footprints scrollbar. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1665–1674, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518957. URL `http://doi.acm.org/10.1145/1518701.1518957`.

Fraser Anderson, Tovi Grossman, Justin Matejka, and George Fitzmaurice. Youmove: Enhancing movement training with an augmented reality mirror. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 311–320, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2268-3. doi: 10.1145/2501988.2502045. URL `http://doi.acm.org/10.1145/2501988.2502045`.

Bon Adriel Aseniero and Ehud Sharlin. The looking glass: visually projecting yourself to the past. In *Entertainment Computing–ICEC 2011*, pages 282–287. Springer, 2011.

Jackie Assa, Yaron Caspi, and Daniel Cohen-Or. Action synopsis: pose selection and illustration. *ACM Trans. Graph.*, 24(3):667–676, July 2005. ISSN 0730-0301. doi: 10.1145/1073204.1073246. URL `http://doi.acm.org/10.1145/1073204.1073246`.

Philip Auslander. Live performance in a mediatized culture, 1999.

Autodesk. The new art of virtual moviemaking. *Whitepaper*, 2009.

Frances Babbage. *Augusto Boal*. Theatre Arts Books, 2004.

Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro. Video snapcut: robust video object cutout using localized classifiers. In *ACM SIGGRAPH 2009 papers*, SIGGRAPH '09, pages 70:1–70:11, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-726-4. doi: 10.1145/1576246.1531376. URL `http://doi.acm.org/10.1145/1576246.1531376`.

Luca Ballan, Gabriel J. Brostow, Jens Puwein, and Marc Pollefeys. Unstructured video-based rendering: interactive exploration of casually captured videos. *ACM Trans. Graph.*, 29:87:1–87:11, July 2010. ISSN 0730-0301. doi: http://doi.acm.org/10.1145/1778765.1778824. URL `http://doi.acm.org/10.1145/1778765.1778824`.

Christoph Bartneck, Mathias Funk, and Martijn ten Bhömer. Dancing with myself: the interactive visual canon platform. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 3501–3502, New York, NY, USA, 2009a. ACM. ISBN 978-1-60558-247-4. doi: 10.1145/1520340.1520512. URL http://doi.acm.org/10.1145/1520340.1520512.

Christoph Bartneck, Mathias Funk, and Martijn ten Bhömer. Dancing with myself. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems - CHI EA '09*, page 3501, New York, New York, USA, April 2009b. ACM Press. ISBN 9781605582474. doi: 10.1145/1520340.1520512. URL http://dl.acm.org/citation.cfm?id=1520340.1520512.

Thomas Baudel and Michel Beaudouin-Lafon. Charade: Remote control of objects using free-hand gestures. *Commun. ACM*, 36(7):28–35, July 1993. ISSN 0001-0782. doi: 10.1145/159544.159562. URL http://doi.acm.org/10.1145/159544.159562.

Allan Baumer and Brian Magerko. An analysis of narrative moves in improvisational theatre. In *Interactive Storytelling*, pages 165–175. Springer, 2010.

Sarah Bay-Cheng, Chiel Kattenbelt, and Andy Lavender. *Mapping intermediality in performance*, volume 4. Amsterdam University Press, 2010.

Phaedra Bell. Dialogic media productions and inter-media exchange. *Journal of Dramatic Theory and Criticism*, (2):41–56, 2000.

Fredrik Bergstrand and Jonas Landgren. Visual reporting in time-critical work: exploring video use in emergency response. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 415–424, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0541-9. doi: 10.1145/2037373.2037436. URL http://doi.acm.org/10.1145/2037373.2037436.

Anastasia Bezerianos, Pierre Dragicevic, and Ravin Balakrishnan. Mnemonic rendering: An image-based approach for exposing hidden changes in dynamic displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 159–168, New York, NY, USA, 2006. ACM. ISBN 1-59593-313-1. doi: 10.1145/1166253.1166279. URL http://doi.acm.org/10.1145/1166253.1166279.

Valentine Tschebotarioff Bill. *Chekhov–the Silent Voice of Freedom*. Philosophical Library, 1987.

Augusto Boal. *The rainbow of desire: The Boal method of theatre and therapy*. Routledge, 1995.

Stefano Bocconi. Semantic-aware automatic video editing. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 971–972, New York, NY, USA, 2004. ACM. ISBN 1-58113-893-8. doi: 10.1145/1027527.1027753. URL http://doi.acm.org/10.1145/1027527.1027753.

Richard A Bolt. *"Put-that-there": Voice and gesture at the graphics interface*, volume 14. ACM, 1980.

R. Borgo, M. Chen, B. Daubney, E. Grundy, G. Heidemann, B. Höferlin, M. Höferlin, H. Leitte, D. Weiskopf, and X. Xie. State of the art report on video-based graphics and video visualization.

Comp. Graph. Forum, 31(8):2450–2477, December 2012. ISSN 0167-7055. doi: 10.1111/j.1467-8659.2012.03158.x. URL http://dx.doi.org/10.1111/j.1467-8659.2012.03158.x.

Oscar Gross Brockett and Franklin J. Hildy. *History of the Theatre (8th Edition)*. Allyn & Bacon, 1998. ISBN 0205281710. URL http://www.amazon.com/History-Theatre-Edition-Oscar-Brockett/dp/0205281710.

Vannevar Bush. As we may think. 1945.

Yaron Caspi, Anat Axelrod, Yasuyuki Matsushita, and Alon Gamliel. Dynamic stills and clip trailers. *The Visual Computer*, 22(9-11):642–652, 2006.

Kai-Yin Cheng, Sheng-Jie Luo, Bing-Yu Chen, and Hao-Hua Chu. Smartplayer: user-centric video fast-forwarding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 789–798, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518823. URL http://doi.acm.org/10.1145/1518701.1518823.

Pei-Yu Chi, Joyce Liu, Jason Linder, Mira Dontcheva, Wilmot Li, and Bjoern Hartmann. Democut: Generating concise instructional videos for physical demonstrations. In *Proc. of ACM UIST*, pages 141–150, 2013. ISBN 978-1-4503-2268-3. doi: 10.1145/2501988.2502052. URL http://doi.acm.org/10.1145/2501988.2502052.

Cinemagram. Cinemagram. URL http://cinemagr.am/.

Herbert H Clark. *Using language*. Cambridge University Press, 1996.

CMU Graphics Lab Motion Capture Database. CMU graphics lab motion capture database. http://mocap.cs.cmu.edu/.

CMU Kitchen Motion Capture Database. CMU kitchen motion capture database. http://kitchen.cs.cmu.edu/.

Gabe Cohn, Daniel Morris, Shwetak Patel, and Desney Tan. Humantenna: using the body as an antenna for real-time whole-body interaction. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pages 1901–1910. ACM, 2012.

Mark Coniglio. Troikatronix. URL http://troikatronix.com/.

Carlos D. Correa and Kwan-Liu Ma. Dynamic video narratives. In *ACM SIGGRAPH 2010 papers*, SIGGRAPH '10, pages 88:1–88:9, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0210-4. doi: 10.1145/1833349.1778825. URL http://doi.acm.org/10.1145/1833349.1778825.

Cyriak. URL http://cyriak.co.uk/blog/.

Nicholas Diakopoulos and Irfan Essa. Videotater: an approach for pen-based digital video segmentation and tagging. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, UIST '06, pages 221–224, New York, NY, USA, 2006. ACM. ISBN 1-59593-313-1. doi: 10.1145/1166253.1166287. URL http://doi.acm.org/10.1145/1166253.1166287.

Julian Dibbell. A rape in cyberspace: How an evil clown, a haitian trickster spirit, two wizards, and a cast of dozens turned a database into a society. *The Village Voice*, December 1993.

Söke Dinkla. The history of the interface in interactive art. In *Proceedings of the 1994 International Symposium on Electronic Art (ISEA)*, 1994.

Alan Dix, Jennifer G. Sheridan, Stuart Reeves, Steve Benford, and Claire OḾalley. Formalising performative interaction. In *Proceedings of the 12th international conference on Interactive Systems: design, specification, and verification*, DSVIS'05, pages 15–25, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-34145-5, 978-3-540-34145-1. doi: 10.1007/11752707_2. URL http://dx.doi.org/10.1007/11752707_2.

Steve Dixon. *Digital Performance: A History of New Media in Theatre, Dance, Performance Art, and Installation*. The MIT Press, 2007.

Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowitcz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. Video browsing by direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 237–246, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1. doi: 10.1145/1357054.1357096. URL http://doi.acm.org/10.1145/1357054.1357096.

A. Engström, M. Esbjörnsson, and O. Juhlin. Mobile collaborative live video mixing. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, MobileHCI '08, pages 157–166, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-952-4. doi: 10.1145/1409240.1409258. URL http://doi.acm.org/10.1145/1409240.1409258.

Arvid Engström, Mark Perry, and Oskar Juhlin. Amateur vision and recreational orientation:: creating live video together. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, CSCW '12, pages 651–660, New York, NY, USA, 2012a. ACM. ISBN 978-1-4503-1086-4. doi: 10.1145/2145204.2145304. URL http://doi.acm.org/10.1145/2145204.2145304.

Arvid Engström, Goranka Zoric, Oskar Juhlin, and Ramin Toussi. The mobile vision mixer: a mobile network based live video broadcasting system in your mobile phone. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, pages 18:1–18:4, New York, NY, USA, 2012b. ACM. ISBN 978-1-4503-1815-0. doi: 10.1145/2406367.2406390. URL http://doi.acm.org/10.1145/2406367.2406390.

Daniel Fischlin, Ajay Heble, and George Lipsitz. *The Fierce Urgency of Now: improvisation, rights, and the ethics of cocreation*. Duke University Press Books, 2013.

Matthew Fong, Abir Al Hajri, Gregor Miller, and Sidney Fels. Casual authoring using a video navigation history. In *Proceedings of the 2014 Graphics Interface Conference*, pages 109–114. Canadian Information Processing Society, 2014.

William Forsythe and Deutsches Tanzarchiv. *Improvisation technologies: a tool for the analytical dance eye*. Zentrum für Kunst und Medientechnologie, 1999.

Matthew N Fotis. *Improvisational Theatre: In the Vanguard of the Postmodern*. PhD thesis, Illinois State University, 2005.

Adam Fourney. Design and evaluation of a presentation maestro: Controlling electronic presentations through gesture. Master's thesis, University of Waterloo, 2009.

Gesa Friederichs-Büttner, Benjamin Walther-Franks, and Rainer Malaka. An unfinished drama: designing participation for the theatrical dance performance parcival xx-xi. In *Proceedings of the Designing Interactive Systems Conference*, DIS '12, pages 770–778, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1210-3. doi: 10.1145/2317956.2318072. URL http://doi.acm.org/10.1145/2317956.2318072.

Anthony Frost and Ralph Yarrow. *Improvisation in drama*. Palgrave Macmillan, 2007.

Daniel Fuller and Brian Magerko. Shared mental models in improvisational theatre. In *Proceedings of the 8th ACM conference on Creativity and cognition*, C&C '11, pages 269–278, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0820-5. doi: 10.1145/2069618.2069663. URL http://doi.acm.org/10.1145/2069618.2069663.

Global Delight Technologies. Game your video. URL https://itunes.apple.com/us/app/game-your-video/id496232649?ls=1.

Dan B Goldman, Brian Curless, David Salesin, and Steven M. Seitz. Schematic storyboarding for video visualization and editing. *ACM Trans. Graph.*, 25(3):862–871, July 2006. ISSN 0730-0301. doi: 10.1145/1141911.1141967. URL http://doi.acm.org/10.1145/1141911.1141967.

Dan B. Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M. Seitz. Video object annotation, navigation, and composition. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pages 3–12, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-975-3. doi: 10.1145/1449715.1449719. URL http://doi.acm.org/10.1145/1449715.1449719.

Daniel R Goldman, David H Adviser-Salesin, and Brian Adviser-Curless. *A framework for video annotation, visualization, and interaction*. University of Washington, 2007.

Live Digital Motion Graphics. Resolume VJ software. URL http://resolume.com/.

Tovi Grossman, Ken Hinckley, Patrick Baudisch, Maneesh Agrawala, and Ravin Balakrishnan. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 861–870. ACM, 2006.

Jerzy Grotowski, TK Wiewiorowski, and Kelly Morris. Towards the poor theatre. *The Tulane Drama Review*, 11(3):60–65, 1967.

Group Shot. Group shot. URL https://itunes.apple.com/ca/app/groupshot/id488709126?mt=8.

Isabelle Guyon, Vassilis Athitsos, Pat Jangyodsuk, Ben Hamner, and Hugo Jair Escalante. Chalearn gesture challenge: Design and first results. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–6. IEEE, 2012. URL http://gesture.chalearn.org/data.

Charna Halpern. *Art by committee: A guide to advanced improvisation*. Meriwether Pub., 2006.

Charna Halpern, Close, and Kim Howard Johnson. *Truth in Comedy: The Manual for Improvisation*. Meriwether Publishing, 1994. ISBN 1566080037. URL http://www.amazon.com/Truth-Comedy-The-Manual-Improvisation/dp/1566080037.

Lone Koefoed Hansen, Julie Rico, Giulio Jacucci, Stephen Brewster, and Daniel Ashbrook. Performative interaction in public space. In *Proc. of ACM CHI EA*, pages 49–52, 2011. ISBN 978-1-4503-0268-5. doi: 10.1145/1979742.1979595. URL http://doi.acm.org/10.1145/1979742.1979595.

Barbara Hayes-Roth and Robert Van Gent. Story-making with improvisational puppets. In *Proceedings of the first international conference on Autonomous agents*, pages 1–7. ACM, 1997.

Otmar Hilliges, David Kim, Shahram Izadi, Malte Weiss, and Andrew Wilson. Holodesk: direct 3d interactions with a situated see-through display. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pages 2421–2430. ACM, 2012.

Guy Hilton. Interference: A performance experiment in internet choreography, call for participation. *Dance & Technology Zone*, 1998. URL http://art.net/~dtz/archive/DanceTech98/0597.html.

Douglas R Hofstadter. Godel, escher. *Bach: An eternal golden braid*, 1979.

Jonathan Hook and Patrick Olivier. Waves: multi-touch vj interface. In *Proc. of ACM ITS*, pages 305–305, 2010. ISBN 978-1-4503-0399-6. doi: 10.1145/1936652.1936733. URL http://doi.acm.org/10.1145/1936652.1936733.

Jonathan Hook, David Green, and Patrick Olivier. A short film about vjs: using documentary film to engage performers in design. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 3491–3492, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-247-4. doi: 10.1145/1520340.1520507. URL http://doi.acm.org/10.1145/1520340.1520507.

Jonathan Hook, David Green, John McCarthy, Stuart Taylor, Peter Wright, and Patrick Olivier. A vj centered exploration of expressive interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1265–1274, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979130. URL http://doi.acm.org/10.1145/1978942.1979130.

Jonathan Hook, Guy Schofield, Robyn Taylor, Tom Bartindale, John McCarthy, and Peter Wright. Exploring hci's relationship with liveness. In *Proc. of ACM CHI*, pages 2771–2774, 2012.

Xian-Sheng Hua, Zengzhi Wang, and Shipeng Li. Lazycut: content-aware template-based video authoring. In *Proc. of ACM MULTIMEDIA*, pages 792–793, 2005. ISBN 1-59593-044-2. doi: 10.1145/1101149.1101318. URL http://doi.acm.org/10.1145/1101149.1101318.

Scott E. Hudson, Chris Harrison, Beverly L. Harrison, and Anthony LaMarca. Whack gestures: Inexact and inattentive interaction with mobile devices. In *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '10, pages 109–112, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-841-4. doi: 10.1145/1709886.1709906. URL http://doi.acm.org/10.1145/1709886.1709906.

Wolfgang Hürst. Interactive audio-visual video browsing. In *Proceedings of the 14th Annual ACM International Conference on Multimedia*, MULTIMEDIA '06, pages 675–678, New York, NY, USA, 2006. ACM. ISBN 1-59593-447-2. doi: 10.1145/1180639.1180781. URL http://doi.acm.org/10.1145/1180639.1180781.

Wolfgang Hürst and Dimitri Darzentas. Quantity versus quality: The role of layout and interaction complexity in thumbnail-based video retrieval interfaces. In *Proceedings of the 2Nd ACM International Conference on Multimedia Retrieval*, ICMR '12, pages 45:1–45:8, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1329-2. doi: 10.1145/2324796.2324849. URL http://doi.acm.org/10.1145/2324796.2324849.

Ann Hutchinson. Labanotation. volume 68, pages 89–89. American Folklore Society; University of Illinois Press, 1955.

Eric Idle. *The Road to Mars: A Post-modem Novel*. Boxtree, 1999.

Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3d object dataset: Putting the kinect to work. In *Consumer Depth Cameras for Computer Vision*, pages 141–165. Springer, 2013.

Gregorio Jimenez, Francisco Sanmartín, and Emanuele Mazza. "DELEM - Delayed Mirror". In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology - ACE '05*, pages 196–199, New York, New York, USA, June 2005a. ACM Press. ISBN 1595931104. doi: 10.1145/1178477.1178506. URL http://dl.acm.org/citation.cfm?id=1178477.1178506.

Gregorio Jimenez, Francisco Sanmartín, and Emanuele Mazza. "delem - delayed mirror". In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, ACE '05, pages 196–199, New York, NY, USA, 2005b. ACM. ISBN 1-59593-110-4. doi: 10.1145/1178477.1178506. URL http://doi.acm.org/10.1145/1178477.1178506.

Keith Johnstone. *Impro for storytellers*. Routledge/Theatre Arts Books, 1999.

Neel Joshi, Sisil Mehta, Steven Drucker, Eric Stollnitz, Hugues Hoppe, Matt Uyttendaele, and Michael Cohen. Cliplets: juxtaposing still and dynamic imagery. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 251–260. ACM, 2012.

Hong-Wen Kang, Xue-Quan Chen, Yasuyuki Matsushita, and Xiaoou Tang. Space-time video montage. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1331–1338. IEEE, 2006.

Thorsten Karrer, Moritz Wittenhagen, and Jan Borchers. Draglocks: handling temporal ambiguities in direct manipulation video navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 623–626, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2207764. URL http://doi.acm.org/10.1145/2207676.2207764.

G Khut. Development and evaluation of participant-centred biofeedback artworks. *Unpublished doctoral exegesis, University of Western Sydney*, 2006.

Tae-Kyun Kim, Shu-Fai Wong, and Roberto Cipolla. Tensor canonical correlation analysis for action classification. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

Don Kimber, Tony Dunnigan, Andreas Girgensohn, Frank Shipman, Thea Turner, and Tao Yang. Trailblazing: Video playback control by direct object manipulation. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1015–1018. IEEE, 2007.

Michael Kirby and Victoria Nes Kirby. *Futurist performance*. Dutton New York, 1971.

Lev Vladimirovič Kulešov. *Kuleshov on film: writings by Lev Kuleshov*. Univ of California Press, 1974.

Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824. IEEE, 2011.

Paul Lapides, Ehud Sharlin, and Mario Costa Sousa. Social comics: a casual authoring game. In *Proceedings of the 25th BCS Conference on Human-Computer Interaction*, BCS-HCI '11, pages 259–268, Swinton, UK, UK, 2011. British Computer Society. URL `http://dl.acm.org/citation.cfm?id=2305316.2305364`.

Brenda Laurel. *Computers as theatre*. Addison-Wesley, 1991.

Elizabeth LeCompte. South bank show. In *The Wooster Group Theater Company*. 1987.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Hyeon-Kyu Lee and Jin-Hyung Kim. An hmm-based threshold model approach for gesture recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(10):961–973, 1999.

Hyowon Lee, Alan F. Smeaton, Catherine Berrut, Noel Murphy, Seán Marlow, and Noel E. O'Connor. Implementation and analysis of several keyframe-based browsing interfaces to digital video. In *Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries*, ECDL '00, pages 206–218, London, UK, UK, 2000. Springer-Verlag. ISBN 3-540-41023-6. URL `http://dl.acm.org/citation.cfm?id=646633.699901`.

Johnny C. Lee, Paul H. Dietz, Dan Maynes-Aminzade, Ramesh Raskar, and Scott E. Hudson. Automatic projector calibration with embedded light sensors. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, UIST '04, pages 123–126, New York, NY, USA, 2004. ACM. ISBN 1-58113-957-8. doi: 10.1145/1029632.1029653. URL `http://doi.acm.org/10.1145/1029632.1029653`.

Jinna Lei, Xiaofeng Ren, and Dieter Fox. Fine-grained kitchen activity recognition using rgb-d. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 208–211. ACM, 2012.

Francis C. Li, Anoop Gupta, Elizabeth Sanocki, Li-wei He, and Yong Rui. Browsing digital video. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, CHI '00, pages 169–176, New York, NY, USA, 2000. ACM. ISBN 1-58113-216-6. doi: 10.1145/332040.332425. URL `http://doi.acm.org/10.1145/332040.332425`.

LiVES. Lives. `http://lives.sourceforge.net/`.

Lian Loke and Toni Robertson. Moving and making strange: An embodied approach to movement-based interaction design. *ACM Trans. Comput.-Hum. Interact.*, 20(1):7:1–7:25, April 2013. ISSN 1073-0516. doi: 10.1145/2442106.2442113. URL http://doi.acm.org/10.1145/2442106.2442113.

Yu-Fei Ma, Lie Lu, Hong-Jiang Zhang, and Mingjing Li. A user attention model for video summarization. In *Proceedings of the Tenth ACM International Conference on Multimedia*, MULTIMEDIA '02, pages 533–542, New York, NY, USA, 2002. ACM. ISBN 1-58113-620-X. doi: 10.1145/641007.641116. URL http://doi.acm.org/10.1145/641007.641116.

I Scott MacKenzie and R William Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI'03 extended abstracts on Human factors in computing systems*, pages 754–755. ACM, 2003.

Brian Magerko and Mark O Riedl. What happens next?: Toward an empirical investigation of improvisational theatre. In *Proceedings of the 5th International Joint Workshop on Computational Creativity*, 2008.

Brian Magerko, Waleed Manzoul, Mark Riedl, Allan Baumer, Daniel Fuller, Kurt Luther, and Celia Pearce. An empirical study of cognition and theatrical improvisation. In *Proceedings of the seventh ACM conference on Creativity and cognition*, pages 117–126. ACM, 2009.

Brian Magerko, Casey Fiesler, Allan Baumer, and Daniel Fuller. Bottoms up: improvisational micro-agents. In *Proceedings of the Intelligent Narrative Technologies III Workshop*, INT3 '10, pages 8:1–8:8, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0022-3. doi: 10.1145/1822309.1822317. URL http://doi.acm.org/10.1145/1822309.1822317.

Magisto. Magisto. http://www.magisto.com/.

Sébastien Marcel. Hand posture and gesture datasets. http://www.idiap.ch/resource/gestures/.

Shinichi Maruyama, 2013. URL http://www.shinichimaruyama.com/.

Michael Mateas and Andrew Stern. Façade: An experiment in building a fully-realized interactive drama. In *Game Developers Conference, Game Design track*, volume 2, page 82, 2003.

Justin Matejka, Tovi Grossman, and George Fitzmaurice. Swift: Reducing the effects of latency in online video scrubbing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 637–646, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2207766. URL http://doi.acm.org/10.1145/2207676.2207766.

Justin Matejka, Tovi Grossman, and George Fitzmaurice. Swifter: Improved online video scrubbing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1159–1168, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2466149. URL http://doi.acm.org/10.1145/2470654.2466149.

Winsor McCay. *Gertie the dinosaur*. Museum of Modern Art, 1909.

Robert McKee. *Story: Style, Structure, Substance, and the Principles of Screenwriting*. It Books, 1997.

Andrew Stern Michael Mateas. Writing façade: A case study in procedural authorship. *Second Person*, 2007.

Yoshiyuki Miwa, Shiroh Itai, Takabumi Watanabe, and Hiroko Nishi. Shadow awareness. In *ACM SIGGRAPH 2011 Art Gallery on - SIGGRAPH '11*, page 325, New York, New York, USA, August 2011. ACM Press. ISBN 9781450309646. doi: 10.1145/2019342.2019347. URL http://dl.acm.org/citation.cfm?id=2019342.2019347.

Dyna Moe. Nobody's sweetheart, 2007. URL http://www.nobodyssweetheart.com/drillpress/index.php/2007/02/03/ready-to-order-harold-poster/.

Mehrnaz Mostafapour and Mark Hancock. Exploring narrative gestures on digital surfaces. In *Proc. of ACM ITS*, 2014.

Jörg Müller, Robert Walter, Gilles Bailly, Michael Nischt, and Florian Alt. Looking glass: a field study on noticing interactivity of a shop window. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pages 297–306. ACM, 2012.

Janet H Murray. *Hamlet on the holodeck: The future of narrative in cyberspace*. Simon and Schuster, 1997.

Miguel A. Nacenta, Yemliha Kamber, Yizhou Qiang, and Per Ola Kristensson. Memorability of pre-designed and user-defined gesture sets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1099–1108, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2466142. URL http://doi.acm.org/10.1145/2470654.2466142.

Michael Nyman. *Experimental music: Cage and beyond*, volume 9. Cambridge University Press, 1999.

Oblong Industries. Tamper. URL http://oblong.com/blog/posts/oblong-at-altitude-sundance-2009.

Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 53–60. IEEE, 2013.

Shogo Okada, Mayumi Bono, Katsuya Takanashi, Yasuyuki Sumi, and Katsumi Nitta. Context-based conversational hand gesture classification in narrative interaction. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 303–310. ACM, 2013.

Jeffrey A Okun and Susan Zwerman. *The VES handbook of visual effects: industry standard VFX practices and procedures*. Taylor & Francis, 2010.

Sara Owsley, David A. Shamma, Kristian J. Hammond, Shannon Bradshaw, and Sanjay Sood. The association engine: a free associative digital improviser. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 766–767, New York, NY, USA, 2004. ACM. ISBN 1-58113-893-8. doi: 10.1145/1027527.1027706. URL http://doi.acm.org/10.1145/1027527.1027706.

Galen Panger. Kinect in the kitchen: testing depth camera interactions in practical home environments. In *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts*, pages 1985–1990. ACM, 2012.

Donovan H Parks and Sidney S Fels. Evaluation of background subtraction algorithms with post-processing. In *Advanced Video and Signal Based Surveillance, 2008. AVSS'08. IEEE Fifth International Conference on*, pages 192–199. IEEE, 2008.

Kadir A Peker and Ajay Divakaran. Adaptive fast playback-based video skimming using a compressed-domain visual complexity measure. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 3, pages 2055–2058. IEEE, 2004.

Mark Perry, Oskar Juhlin, Mattias Esbjörnsson, and Arvid Engström. Lean collaboration through video gestures: co-ordinating the production of live televised sport. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2279–2288, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1519051. URL `http://doi.acm.org/10.1145/1518701.1519051`.

Peter Petralia. Instance: The fragmented stage of virtuoso (working title). *Mapping Intermediality in Performance*, pages 156 – 162, 2010.

Peggy Phelan. The politics of performance. *London and New York: Routledge*, 4, 1993.

Ben Piper and Stefan Agamanolis. Palimpsest: a layered video manuscript of social interaction. URL `http://web.media.mit.edu/~stefan/hc/projects/palimpsest/`.

Andreya Piplica, Christopher DeLeon, and Brian Magerko. Full-body gesture interaction with improvisational narrative agents. In *Proceedings of the 12th international conference on Intelligent Virtual Agents*, IVA'12, pages 514–516, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-33196-1. doi: 10.1007/978-3-642-33197-8_63. URL `http://dx.doi.org/10.1007/978-3-642-33197-8_63`.

Víctor Ponce-López, Sergio Escalera, and Xavier Baró. Multi-modal social signal analysis for predicting agreement in conversation settings. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 495–502. ACM, 2013.

Suporn Pongnumkul, Jue Wang, Gonzalo Ramos, and Michael Cohen. Content-aware dynamic timeline for video browsing. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 139–142, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0271-5. doi: 10.1145/1866029.1866053. URL `http://doi.acm.org/10.1145/1866029.1866053`.

Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6): 976–990, 2010.

Yael Pritch, Alex Rav-Acha, and Shmuel Peleg. Nonchronological video synopsis and indexing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1971–1984, November 2008. ISSN 0162-8828. doi: 10.1109/TPAMI.2008.29. URL `http://dx.doi.org/10.1109/TPAMI.2008.29`.

Gonzalo Ramos and Ravin Balakrishnan. Fluid interaction techniques for the control and annotation of digital video. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, UIST '03, pages 105–114, New York, NY, USA, 2003. ACM. ISBN 1-58113-636-6. doi: 10.1145/964696.964708. URL `http://doi.acm.org/10.1145/964696.964708`.

Jef Raskin. *The Humane Interface: New Directions for Designing Interactive Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000. ISBN 0-201-37937-6.

Alex Rav-Acha, Yael Pritch, and Shmuel Peleg. Making a long video short: Dynamic video synopsis. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 435–441. IEEE, 2006.

Stuart Reeves, Steve Benford, Claire O'Malley, and Mike Fraser. Designing the spectator experience. In *Proc. of ACM CHI*, pages 741–750, New York, NY, USA, 2005. ISBN 1-58113-998-5. doi: 10.1145/1054972.1055074. URL http://doi.acm.org/10.1145/1054972.1055074.

Julie Rico. Evaluating the social acceptability of multimodal mobile interactions. In *Proc. of ACM CHI EA*, pages 2887–2890, New York, NY, USA, 2010. ISBN 978-1-60558-930-5. doi: 10.1145/1753846.1753877. URL http://doi.acm.org/10.1145/1753846.1753877.

Anya Peterson Royce. *Anthropology of the Performing Arts: Artistry, Virtuosity, and Interpretation in Cross-Cultural Perspective*. Rowman Altamira, 2004.

Michael Rubinstein, Ariel Shamir, and Shai Avidan. Multi-operator media retargeting. In *ACM SIGGRAPH 2009 papers*, SIGGRAPH '09, pages 23:1–23:11, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-726-4. doi: 10.1145/1576246.1531329. URL http://doi.acm.org/10.1145/1576246.1531329.

Jaime Ruiz and Yang Li. Doubleflip: a motion gesture delimiter for mobile interaction. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 2717–2720. ACM, 2011.

Kimiko Ryokai, Stefan Marti, and Hiroshi Ishii. Designing the world as your palette. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1037–1049, New York, NY, USA, 2005. ACM. ISBN 1-59593-002-7. doi: 10.1145/1056808.1056816. URL http://doi.acm.org/10.1145/1056808.1056816.

Kimiko Ryokai, Stefan Marti, and Hiroshi Ishii. I/o brush: Beyond static collages. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '07, pages 1995–2000, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-642-4. doi: 10.1145/1240866.1240938. URL http://doi.acm.org/10.1145/1240866.1240938.

Mukesh Kumar Saini, Raghudeep Gadde, Shuicheng Yan, and Wei Tsang Ooi. Movimash: online mobile video mashup. In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 139–148, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1089-5. doi: 10.1145/2393347.2393373. URL http://doi.acm.org/10.1145/2393347.2393373.

Chris Salter. *Entangled: Technology and the Transformation of Performance*. The MIT Press, 2010.

Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. Video textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 489–498, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. ISBN 1-58113-208-5. doi: 10.1145/344779.345012. URL http://dx.doi.org/10.1145/344779.345012.

Julia Schwarz, Charles Claudius Marais, Tommer Leyvand, Scott E. Hudson, and Jennifer Mankoff. Combining body pose, gaze, and gesture to determine intention to interact in vision-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 3443–3452, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2556989. URL http://doi.acm.org/10.1145/2556288.2556989.

Jeffrey Scott. *Improvisation in the theatre: An intersection between history, practice, and chaos theory*. PhD thesis, Texas Tech University, 2014.

William Shakespeare. *Hamlet*. First Folio, 1603.

Masahito Shiba, Asako Soga, and Jonah Salz. A cg projection method of supporting to stage live performances. In *Proceedings of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry*, VRCAI '10, pages 67–70, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0459-7. doi: 10.1145/1900179.1900192. URL http://doi.acm.org/10.1145/1900179.1900192.

Garth Shoemaker, Anthony Tang, and Kellogg S. Booth. Shadow reaching: a new perspective on interaction for large displays. In *Proc. of ACM UIST*, pages 53–56, 2007. ISBN 978-1-59593-679-0. doi: http://doi.acm.org/10.1145/1294211.1294221. URL http://doi.acm.org/10.1145/1294211.1294221.

Garth Shoemaker, Takayuki Tsukitani, Yoshifumi Kitamura, and Kellogg S. Booth. Body-centric interaction techniques for very large wall displays. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, pages 463–472, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-934-3. doi: 10.1145/1868914.1868967. URL http://doi.acm.org/10.1145/1868914.1868967.

Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

Prarthana Shrestha, Peter H.N. de With, Hans Weda, Mauro Barbieri, and Emile H.L. Aarts. Automatic mashup generation from multiple-camera concert recordings. In *Proceedings of the international conference on Multimedia*, MM '10, pages 541–550, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-933-6. doi: 10.1145/1873951.1874023. URL http://doi.acm.org/10.1145/1873951.1874023.

João Silva, Diogo Cabral, Carla Fernandes, and Nuno Correia. Real-time annotation of video objects on tablet computers. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, pages 19:1–19:9, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1815-0. doi: 10.1145/2406367.2406391. URL http://doi.acm.org/10.1145/2406367.2406391.

Single Thread Theatre Company. URL http://singlethread.ca/.

Scott Snibbe. Body, screen and shadow. *San Francisco Media Arts Council (SMAC) Journal*, 2003.

Peng Song, Wooi Boon Goh, William Hutama, Chi-Wing Fu, and Xiaopei Liu. A handle bar metaphor for virtual object manipulation with mid-air interaction. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pages 1297–1306. ACM, 2012.

Viola Spolin. *Improvisation for the theater: A handbook of teaching and directing techniques*. Northwestern University Press Evanston, IL, 1983.

Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

Jim Steinmeyer. The science behind the ghost: A brief history of pepper's ghost, 1999.

Sanghoon Sull, Jung-Rim Kim, Yunam Kim, Hyun S Chang, and Sang U Lee. Scalable hierarchical video summary and search. In *Photonics West 2001-Electronic Imaging*, pages 553–561. International Society for Optics and Photonics, 2001.

Anthony Tang, Saul Greenberg, and Sidney Fels. Exploring video streams using slit-tear visualizations. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '08, pages 191–198, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-141-5. doi: 10.1145/1385569.1385601. URL `http://doi.acm.org/10.1145/1385569.1385601`.

Stuart Taylor, Shahram Izadi, David Kirk, Richard Harper, and Armando Garcia-Mendoza. Turning the tables: an interactive surface for vjing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1251–1254, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518888. URL `http://doi.acm.org/10.1145/1518701.1518888`.

ART Teknika. Colorcode vj. URL `http://colorcodevj.artteknika.com/index_en.html`.

Laura Teodosio and Walter Bender. Salient stills. *ACM Trans. Multimedia Comput. Commun. Appl.*, 1(1): 16–36, February 2005. ISSN 1551-6857. doi: 10.1145/1047936.1047940. URL `http://doi.acm.org/10.1145/1047936.1047940`.

The Builders Association. Xtravaganza, 2002. URL `http://www.thebuildersassociation.org/prod_xtravaganza_info.html`.

The Neutrino Video Project. The neutrino video project. URL `http://wiki.improvresourcecenter.com/index.php?title=The_Neutrino_Video_Project`.

Blast Theory. Ten backwards. DVD, 1999.

James Tompkin, Kwang In Kim, Jan Kautz, and Christian Theobalt. Videoscapes: exploring sparse, unstructured video collections. *ACM Trans. Graph.*, 31(4):68:1–68:12, July 2012. ISSN 0730-0301. doi: 10.1145/2185520.2185564. URL `http://doi.acm.org/10.1145/2185520.2185564`.

Ba Tu Truong and Svetha Venkatesh. Video abstraction: A systematic review and classification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 3(1), February 2007. ISSN 1551-6857. doi: 10.1145/1198302.1198305. URL `http://doi.acm.org/10.1145/1198302.1198305`.

Shuhei Tsuchida, Tsutomu Terada, and Masahiko Tsukamoto. A system for practicing formations in dance performance supported by self-propelled screen. In *Proc. of ACM AH*, pages 178–185, 2013. ISBN 978-1-4503-1904-1. doi: 10.1145/2459236.2459266. URL `http://doi.acm.org/10.1145/2459236.2459266`.

Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18 (11):1473–1488, 2008.

Marina Turco. Instance: Intermediality in vjing: Two vj sets by gerald van der kaap. *Mapping Intermediality in Performance*, pages 56–62, 2010.

Tim Uren. Finding the game in improvised theater. *Second Person*, pages 279–283, 2007.

Cati Vaucelle and Hiroshi Ishii. Play-it-by-eye! collect movies and improvise perspectives with tangible video objects. *Artif. Intell. Eng. Des. Anal. Manuf.*, 23(3):305–316, August 2009. ISSN 0890-0604. doi: 10.1017/S0890060409000262. URL http://dx.doi.org/10.1017/S0890060409000262.

Bret Victor. Inventing on principle. URL http://worrydream.com/#!/InventingOnPrinciple.

Vine. Vine. URL http://vine.co/.

Vjay. Vjay ipad app by algoriddim. http://vjapp.com/vjay-ipad-app-by-algoriddim/.

Daniel Vogel and Ravin Balakrishnan. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *Proc. of ACM UIST*, pages 137–146, 2004.

Daniel Vogel and Ravin Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 33–42. ACM, 2005.

Vyclone. Vyclone. http://vyclone.com/.

Julie Wagner, Stéphane Huot, and Wendy Mackay. Bitouch and bipad: Designing bimanual interaction for hand-held tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2317–2326, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208391. URL http://doi.acm.org/10.1145/2207676.2208391.

David Foster Wallace. *Infinite jest*. Back Bay Books, 2009.

Dan Walsh. Garfield minus garfield. URL http://garfieldminusgarfield.net/.

Robert Walter, Gilles Bailly, and Jörg Müller. Strikeapose: Revealing mid-air gestures on public displays. 2013.

Jinjun Wang, Changsheng Xu, Engsiong Chng, Lingyu Duan, Kongwah Wan, and Qi Tian. Automatic generation of personalized music sports video. In *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA '05, pages 735–744, New York, NY, USA, 2005a. ACM. ISBN 1-59593-044-2. doi: 10.1145/1101149.1101309. URL http://doi.acm.org/10.1145/1101149.1101309.

Jue Wang, Pravin Bhat, R. Alex Colburn, Maneesh Agrawala, and Michael F. Cohen. Interactive video cutout. *ACM Trans. Graph.*, 24(3):585–594, July 2005b. ISSN 0730-0301. doi: 10.1145/1073204.1073233. URL http://doi.acm.org/10.1145/1073204.1073233.

Yu-Shuen Wang, Jen-Hung Hsiao, Olga Sorkine, and Tong-Yee Lee. Scalable and coherent video resizing with per-frame optimization. In *ACM SIGGRAPH 2011 papers*, SIGGRAPH '11, pages 88:1–88:8, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0943-1. doi: 10.1145/1964921.1964983. URL http://doi.acm.org/10.1145/1964921.1964983.

Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.

Feng Xu, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, and Christian Theobalt. Video-based characters: creating new human performances from a multi-view video database. In *ACM SIGGRAPH 2011 papers*, SIGGRAPH '11, pages 32:1–32:10, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0943-1. doi: 10.1145/1964921.1964927. URL http://doi.acm.org/10.1145/1964921.1964927.

Sam Yip, Eugenia Leu, and Hunter Howe. The automatic video editor. In *Proc. of ACM MULTIMEDIA*, pages 596–597, New York, NY, USA, 2003. ISBN 1-58113-722-2. doi: 10.1145/957013.957138. URL http://doi.acm.org/10.1145/957013.957138.

YouTube Capture. Youtube capture. URL http://www.youtube.com/capture.

Nicolas J Zaunbrecher. The elements of improvisation: Structural tools for spontaneous theatre. *Theatre Topics*, 21(1):49–60, 2011.

H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu. Video parsing, retrieval and browsing: an integrated and content-based solution. In *Proceedings of the third ACM international conference on Multimedia*, MULTIMEDIA '95, pages 15–24, New York, NY, USA, 1995. ACM. ISBN 0-89791-751-0. doi: 10.1145/217279.215068. URL http://doi.acm.org/10.1145/217279.215068.

Jamie Zigelbaum, Alan Browning, Daniel Leithinger, Olivier Bau, and Hiroshi Ishii. g-stalt: a chirocentric, spatiotemporal, and telekinetic gestural interface. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, TEI '10, pages 261–264, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-841-4. doi: 10.1145/1709886.1709939. URL http://doi.acm.org/10.1145/1709886.1709939.

Jamie B Zigelbaum. *Mending fractured spaces: external legibility and seamlessness in interface design*. PhD thesis, Massachusetts Institute of Technology, 2008.

Eric Zimmerman. Creating a meaning-machine: The deck of stories called life in the garden. *Second Person*, 2007.

C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM Trans. Graph.*, 23(3):600–608, August 2004. ISSN 0730-0301. doi: 10.1145/1015706.1015766. URL http://doi.acm.org/10.1145/1015706.1015766.

Alexander Zook, Brian Magerko, and Mark Riedl. Formally modeling pretend object play. In *Proceedings of the 8th ACM conference on Creativity and cognition*, C&C '11, pages 147–156, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0820-5. doi: 10.1145/2069618.2069644. URL http://doi.acm.org/10.1145/2069618.2069644.

Vilmos Zsombori, Michael Frantzis, Rodrigo Laiola Guimaraes, Marian Florin Ursu, Pablo Cesar, Ian Kegel, Roland Craigie, and Dick C.A. Bulterman. Automatic generation of video narratives from shared ugc. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, HT '11, pages 325–334, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0256-2. doi: 10.1145/1995966.1996009. URL http://doi.acm.org/10.1145/1995966.1996009.