High Degree of Freedom Input and Large Displays for Accessible Animation

by

Noah Lockwood

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

Abstract

High Degree of Freedom Input and Large Displays for Accessible Animation

Noah Lockwood

Master of Science

Graduate Department of Computer Science

University of Toronto

2006

We explore methods of making computer animation accessible to nontechnical users by utilizing high degree–of–freedom, gestural input and visualization on a large display. Design principles are developed which guide our approach and focus on expressive but comprehensible techniques. We present two applications of these principles to animation tasks: first, to animatic creation, where we develop techniques for rough animation and cinematic communication; and second, to facial animation, where direct manipulation mediates the complexity of a fully expressive virtual face. We present preliminary user evaluation of our techniques, and discuss future directions of our approach.

Contents

Chapter 1

Introduction

In recent years, computer–generated films have evolved from a rarely-seen medium to an almost ubiquitous one. Computer animation has become a fully–fledged medium with both short and feature–length films in a variety of genres, tackling an ever–increasing range of subject matter. The number of computer graphics professionals is increasing, as are the number of computer–animated films released each year.

One of the tremendous advantages of computer animation is also one of its greatest pitfalls. Digital creations aren't limited by any real–world rules and it is a medium where truly anything is possible. However, being wholly removed from the real world has its drawbacks: *everything* must be created by hand, from scratch. Creating a computer–animated film is a multi–year production involving hundreds of artists and technicians steadily creating and animating characters, sets, props and lights. Furthermore, computer graphics methodology and technology have not advanced in a significant way since almost the inception of the field – many things are still done "the hard way" since no one has bothered to invent a better way. Bill Buxton succinctly summarizes the effect of this combination of complexity and status quo: "3D is difficult"[15].

Such difficulty is ultimately dangerous for the field as a whole, since it leaves production in the hands of the technically–skilled, who may not have the creative or artistic

abilities to craft an effective film. One possible solution has recently occurred in live–action film: a preponderance of easy–to–use, powerful and inexpensive equipment and software, which eliminates this technical expertise gap. Robert Redford, among others, calls this "...the democratization of film, because it opens up the world of filmmaking to anyone who wants to do it"[69]. The resulting rise and success of independent films, significantly different from formulaic studio films, has infused the medium with a new energy. Which leads one to ask: where is the democratization of computer graphics? If "anyone" can make a film, why can't anyone create computer graphics?

There has been scattered research in a variety of areas, which has explored potential ways to simplify parts of computer graphics production. 3D models can be created by free–form sketching[43, 111], or by manipulating parameterized models of everyday objects[109] or even the human body[2]. Virtual clothing[41] and hair[67] can be easily manipulated. Cameras can be interactively manipulated[53, 107, 110] or automatically controlled[36, 24] for cinematic effect. Animators can interactively control physically–based characters[61, 113] or objects[84, 85], as well as precisely and expressively orchestrate the animation of characters in new ways[23, 65, 102].

In addition to exploring some of those specific issues, our research further investigates their underlying motivation: to create a computer graphics system which is interactive and usable by non–technical users, which we refer to as *accessibility*. As well, we would like to explore system *expressivity*, which we define as the ability to craft a final result that is close to the user's vision. We develop a set of design principles, grounded in established Human–Computer Interaction research, which guide us towards techniques and methodologies that maximize the capabilities of the user. Our design goal is an animation system that is both accessible and expressive, allowing users to easily create across a broad range of possibilities.

We explore our design idea primarily within the specific implementation of a prototype animatic creation system, dubbed DirectCam. The particular domain of animatic

creation is eminently relevant for our design principles, since it involves non–technical users roughly exploring a broad space of creative possibilities, usually in a group or collaborative setting. We address this issue by implementing our design principles in the form of intuitive bimanual gestural input, fluid control of 3D objects and animation, and visualizations and techniques that facilitate collaboration in addition to viewer–centered production. We also apply our design principles to facial animation, a technically complex problem which is preliminarily explored by expanding upon the DirectCam system.

We have also performed preliminary user evaluation on our DirectCam system. We have run exploratory sessions with several users unskilled in 3D animation, with promising results. In addition, we have explored and discussed the system in–depth with an accomplished animator and computer animation director. These have allowed us to keep in mind the broad target audience for our system, and we have implemented and designed corresponding improvements.

1.1 Contributions

The main contributions of our work arise from the development of our design principles, as well as specific aspects of our DirectCam system implementation.

Bimanual, Gestural Interface

We specifically utilize the familiarity users have communicating and manipulating with their hands, which we use as high degree–of–freedom input. We synthesize novel techniques from well–established research in areas such as bimanual input, direct manipulation, kinesthetic feedback and view manipulation. In addition, we maintain a small but expressive vocabulary of postures for the user to remember, as well as intuitive, continuous gestural control of all system operations.

Group–Appropriate Techniques

We design our system with asymmetric group work in mind: one user who interacts with the system while collaborating with active observers. We utilize a large display and maintain straightforward and informative visualizations for audience comprehension. In addition, we maintain a simple, intuitive and explicit gestural control vocabulary to inform observers not only of the current system operation, but to implicitly instruct them in use of the system.

Fluid, Accessible Control

Our system eliminates the many modes and commands present in modern 3D software. Instead, we present a fluid system where object creation and manipulation, camera control, and animation are fluidly integrated and can easily be switched between. We improve upon conventional object manipulation tools with a direct manipulation–based interface which is simple to understand, and we maintain a through–the–lens perspective to focus on the final 2D appearance of the product over 3D correctness. In addition, we also abstract keyframe animation of objects with high–level direct manipulation including a simple yet powerful rhythmic retiming technique.

1.2 Thesis Overview

Though this thesis covers a variety of issues, this research is ongoing. We present its current state, along with a thorough survey of related literature, discussion of the design principles, details of the prototype implementation, user evaluation, and future direction.

In chapter 2, *Background*, we examine relevant prior work relating to large display interaction, gestural and high degree–of–freedom input, camera control, object selection and manipulation, animation control, and facial animation.

In chapter 3, *Design Principles*, we present a set of principles to guide our research.

We draw these principles from established Human–Computer Interaction literature, as well as relevant and successful prior work in accessible computer graphics systems.

In chapter 4, *Animation Systems*, we describe the implemented systems, guided by our design principles. We describe our methodology for bimanual gestural input, as well as the visualization and interaction techniques for our animatic creation and facial animation systems.

In chapter 5, *User Evaluation*, we discuss the results of our preliminary user tests. We also identify design implications based on the findings.

In chapter 6, *Conclusions and Future Directions*, we summarize our work and its contributions. We discuss future directions for the work, including an in–depth user evaluation.

Chapter 2

Background

Our work draws upon a variety of related research in both Human–Computer Interaction and Computer Graphics. We have grouped these works into six categories: *large display interaction*, *gestural and high degree–of–freedom input*, *camera control*, *object selection and manipulation*, *animation control*, and *facial animation*. With respect to large displays, an enabling technology of this work which allows its group application, we are particularly interested in group–appropriate techniques and methods of at–a–distance interaction. The other important technology for our system is bimanual gestural input, which allows a richer and more powerful input vocabulary than regular input devices. We also gain insight by examining general work concerning high degree–of–freedom input. Controlling the camera view is particularly important, both to give the user an understanding of the virtual space that they're interacting with, but also to orchestrate camera positioning and movements for the final animated product. We examine methods of selecting and manipulating objects to provide the most intuitive and effective methods for a user to directly manipulate our system. Animation control research explores methods to simplify the complex process of character animation, while retaining expressivity. Finally, facial animation is a crucial component of any animated production, and we examine methods of simplifying this process as well.

2.1 Large Display Interaction

Traditionally, computer animation is performed with the same set of devices, regardless of the particular tool or application being used. Input is consists of either mouse or tablet devices, which use standard techniques for a familiar point–and–click interface. Display technology is even more standardized, with animators working on desktop monitors, though sometimes a second monitor is added to reduce window switching. In this work, we wish to explore the potential of other display options for animation tasks. There is previous work using nonstandard displays which is extensible to animation, such as the wealth of virtual environment research[12, 81, 86, 87, 91, 96] or recent explorations into interaction with volumetric displays[31]. However, we note a lack of substantial explorations into using large displays for purposes relevant to animation – in particular, techniques specialized for interacting with 3D scenes on a large display are conspicuously absent.

There is substantial evidence that even single users can gain from utilizing a large display over a traditional monitor, especially for 3D tasks. Tan et al.[99] present research comparing 3D navigation tasks performed on both a monitor and distant large display. For consistency, they position the displays so that both share the same viewing angle from the perspective of the user. They found that the navigation tasks were performed more effectively on the large display, which is is attributed to the increased immersive effect of the large display. This allows a greater sense of "presence" in the 3d environment, which facilitates use of familiar spatial strategies used in everyday experience. They also performed a follow–up investigation which established the much larger contribution of wayfinding and spatial understanding to performance, rather than their particular method of interactivity; however, this finding is likely only relevant for navigation tasks where spatial comprehension is of primary importance, unlike the constant scene interaction required for animation.

Complimentary work by Tyndiuk et al.[104] examines manipulation in addition to

Figure 2.1: Up–close interaction on a large display, using canvas portals[9] to access out–of–reach areas.

similar navigation tasks, however they also frame their work in the context of cognitive science as well. They utilize a well–established test to separate participants into categories of high and low visual attentiveness, the first stage of perception which occurs before spatial reasoning. They confirm the earlier findings of performance benefits with large displays for navigation tasks, and additionally find similar gains for manipulation tasks. Additionally, they note that these gains are larger for individuals with low visual attentiveness. While this implies that cognitive measurement could be used to hone the effectiveness of large display interactive, of primary importance for our work are the benefits demonstrated for both view–centered (egocentric) and object–centric (exocentric) tasks.

Large displays also present a variety of challenges related to interaction techniques, since traditional approaches for a desktop setting may not exploit the potentially high resolution of the screen, or even prove a hindrance. Up–close interaction is especially difficult since only a portion of the display is within easy reach. Bezerianos and Balakrishnan[9]

provide an overview of a variety of approaches to this problem, under the common framework of "canvas portals". Techniques within this framework present views of remote screen areas to the user, which can serve for comprehension of the overall display space or facilitate interaction with distant objects (Figure 2.1). These aids can also be interactively adjusted, allowing control of their size and focal area for maximum user control. However, we note that for animation purposes, consistent comprehension of the entire view is essential, making up–close interaction less appropriate.

Other approaches have examined at–a–distance methods of interaction with large displays. Myers et al.[72] compare a variety of interaction approaches against each other, including up–close interaction, a combined–display approach using handheld devices, conventional mouse control, and laser pointer control. While they unsurprisingly found up–close interaction fastest and most accurate, their work in particular makes clear the difficulty of using laser pointers, including imprecision from hand jitter, though techniques which accommodate for this issue have also been investigated[74]. Malik et al.[68] present a novel technique which uses multi–finger gestural input to control a large display. Interaction can occur at any distance by using either or both hands on a "touchpad", which recognizes hand gestures and postures using computer vision techniques. Zoomed–in workspaces can be created, and both hands can be used cooperatively.

Vogel and Balakrishnan[105] also present hand–based techniques for interacting with large displays. They present a system with a variety of interaction phases dependent on user distance from the screen. In the closer phases, hand gestures are used to navigate menus, make selections, and indicate items of interest. The authors' follow–up work[106] presents a more general view, investigating more distant methods of interaction, again using hand–based techniques. They investigate pointing–based interaction, which utilizes raycasting akin to a laser pointer, which results in similar imprecision difficulties due to hand jitter. However, they also present a relative cursor control method using hand motion and a clutching posture to allow for recalibration, as well as a hybrid rel-

ative/raycasting combination technique. Their experiments show comparable results for these two techniques, though interestingly the majority of users preferred the purely relative technique for ease, speed and accuracy.

Finally, using a large display allows for group interactions impossible with a standard monitor. There is a large body of research within the Computer–Supported Cooperative Work (CSCW) field addressing simultaneous user work in a shared virtual workspace. While this could have interesting applications towards fully collaborative animation where users collaboratively manipulate a scene or character, we choose not to focus on such an approach at present. However, we do wish to investigate asymmetric group interactions, where a group can collaborate while one user at a time interacts with the system. Hawkey et al.[35] present research that, while focusing on collaboration with multiple active participants, holds relevance for our purposes. Their investigation found that user collocation was beneficially for collaboration, even if both users were interacting with the screen from a distance. Users collaborating but separated (one near, one far) had difficulty communicating verbally, and far users were often unable to see or comprehend the direct interaction techniques being used by the near user. In addition, they found that distant interaction was easier using direct input such as a touchscreen, however they note the advantages of indirect input which allows users to concentrate on the same display surface. Their indirect input device, a stylus and tablet, performed poorly though. Interestingly, they do not investigate indirect and relative input, similar to a mouse – Vogel and Balakrishnan[106] cite similarity to mouse input as the likely reason for the preference their users had for their relative technique.

Work not directly within the CSCW field can also be beneficial for our purposes. The PointRight system by Johanson et al.[47] is a versatile system that manages multiple inputs and multiple displays. Their system allows only a single user at a time to interact with a particular display, though they still find beneficial results. For work within a group setting, Khan et al.[54] have developed a “spotlight” technique which allows a user

(a) (b)

Figure 2.2: (a) Videoplace[57]. (b) Charade[8].

to direct group attention on a large display. We can see the potential application of such a technique within an animation task, where particular areas of variable sizes may need to be focused on, making gestural (i.e. pointing) or cursor indication less useful.

2.2 Gestural and High DOF Input

Another technological area that we wish to expand upon is the dimensionality of our input. As noted earlier, computer animation is performed almost exclusively using the standard 2D input devices of either a mouse of stylus/tablet, a distinction between which is usually made by user preference rather than suitability for a particular task. However, our previous examination of Large Display Interaction (Section 2.1, above) revealed that novel interaction methods can be particularly suitable for such a different type of display. In particular, approaches using higher degree–of–freedom input have experienced positive results. These additional input channels can come from spatial data, such as 3D position or rotation, or from greater articulation, such as incorporating controls for individual fingers.

Early work in this area of particular note includes `VIDEOPLACE` by Krueger et al.[57],

which more than 20 years ago utilized full–body input, in the form of silhouette, for interaction with their system. Based on user pose, the system allows different interactions – for example, extending a single finger from a hand allows the user to draw with it, while extending all five fingers erases the drawing. Additional interactions are possible with autonomous "critters" who interact with the user's silhouette and perform different actions, again depending on pose, such as leaping onto an outstretched hand. While the purpose of the work is playful and exploratory, its functionality, expressiveness and immediacy are compelling (Figure 2.2a). Another earlier work, Charade, by Baudel and Beaudouin–Lafon[8], presents gestural input for a more specific task. With one hand in a tethered dataglove, users of their system can control a slide show, issuing commands such as advancing to the next slide or returning to the table of contents (Figure 2.2b). These commands are encompassed by "gestural command sets", where a sequence of starting hand pose, transition movement, and ending hand pose denote a particular command. Based on their work, the authors also present a set of guidelines for designing natural and usable command sets. We especially note their concept of an "active zone", outside of which hand gestures can be used for their normal communicative meaning without inducing system action.

High degree–of–freedom input is also extremely common for controlling or interacting with virtual environments. A variety of approaches are aimed towards facilitating hand–based grasping and manipulating of virtual objects, leveraging the user's familiarity with such interactions from real–world experience. These approaches are generally standardized in terms of input device, involving a 6 DOF (3D rotation and position) glove that can sense finger contraction. We examine these various techniques in detail in Section 2.4, "Object Selection and Manipulation", below. Other approaches involving virtual environments have also focused on using general 6 DOF tracked devices grasped by the user. One such work by Ware and Osborne[107] investigates different view control metaphors with such a device, and is discussed in Section 2.3, "Camera Control", below.

Greater input dimensionality can also be gained with the addition of a second input device, so that each of the user's hands controls multiple degrees of freedom. Such a bimanual approach is unsurprising, given the human tendency for both hands to cooperate in real–world manipulation and interaction tasks. Hinckley[37] provides a thorough overview of this area of research, including the foundational Kinematic Chain theory by Guiard[32]. We further examine this theory in relation to our work in Section 4.3, "System Interaction". This theory is applied by Cutler et al.[21] in their Responsive Workbench, a system for manipulating 3D models in virtual reality. They develop a framework for bimanual interaction, including transitions between tasks that are unimanual, bimanual symmetric, or bimanual asymmetric. Interestingly, they find it helpful to reduce the DOF of input for some tasks; for example, they include the option of constraining rotations to a principal or user–defined axis, instead of allowing full and direct rotation. Furthermore, while Guiard's work is relevant to asymmetric bimanual tasks, where the hands work together but do not perform identical tasks, this does not encompass all possible bimanual operations. Balakrishnan and Hinckley[4] further investigate symmetric bimanual tasks in particular. They found that while the hands can operate in parallel, tasks where the hands do not operate within visual proximity are cognitively taxing. This results in sequential performance instead of parallel, effectively reducing the degrees of freedom of the input.

More theoretical work concerning hand–based input includes Sturman and Zeltzer's[97] design method for "whole hand" interactions. They present an iterative series of procedures, where a designer examines their intended tasks and analyzes possible whole–hand techniques for each element of the task. They also present a whole–hand input taxonomy, which classifies hand input as discrete (snapshot–based) or continuous (motion–based) features. Based on discrete or continuous actions, they categorize possible system interpretations as either direct, mapped, or symbolic. Combinations of these actions and interpretations result in different input styles; for example, continuous mapping applies

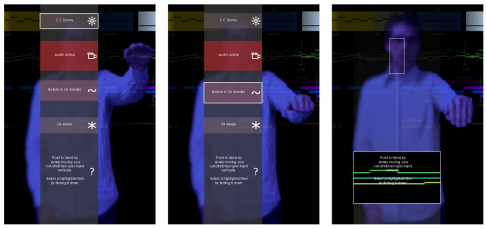
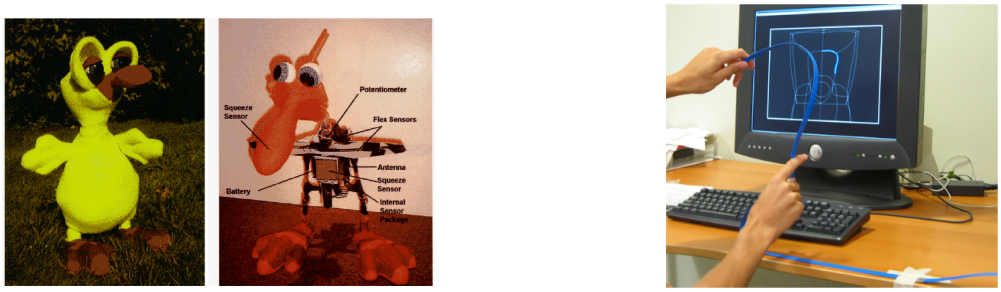


Figure 2.3: Self–revealing help to improve the accessibility of a gestural interface[105].

hand DOF to more abstract system attributes with no kinematic correspondence, while a discrete symbolic style interprets hand poses and motions as input tokens, such as in Charade[8].

Recent work by Vogel and Balakrishnan[105] presents an interactive display controlled by body position and facing as well as hand gestures. The additional degrees of freedom from the user's body are used to indicate interest in the system via proximity, and the system transitions to different "phases" of interaction with appropriately varying levels of user involvement. They present both continuous gestural interaction, such as browsing a list with vertical hand movement, as well as discrete gestures to issue commands, such as dismissing a menu with a flick. One difficulty of gesture–based systems is the often complicated or nonimmediate gestures involved; the authors mediate this with clever visualizations as well as demonstrative, self–revealing help videos (Figure 2.3). Follow–up work by the authors[106] investigates cursor control at a distance using hand input. The authors examine techniques for clicking gestures, as well as a raycasting/relative motion pointing technique reliant on hand posture. Grossman et al.[31] present bimanual gestural interaction techniques for use with a volumetric display. In their system, the user selects the desired 3D operation implicitly with their hand posture, which affords direct, continuous control of that operation. For example, object translation is selected by pinching the fingers of the dominant hand, and subsequent hand translation is mapped to the object. We adopt a similar approach for tool selection and continuous control in



(a) (b)

Figure 2.4: (a) Sympathetic plush toy interface[48], outer and inner views. (b) Shapetape interaction[30].

our system.

High degree–of–freedom input can also come from complex devices which are more freely manipulated, instead of directly interpreting body part positions or rotations. For example, Johnson et al.[48] present a "sympathetic interface" for animating a virtual character. Within a plush doll, they imbed a variety of wireless sensors relaying traditional information such as position and rotation, but also data such as head and limb rotation, as well as squeeze measurements in different locations (Figure 2.4a). The character, a chicken, is controlled sympathetically rather than directly – for example, flapping the doll's wings causes the chicken to fly, executing different flying actions depending on the context. While not a free–form application the system was found to be extremely accessible, especially to children, who connected with the "make–believe" form of interaction better than adults, who had preconceived notions of direct control based on their computer experience. Grossman et al.[30] present techniques utilizing a "shapetape", a flexible ribbon whose bends and twists are sensed by the system. Grasping the tape with both hands allows an variety of shapes and gestures to be formed, which the authors use

to create 3D curves (Figure 2.4b). They also present judicious reduction of input degrees of freedom – for example, a "snapping" gesture of the tape maps its smooth bending, indirectly, to a sharply–cornered line segment.

2.3 Camera Control

A fundamental notion in computer graphics is the mechanism by which three-dimensional data is transformed into a two-dimensional image for the audience to see. Though technically nothing more than a simple matrix representing a projection, the metaphor of the camera is very powerful. It creates an almost immediate comprehension of the projection concept, as well as allowing us to apply more than one hundred years of both still– and motion–picture expertise in camera placement and motion. This knowledge is expanded into the digital realm by the large and well–established body of research in the area of camera control. We identify two primary dimensions along which this work varies. First, *Interactivity* vs. *Automation*; whether the system allows or enhances user control of the view, or if the system controls the camera "intelligently" for a particular purpose. Second, *Assistive* vs. *Compositional*; whether the purpose of the camera is to aid either interaction or spatial understanding, or if the camera emulates a real–life camera with the purpose of generating "cinematic" two–dimensional images. We examine the previous work in all four combinations of these concepts.

Interactive, Assistive Approaches

Interactive camera control began in straightforward way which is still prevalent in 3D applications today. Mouse and keyboard input can be mapped directly to camera degrees of freedom, or DOF, often in a modal way – for example, holding one key may allow mouse movement to control the camera movement in the image plane, while another key allows the camera to truck forwards and backwards, and so on. As well, camera DOF can be

elegantly reduced to allow for simpler and more intuitive control. The common "orbiting" or Arcball technique[94] maps 2D mouse input to the camera's azimuth and elevation around a preset "focus point". While unsuitable for generalized 3D exploration, this technique is particularly effective when examining a fixed point or object, which makes it the de facto standard of camera control for 3D object modeling or painting.

An more general–purpose approach was presented in UniCam[110]. UniCam uses 2D input with a single button, but allows full and direct camera control. The screen is separated into an inner region, where interaction controls camera translation, and a border region, where interaction controls camera orbiting. After selection of a region, subsequent input must be one of a number of possible gestures, which indicates the particular DOF the user wishes to control – for example, gesturing up converts the system into a mode where input controls trucking and horizontal translation. The same stroke can be continued after the gesture to fluidly control the relevant camera DOF, though the direction of control may differ from the selection gesture. Anecdotal evidence is presented towards the technique's effectiveness, however it does not contain any mechanism for scene interaction. Additionally, the large and situation–specific gesture vocabulary requires "a few hours" of training.

Conventional input devices can also be used bimanually to enable synchronous camera control and scene interaction, as shown by Balakrishnan and Kurtenbach[5]. This work used 2D input from the non–dominant hand to control camera orbiting, while 2D input from the dominant hand either selected objects or translated them parallel to the image plane. While benefits in scene comprehension are demonstrated, the authors caution that simultaneous camera control and scene manipulation can present difficulties for the user, at least initially.

In recent years, research into fully interactive camera control using standard input devices has become increasingly rare. This line of research has been expanded, however, with approaches that combine interactive control with real-time *assistive* computations

(a) (b)

Figure 2.5: (a) User view of a virtual environment and WIM[96] model. (b) A user manipulating the WIM (tablet) and interacting with the scene (sphere).

– see "Hybrid Approaches", below.

There has also been a significant amount of investigation into camera control using nonstandard input devices. Most of this work is based at least partially on pioneering work by Ware and Osborne[107], which gave users a handheld six DOF device. The authors investigated three metaphors for camera control with this device. The first, "eyeball in hand", directly maps input DOF to the corresponding camera DOF. The second, "scene in hand", maps input DOF to the corresponding DOF of the scene itself while the camera remains still. Finally, the "flying vehicle" metaphor maps changes in the input's DOF to the velocity of the camera's DOF. The authors identify a continuum of 3D tasks requiring view control: at one end, examining objects, where the "scene in hand" metaphor was found most useful; and at the other end, scene explorations, where the "flying vehicle" metaphor was best suited. The authors also note the importance of scene manipulation techniques (which they do not investigate) that conceptually connect to the camera control metaphor.

An extension of the "scene in hand" metaphor which does incorporate scene manip-

ulation is Worlds In Miniature by Stoakley et al.[96]. The authors investigate issues surrounding immersive displays – in particular, how to gain the benefits of camera control without disorienting the user, as well as how to interact with objects that may be "out of reach", virtually speaking. Their solution was the World in Miniature, or WIM, a miniaturized version of the 3D scene which is suspended in the user's field of view in a heads–up–display manner (Figure 2.5). The WIM is controlled by one hand while the other can be used to manipulate objects in the "real" scene or within the WIM, both of which are coupled so that changes to one appear in the other. In terms of camera control, the WIM approach allowed the user to better understand the scene surrounding their viewpoint while allowing object manipulation, without requiring explicit modes or commands.

Interactive, Compositional Approaches

There is surprisingly little research aimed toward a user who wishes to generate composition–centric camera views and movements. Users who wish to position a camera for maximum effect, whether it is to achieve some image–based goals or to work within cinematic rules, are usually faced with direct camera controls and a system with no "understanding" of the final image. One approach by Gleicher and Witkin[27] offers *through–the–lens* control. In this work, the user can select a visible 3D point in the scene and in real–time, control its resulting image–space projection. Multiple points can be selected and positioned in sequence. One of the reasons for their real-time implementation is the intractability of a closed–form solution, which in turn necessitates an approach using the time derivatives of the user input to solve for the time derivatives of the camera's DOF. However, their interactive approach is quite appealing in the manner that it allows the user to craft their perspective by assembling the final image, one piece at a time. Even more appealing is that the system is entirely reliant on user input – the system doesn't "know" what it's viewing at all.

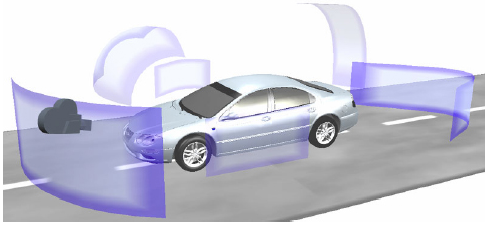
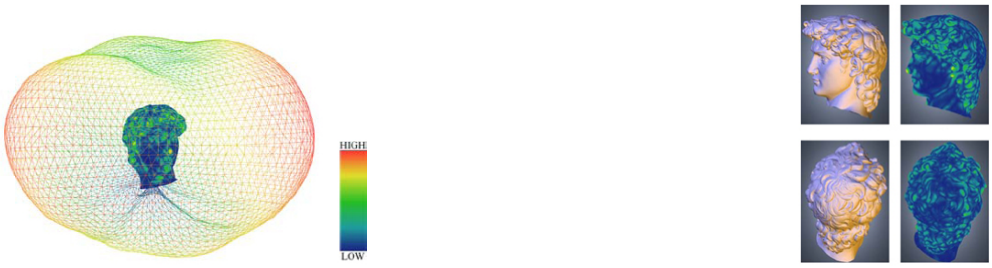


Figure 2.6: Camera surfaces in StyleCam[14].

A more recent approach is the StyleCam by Burtnyk et al.[14]. StyleCam approaches the related issues slightly differently by identifying two different parties in interactive viewing: the author and the user. The system allows the author to use a variety of structures which constrain the interaction of the user, though this restriction is in order to present a more stylized and polished viewing experience. Interactively placed camera surfaces dictate the regions which a camera can occupy, while also controlling the viewing angle and camera speed (through variable control–display gain) for maximum cinematic effect (Figure 2.6). The author can also create stylized transitions between these surfaces, which can include camera paths of any complexity or 2D "slate" transitions. User interaction consists of simple 2D dragging, which moves the camera along the camera surfaces and triggers transitions when edges are reached. The user can also control the playback of the transitions, again by dragging.

Automated, Assistive Approaches

To clarify, fully automated yet assistive camera control systems cannot technically assist user interaction by virtue of their automation – their is, after all, no user interaction to assist. However, these systems can assist a user in comprehending a scene or object, by choosing views that maximize the information presented. This is an unsurprisingly



(a) (b)

Figure 2.7: (a) Measurement of visible mesh saliency from all angles of a model. (b) Top: view maximizing visible saliency. Bottom: view maximizing visible curvature. Regular views and saliency views are shown.

small research area, given the human visual perception and cognition is still not well understood, resulting in a lack of quantifiable rules that can be implemented by a camera control system. Relevant research towards this problem is often in the area of computer vision, such as Weinshall and Werman's valuable work[25] on view likelihood and stability. Their measure of view likelihood may be used to identify typical or characteristic views, while view stability can be used to identify robust or generic views. While concentrating mostly on concave polyhedra and aimed at object recognition, their work draws the important conclusion that the most likely and stable view of an object is the "flattest" view. This work is separate from camera placement issues, but provides metrics for measuring how useful a particular view is.

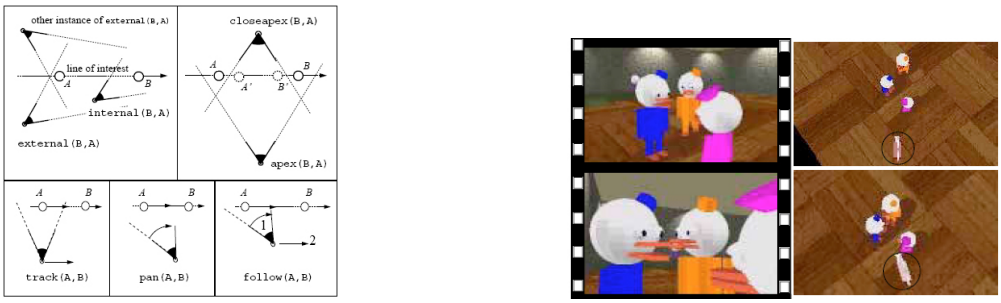
Computer graphics approaches to the issue generally take the approach of determining the optimal view by maximizing some measure of "informativeness". The advantage of such an approach over computer vision–inspired approaches are that, while rigorous perceptual modeling is abandoned, it can more robustly deal with complex object silhou-

ettes and surfaces. A recent approach inspired by perception–based metrics is the work of Lee et al.[63] on mesh saliency. The authors use a scale–dependent method to identify "visually interesting" regions on a mesh in a manner that is robust to surface texture. One application of mesh saliency is in viewpoint selection, where an informative view is defined to be one with maximum saliency of visible features. Using an optimization approach, the system can automatically find views that seem more informative than views selected based on simpler mesh properties such as mean curvature (Figure 2.7).

Automated, Compositional Approaches

Automated approaches which aim for compositional or aesthetic presentation are more common, given that there is a large body of compositional rules for cinematographers, explicitly guiding camera placement and movement in many different situations. Katz's work[52] is an excellent an example of this. These rules, which vary from general to very situation–specific, lend themselves quite well to an encoding that camera control systems can utilize. Furthermore, many camera systems, especially those for interactive systems, can have "meta–knowledge" of the scene: accessing information like the identity of objects such as characters or props, character intentions, and events such as conversations.

One of the foundational works in this area is CamDroid by Drucker and Zeltzer[24]. The authors present the issue of automated camera control as a constraint satisfaction problem, with varying constraints that express both the general and situation–specific rules of cinematography. The constraints are generated by the user in a visual programming language in the form of interconnected "camera modules". Given the constraints their system, in real time, finds an "optimal" solution, making it more robust to variations in the scene, such as character placement. This system is dependent on knowledge of events, such as which character in a conversation is speaking, in order to select the appropriate set of constraints to apply. A minimal amount of user interaction is optional, which can include choosing a specific camera to look through or tracking a particular ob-



(a) (b)

Figure 2.8: (a) Camera modules from He et al.[36]. (b) Camera and overhead views of an automatically–filmed 3–person conversation.

ject. While the networks of camera modules are specifically constructed for two different types of scenes (a two–character conversation and a virtual football game), the system performs well given the large amount of variation within those scenes.

Similar work by He et al.[36] uses a "virtual cinematographer" to automate camera control. Their system also uses camera modules, which dictate camera placement given a high–level direction; for instance, one camera module will place the camera over the shoulder of one character while viewing another (Figure 2.8). The modules are applied by user–written "idioms", which are sets of instructions in a simple programming language that describe which modules to use, given certain conditions. In turn, idioms can also be related by transitions or hierarchically to enable the system to fluidly transition between different situations; for example, a three–person conversation to a two–person one. The system can also identify occlusion between characters, and addresses it by slightly modifying character placement for different shots. The system only generates static camera positions, however, and its user is only demonstrated on a real–time "conversation" application where only character objects are considered.

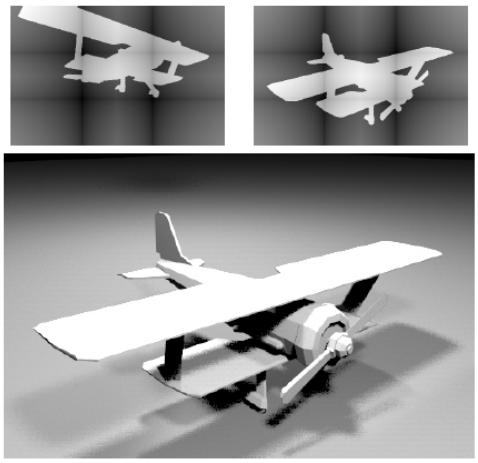


Figure 2.9: Automated composition[28] evaluations and final views. Note the proximity of silhouette edges and corners to grid lines and intersections.

A related approach by Liu et al.[66] applies the idea of encoded rules and camera modules to real–time camera management of a lecture room. Though they approach the problem inspired by live event broadcasting instead of cinematic presentation, the system uses a similar set of guidelines to choose shots based on the situation. In addition, since the system is fully–automated for live, real–life events, the authors cannot rely on meta–knowledge such as specific person locations or the current focus of the presentation. Instead, the situation is classified based on passive information such as image motion (to track the lecturer) and sound levels to locate speaking participants.

There have also been rule-based approaches based on artistic rather than cinematic principles, in order to produce an aesthetically pleasing (rather than informative) view. Gooch et al.[28] present a system which evaluates the final 2D image from the camera based on the "rules of thirds and fifths". These rules, which are actual heuristics used by artists, divide the image into a grid of either three or five divisions to a side and attempt to coincide linear image elements with the gridlines and key features at gridline

intersections. The authors devise an optimization scheme which places the camera to maximize how much of the object's silhouette is close to a grid line (Figure 2.9).

Hybrid Approaches

There has also been work which combines multiple approaches into one coherent application. The aforementioned UniCam system[110] included two partially automated techniques, distinct from their interactive techniques for direct camera control. Using their "click–to–focus" technique, a user can select a point on the surface of an object, after which the camera transitions to a heuristically–defined "oblique view" of the point. In addition, with the "region zoom" technique a user can indicate a region with a bounding sphere that the camera should zoom in on.

McDermott et al.[70] present a system where the user can indicate key shots through interactive camera placement. This interactive placement can also make use of tools to preview shot transitions or to assist in frame composition using a rule–of–thirds grid. After these shots are specified, the system computes transitions between them, creating camera motion or cutting to a new view depending on the relationship between the shots as well as cinematic rules. This system acts as an *assistant* to the user in their task to generate a cinematic scene, rather than leaving camera placement either completely automated or manual.

Finally, the recent HoverCam system by Khan et al.[53] allows simple, direct controls of the camera but integrates a variety of features to assist a user's exploratory viewing of objects. The system maintains a roughly constant distance from the viewed object but maintains a smooth camera path when object cavities, protrusions or sharp edges are encountered (Figure 2.10). In addition, the system incorporates a "look–ahead" feature which prevents the camera from making sharp or jarring turns.

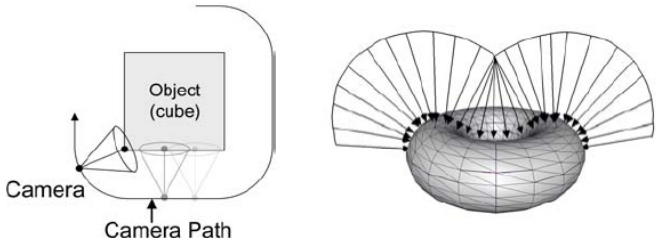


Figure 2.10: HoverCam[53] smooth motion around a cube (left) and over a hole in a torus (right).

2.4 Object Selection and Manipulation

In any system involving animation or manipulation or objects, the method of object selection is of the utmost importance. Typical 2D applications, whether they are word processors, operating systems, or drawing systems, use a simple 2D point cursor which will select the topmost object underneath its activation point. Though there has been research advancing the field of 2D selection past this common approach, we instead focus particularly on techniques appropriate for 3D perspectives and objects because of our 3D animation task. In addition to selection, we also examine on 3D object manipulation. Generally, especially in virtual environments research,"manipulation" refers to rigid transformations; that is, translation and rotation. Scaling (both uniform and non–uniform) is also a common operation in 3D creation systems, and it has been explored in some techniques. Techniques for other potential 3D operations, such as shearing, twisting, or bending, are less common. We examine 3D selection and manipulation techniques using traditional 2D input, as well as higher degree–of–freedom input such as gestural interaction.

Selection and Manipulation with 2D Input

Using 2D input to perform 3D manipulations – or any operation of dimensionality higher than 2 – obviously presents an inherent limitation. The unavoidable approach is to restrict the manipulation to lower–dimensional operations which can be performed consecutively. Houde[39] presents techniques for manipulating 3D objects in a room planning application. This work focused on manipulating virtual replications of real–world objects, which allows the manner of interaction with the objects to dictate the nature of the manipulation. For example, grasping a lamp model by the base indicates rotation, while grasping it by the body indicates translation. Furniture can be directly grasped to translate it along the ground plane, automatically constraining it to two dimensions. Further manipulations are possible through handles which appear on an object's bounding box. A handle on top of the box affords and allows lifting (vertical translation), and handles along the side of the box allow it to be rotated around the vertical axis. These techniques suit the room–planning purpose well, but require special–case preparation depending on the particular object and do not allow freeform manipulation.

A more general approach is presented by Conner et al.[19]. They present a simple widget visualizing an object's local axes with handles, any of which can be selected. Selecting a handle enables translation, rotation, or scaling with that axis, depending on which of three mouse buttons was used. Translation and scaling have straightforward direct manipulations for an axis, but the authors present an elegant method for determining a rotation axis by examining in which direction the user pulls the rotation handle. More complex manipulations are also possible using simple handles, such as twisting, bending, or tapering.

These approaches have been improved upon to allow more varied and precise 3D operations in conventional 3D software, such as Alias' Maya or Autodesk 3D Studio MAX. In these systems, separate modes are selected for one of translation, rotation, or scaling (figure 2.11). Widgets specialized for each operation can be manipulated

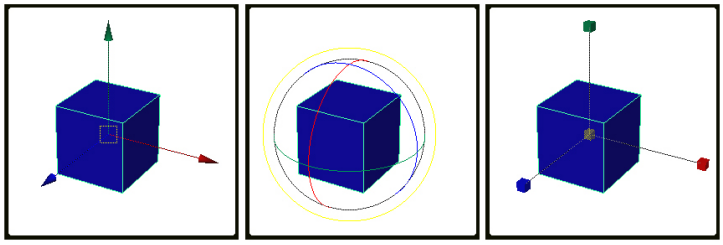


Figure 2.11: *M*aya widgets for (L-R) translation, rotation, and scaling.

similar to Conner et al.'s approach [19], however their operation can be slightly changed with the addition of a keypress. For example, holding a specific key while selecting a translation handle enables translation in the plane normal to the selected axis, as opposed to the axis itself. The frame of reference for the operations can also be selected, so that transformations can apply relative to the world, object, or any arbitrary axes.

Selection and Manipulation with High DOF Input

There is a larger body of work in this area which utilizes higher degree–of–freedom input, usually in the form of one– or two–handed gestural input. Bowman and Hodges[12] present a survey and evluation of their contemporary techniques for selection and manipulation in virtual environments. Because of the immersive nature of these systems, most of these techniques are based on the notion of actually grabbing and manipulating these objects with the hand, like a real–world object. The Go-Go technique[86] provides a non–linear mapping of the hand's distance from the body to virtual arm length, stretching it to allow grasping of distant objects. Bowman and Hodges also present the HOMER technique, which uses raycasting to select an object at any distance. These techniques usually implement direct manipulation of the objects once they are selected, in order to feel consistent with real–world manipulation.

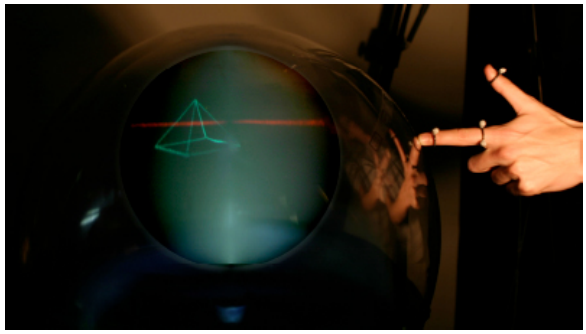


Figure 2.12: Constraining object movement to an arbitrary axis using a volumetric display[31].

While arm–extension or raycasting approaches focus on changing the user's "avatar" or methods of interaction, some techniques instead focus on presenting alternative views or representations of the virtual world to allow for easier manipulation. The Worlds In Miniature (WIM) by Stoakley et al.[96] presents a miniature, manipulable version of the scene in the user's field of view. This miniature view is coupled to the correct view, so that changes in one occur simultaneously in the other (Figure 2.5). The Voodoo Dolls technique[81] provides a similar "handheld" representation, however it instead uses selection to provide an up–close version of the manipulated object in particular. Both of these approaches allow simultaneous up–close and regular views of distant objects, unlike the arm extension techniques which do not modify the view at all. Initially the comparative advantages of arm–extension and miniature representations was unclear, however a study has shown Voodoo Dolls to be more effective than HOMER at precise manipulations of distant objects[80]. We note, however, that task completion and learning time are not compared, which would seem to be a drawback of the Voodoo Doll reliance on abstract "manipulation context" creation.

These techniques for selection and manipulation in virtual environments arise mainly

from the difficulty in changing views – since the display represents the perspective of the user, distortions or changes could prove disorienting. This effect is not present in true 3D displays, however. Grossman et al.[31] present a series of techniques for interacting with a volumetric display system, including object selection and manipulation. They note that without a fixed viewpoint, traditional techniques which rely on selecting "visible" objects become less appropriate since, for example, a ray may intersect multiple objects. They present an augmented raycasting technique where intersected objects can be cycled between using a simple gesture. Object transformations are accomplished with specific hand postures and gestures which directly manipulate the object. To overcome the sometimes imprecise nature of unrestricted 3D input, hey also present a technique where arbitrary axes can be created that restrict transformations to a lower dimensionality (Figure 2.12).

We note the majority of research using high degree–of–freedom input also utilizes some form of 3D display as well. However, Pierece et al.[79] present a set of object selection techniques that, while performed by a user in a virtual environment, rely on 2D visualization. Their Image Plane Interaction Techniques exploit the ultimately 2D user view of a scene by focusing on the hand positions in image space. For instance, their "Sticky Finger" technique selects the object whose image appears under the image of the user's finger. While these techniques are effectively performing raycasting originating at the user's viewpoint, as opposed to their hand position, the intuitiveness and immediacy of these techniques is appealing.

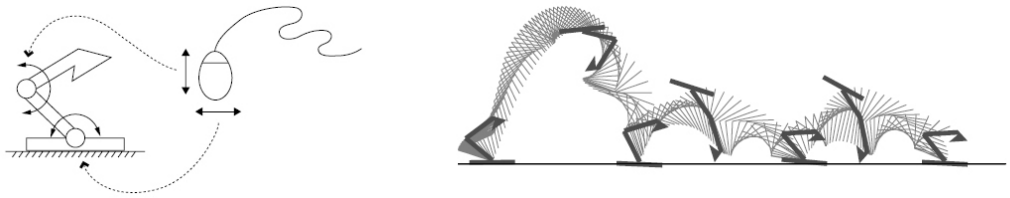
We also note that some manipulation techniques distort user input to achieve greater functionality, rather than providing an absolute mapping or a straightforward metaphor. Similar to philosophy of the Go-Go technique's non–linear arm stretching[86], Poupyrev et al.[87] present non–isomorphic rotation techniques. Expanding on the wealth of research on non–isomorphic translation techniques, they aim to provide an approach which overcomes the inherent difficulties of direct rotation: namely, the limitations of human

joints often require repeated partial rotations, or "ratcheting". Their technique, which linearly amplifies the rotation of their input, proved as accurate as the direct approach but significantly faster, an unsurprising result considering the reduction in the amount of movement required. Interestingly, their test users preferred the non–isomorphic technique over the more real–world accurate direct method.

2.5 Animation Creation/Control

There is a extremely large and varied body of research related to computation animation. We do not examine work in *animation generation*, which we define as processes that create full animation given little user input, or in a fully automated way. Instead, we choose to focus on and draw inspiration from approaches to animation creation and control which we define as *accessible* – that is, methods which are easily learned and used, expressive, and interactive. In addition, we focus more on techniques which allow for quick creation of animation, rather than more involved methods which may provide a finer granularity of control.

Physical simulations are often used to emulate real–life situations in order to produce realistic motion more easily and with less user input. An interactive approach by Popović et al.[85] allows the user to tailor rigid body simulations to have specific events or outcomes – in essence, physical keyframes. Though their approach is limited to simulations simple enough to be carried out in real time or faster, their system uses simple interaction techniques and an elegant motion visualization which allows the user to quickly craft realistic motions while concentrating only on specific events. Some of the authors applied a similar approach to later work[84] where 6 DOF tracked physical proxies are used instead of an interactive system. The user manipulates the object in real–time along a path approximating the desired physical motion, including ballistic arcs and collisions. Their system then generates a realistic–looking animation which emulates the style of the



(a) (b)

Figure 2.13: (a) Interactive control of a physical character[61]. (b)Sample motion generated interactively.

proxy motion from the user, including specific events such as a pencil landing in a cup after a series of bounces. Again, the authors provide an intuitive control for generating realistic motion, though this time their system serializes input and computation rather than a fully interactive approach.

Another physically–based approach is the idea of controlling physical characters in real–time. Laszlo et al.[61] present a system which user mouse motion and keyboard input to affect the DOF of 2D characters. Instead of a direct mapping, however, the input drives a PD controller which generates appropriate force in the character's joints. Using this direct control, which can be operated modally to direct the character for different purposes, a practiced user can execute precise maneuvers in a plain setting or when interacting with an environment (Figure 2.13). However, cinematic–type planning with a focus on particular poses or timings can be difficult. An extension by Zhao and van de Panne[113] provides more abstract character control, in a performance sports setting. They introduce a hybrid control using a gamepad which allows the user to directly control character DOF as well as trigger low–level events such as crouching. In addition, they provide a visualization aid and artificial forces to assist in character

balance. They also present "action palettes", which provide a variety of actions that the user can set the parameters of. The simulation can be re-run and actions precisely timed for maximum effect. Their system strikes an effective balance between expressivity and usability, however the actions are situation–specific and must be developed in advance.

There have also been a variety of approaches which utilize motion capture data to bridge the gap between user input and full character performance. Kovar et al.'s work on Motion Graphs[55] builds a graph out of motion capture data, where the nodes of the graph represent short motion clips. Nodes are connected when their respective motions can be transitioned between with little error. After intelligently pruning the graph, any traversal along the graph generates a smooth sequence of motion different from the captured data. A user can specify a path for the character to move along, and also select motion styles for the character based on the clips – for example, the graph can be reduced to generating motions from the "sneaky walk" capture data. Though the character motion is ultimately limited by the captured data, the quick and high–level control of their system is effective. A related approach by Grochow et al.[29] uses captured motion data to create an augmented inverse kinematics system. Based on the motion data provided to the system, it can produce the most likely pose of a character based on a set of constraints – for example, the hand and feet positions. After precomputation, the system can operate in real–time, allowing the user to specify key frames with minimal effort but maintaining expressivity. Depending on the training data, their system will produce different poses, which are not necessarily interpolations of the input motion. This system can also be used to reproduce poses from photos effectively, or to create complex animations by only animating a few parts of the character.

Some work also combines information from motion capture with physical models in order to "fill in the gaps" for the user as much as possible. Liu and Popović[65] present a system where users need only specify a few keyframes in a physical motion such as hopscotch. Their system uses knowledge of jumping or stepping motions from a set

of training data, in addition to an optimization procedure which produces animations that obey physical laws of momentum and biomechanics. The system also intelligently determines positional constraints for body parts, to prevent visual errors such as foot sliding. The user can also correct the system's interpreted positions for greater control, allowing the system a variable degree of control based on how detailed an unique an animation the user would like to produce.

In addition to providing simple but expressive techniques for manipulating a character spatially, some approaches also deal with more precisely controlling the temporal aspects of a character's motion. Doncheva et al.[23] present a system where a character is animated progressively, in layers. First, a user sketches out rough motion for a character using a 6 DOF tracked object. To select another feature to animate, the user mimics its current motion during playback of the in–progress animation. A second playback of the animation allows the user to provide new motion for the selected feature. An intriguing component of the feature selection is that multiple character DOFs can be selected – for example, multiple legs or even the correlated motion of the shoulder and elbow joints. Character features can also be modified by a similar mimicry technique, however the effects of their change are distributed spatially as well as temporally. For example, adjusting the style of how one particular leg of a spider walks will adjust all of the legs, whenever their walking motion occurs. This process of iterative refinement and simple modification allows a user to produce animations quickly and easily refine desired aspects. In comparison, Terra and Metoyer[101] concentrate solely on the temporal aspect of animation, in particular how to specify desired keyframe timing. They choose timing because a novice animator usually encounters difficulty not in imagining animation timing, but with conveying it using conventional interfaces. Using an object whose path has already been determined with keyframes, the user sketches out a rough facsimile of the path, timing their sketch as they would like the motion to be timed. Their system generates a correspondence between the the path and the sketch, and the

keyframes along the path receive the timing of the corresponding sketch point. Though only applicable to translation animations, their technique is also extensible by timing the animation of inverse kinematics handles or surface points, in order to control object rotation or scaling.

Other approaches emphasize more precise spatial control over the character's poses. Davis et al.[22] present a system where full 3D character pose can be quickly defined from 2D drawings. Using drawings of the character skeleton at keyframes, their system generates a 3D skeleton and candidate corresponding poses. Because of the inherent ambiguity in adding another dimension, their system eliminates many possible but incorrect poses with joint angle and temporal coherence restrictions. Finally, the final pose may require user adjustment since foreshortening results in two possibilities for each joint angle – their system ranks these possible adjustments by a heuristic to maximize the user's efficiency at this correction. In addition, their system accounts for constraints such as foot placement as well as cartoon squash–and–stretch to enhance the final animation. Using this system a user can easily specify key poses in a simple way, and with a minimal amount of adjustment can arrive at pleasing, fully 3D character motion.

Recent work by Igarashi et al.[44] relies on character poses, but instead concentrates on expressive user control of the interpolation between them. Their spatial keyframing system allows each of any number of user–created character poses to be associated with a marker position in 3D or 2D space. Moving a cursor in space between these key points smoothly blends the character between the associated poses, in an expressive way. Users can then craft an animation purely by moving the cursor through space, with pauses, speed changes and repetitions that allow for a simple and direct way to control the character's behaviour (Figure 2.14). Their system can also be used to generate inverse kinematics results that rely on user poses instead of the often awkward–looking solved positions, and model interaction with the ground plane can be used to generate motions such as walking which move the character through space.

Figure 2.14: (Top) Spatial keyframes[44] and associated marker positions (in pink). (Bottom) Resulting poses when interactively moving the control marker.

Finally, there have been approaches to animation that leverage the user's ability to convey motion through rough sketches and paths. Thorne et al.[102] present a system where users can create a humanoid character of any appearance through sketching. The motion of the character is then also controlled by sketching one of a variety of different types of "motion sketches". The sketches consist of paths with different features: sharp climbs or drops, loops, arcs, and so on (Figure 2.15. Each type of motion sketch corresponds to a different type of motion such as skipping or jumping, and particular aspects of the sketch control parameters of the motion, such as a jump's height. These sketches can be strung together or repeated, and the character follows along with the user's sketched instructions. Though their system is somewhat special–case and requires such parameterized animations, it can be very easily learned and provides the user with a method of quickly producing expressive animation, while still allowing a large degree of control.

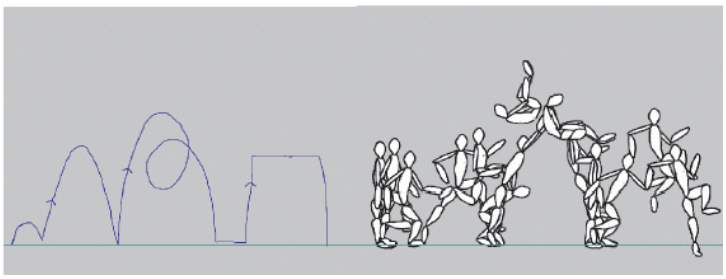


Figure 2.15: (Left) Sketched motion doodle[102]. (Right) Generated animation.

2.6 Facial Animation

A large and well-established area within computer animation concerns the animation of faces – primarily, human faces. While skilled artists can create amazingly lifelike *models* of human faces, whether from their imagination or recreations of a real person, animating those faces is incredibly difficult. The "uncanny valley" effect[71] states that near–lifelike representations of humans can be not only unconvincing, but in fact strange or even disturbing, if not given sufficiently lifelike movement. Given the complexities of facial articulation, producing involving facial animation without meticulous animator effort remains an open research area. Haber and Tertzopoulos[34] provide a thorough overview of the area, including data–driven approaches, physical models, and an in–depth history of previous works. In our work we provide a preliminary exploration into facial animation (Section 4.5) by applying our design principles (Chapter 3). We focus on an interactive approach to facial animation, leveraging the user's familiarity with faces and expressions by providing a direct manipulation interface. Therefore, we examine only related work which explores a similar direct manipulation approach.

Influential work by Pighin et al.[82] addresses the issue of extracting facial geometry, expressions, and textures from photographs. While their model creation process is well

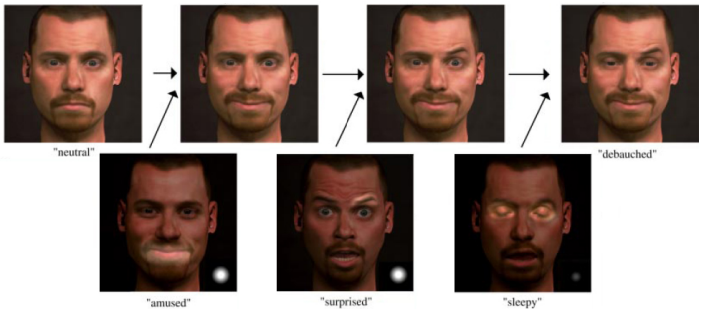


Figure 2.16: An interactive "expression painting" session[82]

executed, of particular interest is their method of interactive expression control. Based on a set of faces showing a particular expression, the user can "paint" parts of these expressions onto an initially neutral face. This allows individual control at a user–determined granularity, of not only the area of influence but the weight of the painted expression as well. This allows parts of different expressions to be mixed together at the user's discretion, which can result in intriguingly communicative expressions being synthesized from simple or seemingly unrelated ones (Figure 2.16).

Another data–driven approach by Zhang et al.[112] creates "spacetime faces" from continually–captured 3D faces, computed using structured light projections and multiple video cameras. They present a "FaceIK" interactive system where a neutral face can be directly manipulated by the user, pushing and pulling different parts of the face. The resulting deformed face is intelligently segmented in real–time and appropriate data is looked up for each segment, which are blended together to provide a convincing result. Again, this feature gives the user fine control by not blending in the entirety of a captured expression, but only near the manipulated area of the face. Controls can also be "pinned", allowing multiple constraints to exist simultaneously, again giving the user an arbitrary level of control over the resulting face (Figure **??**).

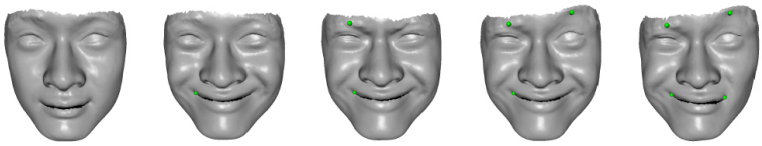


Figure 2.17: A sample "FaceIK" session[112] with a series of user–specified constraints.

Joshi et al.[49] present another data–driven approach, however theirs is much more general since it relies on a sparse set of expression models provided to the system. These expression models, or "blendshape targets", are a common approach to controlling facial animation in conventional 3D animation software. Traditionally, the faces can be blended together by animator control, however the locality of the effect is determined entirely by the model rather than user control. The authors present an automated system which segments the model based on the blendshape targets, allowing only specific areas of the face to be affected by blends. They also present an interactive system for directly manipulating the face, where again multiple constraints can be pinned to craft an expression of arbitrary complexity. We are particularly inspired by this approach, which augments faces prepared for a production animation setting, and we adopt a similar philosophy.

Chapter 3

Design Principles/Goals

We have identified a set of design principles which guide our work towards accessibility and expressiveness. These principles are drawn from well–established Human–Computer Interaction literature, common qualities of effective previous research, and the shortcomings of current computer graphics applications. We examine previous implementations of these principles along with their particular relevance for our work.

Simple Things Done Quickly

In addition to mediating the complexity of operations, we wish to focus particularly on the time taken to perform actions. While complex tasks may task a correspondingly longer time, which is allowable to a point due to their hopefully rare occurrence, we wish to minimize the time taken for simple or frequently performed tasks. We note that in many cases this is separate from task complexity or difficulty. For example, consider a user who would like to create a column for an architectural model in a typical 3D application. The user must go through a large number of steps: finding and selecting the "create cylinder" command, selecting the move tool and moving the cylinder, selecting the rotate tool and orienting the cylinder, and finally selecting the scale tool to give the cylinder the correct proportions. Often these steps require additional work to ensure 3D

correctness; for example, since objects can only be translated in two dimensions at a time, frequent camera movement is needed to gain depth perception of the scene. We note that this scenario is laborious but not necessarily difficult *per se* – a well–designed and clear GUI could make the frequent tool selection straightforward, even for a novice user.

However, Guimbretière et al[33] have concluded that significant time benefits can be gained by merging command selection and direct manipulation in a single gesture or interaction. Continuing the previous example, time is wasted by repeatedly selecting new tools for manipulation, as well as mentally switching task between manipulating the object and searching for the next tool. This consistent time wastage could be significantly mediated by more fluid tool selection mechanism. This combined approach has been used successfully in, for example, the Teddy[43] modeling system. In this system, interaction consists only of simple drawn strokes. Standalone closed strokes create an object, while other operations are context–dependent, such as an open stroke over an object cutting it along the shape of the stroke. While this system is limited in capability, the fluid merging of command selection and manipulation feels quite natural and operations can be performed very quickly.

There are other general approaches to save time that we can adopt. Bimanual interfaces have been shown to save time and reduce errors[51], though Hinckley et al.[38] caution that hands should work in concert instead of independently to maximize efficiency. Mode errors, where a user misclassifies the current situation resulting in undesired system action, can also be reduced by introducing kinesthetic feedback as shown by Sellen et al.[92]. In addition to frustrating the user, mode errors also introduce a significant time wastage since the action must be repeated, in addition to possibly undoing the misunderstood action. We are inspired by these approaches to minimize the potential for user error in our system, as well as eliminating slowing actions such as command selection.

Direct Manipulation

In our effort to create a system as immediately accessible and expressive as possible, we are inspired by research on direct manipulation interfaces. Shneiderman[93] discusses benefits of such systems, including that "novices can learn basic functionality quickly" and that "users gain confidence and mastery because...they feel in control". With these desirable benefits in mind, we adopt his three principles of direct manipulation:

1. Continuous representation of the objects and actions of interest;
2. Physical actions or presses of labeled buttons instead of complex syntax;
3. Rapid incremental reversible operations whose effect on the object of interest is immediately visible.

However, we recognize that even systems founded on direct manipulation can be confusing for users, by involving widgets or other GUI components that are too complicated. Fu and and Gray[98] address this issue by identifying the difference between directly manipulating *interface objects* and *task objects*. Furthermore, the point out that "Direct manipulation works by providing an interface that is *transparent* to the user...[who] do not feel that they are using the device, but are directly accomplishing a task".

We draw upon these observations and principles towards creating an accessible and expressive system. Animation and modeling in particular can have powerful yet simple metaphors of physically manipulating actual objects. We wish to leverage the lifetime experience that users have precisely manipulating objects for a variety of purposes consistent with an animation system: precision positioning, rotation, movement, and so on. Using the principles of direct manipulation, users should be able to as easily utilize the system for purposes which have no straightforward real–world analogue.

Group–Appropriate Techniques

We recognize that most animation productions, like any film production, are the result of a group effort. Even though animators usually do not directly collaborate on the exact same content – a character in a particular shot is is usually animated by a

single person – the final product is the combination of many people's effort. This option is, in effect, "free" in most animation software: different animations or other files can be merged easily. While we do not preclude that functionality from our system, we wish to facilitate more directly collaborative work, akin to a group brainstorming session using a shared workspace. Like in brainstorming sessions, however, we do not approach the problem of users simultaneously manipulating the system; for the time being we leave this issue within the field of Computer–Supported Cooperative Work. Instead, we wish to support an asymmetric group structure: a single participant actively using the system, who can collaborate, discuss or demonstrate with a group or audience. We recognize that animation is often created for a group purpose but its creation and discussion occur separately; we wish to make this entire process more involving.

We can accomplish this with a variety of approaches. A large or distributed display system system allows all participants to understand the current state of the system. A simple and uncluttered interface prevents distraction not only on the part of the user, but the audience as well. We can focus on techniques that lighten the cognitive load on the user, in order to facilitate simultaneous discussion and system usage. We note that techniques performed using kinesthetic feedback instead of visual selection (such as marking menus[59] and its descendants, once mastered by an "expert") can accomplish this, and we seek to enable this capability with as little user experience as possible. We can draw on previous work on "invisible technology"[20], which shares a principle of minimizing distraction. Related work on distraction, particularly concerning in-vehicle interfaces[90], keeps us mindful of the effect of task time and cognitive effort for a multitasking user. Complimentary work also shows how to direct group attention on a large display[53], an essential component of such a collaborative effort.

In addition to reducing cognitive load to promote discussion and group interaction, we also wish to facilitate rapid switching of the primary role between users. We can reduce the learning time of the audience by making the techniques of the primary user

explicit and visible, inspired by approaches on teaching interaction by demonstration[3]. Thus, through participation in a group session, an audience member should be able to understand how to operate the system without being explicitly instructed. Provided that users can observe both the interaction and its effects on the system simultaneously, this "teaching by observation" could prove very beneficial without requiring the audience to actively concentrate on it.

Focus on 2D Appearance

In a system designed to be accessible and expressive, it is important to consider the end goal of the user's interaction. In our case, though the user is manipulating and orchestrating three–dimensional objects and motions, the end effect of this process is a two-dimensional animation. Therefore, we seek to facilitate interaction which results in the desired 2D product, regardless of 3D correctness. Cinematically, this is not an unfounded notion – filmmaking has a long history of using "tricks" such as rear–projection screens, forced perspective, or matte paintings to achieve the appearance of something impractical or altogether impossible. Animated productions frequently use analogous techniques, including frequent composition of independently–generated 2D components. Computer graphics research has also developed new techniques with this idea in mind. Fur and grass can be rendered using camera–aligned "fins" for realism[64] or "graftals" for a non–photorealistic look[56]; though neither approach attempts to create an accurate result, merely an accurate–looking one. Camera perspectives can be warped for an impossible "psychorealistic" effect[18], and view–dependent geometry[88] can be used to create three–dimensionally inconsistent models based on camera position that nevertheless "look" correct.

Furthermore, we also wish to exploit our users' experience in 2D viewing not only as an observer, but as an active participant as well. Though we do not necessarily restrict ourselves to 2D input, it is logical to present users with a 2D visualization of their

work, similar or identical to the viewpoint of the intended audience. As a result of this 2D presentation we can, where necessary, leverage user experience in 2D applications to increase the accessibility of our system. We are inspired by Pierce et al.[79], who present a variety of image plane interaction techniques for use in virtual environments. Their informal tests indicate that users can easily switch between 2D and 3D interaction, and in many situations the users expect 2D–type interactions, even in immersive virtual environments. Similarly, Grossman et al.[31] present gestural techniques for interacting with true 3D objects on a volumetric display, but augment their system with a 2D "surface menu" to provide easier and more intuitive access to commands.

Therefore, in seeking interaction which allows the user to feel as if they are interacting with their final product, 2D input techniques or metaphors may prove may prove as useful as full 3D input. Furthermore, we are inspired by previous approaches to make this as transparent as possible – ideally, the user should not be concerned with or aware of three–dimensional consistency at all.

Exploit High DOF input

Since we are attempting to provide more accessibility and expresiveness in our system than conventional animation processes or software, it is beneficial to examine why in particular the status quo approach often proves so difficult. Common procedure in animation systems, and indeed in computer systems in general, divides manipulation into almost atomic components: for example, translation and rotation must be performed separately in conventional 3D interaction. Arguably, this is due to the discretized, sequential application of operations by the computer system. However, for human understanding this is not necessarily the case. Jacob et al.[45] argue that many tasks are cognitively integrated; for example, focusing on a particular area of an image is conceptually a single operation, rather than two distinct operations of panning and zooming.

An obvious barrier towards this sort of parallel control is input bandwidth – there seems to be a distinctly limited amount of control possible with 2D mouse input, hence the sequential control for higher degree–of–freedom (DOF) tasks. We note that there have elegant techniques to overcome this limitation: for example, speed–dependent zooming[40] combines panning and zooming in an effective and intuitive way, while the Rotate 'N Translate technique[58] allows simultaneous rotation and translation control of 2D objects, using simple 2D input. However, we also note the fine scale of these techniques; developing enough techniques to allow for free–form animation is a prohibitively large problem beyond the scope of this work, though it presents an intriguing challenge.

Instead, we can adopt more unconventional input methods or devices, drawing upon a large body of previous work concerning the effective use of high DOF input (see Section 2.2 for an overview). There are often surprisingly intuitive approaches to combined operations using such input. For example, Jacob et al.[45] present a simple technique using 3D input to intuitively control 2D panning (through movement parallel to the display) and zooming (movement normal to the display) in a combined fashion. Ware et al.[107] present a set of camera–control metaphors using a 6 DOF device, each of which was found to be immediately usable for particular view control tasks. As well, as noted in Section 2.2, input DOF can also be increased by introducing bimanual control. Hinckley et al.[38] argue that not only are two hands often faster because they allow parallel operation, but using both hands changes the way that users think about a task. For complex but cognitively singular tasks, using two hands is often easier than re–thinking the task in terms of lower–dimensional operations.

Therefore, high DOF and/or bimanual input seems a logical application for our system, given our aim of accessibility. However, we note that poor design of such an interface could actually prove an impediment; Hinckley et al.[38] caution against the use of hands for parallel independent tasks, while Balakrishnan and Hinckley[4] note that visually separate bimanual operations similarly divide attention, and result in sequential operation.

To craft an effective high DOF system, we aim to incorporate this type of advice in addition to elements of successful previous approaches.

Chapter 4

Animation Systems

To explore the issue of animation creation, we have thoroughly examined related previous work in a variety of areas. Through this examination, we have formulated a set of design principles to guide us in our development of a creative system that is both accessible and expressive. We began by implementing a "bare bones" system to test and refine our basic visualization and interaction techniques. While successful, it soon became clear that creating a "universal" animation system, suitable for any animation purposes, was too large as a first step prototype. We then examined a broad range of animation tasks, in order to find applications that would benefit most from our design principles and general approach.

Creating *animatics*, or previsualization animations, is a task that matches particularly well with our design principles. The animatic focus on rough animation is ideal for our principle of *Simple Things Done Quickly*, while our emphasis on accessibility allows the director to take an active role. Our *Focus on 2D Appearance* is very appropriate for such a task where cinematic communication is more important than fine details. Furthermore, our development of *Group–Appropriate Techniques* will further the involve the group of cinematic planners responsible for animatic creation.

We also explore the application of our approach to another animation task: facial

animation. Again, we find the particulars of this area a good match with our design principles. For nontechnical users, a *Direct Manipulation* approach can leverage their familiarity with human facial expressions and movements, without having them abstracted to complex animation controls. We can *Exploit High DOF Input* to more easily interact with complex facial controls. Simple facial animation is usually difficult due to the complex controls, so a focus on *Simple Things Done Quickly* would be advantageous. Furthermore, utilizing a large display allows the face model to be seen in appropriate detail, as well as permitting the simultaneous visualization of many possible expressions, a difficult task on the conventional desktop monitor.

We shall first explore the general setup of our system, including its relevant input and display devices. As well, we explain our basic interaction techniques and methodology behind the design of our more specialized interactions. Following that, we describe in detail the animatic creation system, DirectCam, including its components of object creation and manipulation, camera control, and animation. Finally, we present a preliminary extension of our DirectCam system and design principles into another animation domain: facial animation.

4.1 System Setup

To prototype our techniques, we track 3D hand positions using a Vicon optical motion tracking system. While this system is rather costly, we anticipate low-cost tracking to become available in the near future as robust vision algorithms migrate to low-cost camera products. Users wear a pair of gloves augmented with reflective markers. From this the optical motion tracking system provides us with the position and orientation of the hand, as well as relative positions of all hand markers. Other systems for optical tracking [31] use augmented rings, which are less cumbersome. We find gloves to be non–restrictive for our application, as we use whole hand motion for precise control rather than finger

Figure 4.1: A group of users collaborate using *DirectCam*, our animatic creation system.

motion. This also allows users to quickly swap control by exchanging the gloves. We use from six to eight motion tracking cameras to produce a robust reconstruction.

This system has been designed for use in front of a large display. It has successfully been used with a 50–inch plasma screen, a single projector projecting against a screen, and a rear–projected high–resolution array of 12 projectors. The specific screen size is rarely an issue as hand motion can be easily scaled such that there is a near correspondence to perceived screen space motion.

4.2 Postures and Gestures

Users of our system must be able to perform a variety of different actions while creating and refining an animatic. One of the earliest design issues encountered while developing the prototype was how to implement command selection. Part of our design principle of ease of interaction involves leveraging our users' previous experience with computer applications and the techniques involved. We note two common methods in everyday

computer applications. First, a "tool–action" approach involves the user selecting a particular tool before applying its functionality. This allows repeated use, such as the Erase tool in a painting program, since the associated tool remains active. A "target–command" approach presents the opposite – the targets for a command are selected, then the tool selected to apply to them. A common example of this approach are the "Cut" or "Copy" clipboard functions in a word processor. This avoid potential confusion about the current tool, but is correspondingly difficult for actions which must be repeated.

(a) 3D cursor control. (b) Selection. (c) Camera frame control.

Figure 4.2: Main Hand Postures

It is possible to implement either method verbatim in our system, in order to capitalize on users' familiarity with these approaches. However, both approaches have benefits desirable for our system – we wish to avoid user confusion about the current mode or tool, but we also would like users to be able to easily repeat actions, such as refining the position of a 3D object. Similar to previous work involving gestural interaction, we adopt a system of *postures*, or poses, that indicate the user's desired task (Figure 4.2). A user must actively maintain the pose, providing kinesthetic feedback as to which task is currently active. In addition, we design these poses to be intuitively related to their associated tasks. For example, selecting virtual objects for manipulation utilizes a pinching posture, immediately recognizable as a metaphor for grasping an actual object. We adopt a more abstract posture for through–the–lens camera control which is not used in everyday life. However, this posture is familiar to many people as a posture commonly used by filmmakers or photographers to approximate a camera perspective and "frame" shots.

We note that because our system is designed for use with large displays and in a group setting, pointing is a conceivable action by both the user and the audience in order to augment discussion. Pointing is a common posture used in systems with gestural interaction – see Vogel and Balakrishnan[106] for a survey of uses and approaches. Because of its likely use separate from our system, however, we are careful to omit special meaning of the pointing posture. In fact, we have deliberately placed markers only on the gloves' thumbs and index fingers, so that our system cannot recognize postures in which the fingers are posed unequally. This allows us to concentrate on maintaining a small but effective set of simple postures. As a result, a pointing posture will be recognized by the system as a regular "cursor manipulation" posture (see Section 4.3) and will allow the user to control the system cursor, which is visible but does not affect the 3D scene. Because normal pointing can be imprecise between distant people, we find this functionality to be more useful since the cursor can be precisely and unambiguously positioned anywhere on the screen to direct attention, akin to a laser pointer.

Once in a given posture, subsequent hand motion, which we will define generally as *gestures*, dictates the action taken by the system. We emphasize that in most cases our gestures map directly and continuously to manipulations of the scene, rather than being pre–specified movements which initiate specific actions. We are wary of the example of sketch–based systems such as SKETCH[111] and GEdit[60] which maintain a large vocabulary of gestures that, while effective once learned, provide a significant learning curve for the user. While discoverable interfaces can address this issue (Vogel and Balakrishnan[105] for an effective discoverable system with hand gesture interaction), we also wish to avoid the "dead time" resulting from such a feature, since our system is designed to also have an audience present. By avoiding such approaches, our system achieves a feeling of direct manipulation that is immediately accessible. In addition, we keep the system's posture and gesture vocabulary small by reusing many gestures, and "mirroring" postures between hands.

(a) Hierarchy construction

(b) Rotation

(c) Scale manipulation

(d) External Camera

Figure 4.3: Object and Camera Gestures

4.3 System Interaction

In our system users perform three primary creative tasks: creating and manipulating models, manipulating the camera, and creating rough animation for both models and the camera. To accomplish these tasks, we use three main postures: cursor navigation, selection, and through the lens camera manipulation. We will first discuss the interaction issues underpinning our entire system.

Cursors and Hands

To interact with our system a user must be wearing both gloves and standing in front of the screen, at any distance. The precise volume in which interaction can occur is determined by the particular motion capture camera configuration. In the basic type of interaction with the system, cursor manipulation, the system only considers input from a hand when it is above a predetermined vertical threshold. This prevents hands at rest (for example, at the user's side or on their hips) from generating input. Additionally, regardless of height, the system also disregards hands that are at a limp or downward angle. This allows the user to "clutch", or deactivate, the cursor when desired without lowering the hands. We believe this is essential since even though we attempt to tune the parameters of our system to be sensitive to user input, there are extreme tasks (for example, rotating an object by 360 degrees or more) which would require an unusually high system sensitivity to achieve in one motion, due to the joint limits of the human body. Since we wish to maximize the sense of directly manipulating the scene, and

these situations are rare, we believe the the clutch mechanism to be an acceptable trade-off. Further, this may be a more intuitive approach since users are accustomed to this "ratcheting" action in real life situations, such as using a screwdriver. While we found that the proper input sensitivity almost eliminated clutching during cursor manipulation, it was found useful in extreme cases of object and camera manipulation.

While manipulating a cursor, hand movement in the plane of the display maps directly to cursor movement within the display, though the input is scaled by a constant factor so that the perceived, projected motion of the hand and cursor are roughly equal. Given the typical distance at which users comfortably raise their hands, this allows them to navigate the entire display with ease. For simplicity our system does not track user position, and thus cannot compensate for a user with an extreme viewing angle towards the screen. This is not a concern, however, since users in the intended environment for this system will be using a large display, meaning that they will almost always be in front of the display at a moderate distance.

An issue that quickly became apparent when developing this system was how to precisely define the roles of each of the user's hands. An early version of the system assigned each hand an independent, identically–controlled cursor. This was found to be extremely difficult to use in tasks where both cursors needed to coordinate, due to the asynchronous nature of how, when and where each hand might become active. This led to a disjointedness between the motor spaces of the two hands, such that two hands beside each other could result in two very separate cursors, and vice–versa.

Turning to the relevant literature, a common approach in bimanual systems is to apply Guiard's Kinematic Chain theory[32]. This encompasses the notion of an abstract chain of motors, each of which sets the frame of reference for its dependents – for example, the movement of the elbow determines the frame of reference, or broad activity area, of the wrist and subsequent joints. Guiard's model hypothesizes that in an asymmetric bimanual task, where hands perform complementarily, the hands form a functional kine-

matic chain, where the nondominant hand determines the frame of reference for the work of the dominant hand. For example, when writing, most people manipulate the paper with their nondominant hand while writing with their dominant hand. Systems such as the Responsive Workbench[21] have been based upon this model, with successful results.

However, this approach poses some conflicts for our work. Continuously involving the nondominant hand for large–scale positioning is feasible on flat surfaces or using physical input devices, but in our situation the user's hands are held freely. Requiring the use of both hands for tasks which can be performed unimanually would prove unnecessarily fatiguing. In addition, we wish to keep our posture and gesture vocabulary as straightforward as possible to make the system immediately accessible. For example, the Responsive Workbench[21] provides a powerful array of tools, however they encompass a variety of input types (unimanual, bimanual symmetric and bimanual aymmetric) and each tool is controlled by different gestures. For simplicity and accessibility, we focus on unimanual techniques where possible, using a second hand as a modifier only when required. We also note that users can apply postures and gestures from their dominant to using their nondominant unimanually as well. This could allow us to maintain a smaller posture and gesture vocabulary, "mirroring" them for different meaning depending on the hand, as long as the tasks for the nondominant hand are not as dependent on precision.

This methodology, then, allows us to resolve the issue of bi–cursor frame of reference. Given that interactions will primarily occur using the dominant hand, we use its position as the frame of reference for the nondominant hand. Hinckley et al.[38] note the advantage of users understanding the spatial relationship of their hands – this approach means that two hands brought close together results in the intuitive result of proximal cursors as well. When the dominant hand is inactive, we use a standard frame of reference for the nondominant hand, "snapping" it back to its dependent role if the dominant hand is activated. This clearly and concisely indicates the change of reference frame, while a smooth transition might prove more confusing since the cursor would appear to be

moving independently. Furthermore, if the user has activated the dominant hand their attention is focused on it, and the nondominant hand will act as a modifier only when dominant hand action is initiated. We note, however, that this dependent approach means it is more difficult for the nondominant cursor to reach areas of the screen far from the dominant cursor, and techniques must be designed accordingly.

4.4 Animatic Creation System

To more precisely develop and refine our design principles and relevant techniques, we decided to examine the filmmaking process in order to target our system at a particular issue. In film production, the exploration of cinematic composition begins with hand-drawn storyboards. Storyboard artists, in collaboration with the director, develop the visual communication of story that is only suggested in a written script. Camera views and movements are explored and annotated with arrows, borders of subsequent frames, and spatial transition curves [52]. Storyboards are then edited into a *story reel*(Figure 4.4), along with narrative, rough dialog, and temporary sound and music. These reels better convey the dynamic qualities of composition and timing and provide a medium for further refinement.

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

Figure 4.4: Story reel frames from *Ryan*. Images (a) and (b) represent a continuous shot, while images (c)–(e) are subsequent shots.

In computer animation, and increasingly in live-action, story reels are recreated as 3D *animatics* (also known as layout sequences and previsualizations) [76]. These animatics

provide the realistic spatial cues necessary to fully refine a cinematic composition. This composition is built on the staging, timing, and positioning of cameras, characters, and objects. As creative exploration is paramount and many variations will be explored, animation remains in a rough form and 3D models are often constructed as temporary stand-ins.

This process is ideal for our system for a variety of reasons. First, the purpose of animatics is to convey staging, movement and camera composition, rather than precise positioning or animation. By focusing on our design principles, particularly *Simple Things Done Quickly* and *Focus on 2D Appearance*, we can hopefully facilitate quick animatic creation and its continual refinement for maximum cinematic effect. Secondly, our goal of accessibility should allow the director to interact directly with the animatic creation process. Currently, animatic development, like most aspects of a film's visual development, occurs in an iterative process where artists independently produce rough stylistic variations of their subject. The director views these possibilities and guides the artist in their next iteration by pointing out which aspects are desirable or to be avoided – this process continues until the director is satisfied. Our system can be used to directly involve the director in the visual development process, rather than the status quo of attempting to communicate visual ideas verbally. Finally, through our principle of *Group–Appropriate Techniques*, we can exploit the now–common event of group presentations of ongoing animation work, or *dailies*. These sessions allow a large crew to offer input on the broad range of work occurring in a production. In a similar vein, we aim to shorten and enrich the animatic creation process by allowing group discussion and collaboration.

We have called our animatic creation system *DirectCam*, since it focuses on direct manipulation as well as acting as a tool for directors to visually communicate. In addition to the general techniques described earlier in "System Interaction" (section 4.3, we have developed new techniques which aid, but are not necessarily specific to, animatic

creation. We identify the general problem of issuing commands, as well as the three fundamental operations in our system: object creation and manipulation, camera control, and animation.

Shelves

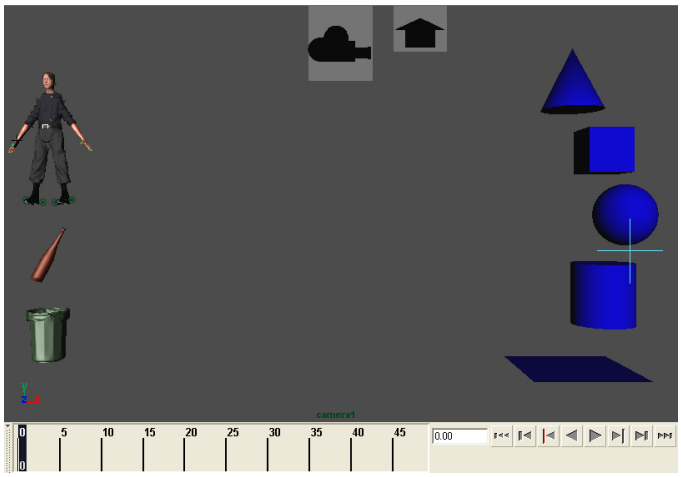


Figure 4.5: Clockwise from left: Model Shelf, Shot Shelf, Primitive Shelf (with cursor), Timeline. Note that in normal operation, only one shelf is visible at a time.

Once focused on the task an animatic creation, an immediate and fundamental issue in our system was how to issue commands. Maximizing the use of direct manipulation while minimizing the number of postures and gestures presented a problem, particularly with object creation where the user should have a large number of choices available. A menu system was a possibility, however conventional linear, nested menus posed a significant awkwardness problem with gestural input, despite the user's likely familiarity with them from desktop computer applications. While there is a wealth of research on menu alternatives, we wanted a system that was immediately accessible and consistent with our direct manipulation approach. A simple but effective solution was our addition

of *shelves* (Figure 4.5), linear arrangements of icons placed along an edge of the display which smoothly slide into place when a cursor approaches. A *model shelf* appears on the left, which provides access to a library of pre–made production models. A *primitive shelf* appears on the right to allow creation of standard 3D primitives shapes. Finally, a *shot shelf* appears at the top of the screen to control shot–specific assets such as the camera, story reel panels and the set. The shelves are only active when desired, leaving the display uncluttered in accordance with our principle of *Focus on 2D Appearance*. Furthermore, the icons on the shelves are not buttons; to extend our approach of direct manipulation, the primitive and model shelves represent repositories of objects that the user can pull into the scene, while the shot shelf presents a camera handle that can be "grasped". The shot and model shelves are discussed in "Model Interaction" (Section 4.4.1), and the shot shelf is discussed in "Camera Control" (Section 4.4.2).

4.4.1 Model Interaction

Object Creation

While we anticipate scenarios for our system where the 3D scene and the objects therein are already prepared, we wish to support a large range of object creation options. Users can create scene geometry in two ways: as references to external files or as entirely new hierarchies of primitive shapes. We use model file references to resolve data for production models and the set, as these are typically in development and changing on a day–to–day basis in a production environment. Primitive hierarchies are stored locally with the animatic, as they are only relevant for exploring cinematic composition and will eventually be replaced by production models. When a user navigates to any shelf, the models they can reference and primitives they can create are brought into view as icons. Pinching the fingers of the dominant hand into the selection gesture while the cursor is over an icon creates a model reference or new primitive as appropriate. This new object is then tied to the cursor for manipulation, and the user's hand in already

in the selection/manipulation posture. In addition, as soon as an icon has been selected, the relevant shelf smoothly slides back off–screen.

One difficulty presented by this approach was precisely how to create the new models, both in terms of their scale and their distance from the camera. For primitive objects, we have manually preset scale values to ensure that the primitives are consistently sized – in our case, we scale each primitive so that each is roughly the size of a unit cube. Production models have a preset scale from their creation which we maintain, though the user can manually change it after creation. Upon creation, in keeping with our principle of *Focus on 2D Appearance*, we place the new object at a distance from the camera such that its projected image precisely matches the size and position of the selected icon. While this results in objects being created at different distances, the consistency between the icon and object feels very natural. This fluid transition from creation to manipulation creates a sense of actually pulling an object from a shelf for immediate use, avoiding the disconnect between shape creation and manipulation present in conventional 3D software.

Object Manipulation

Any objects can be manipulated by changing the dominant hand to the pinching selection posture while the cursor is over the object. Additionally, as objects are created from the user shelf fluidly enters the manipulation posture as described earlier. In both cases, our manipulation techniques remain the same. During object manipulation we directly map its translation from map changes in hand position. Hand motion in the plane (left/right and up/down) of the display is mapped to object translation in the camera's frame of reference, rather than the world axes, which his ensures that the object remains under the hand's cursor. This 2D feel is accessible to users familiar with manipulating objects on a conventional 2D desktop application, and exemplifies our principles of both *Focus of 2D Appearance* and *Direct Manipulation.* For motion normal to the display (forwards/backwards), we translate the object directly away from

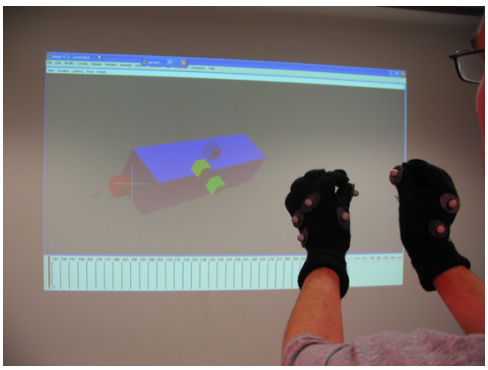


Figure 4.6: Controlling nonuniform scale with the nondominant hand.

or towards the camera instead of along its forward vector. Though this results in a different translation vector depending on the object's screen–space coordinates, this also adheres to our design principles by ensuring that the object's projection remains directly under the cursor.

Object rotation is directly mapped from hand rotation, again occurring in the camera's frame of reference. This follows the metaphor of directly grabbing, positioning, and orienting a real world object. We also wish to enable scaling, another typically–used affine transformation, in our system. Doing so presented a problem since we wished to maintain the direct manipulation metaphor, but in the selection posture the user's hand is reduced to six degrees of freedom, all of which are already used. To maintain a fluid feel for our system, we introduce the use of the nondominant hand for scale control (Figure 4.6). While manipulating an object, raising the nondominant hand and bringing it close to the dominant hand reveals scaling widgets – one for each local axis of the object and one at its center. This is similar to the scaling widgets used in conventional 3D animation software. The user can then select any widget with the nondominant hand's cursor and the selection posture. For the axis–aligned widgets, subsequent hand motion along the image–space projection of the axis non–uniformly scales the object along that axis. To uniformly scale the object, the user can select the central widget, which does

not have an intuitive direction to control it. Instead, we adopt the convention of left or downwards motion indicating a decrease, while motion to the right or upwards increases the scale. While scale is being adjusted, input from the dominant hand is ignored; the object remains locked in position and orientation. Upon release of the nondominant hand selection, the object only returns to a manipulation context if the user has maintained the selection posture with their dominant hand.

This addition of the nondominant hand to object scaling is in keeping with our direct manipulation approach. Real–life objects can be positioned and oriented with one hand, as in our system. However, activities similar to scaling, such as stretching or compressing a deformable object, require two hands. This correspondence between real–life actions and in–system manipulations is easily understood by users, in keeping with our design principles of *Direct Manipulation*, as well as *Simple Things Done Quickly*.

Hierarchies

Our presented techniques allow the user to both create rough representational objects and import more precise production models. However, we recognize that there are situations where a user may wish to use more detailed that have been developed from scratch, in order to explore possibilities beyond those provided by the production models. Instead of introducing a separate set of more detailed object creation tools, we enable the user to construct object hierarchies. When manipulating a primary object with the dominant hand, the user can raise the nondominant hand to control the second cursor. Pinching and releasing the nondominant hand over a secondary object creates a hierarchical link between the two objects, with the secondary as parent of the primary. This allows a more complex object to be quickly assembled from simpler pieces in a building block fashion. We maintain a typical hierarchical structure – parent transformations affect child objects but not vice–versa – which allows for individual parts of a hierarchy to be animated. When manipulating an object, the user can sever hierarchical links of the parent or child

variety by again momentarily selecting them using the nondominant hand.

To avoid visual clutter we avoid visualization of hierarchies, whose structure is usually simple as well as evident from object placement. In addition to the user-created objects, imported production models can also contain pre–set hierarchies. Typically, though, production hierarchies are more complex than is necessary for animatic production, so our system supports a tagging mechanism to restrict which parts of a production model hierarchy are selectable, usually to only a few objects. Such tagging can be done by a technical character developer specifically for animatic production, similarly to how controls are designed specifically for animators.

4.4.2 Camera Control

In keeping with our design principle of focusing on the final 2D appearance of the work, we model our interaction with cameras on the workflow of live action directors. During filming, directors are more concerned with the composition and effect of camera movement, rather than the specifics of how or where the camera actually moves. Shots are planned by arranging motion through the set to capture desired spaces while repeatedly looking through the lens to maintain appropriate composition. Inspired by this, we allow users to navigate locally with a through–the–lens perspective, as well as globally, using an external point of view.

Through–the–Lens Camera Control

By default, the system displays the perspective of the *scene camera* – that is, the camera from which the scene is rendered, and through which the audience will view the scene. To manipulate the scene camera, users position their hands into a framing posture (Figure 4.2c). Horizontal and vertical translation of this posture is then mapped to the pan (left/right rotation) and tilt (up/down rotation) controls of the camera. It is important to note that we map hand translation to corresponding camera motion,

instead of the apparent scene motion. For example, moving the frame posture upwards tilts the camera upwards, which results in the image of the scene moving downwards. At first, this seems to contradict our goal of directly manipulating the scene wherever possible. However, by directly manipulating the camera in this manner instead of the camera view, we remind the user that they are viewing the scene through the camera, as the audience would. In addition, this arrangement ensures that the scene area visible to the user through their framing posture changes, as it would on a real life set.

The framing posture control allows the user to precisely line up a shot, but only from a stationary position – as if the camera were mounted on a tripod. To control the camera position, the user enters the cursor manipulation posture with their dominant hand and moves the cursor to the top of the screen to reveal the camera shelf. The user can then select the "camera handle" icon by pinching it. While the selection posture is maintained, we map hand motion directly to camera motion in the camera's frame of reference. For example, grasping the camera handle and moving the hand forwards will translate the camera directly ahead, regardless of the camera's worldspace orientation. We support clutching directly in this case, so for long camera moves a user does not have to release the camera handle and return to cursor manipulation in order to reposition their hand.

We have deliberately restricted the degrees of freedom of the scene camera and its associated control techniques, in order to maintain a set of controls which are very quickly comprehended. We restrict the camera rotation to only two degrees of freedom – pan and tilt – by enforcing that the camera's up direction matches the scene's. This removes control of the camera's roll (rotation around its forward vector) and ensures that the "horizon" of the scene remains level. An early version of the system used strictly rotation around the camera's local axes, which resulted in a visible lack of control over camera roll, despite the fact that the manipulation of the camera was completely direct. This, combined with the fact that non–level camera angles are rarely used in live or animated films, led to removal of roll control. The camera's aperture, or field of view, was also

controllable by the distance between the hands while in the framing posture. This feature was also removed because of the resulting visible jitter, due to noise from both the motion capture system and slight hand movements, and its rare utility.

Additionally, we have restricted the framing posture to two gestures: horizontal and vertical translation, which control only the camera's rotation. Even treating the posture as completely rigid, it still has six degrees of freedom, of which we only use two. An early version of the system allowed the camera to truck (move directly forwards/backwards) by translating the posture directly towards or away from the screen. This had an unintended effect since users panning or tilting the camera will not move their hands completely parallel to the screen, resulting in slight camera movement when it wasn't desired. In addition, only partially decoupling control of the camera's rotation and translation was found to be confusing. This feature was removed as well.

External Camera Control

We recognize, however, that there are situations where a user may wish to view the scene from a distant perspective, or to more precisely understand the placement of the scene camera in relation to the scene itself. To facilitate this, we allow the user to enter an *external view* by making a fast "pull back" gesture directly away from the display, while controlling the scene camera in the framing posture (Figure 4.3d). We then create an animated transition to a new external view behind the scene camera, which becomes visible in this new view. The external view is controlled with the framing posture and by selecting its own camera handle, in a manner identical to the scene camera. To move the perspective back into the scene camera, the user makes a corresponding "push in" gesture, which results in a similar animated transition.

While this new perspective can be valuable to the user for comprehension purposes, we also allow the user to directly manipulate the scene camera. In accordance with our design principle of ease of interaction, we seek to avoid user confusion by leveraging their

pre-existing experience manipulating objects with our system. For that reason, the user can select the scene camera with the dominant hand cursor, as if it were a regular scene object, and the manipulation gestures (hand translation and rotation) remain the same. As before, translation of the scene camera occurs in the external view's frame frame of reference. Additionally, we continue to restrict the scene camera rotation to only pan and tilt control, which we map directly from the wrists twisting and up/down rotation, respectively. This control is slightly different than rotation control for regular objects, which are rotated within the view's frame of reference. However, we found this control to be more intuitive because it gives the sense that the user's hand has "become" the scene camera. In keeping with our design principle of *Focus on 2D Appearance*, we disable selection of any non–camera object, reasoning that the scene should only be modified when it directly affects the view of the scene camera.

An intended use of the external view is to allow the user to easily aim the scene camera at objects that it cannot currently see. In order to accommodate this, we wish to main user awareness of the scene camera's current perspective when in the external view. One possible solution was to add a second view to the display, which would concurrently display the view through the scene camera as it is directly manipulated. However, this would require a user to constantly switch attention between the external and internal views. In addition, we felt that the user would concentrate too much on fine placement of the scene camera, which is better accomplished using the through–the–lens tools. Another possibility was to visibly represent the camera's view through additional geometry. A translucent polygonal representation of the camera's view frustum was found to be too large and distracting, and put uneven emphasis on the borders of the frame since their intersection with scene geometry was clearly visible. An effective alternative, however, was to add a translucent "beam" to the scene camera model which extends directly forward from it. This allows the user to quickly comprehend which part of the scene lies at or near the center of the scene camera's view by seeing which objects the

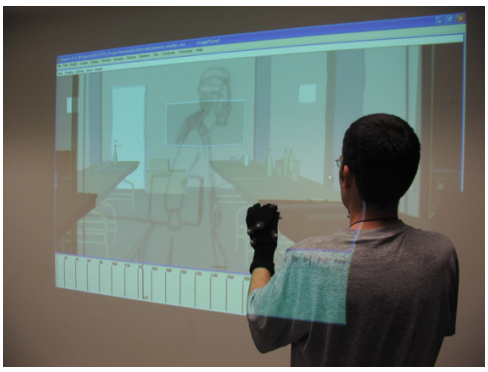


Figure 4.7: A story reel frame is overlaid on the main camera view of the scene.

beam intersects. In addition, this more clearly defines the role of the external view as a quick method to make broad camera changes, while the through–the–lens tools are more appropriate for precise refinement.

Story Reel Integration

Because our system is designed to integrate into the traditional pre–production pipeline, we wish to allow the user in–system access to supplemental materials. A user may import story reel footage, which consists of a sequence of storyboard panels. This is displayed as as a semi-transparent plane, constrained to the camera view frustum, but at a slight distance in front of the camera so that objects of primary importance, such as characters, are not obscured as all (Figure 4.7). We also maintain the story reel timing by displaying the appropriate frame for its designated frames when the user is manipulating the current frame or playing back the scene (see "Time Scrubbing", below). In addition, the story reel frames are only visible through the scene camera, and not the external view.

4.4.3 Animation

Our techniques for Object Manipulation and Camera Control are suitable for quickly and easily creating rough scenes, in accordance with our design principles. However, these scenes remain static – they contain no animation. Our Animation techniques seek to add this control, but in keeping our goal of accessibility we wish to integrate these as smoothly as possible. To that end, we introduce only one new posture (see "Animation Retiming", below) and most gestures are extremely similar to those used previously.

One important distinction, though, is that all time–centric tasks are accomplished using primarily the non-dominant hand instead. We recognize two fundamentally different aspects of this work, and animation systems in general: spatial control and temporal control. We avoid using an "animation mode" which modifies the meaning of spatial manipulation, as many 3D animation packages do, because this can lead to mode confusion and unintended actions. Instead, we again rely on kinesthetic feedback by completely differentiating the roles of the two hands, leaving no doubt for the user as to whether they are making temporal or spatial modifications. From a practical standpoint, this is easily integrated into the system because our animation–related actions require significantly less precision, and are thus appropriate for the nondominant hand. Additionally, in an abstract sense, this approach is consistent with the Kinematic Chain model of bimanual tasks[32], where the nondominant hand is used to determine the context of interaction for the dominant hand. While previous approaches use this to determine spatial context, we instead use it to determine a temporal context. This is also consistent with our decision to maintain a small set of postures and gestures, many of which are shared between hands.

In keeping with another principle, *Simple Things Done Quickly*, we specifically avoid control for creating refined motion. Producing refined, believable motion is not the purpose of our work since it is not essential to the animatic process; additionally, our intended users may not have any previous animation experience. Our approach instead

allows and motivates the user to quickly and creatively explore the cinematic composition of the scene over time. We emphasize a pose–to–pose approach to encourage conformity with the story reel and generate simple animation which is more easily evaluated for its cinematic effectiveness.

(a) Keyframe creation (b) Time scrubbing (c) Key retiming

Figure 4.8: Animation Gestures

Time Scrubbing

At the bottom of the display is the animation timeline, which indicates the the entire timeframe of the animation as well as the frame number currently being displayed. Currently the duration of the animation is determined by the length of the story reel, though it could be changed to any desired value by the normal Maya means - we found this so unlikely an occasion that adding functionality for it would have been needlessly complicating. The timeline can be "grasped" by the nondominant hand exactly as the dominant hand can select and then manipulate an object. Due to limitations of our implementation the cursor cannot be displayed over the timeline, therefore we recognize when the hand is being directed toward the screen and is below a preset vertical threshold. This process, while approximate, has performed well in practice. Furthermore, we allow the user to select the timeline anywhere, not only near or at the current frame.

The timeline can be "scrubbed" interactively while the selection posture is maintained by gesturing to the left or right, which moves the current frame earlier or later respectively. We stop the current frame at the beginning and end of the scene's timeframe instead of looping around, to maintain the metaphor of an actual slider. In addition,

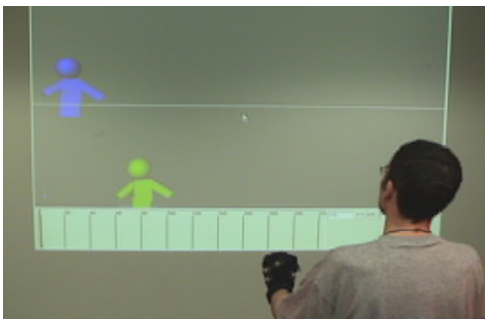


Figure 4.9: A user prepares to set a key by releasing the object's temporal icon on the timeline.

because we wish this technique to produce both large–scale (such as "fast-forwarding" quickly to the end of a scene) and small–scale (such as selecting a particular frame) changes, we scale the sensitivity of the function proportionally with the speed of the gesture. Thus, a fast flick can be used to navigate to the general area of the desired frame, and the gesture can fluidly transition into a slower movement for precise control.

Keyframe Animation

Since our system emphasizes coarse pose–to–pose animation as dictated by the story reel, our techniques integrate into the standard Maya keyframe animation system. This provides sufficient control for our purposes while retaining a relatively simple and easy–to–learn motion model. To set a keyframe for a particular object, the user advances to the appropriate frame and manipulates the the object into its desired configuration. The user then selects the object with the nondominant hand's cursor, and a semi–transparent *temporal icon* of the object in its current pose is created underneath the cursor. To create the key, the user then "drops" the icon onto the timeline by moving it to the bottom of the screen and releasing the selection posture (Figure 4.9). This action can be also be more easily and quickly performed as a downwards throw gesture.

Keys can be redefined by repeating this process. In intermediate frames, the Maya

"flat" interpolation is used to position objects between their keys in a smooth manner that doesn't suffer from the unintuitive overshooting problems that can occur in some types of interpolating curves. The user is free to re–position an object from its interpolated state at a frame in between two keys, and can add a new key if desired. If the user decides not to use the new pose, changing the current frame snaps the model back to its interpolated state, indicating that the potential change was discarded.

Retiming

One common difficulty users have with the keyframing paradigm is with revising already–keyed motion. While experienced animators are comfortable editing motion curves and meticulously adjusting keyframe timings, we seek a quick, coarser method in keeping with our design principle of *Simple Things Done Quickly.* To accomplish this, we enable our users to retime their animations in a variety of ways. First, an object's keys can be scaled uniformly to control its speed. When the user is manipulating an object's temporal icon, tick marks are shown along the timeline to mark the object's keyframes. By bringing the dominant hand up and pinching it, the user can scale the keys up or down from their origin from moving the dominant hand right or left respectively. The object's key ticks are visibly stretched or compressed. Similarly, pinching the dominant hand and translating both hands in the same direction translates all of the keys equally forward or backward in time. These animation scaling and translation tasks can be performed concurrently as well by translating both hands in different directions and/or by different amounts.

These techniques lack the precise direct manipulation feel of the time scrubbing, where the user actually reaches *towards* the timeline. In this case, the user is still manipulating a part of the timeline, but their hands could be in virtually any position. However, while using these techniques the user doesn't need to see their hands at all - kinesthetic feedback is enough, combined with direct observation of the keys along the time. We have

found that despite its slight indirection, using these techniques "feels" like the timeline is being directly manipulated by one's hands – either stretched, compressed, or moved. We also note the reversed usage of the hands: in this case the nondominant hand plays a primary role and the dominant hand is used to modify its function, while this relationship is reversed in object manipulation (i.e. spatial) tasks. We have found this separation more clearly distinguishes spatial and temporal manipulation, while the reversal is easy to understand and allows us to maintain an uncomplicated set of postures and gestures.

Users can also non–uniformly scale keys in time, in order to change the rhythm of a movement and to modify the spaces between keys. When manipulating an object's temporal icon, a user can tap the fingers of the nondominant hand against the palm of the dominant hand, with one tap for each of the object's preexisting keys. Retiming can be aborted at any time prior to completion by releasing the nondominant hand's selection posture. The retiming occurs as follows:

Our object has n keys at frame numbers $f_1, \ldots, f_n$. There are also n taps, which occur at times $t_1, \ldots, t_n$.
Let $f'_1, \ldots, f'_n$ be the new frames for the n keys. Then for $1 \leq i \leq n$,

$$f'_i = f_1 + \frac{t_i - t_1}{t_n - t_1} \cdot (f_n - f_1)$$

In effect, this replaces the original key sequence with the tapping sequence, which is uniformly scaled to match sequence length. We note that the order of keys remains the same, and the first and last keys in the sequence are unaffected. For intermediate keys, the above equation yields real–valued frame numbers which are rounded to the nearest integer. Because of this rounding, we must sometimes displace keys forward from their intended frame if that frame is already occupied by another key.

We have also experimented with a straightforward mapping which replaces the key times with the tap times verbatim. While such an approach is more direct, we have found the non–uniform technique is more robust to key sequence length: with a direct

mapping, quick successions of keys becomes difficult to tap out accurately, and sequences which span a long time take correspondingly long to retime. The non–uniform technique, however, focuses on the rhythm of the animation while allowing its length to be modified by the separate scaling technique. This technique is similar to an effective technique presented by Terra and Metoyer [101], although we decouple timing control from pose manipulation and use a relative mapping. We plan to more precisely investigate the effectiveness of our technique, particularly in comparison to the direct mapping – see "Future Directions" (Section 6.2) for more information.

4.5 Facial Animation System

In addition to animatic creation, we wished to further test and refine our design principles by applying them to another animation problem: facial animation. Animating faces at a production level is incredibly difficult, time–consuming and precise work. A typical difficulty with animating characters is that very fine detail is required, since the audience is intimately familiar with human movement through everyday experience. This difficulty is exacerbated when animating faces, which are often the focus of our visual attention – there is a variety of evidence that facial perception is a special process in the human brain, which regardless if fully active in young infants and may, in fact, be innate[73]. This is made even more difficult by the arcane and extremely complicated control systems for animated characters' faces.

However, it is possible that our intense familiarity with faces can, instead of serving as a barrier for effective facial animation, be leveraged as an advantage. Guided primarily by our design principle of *Direct Manipulation*, we envision an animation system where the user can craft facial expressions directly instead of manipulating cognitively distant values and sliders. Furthermore, our principle of *Focus on 2D Appearance* is applicable here, since in almost all cases facial movement can be considered two–dimensional, and

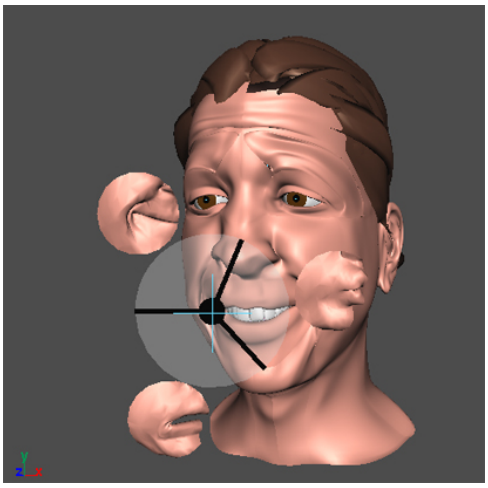


Figure 4.10: Our facial animation system, with an expression menu to allow control of mouth expression.

we can exploit techniques for both 2D interaction and visualization. Previous facial animation research has included direct manipulation components[49, 112] (see section 2.6 for a more in–depth view), but these systems usually require specialized input, or are purely interactive and unsuitable for animation purposes. As in the DirectCam system, we integrate our facial animation system (Figure 4.10) into a commercial 3D animation package to take advantage of pre–existing facial rigs, as well as maintaining a animation format that can be manually adjusted for maximum precision, using conventional techniques. In addition, we seek not to completely conceal the underlying animation system, but through our interaction and visualization techniques allow the user to understand it, and as a result manipulate it more easily and to greater effect.

Interaction Techniques

For this system, general behaviour regarding postures, gestures and cursor manip-

ulation remains unchanged from that described in "System Interaction" (section 4.3). Since animation is now the sole concern, however, we do not include our through–the–lens camera control techniques, which are more appropriate for finer control over camera movement for cinematic effect. Instead, we adopt a simple technique for orbiting the camera view around the head, so that any view can be gained while keeping the head centered and oriented correctly in the view. Pinching with the nondominant hand while it is raised, or the dominant hand while its cursor is over empty space and not the head model, allows the view to be manipulated. While the posture is maintained, horizontal hand movement controls the camera azimuth (i.e. rotation around the vertical, or orbiting side–to–side), and vertical hand movement controls camera elevation (i.e. orbiting up and down). We map hand direction to camera direction – for example, moving the hand to the user's right will rotate the camera to its relative right as well. In addition, hand movement normal to the plane of the display controls camera distance from the head in a similarly direct manner. This gives a feeling of directly manipulating the camera, similar to Ware et al.'s "camera in hand" metaphor[107]. We allow dominant hand control of the camera for situations where the user wishes to explore their view, and is solely concentrated on camera control. In this case, we allow the user control with their more familiar and precise dominant hand. Controlling the camera with the nondominant hand allows for simultaneous camera control and scene manipulation bimanually, which has been shown by Balakrishnan and Kurtenbach[5] to be both beneficial and manageable.

Because this system works on top of a pre–existing facial animation rig, we have designed it to provide accessibility to the underlying controls without overly abstracting them. Unfortunately we cannot guarantee the robustness of our system, and since facial posing is so dependent on precision, low–level control of some form is essential for error correction as well as specifying details. Thus, in this initial exploration we provide direct control of the facial expressions, or blendshapes, in the rig. When manipulating the dominant hand cursor, pinching into the selection posture over a point on the model's face

will bring up an "expression menu", similar to a control menu[83]. The circle surrounding the cursor is separated into "slices" which correspond to the expressions that affect the selected surface point. Furthermore, the angular ranges of these slices correspond roughly the "direction" of the expression. For example, selecting the left corner of the character's mouth will display a menu with three slices: smile, frown and pucker. Outside the circle at the middle of each slice are thumbnail views of the expression at its maximum setting. The thumbnails can be viewed in different sizes, each showing a different amount of the face with the corresponding expression (Figure 4.11).

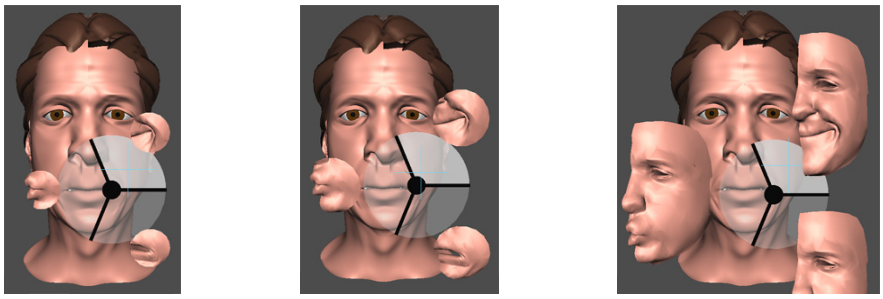


Figure 4.11: Expression menu preview sizes.

Similar to the control menu, moving the cursor outside of the circle while maintaining the selection posture selects a particular expression for manipulation. The expression is increased by cursor movement along the vector bisecting the slice, and likewise opposite for decreasing the expression. Expressions have preset maximum and minimum values that lie roughly within the range of believability. We note that this technique differs from control menus in two ways: first, we allocate different angular ranges for each expression based on its effect on the selected surface point. Secondly, unlike most marking menu–type implementations, the directions for expression selection within the menu are extremely relevant. A marking–type menu may have system commands like "Open" or "Save" scattered around the menu – however there is no natural association of these

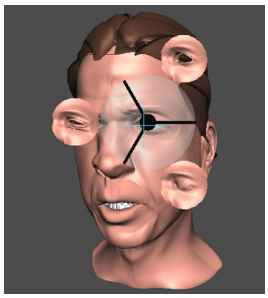


Figure 4.12: Eye controls: (clockwise from left) squint, open, close. The open and close controls affect the same underlying *Maya* value.

functions with their compass direction. Familiarity and speed comes only from repetition building a "muscle memory". With our expression menus, however, the direction of a particular option is directly related to the option itself – in this case, it represents the movement of the selected surface point during the expression. This direct relevance of the menu option direction brings it closer to direct manipulation, while maintaining fine–level control.

However, our system does compensate for the straightforward and often unintuitively unrestricted nature of blendshape control. Some expressions can work in two directions – eyelid control is often one blendshape, for example, which controls eyelid movement up *and* down since the normal value is somewhere in between. While we could assign this blendshape one arbitrary direction in the expression menu, this would lead to awkward interactions as the blendshape is selected and then hand movement is reversed to control it in the opposite direction. Instead, we create a second slice in the expression menu with an appropriate preview, whose manipulation is mapped inversely to the blendshape control (Figure 4.12). In addition, we enforce that some blendshapes are mutually exclusive – for example, furrowing and raising the eyebrows. Unintentionally combining these

expressions in a normal system would result in an odd appearance since they do not neutralize each other. Instead, we enforce that the manipulation of one blendshape while its opposite is active first descreases the active blendshape to zero before increasing the selected one.

4.6 Implementation Details

Using the motion capture system we track 24 degrees of freedom: 12 from reconstructed positions and rotations of the hands and three for each thumb and index finger position relative to the appropriate hand. Each posture is recognized by similarity to a particular degree of freedom configuration. If the pose is within some threshold, we assign a given posture. To resolve ambiguities where distance to multiple poses is within their corresponding thresholds, we prioritize the poses by how commonly they are used in practice.

These systems have been integrated into the *Maya* animation system by using its geometry API and by mapping our poses to Maya's keyframe curve representation. Maya algorithms are used for generating, manipulating and interpolating keyframes.

For the facial animation system, regions of the face have been manually annotated with their relevant expressions, and the angular ranges are hand–set for the expression menus. However, we envision a straightforward automated process that can accomplish this same task, and even interpolate values so that the menus options change continuously depending on selected surface point.

Chapter 5

User Evaluation

During the development of our DirectCam system, we performed a variety of user evaluations to gauge the accomplishment of our goals and design principles. Though these evaluations were not rigorous or quantitative, they provided valuable feedback on the system. There were three types of evaluations. First, we had a number of trial users explore the system to provide general feedback about its accessibility and expressiveness. Most of these users had strong computer skills, but were inexperienced in 3D software or filmmaking. Some of these users were asked specifically to evaluate the system, while others observed the system in use during development and were eager to try it. These users received little to no instruction in the use of the system – some opted to observe a more experienced user before trying it themselves. For our second evaluation, we invited a professional animation director with a background in character animation to use the system. While he provided general feedback on the system's techniques, he also shared his views on the usefulness of our system in an animation production pipeline. He explored the use of the system as well as received explicit instruction, in order to evaluate all aspects. For our third evaluation, we tested our system by creating an animatic shot from an actual story reel. This allowed us to test the appropriateness of system for the processes specifically involved in animatic creation, and was performed by an expert user.

We discuss the results of all evaluations, followed by their implications on the design of our system.

5.1 Results and Observations

5.1.1 Interaction Techniques

All users were able to use our basic interaction techniques easily. After some quick experimentation using both hands, all the users switched to their dominant hand for primary interaction, as we expected. Cursor manipulation was easily understood and required no instruction – due to their familiarity of cursor manipulation from regular computer applications, users quickly grasped (or perhaps correctly assumed) that the cursor was affected solely by horizontal and vertical motion. The relative frame of reference for the hands was understood quickly, and they were able to find a comfortable arm pose for interaction. Users did not use the clutch feature very often, however, and as a result raising their hand up to chest or eye level activated the cursor prematurely, because of the roughly waist–level vertical activity threshold. This resulted in cursors often beginning interaction at the top of the screen, which proved occasionally inconvenient.

Object creation was handled easily by all users – the shelves proved intuitive, and many users singled them out as one of their favourite features. When manipulating the objects, users found object translation immediately intuitive. In particular, moving objects directly towards or away from the camera instead of along a fixed vector was used without comment, presumably since it "feels" correct. Object rotation proved troublesome in some cases, where the joint limits of the user's hand impeded rotation or an odd angle mixed up the internal *Maya* rotations. The director in particular expressed an interest in alternate rotation techniques that could offer greater precision. As well, by default we rotate the object about its predefined pivot point, which usually occurs at the center of the object's bounding box. This proved unintuitive for more irregular shapes

like the cone, or when manipulating particularly long and thin objects. However, users still found the "building block" approach easy to use and thoroughly enjoyed constructing scenes and objects out of different combinations. Some expressed interest in a more robust version aimed specifically at constructing 3D scenes – we discuss this in Section 6.2, "Future Directions".

Our camera control techniques proved easy to use and control, and was by far the favourite feature of the system amongst all users. We believe that this is because of the correspondence between hand motion and virtual camera motion, which is amplified by the large display – users very much enjoyed taking the role of director in such a, for lack of a better term, direct way. Some users did comment, however, that they would also be interested in a scene–centric control metaphor where apparent scene motion corresponds to hand motion. The external view was also unanimously approved of, though some users were more enamoured with the flying transition between camera views than any actual utility of the new view. The director found the techniques appropriate for precise control, but remarked that gross control was difficult. This is true, as our constant control–display gain allows a rotation of about 90 degrees before clutching and repeating the gesture is required. He also felt that moving the camera control posture parallel to the display was less intuitive than rotating in place, which would allow him to position the camera at any angle in one gesture.

Our animation techniques were not tested as thoroughly as the rest of our system. Since many test users were unfamiliar with 3D animation and the keyframing paradigm, we didn't want to bias evaluations of our techniques based on lack of familiarity of the underlying ideas or a poor explanation of the concept. A small number of tests with the animation techniques have shown them to be adequate, and users are able to easily understand the distinction in roles of the two hands manipulating objects spatially or temporally. However, because the animation techniques directly interface with the underlying *Maya* keyframe data. We would like to develop abstractions of this mechanism,

similar to how our other techniques intuitively, rather than directly, control internal camera or object attributes. We discuss this further in Section 6.2, "Future Directions". We are optimistic about our rhythmic retiming technique, which has performed well preliminarily, however it is unintuitive and requires a more complicated explanation than any other technique in the system.

5.1.2 General Results and Observations

Some users observed that, even though their interaction was short enough to prevent discomfort, arm fatigue could be an issue after extended use. To alleviate this, the director suggested the use of a director's chair or similar furniture with high armrests. This would allow the user to rest their elbows on the armrests, drastically reducing fatigue but still allowing a comfortable and large range of motion.

Evaluation of our system occurred on both a single–projector wall–projected display as well as a ten–projector back–projected display. The single–projector setup worked well, though the user sometimes occluded the projection. The combined display functions at a higher resolution than a single projector, however users made no remark about the display resolution in either case. We believe that a low resolution display is entirely sufficient for animatic creation purposes, since gross motion and composition are the most important factors. Furthermore, utilizing a higher–resolution combined display can cause lag due to bandwidth usage, an effect that users were very sensitive to.

Though the system was not tested in the particular type of group collaborative scenario that we are targeting, all of our group–appropriate techniques proved successful. Some users were passersby who observed an expert user operating the system and wanted to try it, where they excelled without instruction. As well, our techniques seem to be intuitive and cognitively light, since all of our test users were easily able to converse with others while simultaneously operating the system.

To technically evaluate the system, we set the goal of recreating animatic quality

motion using an existing story reel from an animated film, *Ryan*, as well as production models from the film. We assume the shot has been initialized to reference appropriate assets such as reel footage and a model library, as is typical in animation production. In about three minutes, a user experienced with the system was able to load the reel, load the set and primary character, and produce rough keyframe animation for both the camera and main character. Additionally, stand–in geometry was created to layout out positions of secondary characters who move little during the shot. Frames from the reel are shown in Figure 5.1, and the corresponding frames from the created animatic are shown in Figure 5.2.

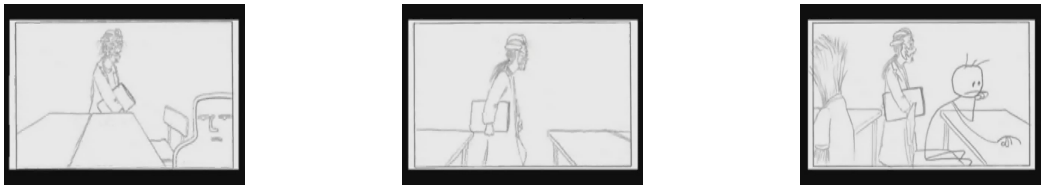


Figure 5.1: Story reel frames depicting a continuous shot from *Ryan*.

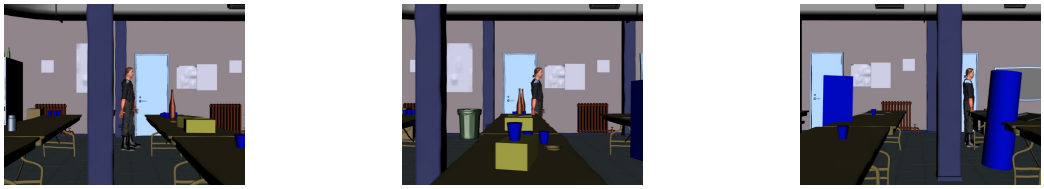


Figure 5.2: Frames from the animatic corresponding to the reel shown in Figure 5.1. Note the use of stand–in geometry to position as yet unmodeled characters.

5.2 Design Implications

Based on our preliminary user evaluations, our system techniques and design principles have been successful. Therefore, we will focus on small adjustments and refinements rather than a large restructuring of our approach. In particular, the posture and gesture

methodology proved very successful and in most cases, was not explicitly noticed by the user. This approach has proven very accessible, much more so than we expect that a more complicated and abstract gesture vocabulary would be. We intend to frame any additions to the system within this methodology, even though it will be a challenge to maintain system simplicity.

Object creation as well as object and camera manipulation proved straightforward for most users, with the exception of rotation. The rotation mapping was sometimes unintuitive, even though hand motion was directly mapped to the object, and ratcheting proved more awkward than initially thought. However, we now note that we can still maintain our principle of *Direct Manipulation* without necessarily having a direct mapping. The director suggested an alternate rotation method where the object could be mounted on a virtual spindle and spun with the hand – Cutler et al.[21] as well as Grossman et al.[31] implement a similar feature in their systems. We plan to investigate such possibilities along with more transparent non–isomorphic techniques, such as those examined by Poupyrev et al.[87].

There are a variety of settings within our system, mostly control–display gain values for the various techniques, which have been hand–tuned. In our evaluations these values worked very well, with no users commenting that the controls either felt imprecise or overly sensitive. However, we do note that these values were tuned, as well as the evaluations performed, at a relatively close distance to the display. Given free choice of their position, users unsurprisingly positioned themselves centered in front of the screen, but also close enough that the display roughly filled their field of view. We plan to investigate the effectiveness of our settings from different viewing angles and distances – a discrete assortment of settings may be necessary to maintain a smooth feeling to the interaction. Alternatively, it may also be worth investigating automated and continuous adjustment, and perhaps user–guided adjustment as well.

The director in particular brought up the point that while it makes sense that object

manipulation gestures occur parallel to the display, this is not always the case. He felt it was more intuitive for camera control gestures, in particular with the framing gesture which controls camera rotation, to be controlled by user rotation as well – or, equivalently, hand rotation about the body, relative to the user's front. This does make sense – in the framing posture with both hands rigidly together, rotation is more comfortable than planar translation. We note that given our current implementation, detecting this is very difficult – by only tracking the user's hands, our system cannot infer the user's position. For example, a user with a forward–extended hand would appear identical to a user with that hand simply raised, but standing closer to the display. However, it may be possible to implement heuristics which could recognize and interpret rotation in the framing posture.

We can also add a simple marker to the user to track their position – for instance, a velcro patch or hat (such as used by Vogel and Balakrishnan[105] to track users) could easily be traded between users and would provide the rough information we would require. Such knowledge also allows for a variety of possible extensions to our techniques. We could more precisely know distance from the display and viewing angle, and adjust control–display gain or other settings to compensate. Tracking a hat would allow us to determine if hands are held in front of the face, which we could leverage by implementing techniques similar to Pierce et al.'s image plane techniques[79]. We could also roughly track user gaze direction, which could also prove useful. This would also allow us to detect arm extension, perhaps after an initial calibration phase. This could lead to adding isometric rate control near the extent of arm reach, similar to a laptop track pad, to avoid clutching.

Chapter 6

Conclusions and Future Directions

In this thesis we have presented a novel 3D creation and animation system designed for use by a nontechnical user. Our aim was to develop a system that is not only immediately usable and comprehensible by novice users, or *accessible*, but also also able to easily craft results in tune with the user's vision, or *expressive*.

We identified design principles that would guide us toward that goal. We aim to have *Simple Things Done Quickly*, to allow for a broad range of creation quality without penalizing simple work. We adopt a *Direct Manipulation* approach to every aspect of our system, to make its use as immediate and intuitive as possible. We are mindful of *Group–Appropriate Techniques* to facilitate collaboration and discussion during creation, and also "teaching by observation". We *Focus on 2D Appearance* in order to ensure the user's awareness of the final product, as well as leveraging their familiarity with 2D interaction. Finally, we *Exploit High DOF Input* to simplify interaction and eliminate the serialized manipulation and mode selection prevalent in conventional animation software.

We developed a prototype system, DirectCam, which embodied these principles and focused on the particular animation production task of creating animatics. We used bimanual gestural input, combined with our design principles for an effective and accessible interface. Our posture and gesture designs combine with large display visualization for

group–appropriate work. We also applied our design principles to another animation area, facial animation, resulting in a different but effective interface.

We have conducted a variety of user evaluations to test the effectiveness of DirectCam. A series of trial users experimented with the system, and provided valuable observations and feedback about our techniques. An accomplished animator and director thoroughly tested the system, and gave his approval and feedback. Finally, we tested the system by creating an animatic from a story reel, as in actual production. These have so far shown that our system is both accessible to new users, and expressive enough to fit the needs of animatic production.

6.1 Contributions

Our design principles were formulated to assist in the development of accessible and expressive animation systems. Our goal was not to create a rigid step–by–step framework for system development, nor was it to present a "super–system" capable of addressing the needs of all possible users. Instead, we applied the principles to particular animation problems, utilizing display and input technology not often used for animation. In addition to our principles, grounded in well–established computer graphics and human–computer interaction research, we have also contributed original techniques and approaches, which are detailed below.

Bimanual, Gestural Interface

While there have been previous explorations of bimanual and gestural input, we integrate and expand on them. We have developed a very simple but powerful set of postures that rely on continuous direct manipulation of task objects, instead of manipulating controls or gesturally issuing commands. We have leveraged the innate human distinction between dominant and nondominant hand, allowing the user to comprehend and manipulate objects both spatially and temporally. We also present through–the–lens camera

control techniques which are immediately intuitive to users.

Group–Appropriate Techniques

The specific task of animatic creation allows us to optimize our techniques for an unusual situation: an asymmetric group where only one user operates the system, but who also collaborates with a group of active observers. Our direct manipulation techniques and uncluttered interface allow for straightforward visualization of operations, clearly communicating the user's interaction to the audience. Our gestural control vocabulary also communicates current user operation. In addition, user evaluation has shown that, even with no prior experience with 3D software, users can comfortably operate the system after very brief observation and with no explicit instruction.

Fluid, Accessible Control

Our system overcomes the many modes and commands present in modern 3D software, which is often very confusing for novice users. Our system is modeless, and all system aspects (including object creation and manipulation, camera control, and animation) can be fluidly and intuitively switched between at any time. Our direct manipulation approach allows for more comprehensible continuous and responsive control, rather than issuing discrete commands. Our through–the–lens approach focuses on the user on the end product of their creation, a final 2D animation, rather than 3D correctness. We also leverage the familiarity of our users with 2D interfaces and visualizations to increase the accessibility of our system. Finally, we also abstract keyframe animation of objects with high–level direct manipulation including a simple yet powerful rhythmic retiming technique.

6.2 Future Directions

The most important addition to this work would be more thorough user evaluation, to more definitively gauge the effectiveness of our techniques. First, we wish to pursue additional qualitative evaluation by having users perform a variety of tasks with the systems, and giving their feedback in a thorough questionnaire. We would also like to try some group tasks as well, replicating the animatic creation task, but with a group of users unfamiliar with the system, analogous to our target situation. This would allow us to test the group–appropriateness of our system collectively, instead of on a technique–by–technique basis as we have so far. We would also like to perform more quantitative evaluations of our techniques individually, as well as the entire system. Object creation and manipulation tasks can be performed in our system and in *Maya* by novices after receiving a short amount of instruction. We are inspired by the variety of object manipulation evaluations provided in the related literature, and feel that tasks could be designed which could reliably measure task completion time as well as correctness. We make no contention that *Maya* can produce the same results – our system is integrated into *Maya*, after all – and likely perform faster, in the hands of an expert user. However, choosing subjects unfamiliar with 3D animation software and providing equivalent instruction in both systems would also allow us to gauge how accessible the system is.

One difficulty with the system currently is animation. Keyframing seems an appropriate paradigm for animatic creation in particular, which emphasizes pose–to–pose motion as dictated by the key poses in the storyboards and story reel. However, we wish to explore whether animation control can be abstracted and made fluid to the degree of our other techniques, since the keyframe paradigm itself requires sufficient understanding on the part of the user in order to be used effectively. We are encouraged by our current retiming technique, which focuses on the rhythm of animation rather than the particular poses or frame numbers. Perhaps other, higher–level controls for animation can work correctly – in addition to focusing on rhythm, perhaps we could focus more particularly

on the motion or direction of the animation. Users could grab an object and animate it in real time, bridging the conceptual gap between manipulation and animation as recently pointed out by Buxton[15]. In addition, we can investigate ways to more easily visualize and edit the entire animation of an object – for example, the system could generate a motion path from initial animation that would be more easily editable. This approach seems promising, especially considering recent work in bimanual curve creation[30] and manipulation[75].

A specific choice early in our system was to minimize the amount of non–scene imagery onscreen, including menus, icons, and instructional text or diagrams. In accordance with our design principles, providing an uncluttered view facilitates a greater feeling of *Direct Manipulation* as well as a *Focus on 2D Appearance*, in addition to making the view more comprehensible to viewers (*Group–Appropriate Techniques*). However, we may be approaching an "upper bound" of sorts with our techniques, such that maintaining such a sparse design may require too complex a set of postures and/or gestures. Techniques such as self–revealing help[105] or icons integrated in a control menu[1] may prove a useful way to make more complicated functionality accessible, without adding too much to user cognitive load. Previous work on Worlds In Miniature[96] for virtual environments indicate that users can observe and manipulate "heads up display"–style objects positioned in screen space, while maintaining a sense of the overall scene. Perhaps to mediate audience distraction, we could also utilize positional information about the user and attempt to position such onscreen aids so that the user occludes the audience view of them.

Finally, we wish to explore the potential of these techniques for generic 3D scene creation. Many of our trial users thoroughly enjoyed creating and manipulating simple scenes, and expressed an interest in potential system features designed for that purpose. Our design principles seem applicable to 3D applications with different purposes, and we wish to explore their potential using a similar fully 3D bimanual interface, as well as

using traditional 2D input. Given the ubiquity of 3D acceleration in common desktop computers, we envision a complete 3D creation system accessible to all users. We are encouraged by our current progress and look forward to advancing the democratization of computer graphics.

Bibliography

[1] A. Agarawala and R. Balakrishnan. Keepin' it real: Pushing the desktop metaphor with physics, piles and the pen. In *Poceedings of CHI 2006*, 2006 (in press).

[2] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22(3):587–594, 2003.

[3] R. Baecker. Showing instead of telling. In *Proceedings of SIGDOC 2002*, pages 10–16, 2002.

[4] R. Balakrishnan and K. Hinckley. Symmetric bimanual interaction. In *Proceedings of CHI 2000*, pages 33–40, 2000.

[5] R. Balakrishnan and G. Kurtenbach. Exploring Bimanual Camera Control and Object Manipulation in 3D Graphics Interfaces. In *Proceedings of CHI 1999*.

[6] W. Bares and B. Kim. Generating Virtual Camera Compositions. In *Proceedings of IUI 2001*, pages 9–12, 2001.

[7] W. Bares, S. McDermott, C. Boudreaux, and S. Thainimit. Virtual 3D Camera Composition from Frame Constraints. In *Proceedings of Multimedia 2000*, pages 177–186, 2000.

[8] T. Baudel and M. Beaudouin-Lafon. Charade: remote control of objects using free-hand gestures. *Commun. ACM*, 36(7):28–35, 1993.

[9] A. Bezerianos and R. Balakrishnan. View and Space Management on Large Displays. *IEEE Computer Graphics and Applications*, 25(4):34–43, 2005.

[10] J. Blinn. Where Am I? What Am I Looking At? *IEEE Computer Graphics and Applications*, 8(4):76–81, 1988.

[11] R. A. Bolt and E. Herranz. Two–Handed Gesture in Multi–Modal Natural Dialog. In *Proceedings of UIST 1992*, pages 7–14, 1992.

[12] D. Bowman and L. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of I3D 1997*, 1997.

[13] D. A. Bowman, D. B. Johnson, and L. F. Hodges. Testbed Environment of Virtual Environment Interaction. In *Proceedings of VRST 1999*, pages 26–33, 1999.

[14] N. Burtnyk, A. Khan, G. Fitzmaurice, R. Balakrishnan, and G. Kurtenbach. StyleCam: Interactive Stylized 3D Navigation Using Integrated Spatial and Temporal Controls. In *Proceedings of UIST 2002*, pages 101–110, 2002.

[15] B. Buxton. Thoughts on the State of 3D CG in Film and Video. *IEEE Computer Graphics and Applications*, 25(3):80–83, 2005.

[16] D. Christianson, S. Anderson, L. He, D. Salesin, D. Weld, and M. Cohen. Declarative Camera Control for Automatic Cinematography. In *Proceedings of AAAI 1996*.

[17] M. Christie and J.–M. Normand. A Semantic Space Partitioning Approach to Virtual Camera Composition. In *Proceedings of Eurographics 2005*, pages 247–256, 2005.

[18] P. Coleman and K. Singh. RYAN: Rendering Your Animation Nonlinearly Projected. In *Proceedings of NPAR 2004*, pages 129–156, 2004.

[19] B. Conner, S. Snibbe, K. Herndon, D. Robbins, R. Zeleznik, and A. van Dam. Three-dimensional widgets. In *Proceedings of I3D 1992*, pages 183–188, 1992.

[20] J. Cooperstock. Making the user interface disappear: the reactive room. In *Proceedings of CASCON 1995: conference of the Centre for Advanced Studies on Collaborative research*, page 15, 1995.

[21] L. Cutler, B. Frölich, and P. Hanrahan. Two-handed direct manipulation on the responsive workbench. In *Proceedings of I3D 1997*, pages 107–114, 1997.

[22] J. Davis, M. Agrawala, E. Chuang, Z. Popović, and D. Salesin. A sketching interface for articulated figure animation. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 320–328, 2003.

[23] M. Dontcheva, G. Yngve, and Z. Popović. Layered Acting for Character Animation. *ACM Trans. Graph.*, 22(3):409–416, 2003.

[24] S. Drucker and D. Zeltzer. CamDroid: A System for Implementing Intelligent Camera Control. In *Proceedings of I3D 1995*, pages 139–144, 1995.

[25] D.Weinshall and M. Werman. On view likelihood and stability. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):97–108, 1997.

[26] J. Funge, X. Tu, and D. Terzopoulos. Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters. In *Proceedings of SIGGRAPH 1999*, pages 29–38, 1999.

[27] M. Gleicher and A. Witkin. Through–the–Lens Camera Control. In *Proceedings of SIGGRAPH 1992*, pages 331–340, 1992.

[28] B. Gooch, E. Reinhard, C. Moulding, and P. Shirley. Artistic Composition for Image Creation. In *Proceedings of the Eurographics Workshop on Rendering*, 2001.

[29] K. Grochow, S. Martin, A. Hertzmann, and Z. Popović. Style-based inverse kinematics. *ACM Trans. Graph.*, 23(3):522–531, 2004.

[30] T. Grossman, R. Balakrishnan, G. Kurtenbach, G. Fitzmaurice, A. Khan, and B. Buxton. Creating principal 3d curves with digital tape drawing. In *Proceedings of CHI 2002*, pages 121–128, 2002.

[31] T. Grossman, D. Wigdor, and R. Balakrishnan. Multi–Finger Gestural Interaction with 3D Volumetric Displays. In *Proceedings of UIST 2004*, pages 61–70, 2004.

[32] Y. Guiard. Asymmetric division of labour in human skilled bimanual action: the kinematic chain as a model. *Journal of Motor Behavior*, 19:486–517, 1987.

[33] F. Guimbretière, A. Martin, and T. Winograd. Benefits of merging command selection and direct manipulation. *ACM Trans. Comput.-Hum. Interact. (TOCHI)*, 12(3):460–476, 2005.

[34] J. Haber and D. Terzopoulos. Facial modeling and animation. In *Proceedings of SIGGRAPH 2004 Course Notes*, 2004.

[35] K. Hawkey, M. Kellar, D. Reilly, T. Whalen, and K. M. Inkpen. The proximity factor: impact of distance on co-located collaboration. In *GROUP '05: Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, pages 31–40, 2005.

[36] L.–W. He, M. Cohen, and D. Salesin. The Virtual Cinematographer: A Paradigm for Automatic Real–Time Camera Control and Directing. In *Proceedings of SIGGRAPH 1996*, pages 217–224, 1996.

[37] K. Hinckley. *Haptic Issues for Virtual Manipulation.* PhD thesis, University of Virginia, 1996.

[38] K. Hinckley, R. Pausch, D. Proffitt, and N. Kassell. Two-handed virtual manipulation. *ACM Trans. Comput.-Hum. Interact. (TOCHI)*, 5(3):260–302, 1998.

[39] S. Houde. Iterative design of an interface for easy 3-d direct manipulation. In *Proceedings of CHI 1992*, pages 135–142, 1992.

[40] T. Igarashi and K. Hinckley. Speed-dependent automatic zooming for browsing large documents. In *Proceedings of UIST 2000*, pages 139–148, 2000.

[41] T. Igarashi and J. Hughes. Clothing manipulation. In *Proceedings of UIST 2002*, pages 91–100, 2002.

[42] T. Igarashi, R. Kadobayashi, K. Mase, and H. Tanaka. Path Drawing for 3D Walkthroughs. In *Proceedings of UIST 1998*, pages 173–174, 1998.

[43] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3d freeform design. In *Proceedings of SIGGRAPH 1999*, pages 409–416, 1999.

[44] T. Igarashi, T. Moscovich, and J. F. Hughes. Spatial Keyframing for Performance–Driven Animation. In *Proceedings of SCA 2005*, pages 107–115, 2005.

[45] R. Jacob, L. Sibert, D. McFarlane, and M. P. Mullen. Integrality and separability of input devices. *ACM Trans. Comput.-Hum. Interact.*, 1(1):3–26, 1994.

[46] F. Jardillier and E. Languénou. Screen–Space Constraints for Camera Movements: The Virtual Cameraman. In *Proceedings of Eurographics 1998*, 1998.

[47] B. Johanson, G. Hutchins, T. Winograd, and M. Stone. Pointright: experience with flexible input redirection in interactive workspaces. In *Proceedings of UIST 2002*, pages 227–234, 2002.

[48] M. Johnson, A. Wilson, B. Blumberg, C. Kline, and A. Bobick. Sympathetic interfaces: using a plush toy to direct synthetic characters. In *Proceedings of CHI 1999*, pages 152–158, 1999.

[49] P. Joshi, W. Tien, M. Desbrun, and F. Pighin. Learning controls for blend shape based realistic facial animation. In *Proceedings of SCA 2003*, pages 187–192, 2003.

[50] S. Jul and G. Furnas. Critical Zones in Desert Fog: Aids to Multiscale Navigation. In *Proceedings of SIGGRAPH 1998*, pages 97–106, 1998.

[51] P. Kabbash, W. Buxton, and A. Sellen. Two-handed input in a compound task. In *Proceedings of CHI 1994*, pages 417–423, 1994.

[52] S. Katz. *Film Directing Shot by Shot: Visualizing from Concept to Screen.* Michael Wiese Productions, 1991.

[53] A. Khan, B. Komalo, J. Stam, G. Fitzmaurice, and G. Kurtenbach. Hovercam: interactive 3d navigation for proximal object inspection. In *Proceedings of I3D 2005*, pages 73–80, 2005.

[54] A. Khan, J. Matejka, G. Fitzmaurice, and G. Kurtenbach. Spotlight: directing users' attention on large displays. In *Proceedings of CHI 2005*, pages 791–798, 2005.

[55] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *Proceedings of SIGGRAPH 2002*, pages 473–482, 2002.

[56] M. Kowalski, L. Markosian, J. Northrup, L. Bourdev, R. Barzel, L. Holden, and J. Hughes. Art-based rendering of fur, grass, and trees. In *Proceedings of SIGGRAPH 1999*, pages 433–438, 1999.

[57] M. Krueger, Thomas Gionfriddo, and Katrin Hinrichsen. VIDEOPLACE – An Artificial Reality. In *Proceedings of CHI 1985*, pages 35–40, 1985.

[58] R. Kruger, S. Carpendale, S. Scott, and A. Tang. Fluid integration of rotation and translation. In *Proceedings of CHI 2005*, pages 601–610, 2005.

[59] G. Kurtenbach. *The Design and Evaluation of Marking Menus.* PhD thesis, University of Toronto, 1993.

[60] G. Kurtenbach and B. Buxton. GEdit: a Test Bed for Editing by Contiguous Gestures. *SIGCHI Bull.*, 23(2):22–26, 1991.

[61] J. Laszlo, M. van de Panne, and E. Fiume. Interactive Control for Physically–Based Animation. In *Proceedings of SIGGRAPH 2000*, pages 201–208, 2000.

[62] C. Latulipe, C. Kaplan, and C. Clarke. Bimanual and unimanual image alignment: an evaluation of mouse-based techniques. In *Proceedings of UIST 2005*, pages 123–131, 2005.

[63] C. Lee, A. Varshney, and D. Jacobs. Mesh saliency. *ACM Trans. Graph.*, 24(3):659–666, 2005.

[64] J. Lengyel, E. Praun, A. Finkelstein, and H. Hoppe. Real-time fur over arbitrary surfaces. In *Proceedings of I3D 2001*, pages 227–232, 2001.

[65] C. K. Liu and Z. Popović. Synthesis of complex dynamic character motion from simple animations. In *Proceedings of SIGGRAPH 2002*, pages 408–416, 2002.

[66] Q. Liu, Y. Rui, A. Gupta, and J. J. Cadiz. Automating camera management for lecture room environments. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 442–449, 2001.

[67] S. Malik. A sketching interface for modeling and editing hairstyles. In *Proceedings of Eurographics Workshop on Sketch-Based Interfaces and Modeling*, 2005.

[68] S. Malik, A. Ranjan, and R. Balakrishnan. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *Proceedings of UIST 2005*, pages 43–52, 2005.

[69] T. McCarthy. Redford sunny about 23 years of 'dance. *Variety*, January 2004.

[70] S. McDermott, J. Li, and W. Bares. Storyboard frame editing for cinematic composition. In *IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces*, pages 206–207, 2002.

[71] M. Mori. Bukimi No Tani. *Energy*, 7:33–35, 1970.

[72] B. Myers, R. Bhatnagar, J. Nichols, C. H. Peck, D. Kong, R. Miller, and A. C. Long. Interacting at a distance: measuring the performance of laser pointers and other devices. In *Proceedings of CHI 2002*, pages 33–40, 2002.

[73] C. Nelson. The development and neural bases of face recognition. *Infant and Child Development*, 10:3–18, 2001.

[74] D. Olsen and T. Nielsen. Laser pointer interaction. In *Proceedings of CHI 2001*, pages 17–22, 2001.

[75] R. Owen, G. Kurtenbach, G. Fitzmaurice, T. Baudel, and B. Buxton. When it gets more difficult, use both hands: exploring bimanual curve manipulation. In *Proceedings of Graphics Interface 2005*, pages 17–24, 2005.

[76] R. Parent. *Computer Animation: Algorithms and Techniques*. Morgan Kaufmann, 2001.

[77] V. Pavlovic, R. Sharma, and T. Huang. Visual Interpretation of Hand Gestures for Human–Computer Interaction: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):677–695, 1997.

[78] K. Perlin and A. Goldberg. Improv: A System for Scripting Interactive Actors in Virtual Worlds. *Proceedings of SIGGRAPH 1996*, pages 205–216, 1996.

[79] J. Pierce, A.Forsberg, M. Conway, S. Hong, R. Zeleznik, and M. Mine. Image plane interaction techniques in 3d immersive environments. In *Proceedings of I3D 1997*, pages 39–43, 1997.

[80] J. Pierce and R. Pausch. Comparing voodoo dolls and homer: Exploring the importance of feedback in virtual environments. In *Proceedings of CHI 2002*, 2002.

[81] J. Pierce, B. Stearns, and R. Pausch. Voodoo dolls: seamless interaction at multiple scales in virtual environments. In *Proceedings of I3D 1999*, pages 141–145, 1999.

[82] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin. Synthesizing realistic facial expressions from photographs. In *Proceedings of SIGGRAPH 1998*, pages 75–84, 1998.

[83] S. Pook, E. Lecolinet, G. Vaysseix, and E. Barillot. Control menus: execution and control in a single interactor. In *Proceedings of CHI 2000 Extended Abstracts*, pages 263–264, 2000.

[84] J. Popović, S. Seitz, and M. Erdmann. Motion sketching for control of rigid-body simulations. *ACM Trans. Graph.*, 22(4):1034–1054, 2003.

[85] J. Popović, S. Seitz, M. Erdmann, Z. Popović, and A. Witkin. Interactive manipulation of rigid body simulations. In *Proceedings of SIGGRAPH 2000*, pages 209–217, 2000.

[86] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The go-go interaction technique: Non-linear mapping for direct manipulation in vr. In *Proceedings of UIST 1996*, pages 79–80, 1996.

[87] I. Poupyrev, S. Weghorst, and S. Fels. Non-isomorphic 3d rotational techniques. In *Proceedings of CHI 2000*, pages 540–547, 2000.

[88] P. Rademacher. View-dependent geometry. In *Proceedings of SIGGRAPH 1999*, pages 439–446, 1999.

[89] B. Robertson. Deep Background. *Computer Graphics World*, 22(7), 1999.

[90] D. Salvucci, M. Zuber, E. Beregovaia, and D. Markley. Distract-r: rapid prototyping and evaluation of in-vehicle interfaces. In *Proceedings of CHI 2005*, pages 581–589, 2005.

[91] J. Segen and S. Kumar. Gesture VR: Vision–Based 3D Hand Interace for Spatial Interaction. In *Proceedings of Multimedia 1998*, pages 455–464, 1998.

[92] A. Sellen, G. Kurtenbach, and W. Buxton. The prevention of mode errors through sensory feedback. *Human Computer Interaction*, 7(2):141–164, 1992.

[93] B. Shneiderman. Direct manipulation for comprehensible, predictable and controllable user interfaces. In *Proceedings of IUI 1997, the 2nd International Conference on Intelligent User Interfaces*, pages 33–39, 1997.

[94] K. Shoemake. Arcball: a user interface for specifying three-dimensional orientation using a mouse. In *Proceedings of the conference on Graphics interface '92*, pages 151–156, 1992.

[95] K. Singh, C. Grimm, and N. Sudarsanam. The IBar: A Perspective–Based Camera Widget. In *Proceedings of UIST 2004*, pages 95–98, 2004.

[96] R. Stoakley, M. Conway, and R. Pausch. Virtual Reality on a WIM: Interactive Worlds in Miniature. In *Proceedings of CHI 1995*, pages 265–272, 1995.

[97] D. Sturman and D. Zeltzer. A design method for "whole-hand" human-computer interaction. *ACM Trans. Inf. Syst.*, 11(3):219–238, 1993.

[98] W. t. Fu and W. Gray. Redirecting direct manipulation or what happens when the goal is in front of you but the interface says to turn left? In *Proceedings of CHI 1999*, pages 226–227, 1999.

[99] D. Tan, D. Gergle, P. Scupelli, and R. Pausch. Physically large displays improve path integration in 3d virtual navigation tasks. In *Proceedings of CHI 2004*, pages 439–446, 2004.

[100] D. Tan, G. Robertson, and M. Czerwinski. Exploring 3D Navigation: Combining Speed–Coupled Flying with Orbiting. In *Proceedings of CHI 2001*, pages 418–425, 2001.

[101] S. Terra and R. Metoyer. Performance Timing for Keyframe Animation. In *Proceedings of SCA 2004*.

[102] M. Thorne, D. Burke, and M. van de Panne. Motion Doodles: an Interface for Sketching Character Motion. *ACM Trans. Graph.*, 23(3):424–431, 2004.

[103] B. Tomlinson, B. Blumberg, and D. Nain. Expressive Autonomous Cinematography for Interactive Virtual Environments. In *AGENTS '00: Proceedings of the Fourth International Conference on Autonomous Agents*, pages 317–324, 2000.

[104] F. Tyndiuk, G. Thomas, V. Lespinet-Najib, and C. Schlick. Cognitive comparison of 3d interaction in front of large vs. small displays. In *Proceedings of VRST 2005*, pages 117–123, 2005.

[105] D. Vogel and R. Balakrishnan. Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple Users. In *Proceedings of UIST 2004*, pages 137–146, 2004.

[106] D. Vogel and R. Balakrishnan. Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays. In *Proceedings of UIST 2005*, 2005.

[107] C. Ware and S. Osborne. Exploration and virtual camera control in virtual three dimensional environments. In *SI3D '90: Proceedings of the 1990 symposium on Interactive 3D graphics*, pages 175–183, 1990.

[108] E. Wolff. Tool Time at Pixar. *Millimeter*, 32(11):33–35, 2004.

[109] C. Yang, D. Sharon, and M. van de Panne. Sketch–based modeling of parameterized objects. In *Proceedings of Eurographics Workshop on Sketch-Based Interfaces and Modeling*, 2005.

[110] R. Zeleznik and A. Forsberg. UniCam: 2D Gestural Camera Controls for 3D Environments. In *Proceedings of I3D 1999*, pages 169–173, 1999.

[111] R. Zeleznik, K. Herndon, and J. Hughes. SKETCH: an interface for sketching 3D scenes. In *Proceedings of SIGGRAPH 1996*, pages 163–170, 1996.

[112] L. Zhang, N. Snavely, B. Curless, and S. Seitz. Spacetime faces: high resolution capture for modeling and animation. *ACM Trans. Graph.*, 23(3):548–558, 2004.

[113] P. Zhao and M. van de Panne. User interfaces for interactive control of physics-based 3d characters. In *Proceedings of I3D 2005*, pages 87–94, 2005.